UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Video distribution in a D2D enabled 5G network supporting Public Safety Services

Richard Siebel
M.Sc. Thesis
August 29, 2017

Supervisors:
prof. dr. J.L. van den Berg
prof. dr. ir. G.J. Heijenk
dr. ir. M.J. Bentum

Design and Analysis of Communication
Systems (DACS) group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

5G mobile networks, which will become available around 2020, aim to support voice, video and other high demanding communication services for billions of connected devices, such as smartphones, sensors, vehicles and other Internet or Things (IoT) devices. Therefore the capabilities of 5G must extend far beyond previous generations of mobile communication. Examples of these capabilities include very high data rates, very low latency, ultra-high reliability, energy efficiency and high capacity. One of the key technologies for supporting these 5G capabilities is device-to-device (D2D) communication. D2D enables devices to communicate with each other without using the infrastructure of the network.

Due to significant investments involved, governments are reluctant to renew Public Safety Networks (PSNs). Also for manufacturers and providers Public Mobile Networks (PMNs) offer a larger market and thereby delivering more profits. As a result, the technological developments for PSNs are lagging behind compared to PMNs. However, synergies can produce a number of benefits, including increased aggregate capacity, improved resiliency and enhanced radio coverage and up to date technological implementations for Public Safety Services (PSS). The convergence of both networks begun with the introduction of Proximity Services at 4G LTE exclusively offering D2D capabilities to PSS officials. This trend is continued in 5G, where Public Safety Services is one of the use cases which will have to be supported.

This thesis focusses on the use of D2D communication for Public Safety Services purposes. In particular, we focus on spectral resource allocation for a group of first responders who are supported by a relay station. User equipment (UE) can be directly linked to the base station or indirectly via the relay station, with the link from the UE to the relay station being a D2D link. It is assumed that all first responders send live video streams to a Central Command Post.

Our goal is to have as many UEs as possible sent their live video streams with a high as possible video quality level. The challenge here is to determine the resource allocation for all UEs and whether they should send their video streams directly to the base station or via the relay station. This depends on the video quality to be obtained, the distance from the UEs to the base station and relay station and whether it is more efficient, with respect to spectral resources, to send directly to the base station or via the relay station. All allocations of resources and route choices for all UEs should be considered in conjunction, which makes it very difficult.

To this end we have investigated what the most efficient method is to allocate spectral resources for streaming video in a 5G mobile network. We also examined what the effects are of a number of key parameters, such as transmit power, required throughput and distance from the relay station to the base station, on the route choice for a UE.

Based on these investigations, we have developed a heuristic resource allocation algorithm. The algorithm bases its choices on the calculations regarding the required resources and route to the base station (i.e. direct or via relay) for a single UE in isolation. When for all UEs choices have been made, corrections in the resource allocation are carried out taking the calculations of all UEs into account. The heuristic algorithm is evaluated by comparing its performance to the performance of the most optimal scheduling. The optimal scheduling is not suitable for implementation as it is not scalable and calculating the most optimal resource allocation takes a long time.

The results, based on simulation, show that the heuristic algorithm is a very promising, efficient and fast method for performing recourse allocation for a clustered D2D enabled 5G network for supporting Public Safety Services. In almost the entire range of the test scenarios, the UEs for both the heuristic algorithm and the optimal scheduling meet their requirements. Only when de distances to the base station become very large, it becomes clear that the heuristic algorithm performs less than the optimal scheduling. As a result, the area where the UEs meet their requirements for the heuristic algorithm is slightly smaller than that of the optimal scheduling. The resource usage of the heuristic algorithm is somewhat higher than optimal scheduling even when both meet the throughput requirements.

Acknowledgement

This thesis marks the end of my three-year pre-master and master program in Electrical Engineering at the University of Twente. I have conducted this thesis at the Design and Analysis of Communication Systems (DACS) group which is part of both the Computer Science and the Electrical Engineering department of the Faculty for Electrical Engineering, Mathematics and Computer Science.

I would like to express my gratitude to a number of people who have supported me in carrying out this thesis. First of all I like to thank my supervisor Prof. Dr. Hans van den Berg for his guidance during the project. His sharp questions and observations always gave me food for thought and kept me going in the right direction. Second, I like to thank committee member Prof. Dr. Ir. Geert Heijenk for his contributions and suggestions during the presentation moments which gave me direction throughout my project. The third person I would like to thank is Dr. Ir. Mark Bentum. Not only for his feedback as a committee member, but also for his encouraging words throughout my years at the university. I would also like to thank M.Sc. Mozhdeh Gholibeigi for her advice, insights and company during the working days at the university and all other members from the DACS group.

Finally, I would like to express my very profound gratitude to my wife, children, parents and other family members for their never-ending support and encouragements throughout my years of study. This accomplishment would not have been possible without them.

Richard Siebel

Table of content

List of figures

List of tables

List of acronyms and abbreviations

4G	4th generation network
5G	5th generation network
BS	Base Station
CIF	Common Intermediate Format
CSI	Channel State Information
D2D	Device to Device communication
dB	Decibel
FBMC	Filter Bank Multi-Carrier
FPS	Frames Per Second
GFDM	Generalized Frequency Division Multiplexing
Hetnet	Heterogeneous network
HF	High Frequency
IoT	Internet of Things
kHz	Kilohertz
km	Kilometer
LTE	Long Term Evolution
Mbps	Megabit per second
MEC	Mobile Edge Computing
MHz	Megahertz
ms	Millisecond
NFV	Network Function Virtualisation
NOMA	Non-Orthogonal Multiple Access
OFDM	Orthogonal Frequency Division Modulation
PMN	Public Mobile network
PSN	Public Safety Network
PSS	Public Safety Services
Ptx	Transmit power
QoE	Quality of Experience
QoS	Quality of Service
RAT	Radio Access Technology
RB	Resource Block
RS	Relay Station
SC-FDMA	Single Carrier Frequency Division Multiple Access
SINR	Signal to Interference Ratio
UE	User Equipment
UFMC	Universal Filtered Multi-Carrier
V2V	Vehicle to Vehicle communication
VHF	Very High Frequency
VR/AR	Virtual reality / Augmented reality
Wifi	Wireless Fidelity

1. Introduction

Mobile technology has experienced a number of generation changes, where mobile communication has transformed from clunky, heavy and not so portable devices that supported a single service (voice) into a complex interconnected environment, built on integrated multi-technology networks that support millions of applications and billions of subscribers, delivering content to a multitude of devices and screens, to enterprises and consumers, with a potentially massive benefit to society. 5G mobile networks which will become available around 2020 has to support voice, video and a complex range of communication services for billions of connected devices, such as smartphones, sensors, vehicles and other Internet or Things (IoT) devices [1].

The capabilities of 5G must extend far beyond previous generations of mobile communication. Examples of these capabilities include very high data rates, very low latency, ultra-high reliability, energy efficiency and high capacity, and will be realized by the development of LTE in combination with new radio-access technologies. Key technology components include extension to higher frequency bands, access/backhaul integration, flexible duplex, flexible spectrum usage, multi-antenna transmission, ultra-lean design, and device-to-device (D2D) communication [2].

This thesis focusses on the use of D2D communication for Public Safety Services purposes. Here our goal is to have first responders send live video streams to a Central Command Post. The main question is therefore aimed at D2D, Public Safety Services and video distribution.

This chapter is organised as follows. Section 1.1 gives a description of the problem we want to solve in this thesis and Section 1.2 contain the research questions we want to answer. Section 1.3 describes our approach and contributions. In Section 1.4 related work is discussed which served as the starting point for this research and in Section 1.5 the outline of the thesis is given.

1.1. Public Safety Services

Up till now there was a wide consensus among Public Safety Agencies (PSA) regarding the need of dedicated Public Safety Networks (PSNs) for mission-critical communications because commercial Public Mobile Networks (PMNs) are not considered able to provide the required degree of service availability, reliability, and security. However, the significant investment required to rollout dedicated PSNs may not be affordable for some governments. Hence, while some countries can deploy new dedicated PSNs with nationwide coverage, others may decide to cover only some critical areas with dedicated infrastructures or to rely exclusively on PMNs [3]. In addition, due to significant investments, governments are less willing to renew their networks and keep up with the technological developments. Therefore the technological developments for PSNs are lagging behind compared to PMNs, as they provide a larger market for manufacturers and providers and thereby delivering more profits.

Even when dedicated PSNs can be rolled out, the unpredictable nature of the time, place, and scale of an incident renders it virtually impossible to ensure that first responders will have proper support from the PSNs during an emergency (e.g., due to lack of coverage, capacity, or damaged infrastructure). Examples of these are “The shooting at the beach in Hoek van Holland” and “The crash of flight 1951 of Turkish airlines”, also known as the Polder crash. Both happened in 2009. In both cases, the PSN did not deliver the functionality that was expected.

In this context, significant opportunities for creating and exploiting synergies between PMNs and PSNs arise. Synergies can produce a number of benefits, including increased aggregate capacity, improved resiliency and enhanced radio coverage and up to date technological implementations for Public Safety Services (PSS). The merging of PMNs and PSNs has begun by the introduction of Proximity Services at 4G LTE exclusively offering D2D capabilities to PSS officials even if all base stations are out of service [4]. This trend is continued to 5G. Although PSS is not referred to as one of the five verticals identifying the key requirements for 5G, it is one of the use cases (lifeline communication) which will have to be supported by 5G [5].

For handling accidents and combating disasters first responders of multiple Public Safety Agencies can be deployed like police officers, firefighters and paramedics. For the most effective deployment of these services coordination between the first responders is of great importance. To facilitate this coordination a shared PSN is used and command and control is carried out from a central location. For creating situational awareness to support the decision making process life video streams and high resolution images can be sent from the first responders to the Central Command Post. In turn, the Central Command Post can send out orders supplemented with maps of the area and pictures or other critical information about the incident or disaster which is needed in the deployment area. Also video streams of others present on the site can be shared.

However, there may be situations where there is no network coverage in the deployment area or due to poor radio link quality many spectral resources are needed to maintain a communication link, which drastically reduces throughput and network capacity. For example buildings blocking the signal, broken down base stations or areas that lay partially or completely outside the coverage of a base station. As a result first responders are not able to communicate with others, which can put them in life threatening situations. Further the Central Command Post is not able to receive (video) information from the PSS operators and distribute orders and other critical information, which can lead to chaos among the first responders.

In order to extend coverage or improve throughput of the network in the deployment area, a system is proposed in which clustered D2D communication is used where a cluster head will act as a relay station, see Figure 1-1.

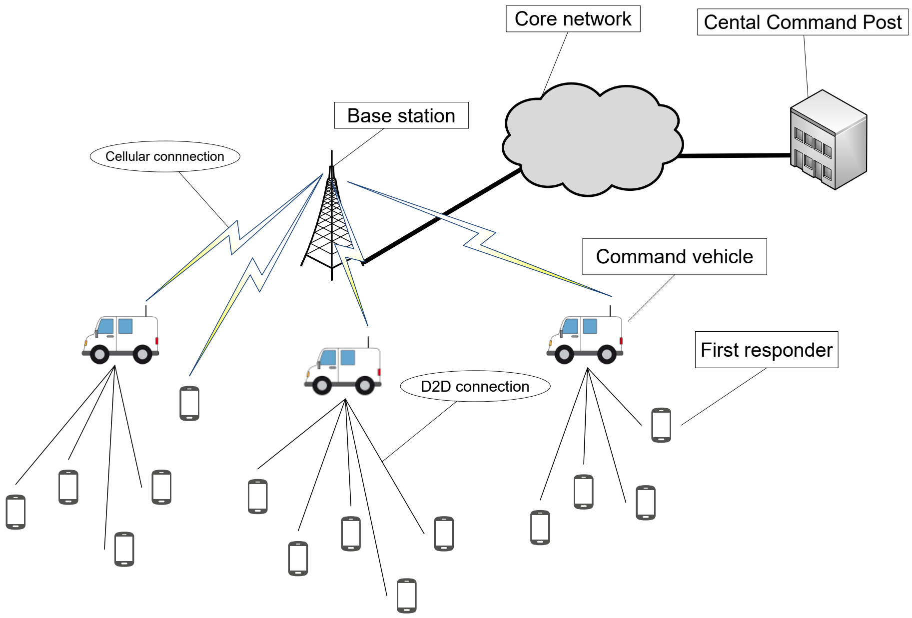

Figure 1-1. Clustered D2D Public Safety Network.

In case a first responder does not have cellular coverage, a D2D link is established to the cluster head. For this configuration the command vehicle is chosen to be the relay station, although this may also be a separate relay station. Even though the command vehicle may not have the best quality link (SINR, throughput) with the cellular network, there are several reasons why it is considered cluster head: First, there is no need for an additional vehicle and personnel. Second, the command vehicles can power the communication equipment, so acting as a cluster head will not drain the radios battery; third, the command vehicle will likely be featured with redundant means of communication in case there is no cellular network available. This could include a satellite link or VHF/HF radio connection; last, in case all links to the Central Command Post fail, Command and control can be taken over from there.

In this thesis we will develop an uplink radio resource allocation algorithm for the architecture given in Figure 1-1 where all UEs want to send a video stream to the Central Command Post, either directly through the base station or via the relay station. In both cases the communication channels for the relay station and UEs are separated in time and frequency, so these will not be used by others. In this thesis we focus on a single cluster containing multiple UEs, one relay station and one base station, as shown in Figure 1-2. The reason for this setup is twofold. First, the choice for a single base station represents a realistic situation where first responders are active at the edge of a 5G network. One of the ambitions for 5G is to have network coverage everywhere, however due to a disaster it can be that a part of the network is malfunctioning, limiting network coverage. Second, conducting a research with this setup can be completed in the available time and contains all the key elements to make a significant contribution to video streaming capabilities for public safety services in 5G.

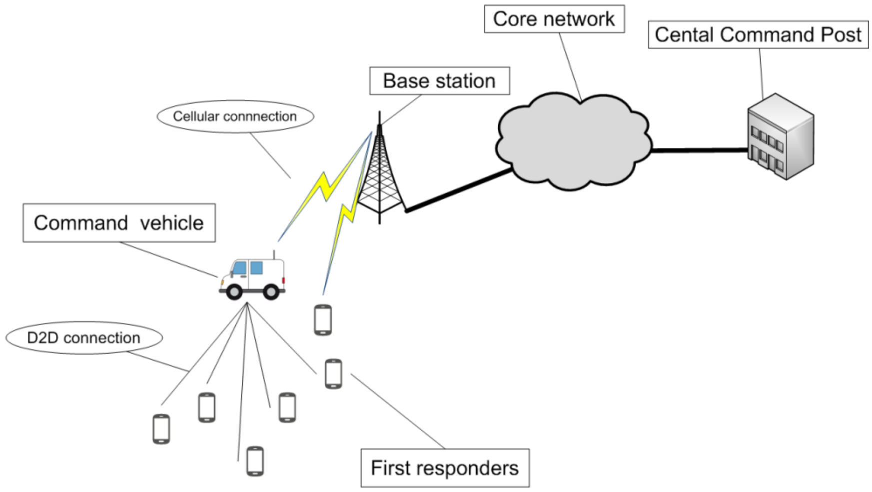

Figure 1-2. Research architecture.

1.2. Research questions

In a 5G network communication channels are divided over frequency and time. So for every device active on the network it has to be decided at which time and at which frequency it is allowed to transmit its information. This process is called resource allocation. Factors that greatly affect resource usage are the distance from the transmitter to the receiver, transmit power and the possibility to use a relay station. With a lot of active devices, limited bandwidth and demands for high data rates resource allocation is very challenging.

The main research question we want to answer in this thesis is: How to create a suitable low complexity resource allocation algorithm for the distribution of live streaming video in a clustered D2D enabled 5G network supporting Public Safety Services? To answer the main research question, the following sub-questions are formulated:

1. Which requirements must be taken into account when developing and evaluating the proposed algorithm?
    a. What are the requirements for communication systems and video used by public safety services?
    b. What are the requirements for streaming different quality type videos?

2. What can be a suitable resource allocation algorithm for meeting the video service requirements and leading to high network efficiency?

3. How does the proposed resource allocation algorithm perform compared to the optimal resource allocation scheme?
    a. To which extent does the proposed resource allocation algorithm deliver the same video quality as the optimal scheme?
    b. What is the resource usage of the proposed resource allocation algorithm, and how does it compare to the optimal scheme?

1.3. Approach and contributions

In this thesis, we have investigated uplink video distribution for public safety services application in a D2D enabled 5G network. In particular, we have looked at how to perform resource allocation when using a relay station, which is very challenging as all allocations of resources and route choices for all UEs have to be considered in conjunction. Our basic scenario consists of a cluster of UEs, where each UE can directly send its video stream to the base station or indirectly via the relay station, depending on the throughput and resources used. Our research has resulted in the following contributions.

- We have investigated what the requirements are for using streaming video for Public Safety Purposes.

- We have developed a numerical approach for calculating the amount of resources needed for transmitting streaming video via a 5G network.

- We have investigated which transmission power is most suitable for the relay station compared to the transmit power of the UEs.

- We have developed a heuristic algorithm for performing resource allocation in a clustered D2D enabled 5G PSS network.

- We have designed and implemented multiple scenarios in Matlab that simulates a PSS communication environment.

- We have developed an optimal resource scheduling algorithm which is used to evaluate the heuristic resource allocation algorithm.

- We have evaluated the performance of the heuristic resource allocation algorithm.

1.4. Related work

In reference [6] a D2D clustering approach is used to enhance the performance of public safety networks. In each cluster, a single device, the cluster head, is selected to communicate with the base station in either uplink or downlink direction, or both. The cluster head relays the information from and or to the other cluster members. Neighboring UEs use orthogonal resources, and thus interference is not an issue. The cluster head is the UE which can achieve the highest throughput from and/or to the base station. On the one hand, this is beneficial as this provides the highest possible throughput to the cluster. On the other hand, not all devices are suitable for serving as cluster head. Dismounted personnel wear small communication means with limited battery capacity. Choosing one of them as a cluster head can severely impact their UEs battery life and ultimately leave a first responder without a working means of communication. Our preference is to use a dedicated cluster head with suitable communication equipment and power supply to perform this task.

1.5. Outline of the thesis

The remainder of this thesis is ordered as follows. Chapter 2 describes background information on 5G networks and D2D communication which are essential to this research. Chapter 3 describes the requirements for the use of video for PSS and throughput requirements for various quality type video streams. Chapter 4 discusses how to determine the amount of resources needed for streaming video in a 5G network. Chapter 5 describes the main challenge for resource allocation in a clustered D2D enabled 5G network and covers the optimal solution and the proposed heuristic algorithm. The heuristic algorithm is evaluated in Chapter 6. Finally in Chapter 7 the conclusions are discussed and suggestions are provided for future work.

2. 5G mobile networks

The background information provided in this chapter is essential as it forms a basis for this research. Section 2.1 discusses background information on the concept of 5G, the requirements set for 5G and which technologies are needed to meet these requirements. D2D, which is one of the key technologies for 5G, plays an important role in this research and is described in Section 2.2.

2.1. What is 5G?

5G can be seen as a shift in mind-set where we are moving from one-size-serves-all networks with vertical infrastructures to agile networks that can be programmatically deployed for specific high-level use cases [1]. Initially, cellular networks provided people with a way to talk to each other free of location constraints. Today's 4G cellular networks provide people and businesses with access to information and entertainment instantly. As a result, industries are transforming, creating new business cases that use connectivity and in some cases abandoning traditional ones. By providing a flexible and adaptable network, 5G will offer a platform that will support many use cases that we cannot even imagine possible today.

An important differentiation between 4G and 5G is the integration of verticals in the design of 5G. These verticals are: Factories of The Future, Automotive, Health, Energy and Media & Entertainment. Use-cases originating from these verticals are considered as drivers of 5G and must be covered by the design and standardisation process. As a result vertical industries will have enhanced communication technology available to trigger the development of new products and services. With 5G, networks will be transformed into intelligent orchestration platforms [7].

Not all future 5G application will require networks that are ultra-fast, super smart, and must have the capacity to support massive numbers of devices. For example, very-high-rate applications such as streaming high-definition video may have relaxed latency and reliability requirements compared to driverless cars or public safety applications, where latency and reliability are paramount but lower data rates can be tolerated [8]. That's why networks will have to be built in a flexible way by introducing logical network slices to meet the specific demands of each use case from one of the vertical industries [9]. A network slice can be composed out of a collection of 5G network functions and RAT settings.

There is a general consensus about the demands that 5G systems will have to meet in comparison to the current 4G standard. A 5G system should deliver 1000 times more data per area, up to a 100 times higher user data rate, and up to a 100 times more connected devices. Ground-breaking technological innovations are needed for meeting the ambitious requirements set for 5G. In Table 2.1 an overview is given of the most important concepts of these technological innovations which are described in more detail in Appendix A. Since D2D has an important role in this thesis, it is described in detail in the following section. For an even more detailed explanation on these technologies, see references [10], [8], [9], [11], [12], [13], [14] and [15].

Millimeter wave spectrum	Mobile edge Computing
Massive MIMO and beamforming	Radio Access Techniques
Wireless Software Defined Networking	Network densification
Network Function Virtualisation	Device to device communication

Table 2.1. Supporting technologies for 5G networks.

To finally meet the requirements set for 5G all these technologies will have to be integrated so that they complement each other. Figure 2-1 shows a general 5G cellular network architecture where, amongst other emerging technologies, most of the technologies indicated above are operating in an interconnected manner.

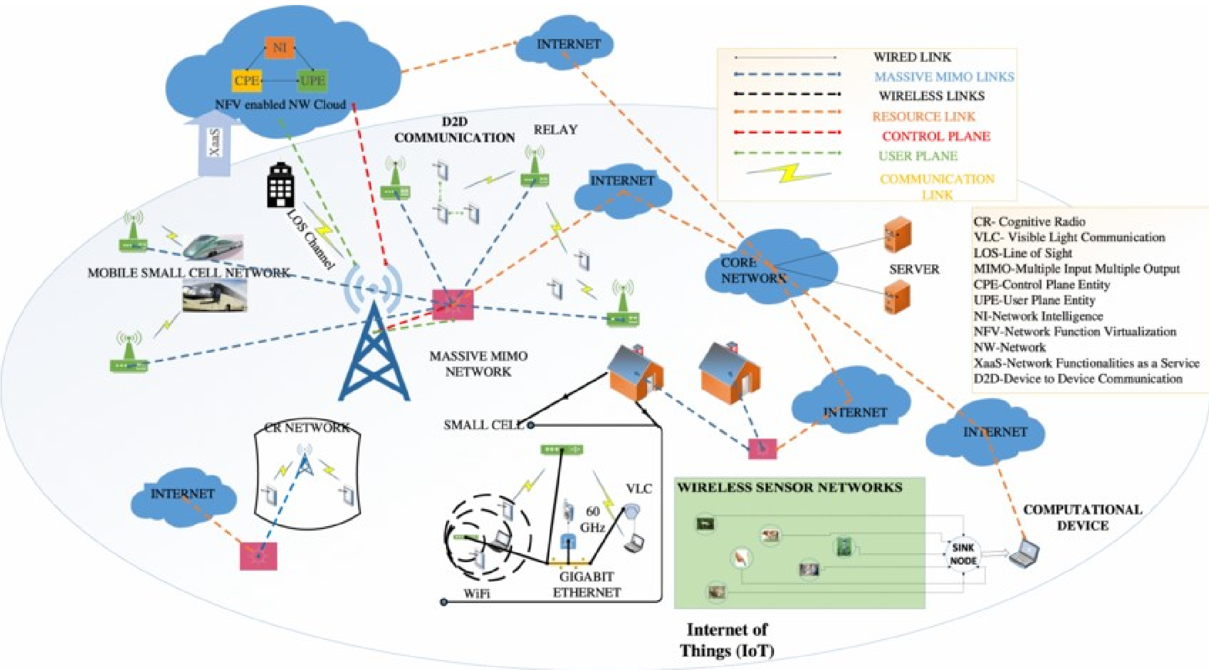

Figure 2-1. General 5G cellular network architecture [13].

2.2. Overview of D2D communication

D2D communication refers to direct communication between devices, without their traffic going through any network infrastructure. Under normal conditions the base station is controlling the radio resource usage of the D2D links to minimize interference effects. In this setup D2D can help to increase spectrum efficiency and hence, network capacity. D2D can also be used for fall-back connectivity for an out of coverage device by using an in-coverage device as a relay.

In conventional cellular networks all communication takes place via a base station. Even when two devices are in close proximity of each other, communication takes place via the base station. In situations where real time applications are used like VR/AR or live video streaming, demanding a low latency and high data throughput this is not optimal. One of the solutions for this is D2D communication. D2D communication can be applied in cellular networks in several ways. The most obvious ones are shown in Figure 2-2. (a) Device relaying with base station controlled link establishment; (b) direct D2D communication with base station controlled link establishment; (c) direct D2D communication with device controlled link establishment; and (d) device relaying with device controlled link establishment. A D2D application that is not shown is range extension. Here, a UE, which is out of range of a base station, gets connected to the network via another relaying UE [16].

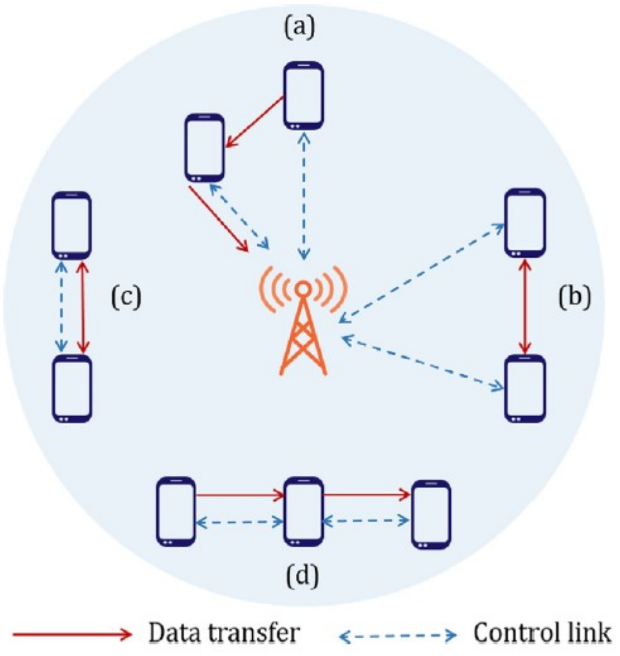

Figure 2-2. Device to device scenarios [16].

For D2D communication a number of choices can be made concerning the allocation of spectral resources, each with its advantages and disadvantages.

In-band D2D

In In-band D2D the cellular spectrum is used for both cellular and D2D users. The motivation for in-band D2D is that the cellular frequency spectrum can be controlled and so the QoS. Here a further breakdown can be made in underlay and overlay. In overlay D2D a part of the cellular frequency band is dedicated for D2D users, which means that there are less frequencies left for cellular users. The advantage of in-band overlay is that there is no interference between cellular and D2D users. The downside is that the frequency band is not used optimally. A more efficient use of the frequency band can be achieved by using underlay D2D, in which frequencies for cellular users are reused for D2D communication. This requires tight control over the frequencies by the base station. The key disadvantage of in-band underlay is the interference caused by D2D users to cellular users and vice versa. However, this interference can be minimized when D2D communication is allowed in the uplink spectrum, as will be discussed in the next paragraph.

The cellular uplink spectrum is often under-utilized compared to the downlink, therefore D2D communication uses uplink resources to improve the resource utilization. Moreover, when D2D communication share downlink resources, base stations become fairly strong interferers for D2D receivers, and D2D transmitters may cause high interference to nearby co-channel cellular UE's. This may significantly degrade the network performance. When D2D communication uses uplink resources, the interference from D2D to cellular transmissions can be better handled, since base stations are more powerful than UEs and therefore suffer less from D2D interference. In addition, SC-FDMA that is used in LTE for uplink communication is less complex and consumes less energy [17]. Therefore, sharing the uplink spectrum for D2D communication is preferred [18], [4], [19].

Out-band D2D

In out-band D2D an unlicensed frequency band (ISM) for D2D communication is used, lifting the problem of mutual interference between D2D and cellular users. Here a choice can be made between controlled and autonomous. In the first case the cellular network advanced management features are used to improve efficiency and reliability of D2D communication. In the autonomous case the choice to use D2D is left to UEs, reducing the overhead of the cellular network. The most known access technologies for out-band D2D communication include Wifi direct, Bluetooth and ZigBee. The disadvantage is that these systems cannot provide security and QoS guarantee as cellular networks do [20]. In addition, devices must have an additional interface, which can be problematic for low cost sensor devices. For current and future smartphones this will be no problem. They are usually equipped with multiple interfaces [21].

3. Requirements for video streaming for PSS usage

In this chapter the requirements for video usage for public safety services purposes are described. Section 3.1 the most relevant performance specifications are given and the data rate requirements for different resolution frame rate combinations in Section 3.2. The video codec used in this study is one of the most common codecs used in video surveillance [22]. Although there may be qualitatively better codecs, this research does not discuss other video codecs as we primarily focus on radio resource allocation.

3.1. Quality requirements

In reference [23] a video quality test was conducted to estimate the level of video quality that first responders find acceptable for tactical video applications. For this they have performed experiments involving first responders for the determination of a basic quality thresholds for public safety tactical video applications. The most relevant performance specifications for tactical video are summarized in the table below.

Maximum one-way video delay	1 second
Minimum frame rate	10 frames per second
Suitable image size	CIF (352 x 288), SIF (360 x 240), and QVGA (320 x 240)

Table 3.1. Performance specifications for tactical video.

For the case that the one-way delay exceeds the recommended maximum delay it is indicated that "more delay and a better picture quality" is preferable to "less delay and a worse picture quality" for tactical video.

This study dates back to 2007. Today, the use of high-resolution video in small mobile devices, such as the smartphone, has become the norm. This study therefore assumes 4CIF (704 x 576) image resolution, which is comparable to DVD quality, for use by public safety services. It is stated that "more delay and a better picture quality" is preferable to "less delay and a worse picture quality" for tactical video. In this research this is interpreted as "less frames per second and a better picture quality" is preferable to "more frames per second and a worse picture quality"

3.2. Requirements for various video formats

In this study we do not investigate the implementation of different codecs. However we prefer the use of a codec because the required bandwidth for a raw single SD or HD video stream is enormous. A widely used codec for surveillance cameras is CIF [22]. In [24] it is indicated that 4CIF is comparable to DVD quality video. These codecs we will use in this study. Table

3.2 gives an overview of the required data rates for different types of resolution and frame rate combinations which we will use in this research.

Resolution	5 FPS	10 FPS	15 FPS	25 FPS
CIF	100 kbps	200 kbps	300 kbps	500 kbps
4CIF	400 kbps	800 kbps	1200 kbps	2000 kbps

Table 3.2. Bitrate for different quality and framerate.

4. Resource allocation for streaming video in 5G

In this chapter, we investigate how to allocate spectral resources to users for streaming video in a D2D enabled 5G cellular network supplemented with a relay station. For this we first look at resource allocation in 4G LTE, which serves as a basis for 5G and so for our approach. Then we will examine how much data a resource block contains. Next, we will come up with a method for determining the amount of resources needed to support streaming video, with and without the deployment of a relay station.

4.1. Resource allocation in 5G LTE

One of the principle technologies for (initial) 5G radio access networks is 4G LTE [25]. Therefore this section explains how spectral resources are allocated in LTE. In 4G LTE, E-UTRA uses OFDMA for the downlink channel and SC-FDMA for the uplink channel. The latter is used to overcome the high peak-to-average power ratio and thereby safes power at the UE [17], [16]. In LTE resources are divided over time and frequency. In the frequency domain LTE can work with bandwidths of 1.4MHz, 3MHz, 5MHZ, 10MHz, 15MHz or 20MHz containing 6, 15, 25, 50, 75 and 100 so called resource blocks respectively. To calculate the amount of available resource blocks, the following calculation can be used [17]:

$$\text{Available } RBs = \frac{bandwidth * 0.9}{180\ kHz} \tag{4.1}$$

Here the bandwidth is the total available bandwidth, the factor 0.9 takes into account 10% guard band and 180 kHz is the bandwidth of a single resource block which consists of 12 subcarriers with a bandwidth of 15 kHz. It must be noted that for the 1.4 MHz band this calculation is not correct [17].

In the time domain LTE uses radio frames of 10ms which include 10 sub-frames of 1ms. Each sub-frame contains two resource blocks with a duration of 0.5ms [17]. A resource block is the smallest resource allocation unit which can be assigned to a specific device. Figure 4-1 shows an example of assigned resource blocks using a 1.4 MHz frequency band.

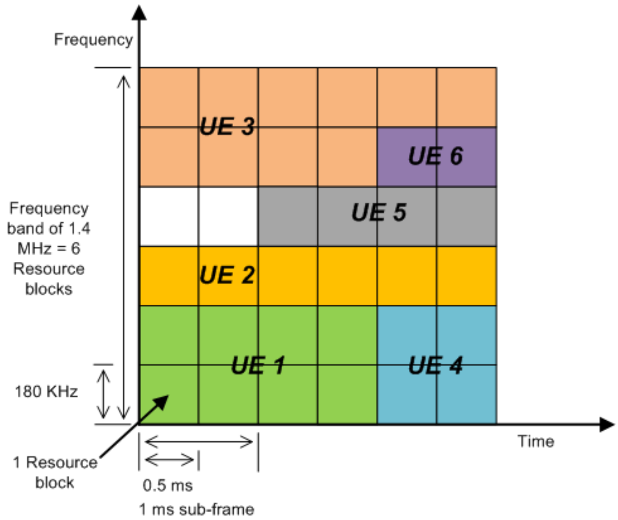

Figure 4-1. Example of resource allocation in LTE [26].

4.2. Determining the required resources for supporting streaming video

In this section we examine how much resources are needed for a given quality video stream. This section is structured as follows: in Subsection 4.2.1 the key parameters and formulas are discussed for calculating data rates in LTE. In Subsection 4.2.2 we use these formulas to examine the relation between LTE resource blocks and data rate. Finally in Subsection 4.2.3 formulas are given to determine the required amount of resource block for a given quality video stream.

For determining the required amount of resources needed by a first responder who wants to send a video stream with a certain resolution and frame rate use is made of calculations and parameter settings from [27] and the references in there.

4.2.1. Key parameters and formulas for calculating data rates in LTE

For this part of the study the path loss for both the links to the base station and to the relay station is given by:

$$L(d) = 146.1 + 10 * n * log_{10}(d_{(km)}) \tag{4.2}$$

This path loss formula , taken from [28], consists of a fixed part which depends on the height of the antenna and the frequency used. n is the path loss exponent set to 3.53 and d is the distance between the transmitter and receiver in kilometers. For the evaluation of the proposed heuristic algorithm separate path loss values are used for the links to the base station and the relay station. This is described in Section 4.4.

The formula for the signal to interference and noise ratio is given by:

$$SINR = \frac{P_{TX} / m}{L(d) * N} \qquad (4.3)$$

Here P_{TX} is the transmitting power, m is the number of resource blocks used by the transmitting device during transmission, $L(d)$ is the path loss and N is the noise component set to -146.45 dB [27]. The noise component consists of of two parts. The first is the thermal noise of -151.45 dB for a single resource block of 180 kHz and the second part is a 5dB noise noise figure.

The data rate r is given by the Shannon formula which has been modified with an implementation factor sigma [27], [29]. For 16 QAM modulation, sigma is set to 0.4.

$$r = (m\, x\ 180kHz) * \sigma * log_2(1 + SINR) \qquad (4.4)$$

In order for the receiver to be able to distinguish the information in a resource block from noise and distorting signals, the receive power must be of a minimum level. In this thesis we use fixed transmit powers for the relay station and the UEs. Because of this, the transmitted power is distributed over all the allocated resource blocks. This means that there is a maximum number of resources that can be assigned to the relay station or a UE. If more resources are allocated then the receiving power drops below the level for which it is still possible to distinct the information from the noise and interference. The maximal number of resource blocks can be calculated by rewriting the SINR formula the following way:

$$m_{max} = \frac{P_{TX}}{SINR_{min} * L(d) * N} \qquad (4.5)$$

$SINR_{min}$ constraints the maximum number of usable resource blocks for a transmitting device. Using more resource blocks than m_{max} will result in an unacceptably low signal quality at the intended receiver. $SINR_{min}$ is set to -10 dB [27].

4.2.2. Relation between data rate and LTE resource blocks

Initially, the most obvious method for determining the required resources seems to be a rewriting of the formulas given in Subsection 4.2.1. However, this is not the case. Completely written out, the formula for calculating the data rate is as follows where *m* is the parameter we are searching for:

$$r = (m\, x\ 180kHz) * \sigma * log_2(1 + \frac{P_{TX} / m}{L(d) * N}) \qquad (4.6)$$

Since *m* is present both inside and outside of the log function of (4.6), it is very complex to determine its value. A numerical approach is therefore more obvious. Using the given formulas, it can be determined what the data rate is for a given number of resource blocks. Figure 4-2 shows the data rate for a given number of resources used where a 2 Watt transmitter is located at various distances from its destination, ranging from 100 to 1000 meters. The maximum number of available resource blocks at any given moment is 50.

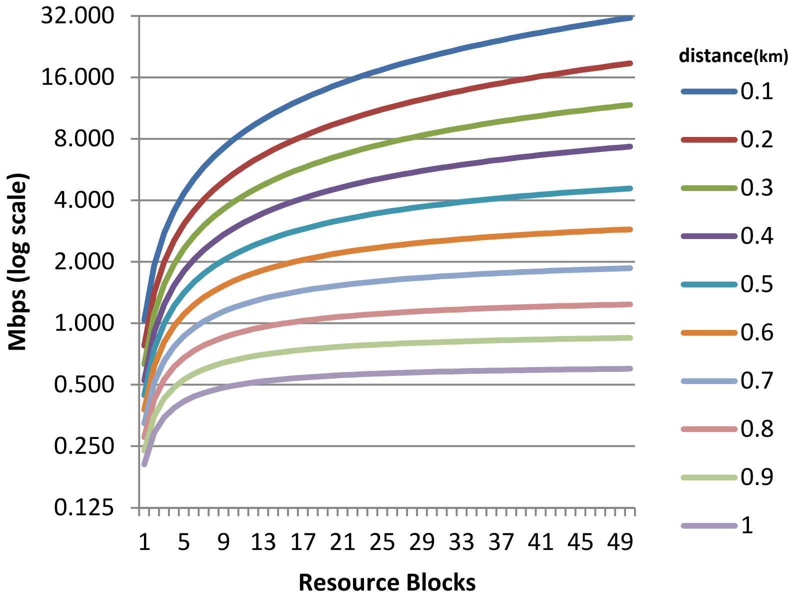

Figure 4-2. Achievable data rate as a function of the number of resource blocks for various distances.

It can be seen that for all distances the data rate increases very rapidly for the first few resource blocks, but starts to level off at some point. From this point on a lot of resources have to be added for only a slight increase in date rate. When the transmitter is further away from the receiver, this flattening becomes more significant. This phenomenon can be examined even better in Figure 4-3. Here 100% indicates the maximum achievable data rate. When a transmitter is located at a close distance of its intended receiver one resource block contains only a small portion of the maximal achievable data rate and each added resource block adds an almost equal increase in data rate. However, when the transmitter is located further away the first few resource blocks contain a very large portion of the achievable data rate after which adding more resource blocks almost give no further improvement in throughput.

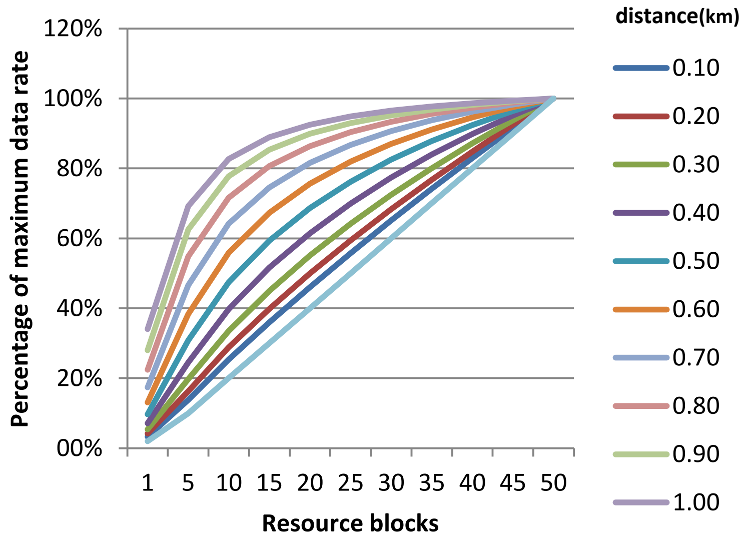

Figure 4-3. Achievable data rate as a function of the number of resource blocks for various distances in percentage.

In general, it can be said that two resource blocks that are used at the same time contain less data than two consecutive resource blocks. This is because in the first case the transmit power is divided over two resource blocks, where in the second case both resource blocks use the total available transmit power. Based on this principle, it is more advantageous, in terms of resources, to use fewer resources for a longer period of time for a transmission than a lot of resources for a short period of time.

4.2.3. Calculating the required amount of resources for a video stream

In the previous Subsection, it has been found that it is more efficient to use as few resource blocks as possible at the same as this will result in the highest amount of data per resource block. In this subsection we draw up a formula for determining the amount of resources for a given quality video stream for which we want to use as few resources as possible for a longer period of time.

Unlike the frequency domain, the time domain is linear, as can be seen by the straight line in Figure 4-3. If a UE is transmitting 1 Mbps but only gets resources half of the time, this will leave a throughput of 0.5 Mbps. This linearity is used for determining the amount of resources needed for sending a certain quality video stream to an intended receiver. The formula is as follows:

$$\left((1 - t_{ue}) * Z^{-1}_{(R_T,d_{ue})}\right) + \left(t_{ue} * Z_{(R_T,d_{ue})}\right) \geq R_T \tag{4.7}$$

Were R_T is the target rate related to a particular video quality, $Z_{(R_T,d_{ue})}$ is the data rate gained with minimal number of resource blocks needed to match or exceed R_T, and $Z^{-1}_{(R_T,d_{ue})}$ is the data rate gained with one resource block less than $Z_{(R_T,d_{ue})}$. The distance of the UE to the receiver is indicated by d_{ue}. The time the higher data rate from $Z_{(R_T,d_{ue})}$ is needed in addition to $Z^{-1}_{(R_T,d_{ue})}$ to match R_T is indicated by t_{ue} and is calculated in the following way:

$$t_{ue} = \frac{R_T - Z^{-1}_{(R_T,d_{ue})}}{Z_{(R_T,d_{ue})} - Z^{-1}_{(R_T,d_{ue})}} \tag{4.8}$$

Which is the difference in data rate between the target rate and $Z_{(R_T,d_{ue})}$ divided by the difference of the rate $Z_{(R_T,d_{ue})}$ and the rate gained with one resource block less, as shown in Figure 4-4.

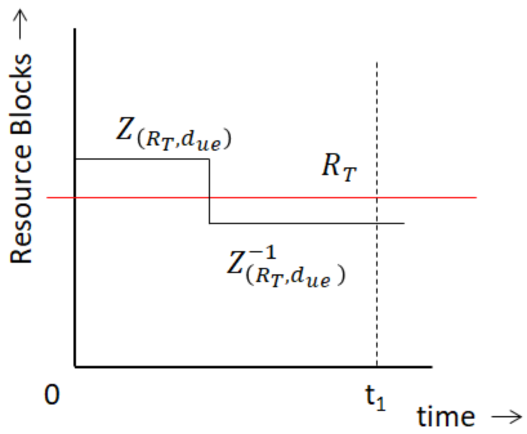

Figure 4-4. Determining the required number of resource blocks.

Formula (4.7) can also be used for a deployed relay station. When the relay station is supporting one UE they both have the same target rate. When supporting multiple UEs the target rate for the relay station is the sum of the target rates of all UEs linked to it. The required amount of resources for a relay station supporting multiple UEs is calculated using the following formula:

$$\left((1 - t_{rs}) * Z^{-1}_{(R_{T_{rs}},d_{rs})}\right) + \left(t_{rs} * Z_{(R_{T_{rs}},d_{rs})}\right) \geq \sum_{k=1}^{K} R_{T(k)} \tag{4.9}$$

Where $R_{T(k)}$ is the target rate of UE(k), d_{rs} is the distance of the relay station to the base station, and t_{rs} is the time the higher data rate from $Z_{(R_{T_{rs}},d_{rs})}$ is needed in addition to $Z^{-1}_{(R_{T_{rs}},d_{rs})}$ to match the sum of all target rates of the UEs linked to the relay station. The total amount of resources needed for the relay station and the UEs linked to it can be determined in the following way:

$$\begin{aligned}\sum_{k=1}^{K}\left(\left((1 - t_k) * Z^{-1}_{(R_{T(k)},d_k)}\right) + \left(t_k \text{ x } Z_{(R_{T(k)},d_k)}\right)\right) \\ \leq \left((1 - t_{rs}) * Z^{-1}_{(R_{T_{rs}},d_{rs})}\right) + \left(t_{rs} * Z_{(R_{T_{rs}},d_{rs})}\right)\end{aligned} \tag{4.10}$$

It must be taken into account that the total number of resource blocks used by the UEs linked to the relay station, the relay station itself and the UEs directly linked to the base station must not exceed the number of available resource blocks. It also must be noted that we do not describe in detail the manner in which the allocated resources have to be assigned to the UEs and the relay station per unit time. We only indicate that the allocated resources fit the available resources.

4.3. Critical distance

When a UE is located close to a base station, for a given data rate, the UE will us a direct link to the base station as this link will use the least resources. When the UE is located at a distance from the base station it may be that an indirect link via the relay station uses fewer resources than a direct link. However, there is also a distance where the resource usage for both paths is the same, which is called the critical distance [27]. Insight to this critical distance is of importance to examine the parameters which are of influence on the choice for a transmission path for a UE. The critical distance will be explained with reference to Figure 4-5. Here X-Y indicates the link between entity X and Y and $d_{X\text{-}Y}$ the distance between entity X and Y. The distance d_{UE-BS}, where the direct link UE-BS uses as much resources as an indirect link via the relay station UE-RS + RS-BS, is called the critical distance. Unlike shown in Figure 4-5 it is also possible that the critical distance lies beyond the relay station, so $d_{UE-BS} > d_{RS-BS}$.

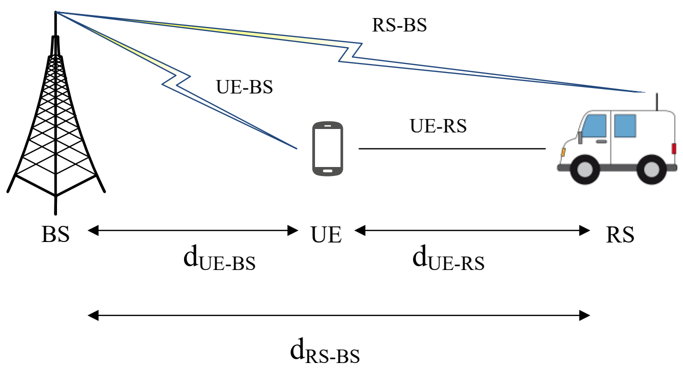

Figure 4-5. Determining the critical distance.

Amongst others, the critical distance depends on the distance of the relay station to the base station, the transmit power of the UE and the relay station and the required throughput. The higher the transmit power of the UE, the further the critical distance lies. If the transmission power of the relay station increases, it becomes more efficient, with respect to resources, to send via the relay station and so the critical distance is reduced. When the required throughput increases, more resources are needed at the same time, reducing the available transmission power per resource block. This will also decrease the critical distance. In Figure 4-6 three relay stations are located at three different distances to the base station, namely 0.3, 0.6 and 0.9 kilometers. Then for a UE directly linked to the base station and one to every relay station it is determined how many resources for that particular transmission path are required for various distances of the UEs relative to the base station. This includes the resources required for the relay stations. The required throughput is 0.4 Mbps. The distances where the three lines from the UEs linked to the relay stations cross the line of the UE directly linked to the bases station are the critical distances.

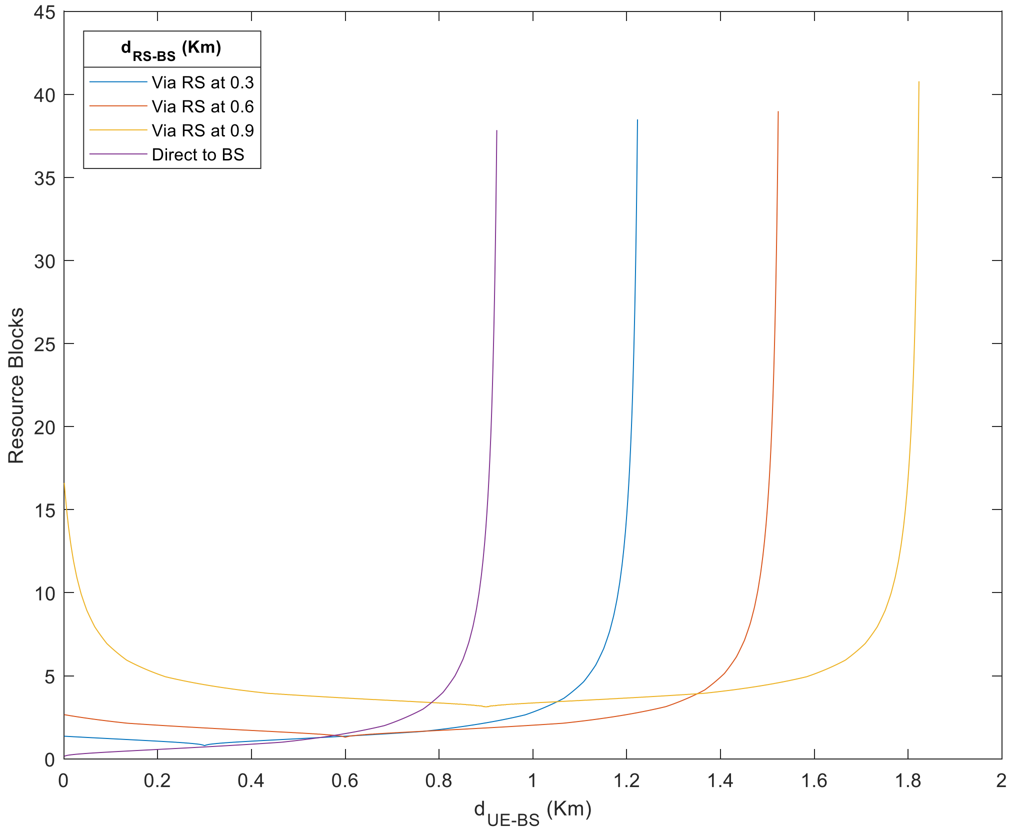

Figure 4-6. Critical distance.

Seen from the relay stations at 0.3, 0.6 and 0.9 km, the lines which indicate the total resource usage are symmetrical in both directions. The farther away the relay station is located from the base station, the farther the critical distance will lie. As the distance of the relay station to the base station increases it will require more resources to forward the data from de UE. In Figure 4-6 this is clearly shown for the relay station located at 0.9 kilometer. Regardless of the distance from the UE to the relay station and the necessary resources needed for that link there is always a fixed amount of resources needed ($\approx 4\ RB$) by the relay station to support this UE. If the relay station is located very far from the relay station it can be the case that the line for the UE linked to a relay station and the line for the UE directly linked to the base station do not cross each other anymore. This means that there is an area where a UE is not able to communicate to the network. This area is called a skip zone or silent zone.

As stated before the transmit power of the relay station is also of influence on the critical distance. Figure 4-7 shows the critical distance for a 0.4 Mbps link via a relay station located at 0.5 kilometer from the base station. Here we used various transmission powers for the relay station to examine the effect on the critical distance. The transmission power of the UE is 1 Watt.

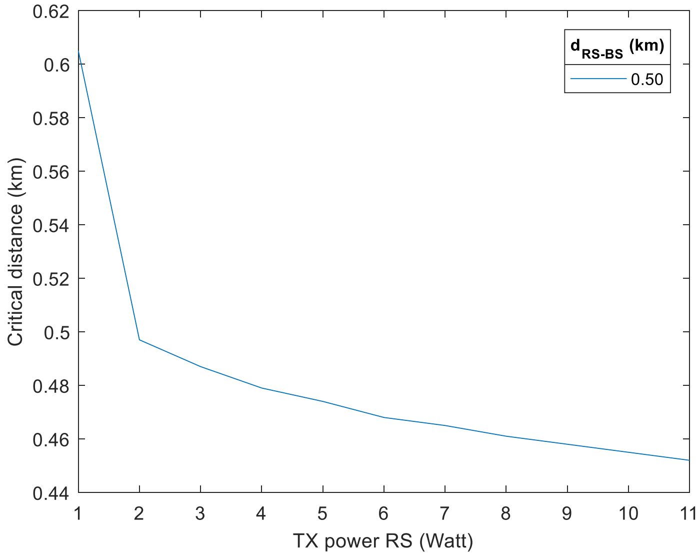

Figure 4-7. Influence transmit power Relay station on critical distance.

What is noticeable is that there is a rapid decline of the critical distance at first, but this decline decreases as transmission power of the relay station further increases. A reason for this is that when the transmission power of the relay station is increased, this reduces the amount of resources needed for the link RS-BS. But when the amount of resources required for the RS-BS link is reduced, it also reduces the effect of further increasing the transmission power as the number of required resources is small already. Using a transmission power of 2 Watt for the relay station, which is double the transmission power of the UE, is giving a significant reduction of the critical distance. This is also observed, to a greater and lesser extent, using other throughputs, other transmit powers for the UEs and distances for the relay station.

4.4. Path loss model

In the precious sections a path loss model is used assuming a fixed antenna height. For examining the topics discussed in these sections that path loss model was sufficient. However, for the evaluation of our proposed resource allocation algorithm we want to adjust the antenna height of the relay station. For this, we will use the path loss model from [30]. This model is applicable for urban and suburban areas. The path loss is calculated using the following formula:

$$L = 40(1 - 4 * 10^{-3}\Delta h_b)log_{10}(d) - 18log_{10}(\Delta h_b) + 21log_{10}(f) + 80\ dB \qquad (4.11)$$

Where d is the distance from the UE to the base station in kilometers, f is the carrier frequency set to 2000 MHz and Δh_b is the base station antenna height measured from rooftop level in meters.

For cellular links Δh_b is set to 15 meter. Entering this value in formula (4.11) gives:

$$L(d) = 128.1 + 37.6log_{10}(d) \tag{4.12}$$

For D2D links to the relay station there are two path loss values. One for when the relay station is located at a fixed position and a small antenna mast can be put down. The second path loss value is used for the situation where the relay station is mobile, moving through an area, and the antenna is at vehicle height. For the path loss value for the small antenna mast links Δh_b is set to 10 meter giving:

$$L(d) = 131.3 + 38.4log_{10}(d) \tag{4.13}$$

For the mobile situation Δh_b is set to 0 meter giving:

$$L(d) = 148 + 40log_{10}(d) \tag{4.14}$$

Although this value for Δh_b is debatable, this way of using of the path loss model for D2D application is also applied in other publications like [31].

5. Proposed resource allocation algorithm

In this chapter we want to answer the research question: What can be a suitable resource allocation algorithm for meeting the video service requirements and leading to high network efficiency? To formulate an answer to this question first the challenge in assigning resources in a scenario with multiple video streams is described. Next, a mathematical approach is drawn up where the scheduling of resources and the choice for the relay station is considered as a combinatorial optimization problem. Because this optimization problem is very complex to solve and mathematically comprehensive, a heuristic resource scheduling algorithm is proposed.

The following assumptions are made for developing the resource allocation algorithm:

1. Within a cluster, there is no reuse of resources.
2. All UEs in the network want to send their video streams to the Central Command Post via the base station. UEs can send their video streams directly to the base station or via the relay station to the base station.
3. The base station is aware of the locations of all relay stations and UEs in the network.

5.1. Radio resource assignment challenge

When designing an appropriate algorithm, we assume that the base station has knowledge of all the full instantaneous channel state information (CSI) of all cellular and D2D links. This assumption is realistic since the movements through the region are relatively slow, and the transfer of data is less time critical in comparison to V2V communication for example, where decisions have to be made on a per millisecond basis.

In the previous chapter it is shown that when a higher data rate is required this requires more than a proportionate number of resource blocks. For the UEs directly linked to the base station, that's no problem because the number of resource blocks calculated is also what is needed to send their video streams. For the UEs who want to send their video streams via the relay station, this is different. For the part of the UE to the relay station, the same applies to the UEs that are connected to the base station. However, the challenge lies with the part of the relay station to the base station. As long as only one UE wants to send its video stream via the relay station to the base station, the calculation as in Formula (4.7) will suffice. But when multiple UEs want to use the relay station, which is plausible, then this way of determining the required number of resources does not apply anymore. The number of resource blocks required must then be calculated on the sum of the data rates of the UEs that are linked to the relay station as given in Formula (4.9). This is schematically shown in Figure 5-1.

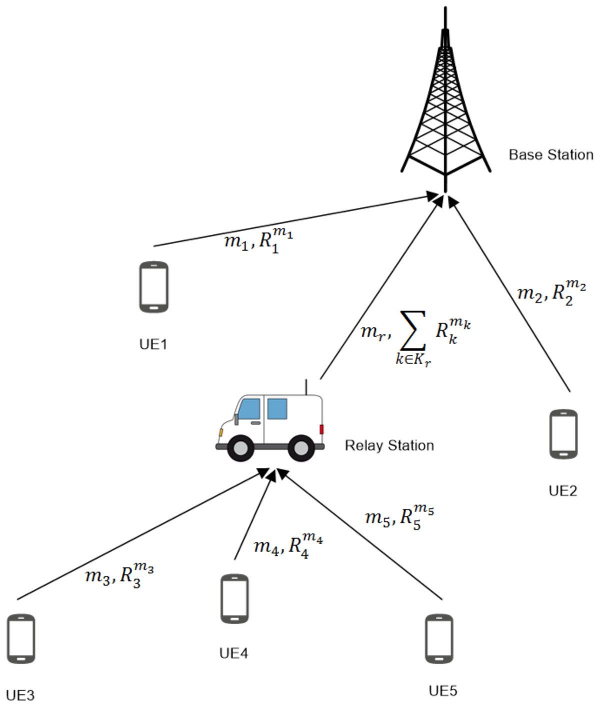

Figure 5-1. Resource distribution in a D2D enabled LTE network with relay station.

Here m_x, $(x = 1 - 5)$ is the amount of resource blocks assigned to a particular UE x and $R_x^{m_x}$ is the data rate that the UE can reach using m_x resource blocks. K_r is the set of UEs connected to the relay station. The number of resource blocks needed by the relay station to support these UEs is indicated by m_r and depends on the summation of their data rates $(R_3^{m_3} + R_4^{m_4} + R_5^{m_5})$. Thus, the number of resource blocks required for the link from the relay station to the base station is only known after determining which UEs will send their video stream via the relay and at which data rate. However, this is only clear after determining how many resource blocks are available for a UE and what data rate it needs or can reach. The amount of resources available for the UEs and needed for the relay station depend on each other.

The challenge is therefore to determine for all UEs how many resource blocks they are assigned and whether they send their video stream directly to the base station or through the relay station. This depends on the data rates to be obtained, the distance from the UEs to the base station and relay station and whether it is more efficient, with respect to spectral resources, to send directly to the base station or via the relay station. All allocations of resources and route choices of all UEs should be considered in conjunction, which makes it very difficult.

5.2. Optimal resource allocation

In this section, we want to find an optimal mathematical solution for the resource allocation challenge described in the previous section. To this end, we will provide a solution for the case of one video quality level in Subsection 5.2.1 which we will extent in Subsection 5.2.2 for using multiple video quality levels and a group of priority and non-priority class users.

It is important that UEs which are located farther away from the base station also get enough resources to reach a certain target rate. So there has to be a certain level of fairness in assigning resources to the UEs. It is explicitly not the intention to maximize the overall data rate like in opportunistic scheduling. The situation that may arise in that case is that UEs that are located near the base station are allocated more resources because they will achieve high data rates with relatively few resources, unlike UEs which are further away from the base station, that will need more resources for a lower data rate [32]. With opportunistic schedulers, it may be that UEs who are located farther away will hardly be allocated any resources, which for PSS usage is unacceptable.

5.2.1. Single video quality level

For the case of one video quality level our goal is to have as many UEs as possible to achieve this quality level, while using a minimal amount of resources. This quality level corresponds to a required data rate, which is indicated by the target rate R_T.

We consider a network with a single cluster (as in Figure 5-1), with K users (UEs) ($1 \leq k \leq K$), a relay station r and a base station b. A user can only be connected to either the relay station or base station at one point in time, so they are divided into two groups. The first group of users, indicated with K_r, send their video streams through the relay station. The second group of users, indicated with K_b, send their video streams directly to the base station. The frequency spectrum is divided into resource blocks (RBs). The total number of resource blocks available is M ($1 \leq m \leq M$).

The achievable rate $R_k^{m_k}$ in bits per second (bps) for the kth user using m_k resource blocks per unit of time is given by:

$$R_k^{m_k} = \begin{cases} \min[(m_k * 180Khz) * \sigma * log2\left(1 + SINR_{k,r}^{m_k}\right), \\ \left((m_r * 180Khz) * \sigma * log2\left(1 + SINR_{r,b}^{m_r}\right)\right) / |K_r|]\,, via\ relay\ station \\ (m_k * 180Khz) * \sigma * log2\left(1 + SINR_{k,b}^{m_k}\right), direct\ to\ base\ station \end{cases} \quad (5.1)$$

Where $|K_r|$ is the number of users in group K_r and m_r is the amount of resource blocks used by the relay station per unit of time to send $|K_r|$ video streams to the base station. $SINR_{k,r}^{m_k}$, $SINR_{r,b}^{m_r}$ and $SINR_{k,b}^{m_k}$ are calculated according to Formula (4.3), depending on the distances k-r, r-b and k-b, respectively, and the number of resource blocks used on the particular link.

The received power must be higher than the minimum threshold so that the receiver can reconstruct the data from the received signal. On the other hand, the power of the receive signal must not be too high so that it starts to interfere with carriers of adjacent resource blocks. The minimum receive power at the relay station and the base station per resource block is described by P_{min}^{rec}. Because we are using fixed transmission power for all UEs and the relay station P_{min}^{rec} limits the number of resource blocks that can be used for a certain link. The maximum receive power at the relay station and the base station per resource block is described by P_{max}^{rec} which limits the fixed transmission power. This latter receive power constraint is less important given the low transmission powers used.

The total number of resources used must not exceed the total available resources. When the collection of users connected to the base station is notated as K_b and the users connected to the relay station as K_r the resources constraint can be described as:

$$\sum_{k \in K_b} m_{k,b} + \sum_{k \in K_r} m_{k,r} + m_{r,b} \leq M \tag{5.2}$$

Here $m_{k,b}$ indicates the amount of resources used by a user connected to the base station, $m_{k,r}$ the amount of resources used by users connected to the relay station on that link, and $m_{r,b}$is the amount of resources used by the relay station to support the total rate of the video streams routed via the relay station.

The optimization problem can be formulated using the objective function below. Here $\mathcal{K}$ is the set of subsets K_b and K_r $\{1, \dots, k\}$ and $\mathrm{I}_{\{R_k^{m_k} \geq R_T\}}$ is the indicator function which equals 1 if $R_k^{m_k} \geq R_T$ and 0 otherwise.

$$\max_{\substack{K_r \in \mathcal{K} \\ (m_1, \dots, m_k, m_r)}} \sum_{k=1}^{K} \mathrm{I}_{\{R_k^{m_k} \geq R_T\}} \tag{5.3}$$

subject to:

C1: $\sum_{k \in k_b} m_{k,b} + \sum_{k \in k_r} m_{k,r} + m_{r,b} \leq M$
C2: $P_k^m \geq P_{min}^{rec}, \forall m, k$
C3: $P_r^m \geq P_{min}^{rec}, \forall m, r$
C4: $P_k^m \leq P_{max}^{rec}, \forall m, k$
C5: $P_r^m \leq P_{max}^{rec}, \forall m, r$

Constraint C1 describes the resource block allocation constraint. Constraint C2 till C5 are the minimum and maximum power constraints.

The maximization in (5.3) results in a list with possible UE distributions across the groups K_r and K_b and associated resource allocations $(m_1, \dots, m_k, m_r)$. Selecting the distribution that uses the least amount of resources in total is the solution where the maximum number of UEs have a data rate that is equal to or higher than the target rate while using a minimum amount of resources.

Reference [33] also provides a solution for resource allocation in a network that consists of UEs, relay stations and a receiver. In our case, the latter is the base station. The difference is that in [33] the network contains multiple relay stations and multiple transmission paths are used at the same time. In our research only one relay station is used and a UE can only use one transmission path at the same time (i.e. via the relay or direct to the base station). In reference [33] the optimisation problem is solved with a heuristic approach where they use equal power allocation across the subcarriers of a resource block. Because of this the power constraint is ignored which simplifies the original problem. This solution is not suitable to our research. We use fixed transmission power for all UEs and the relay station so if a transmitting entity is closer to its intended receiver it will have more power per resource block and ultimately send more data within a resource block. Also, it is indicated that when it is not possible for all UEs to meet their rate requirements that UEs are given a rate close to their requirement. In our research this would be a waste of resources because when the target rate for a certain quality video stream is not met the target rate is lowered to a lower video quality level. As a result of the approach in [33] the UE is receiving way too many resources for the new target rate. The excess of resources assigned to a particular UE can be better assigned to other UEs, which will allow as many UEs as possible to achieve the highest possible target rate.

5.2.2. Multiple video quality levels

When using one video quality level it will likely occur that some UEs will meet the target rate and are able to send their video streams, while others will not meet the target rate and have no video connection at all. For public safety applications this is totally unacceptable. Therefore several video quality levels are applied, as described in chapter 3. For using multiple video quality levels our first goal is to have as many UEs as possible to achieve the minimal video quality level. The second goal is to have as many UEs as possible to achieve a higher video quality level. So "more UEs having less increase in video quality" is preferable to "less UEs having much increase in video quality". The last goal is to achieve the prior goals while using a minimal amount of resources.

The multiple video quality levels correspond to required data rates as indicated in section 3.2, which are denoted as $R_{T(i)}$. Here the i ($1 \leq i \leq I$) indicates which target rate it concerns. Determining the achievable rate $R_k^{m_k}$ is harder compared to the single video quality case since it is likely that the UEs in group K_r will have different quality video streams and so do not use an equal part of the relays station's resources. As a consequence, the UEs connected to the relay station get a proportionate share of the throughput of the relay station which depends on the throughput of the link from the UE to the relay station, as shown in Formula (5.6). The rate of a UE directly linked to the base station using m_k resource blocks, indicated with $R^{m_k}_{k,UE-BS}$, is calculated according to Formula (5.4). The rate for the link from a UE to the relay station, indicated by $R^{m_k}_{k,UE-RS}$, is calculated according to Formula (5.5).

$$R^{m_k}_{k,UE-BS} = (m_k * 180Khz) * \sigma * log2\left(1 + SINR^{m_k}_{k,b}\right) \tag{5.4}$$

$$R^{m_k}_{k,UE-RS} = (m_k * 180Khz) * \sigma * log2\left(1 + SINR^{m_k}_{k,r}\right) \tag{5.5}$$

$$R^{m_k}_{k,RS-BS} = \left((m_r * 180Khz) * \sigma * log2\left(1 + SINR^{m_r}_{r,b}\right)\right) * \frac{R^{m_k}_{k,UE-RS}}{\sum_{i \in K_r} R^{m_k}_{k,UE-RS}} \tag{5.6}$$

The achievable rate $R^{m_k}_k$ in bits per second (bps) for the kth user using m_k resource blocks per unit of time is then given by:

$$R^{m_k}_k = \begin{cases} \min\left[R^{M_k}_{k,UE-RS}, R^{M_k}_{k,RS-BS}\right], via\ relay\ station \\ R^{M_k}_{k,UE-BS}, direct\ to\ base\ station \end{cases} \tag{5.7}$$

Although all first responders contribute to the same end status, not all of them are equally interesting to be followed by the Central Command Post using a high quality video stream. By applying priority amongst the first responders, scarce spectral resources can be assigned to the most interesting cases. Therefore we define two types of users: a priority class k_1 and a priority class k_2. The users of priority class k_1 are assigned resources to achieve the highest possible data rate and are thus able to send a video stream with the highest quality possible. Priority class k_2 users must be assigned enough resources to achieve the minimum target rate. If enough resources are available priority class k_2 users can even achieve a higher target rate.

This optimization problem is solved in three steps. First, it must be determined which resource allocations and route combinations result in the priority UEs achieving the maximum target rate and the non-priority UEs a throughput equal to or higher than the minimum target rate. Then it has to be determined which of these combinations give as many non-priority UEs as possible a high as possible target rate. Finally, the combination that uses the least amount of resources is chosen as the optimal resource scheduling. The constraints as described for the case with one video quality level in Subsection 5.2.1 also apply here.

The optimization problem described above is computational very hard to solve. Going through all possible resource assignments and choose the optimal one, with 7 video quality levels and 2 possible routes to the base station, gives K^{14} possible combinations. The time required to find the optimal resource allocation is so large that this is not a practical solution. Therefore a heuristic approach will be discussed in the next section.

5.3. Heuristic resource allocation algorithm

Given the locations of the first responders and the relay station, we are looking for a suitable low complexity resource allocation algorithm so that all priority and non-priority first responders are able to send video streams with the highest resolution and frame rate possible. As discussed in the previous section, the optimisation problem is hard to solve mathematically and very extensive. Therefore this algorithm is a heuristic numerical approach to find a near optimal resource allocation.

The proposed algorithm works according to critical distance principle. The algorithm bases its choices on the calculations regarding the required resources and route to the base station (i.e. direct or via relay) for a single UE in isolation. Then, when for all UEs choices have been made, corrections are carried out. These corrections include adjusting the throughput, which result in a higher or lower quality video stream, or choosing the route directly to the relay instead of via the relay station. The algorithm takes into account the presence of priority UEs and non-priority UEs. The reason for using priority levels is that the amount of available resources is very limited, which means that video quality per user can be relatively low after allocation of resources. By forming priority and non-priority groups, the video quality of the most interesting cases can remain as high as possible at the expense of the video quality of the other users.

A high level overview of the algorithm is shown by the flowchart in Figure 5-2 with a global description of the algorithm below. Next, a more detailed description of all steps of the algorithm will be discussed.

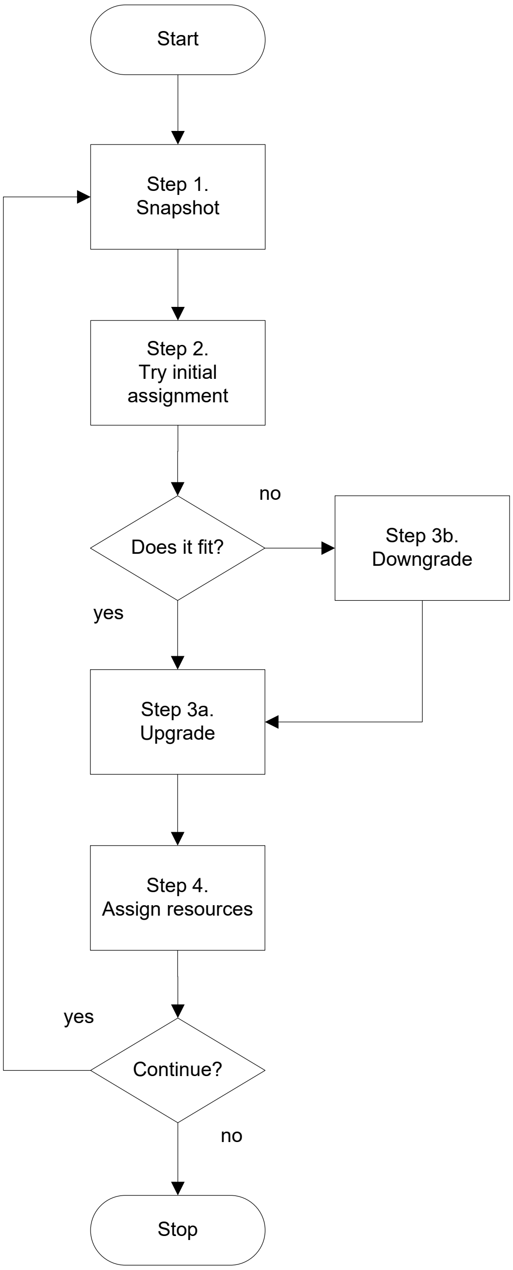

Figure 5-2. Proposed resource allocation algorithm.

The algorithm starts in Step 1 by taking a snapshot of the locations of the relay station and the UEs. Based on this information the base station can determine which UEs can support which data rate (see Table 3.2) and if it is more efficient, in terms of resources, to send the video stream directly to the base station or via the relay station. This is done for all UEs individually not taking into account the influence of other UEs who also want to send their video stream via the relay station as well. Step 2 attempts an initial assignment of resource blocks, giving the priority UEs the highest achievable data rate and the non-priority UEs a lower but acceptable data rate. If this initial assignment does fit the total amount of available resource blocks it is highly likely that there are some unused resources left. Since the priority UEs already have the highest achievable data rate, the remaining resources are used in Step 3a to try to upgrade the non-priority UEs to a higher data rate. If the initial assignment setup does not fit the total amount of available resource blocks then all UEs will be downgraded in Step 3b according to a specific order. When after downgrading the adjusted assignment of resources does fit the available resource blocks it is likely that there are unused resources left. Therefore it is checked whether an upgrade can take place. After each change in resource allocation in Step 3a and Step 3b, an efficiency check is performed to determine if it is more efficient to reallocate UEs from the relay station to the base station. Then in Step 4 the resources are actually assigned to the UEs and relay station so that they can send their data. As long as there are PSS officials active in the area the situation continues and after a specified time the algorithm starts with Step 1 again. In the following a more detailed description of the heuristic algorithm is given.

Step 1. Snapshot

A snapshot is made of the locations of the UEs and the relay station. The assumption here is that the base station is able to receive geolocation information and/or channel quality information from all users and the relay station. Using this location information the base station can determine for all supported data rates if the UEs can connect directly to the base station and can connect to the base station via the relay station. For all routes and data rates the amount of resources needed to support all video resolutions and framerates are calculated. Based on these overviews it can be determined whether it is more efficient, in terms of resources, to send directly to the base station instead of via the relay station or vice versa for a given resolution and frame rate.

Step 2. Initial resource assignment

To come up with a resource allocation scheme fast, it is chosen to assign the priority UEs the highest video quality and the non-priority UEs a video quality which just meet the requirements. The idea behind this is that when a relay station is deployed the available resources for non-priority UEs will probably be limited. Therefore it is likely that not all UEs will achieve the highest video quality.

It is examined if all non-priority UEs are able to support 4CIF 10 frames (0.8 Mbps) and the priority UE is able to support 4CIF 25 frames (2 Mbps). If the maximum resolution and frame rate for a UE is lower than the one chosen for the initial resource assignment, the lower resolution and frame rate is set for that user. Then an efficiency check, which will be discussed at the end of this section, is performed and it is checked whether the basic setup fits the amount of available resources.

Step 3a. Upgrade non-priority UEs

If the base station and relay station can support the allocation of Step 2 it is likely that there are some unused resources left. Since the priority UE already has its maximum quality video link assigned, the unused resources can be appointed to the non-priority UEs. A schematic overview of the upgrading process is given in Figure 5-3. Raising the video quality of the non-priority UEs is done one step at a time and in ascending order of required RBs, so that as many UEs as possible can be provided with a higher frame rate or resolution. So first all non-priority UEs have to be assigned 4CIF 15 frames before allocating 4CIF 25 frames. The exception to this is that when a particular UE cannot handle a higher data rate due to its link quality. Then other UEs will be further upgraded.

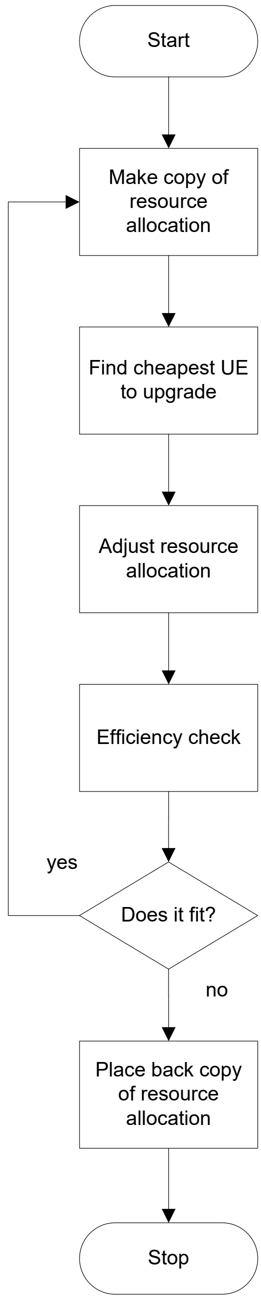

Figure 5-3. Upgrading process of Step 3a of the heuristic resource allocation algorithm.

Step 3b. Downgrade UEs

If the system cannot support the basic setup, UEs will have to be downgraded to lower resolution and/or frame rates. The schematic overview of the downgrading process is shown in Figure 5-4. The downgrading process starts by lowering the frame rate for non-priority UEs step-by-step starting with the UE that releases the most resources. By doing so we try to keep the number of UEs that need to be downgraded as low as possible.

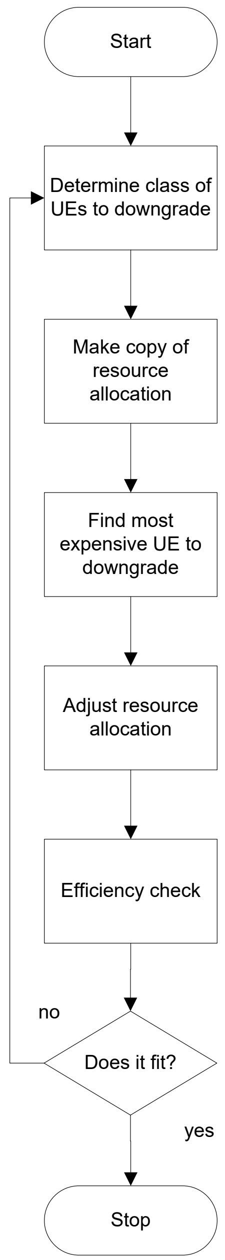

Figure 5-4. Downgrading process of Step 3b of the heuristic resource allocation algorithm.

If downgrading the non-priority UEs is not enough, the priority UEs frame rate is scaled back to 10 frames per second. If this is still not sufficient, then the resolution of the non-priority UEs is downgraded to CIF. Again starting with the UE releasing the most resources. This downgrading mechanism attempts to provide the priority UEs with the highest resolution and framing rate possible. However, this must not lead to a situation where a non-priority UE is not able to send video to the Central Command Post anymore. A requirement is that non-priority UEs must have a video connection as long as the link quality allows them to send video streams at the lowest resolution and frame rate. Table 5.1 shows the order of the downgrading process. The P indicates the path of the priority UEs. The N indicates the path of the non-priority UEs. When a UE achieves a lower maximum resolution or frame rate than the other UEs, first these other UEs are downgraded before the former UE also participates in the downgrading process.

Quality/step	1	2	3	4	5	6	7	8	9	10	11
4CIF 25f	P	P									
4CIF 15f			P								
4CIF 10f	N			P	P	P	P				
4CIF 5f		N	N	N				P			
CIF 15f					N				P		
CIF 10f						N				P	
CIF 5f							N	N	N	N	P/N

Table 5.1. Order of the downgrading process of Step 3b of the heuristic resource allocation algorithm.

After downgrading a certain UE it is likely that there are some unused resources left. They can be assigned to non-priority UEs via the upgrading process.

Efficiency check

The choice for a UE to send its video stream directly to the base station or via the relay station is based on the amount of resources used for both routes in Step 1. This calculation only takes into account the resources needed to support the video stream of one particular UE in isolation. However, it is very likely that for some other UEs it is also more efficient to use the relay station. Therefore the amount of resources needed by the relay station to support all those users is higher than the sum of the resources needed for the individual streams as explained in Section 4.2. For example if the relay station is located at 0.5 km and has to support one user sending a video with 4CIF resolution at a frame rate of 15 FPS (1.2 Mbps) the relay uses 1.722 resource blocks per second. However if the relay has to support 2 UEs for the same video quality it uses 3.897 resource blocks per second, which is more than twice the amount of RBs to support one video stream. Therefore, it can be more efficient to let a UE, for who on an individual basis it was determined to use the relay, send its video stream directly to the base station.

The efficiency check analyses if a UE who is intended to send its video stream via the relay station can send its video directly to the base station and by doing so reduces the total amount of resources used. To increase the chance of success and to try to switch as much UEs as possible the check starts with the UE who, based on the calculations in Step 1, requires the least additional resources to transfer from the relay station to a direct link to the base station. If after the switch of the first UE it appears that fewer resources are used and the total amount of resources used are equal or less than the total amount of resources available, the following UE is transferred and so forth. If after the switch of a UE it appears that more resources are used than before, then this switch will be made undone and the efficiency check is ended. If before and after the switch of a UE the required amount of resources do not fit the total amount of available resources, the efficiency check is ended as well. The reason for this is that it is unknown what the effect of the switch of that particular UE has on the amount of required resources.

6. Numerical results

In this chapter the heuristic uplink resource allocation algorithm is evaluated on performance. This is done by comparing the performance of the heuristic algorithm to the most optimal resource scheduling. We evaluate the algorithm by applying various different settings to get insight into the behaviour of the algorithm in terms of the resulting throughputs and efficiency.

6.1. Basic scenario

The evaluation is performed using a single LTE base station, one relay station and 6 UEs, as shown Figure 6-1. The test scenario, described in more detail below, offers combinations between operating in a very small area and widespread across a larger area, in an area with good link quality to the base station as well in an area with poor link conditions. Calculating the optimal resource scheduling requires extensive calculations. Calculating the optimal scheduling for more than 6 UEs would take a very long time. Therefore the number of UEs for calculating the optimal scheme is limited to 6, being 1 priority UE and 5 non-priority UEs.

The relay station in Figure 6-1 is placed on a distance d_{RS-BS} from the base station, starting at 0.1 km which will be varied. 6 UEs are randomly dropped in an area around the relay station with a radius d_{UE-RS} starting at 0.1 km which will also be varied. Then the measurements are performed. The random dropping of UEs and taking measurements is performed 50 times to get representative results. Then the area where the UEs are dropped is enlarged to an area with a radius of 0.2 km and new measurements are performed, and so on until the area reaches a radius of 1 km. After the radius of the area where the UEs are dropped reaches 1 km the relay station is moved 0.1 km further away from the base station and the dropping of UEs and taking measurements start over again. Unless indicated otherwise, it is assumed that the relay station uses a (small) antenna mast to lower the path loss towards the UEs and increase the SINR. The transmission power of the UEs is 1 Watt and the transmission power of the relay station is 2 Watt. The maximum number of available resource blocks at any given moment is 50.

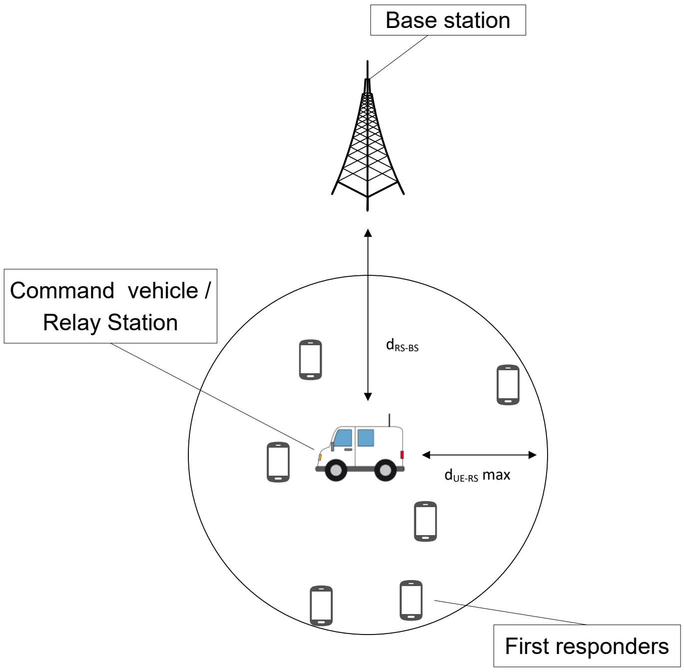

Figure 6-1. Scenario setup.

6.2. Scenario variations

In addition to the basic scenario, which we will use as a reference, we want to vary some of the key parameters to further examine the behaviour of the heuristic algorithm. These variants consist of varying the transmit power of the relay station, varying the antenna height of the relay station and increasing the number of UEs. In the subsections below, these variants are further explained.

6.2.1. Increased transmit power of the relay station

In Section 4.3 it is argued that for the relay station a transmit power of twice the transmit power of the UEs gives a fair decrease of the critical distance, while further increasing of the transmit power of the relay station reduces this effect. That measurement was performed with a single user connected to the relay station. Having more UEs in the area would suggest that when the transmit power of the relay station is increased more UEs will want to send via the relay station.

6.2.2. Increased number of UEs in the area.

As stated in Section 6.1 calculations for the optimal resource scheduling is limited to 6 UEs. However, to get an idea of how the heuristic algorithm performs with more UEs we let it run with 10 UEs without the optimal scheduling and compare it to previous measurements. This scenario uses 1 priority UE and 9 non-priority UEs.

6.2.3. Antenna of the relay station at vehicle height.

In the basic scenario a small antenna mast is used at the relay station to improve SINR of the UEs linked to it. In this variant a path loss model is used which implies that the antenna of the relay station is at vehicle height which in turn corresponds to a mobile relay station that moves through the area.

6.2.4. No relay station.

By allowing the UEs only to communicate with the base station insight is gained on two subjects. First, because there are no UEs attached to the relay station the efficiency check of the heuristic algorithm is not used. This makes it possible to only assess the performance of the upgrading and downgrading process. And secondly, by comparing the results for this scenario to the scenarios with relay station, the effective work area of the relay station can be observed. This is the area where the UEs benefit from deployment of the relay station.

6.3. Performance metrics

To measure the performance of the heuristic algorithm two metrics are used, failure rate and efficiency.

6.3.1. Failure rate

This is the number of times a UE does not meet the minimal required resolution and frame rate expressed in percentage calculated over 50 iterations. This failure rate is considered for both categories of users, priority users and non-priority users. If one or more non-priority UEs do not meet their requirement, this is seen as one failure. By comparing the failure rate of the heuristic algorithm to the optimal scheduling, it is possible to examine how effectively the heuristic algorithm works.

6.3.2. Efficiency

The efficiency is determined by the average amount of resources needed per Megabit per second which is calculated using the total throughput generated by all UEs combined and the total amount of resources needed to achieve this throughput. By comparing the efficiency of the heuristic algorithm to the optimal scheduling makes it possible to examine how efficient the heuristic algorithm works.

6.4. Results

This section discusses the results of all scenarios discussed in Section 6.2. First, in Subsection 6.4.1, the results for the basic scenario are discussed. Then in Subsection 6.4.2 the results for scenario with the increased transmission power of the relay station is discussed. Subsection 6.4.3 discusses the results for the scenario where the number of UEs is increased. Next the results for the scenario where the antenna of the relay station is at vehicle height is discussed in Subsection 6.4.4. Finally, Subsection 6.4.5 discusses the results for the scenario without relay station.

6.4.1. Results for the basic scenario

Figure 6-2 to Figure 6-5 show the results for the basic scenario described in section 6.1. The results are displayed using 3D bar charts. The x-axis represents the radius of the area around the relay station in which the UEs are located, denoted by $d_{UE-RS}\ max$. The y-axis represents the distance from the relay station to the base station, denoted by d_{RS-BS}. For Figure 6-2 and Figure 6-3 the z-direction shows the failure rate for the priority UE and the non-priority UEs respectively. In Figure 6-4 the z-direction represents the average amount of resources required per Mbps and in Figure 6-5 the z-direction shows how many UEs on average make use of the relay station. On each point on the 3D bar charts of two values will be displayed. One for the heuristic algorithm and one for the optimal scheduling.

Remarkable is the sharp distance of the relay station in Figure 6-2 from where the priority UE is not reaching its requirement anymore at 1.4 kilometer. This is two hundred meters closer than where the priority UE according to the optimal scheduling is no longer meeting its requirement. The same applies to the non-priority UEs in Figure 6-3 at a distance of 1.2 kilometer. However, when the relay station is located at 1.5 kilometer, the non-priority UEs mostly meet their requirements when they are distributed at close distance to the relay station. This is in contrast to the optimal scheduling. The reason for this is that the priority UE has a lower throughput for the heuristic algorithm, and so more resources are distributed over the non-priority UEs which will still meet their requirement.

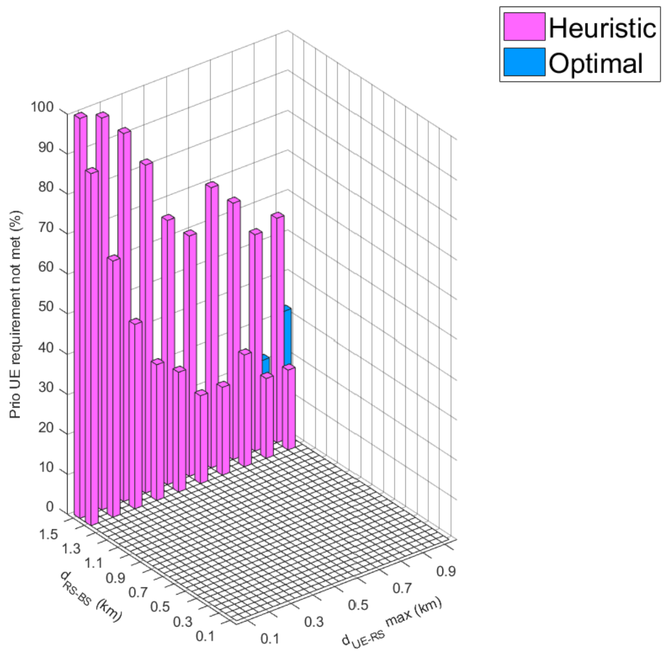

Figure 6-2. Number of times (in percentage) the priority UE is not meeting its requirement for the basic scenario.

That for the heuristic algorithm the priority UE often does not meet its requirement when distance $d_{UE-RS}\ max$ is small while d_{RS-BS} is large (1.3 and 1.5 km), is because for all UEs and all data rates it is more efficient to use the relay station. For the relay station it is not possible to support all UEs because the amount of resources needed is huge. The efficiency check stops immediately after 1 UE is transferred to the base station as the resources needed still does not fit the available resources. Only after all UEs have been downgraded multiple times and trying to reallocate UEs to the base station, will the amount of resources required fit within the amount of available resources. UEs can then be upgraded. However, the priority UE does not participate in the upgrading process because the assumption was that it would already have the maximum throughput. In this case, however, this is not correct.

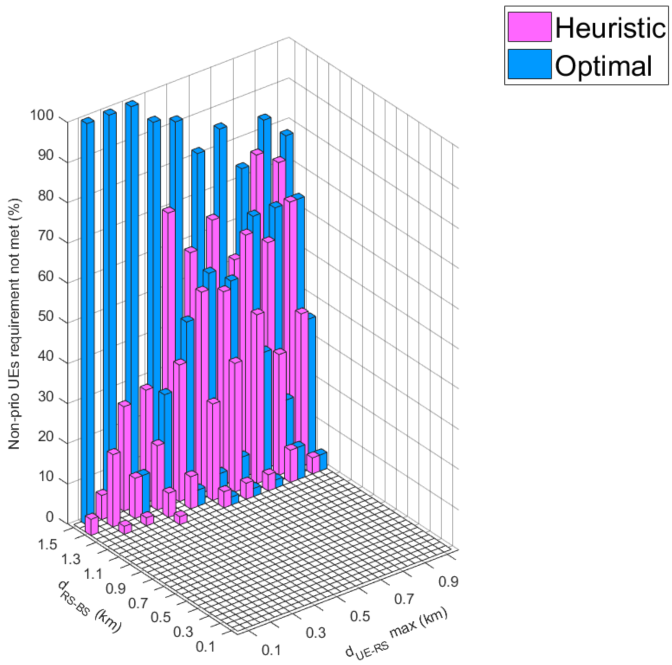

Figure 6-3. Number of times (in percentage) the non-priority UEs are not meeting their requirements for the basic scenario.

The resource usage is equal for the heuristic and optimal scheduling at close distance to the base station, as shown in Figure 6-4. In this area, The UEs are often directly linked to the base station. Then, from 0.7 kilometers onwards, the resource usage begins to differ because the heuristic algorithm is less accurate than optimal scheduling. What stands out is that at $d_{RS-BS} = 1.5\ km$ and $d_{UE-RS} = 0.1\ km$ the heuristic algorithm needs fewer resources than the optimal scheme per Mbps. As stated before, this is because the priority UE for the heuristic algorithm is downgraded, causing the non-priority UEs to reach a higher throughput, which results in a higher overall throughput than that of the optimal scheduling. As a result, the average number of resources per Mbps is lower than that of the optimal scheduling.

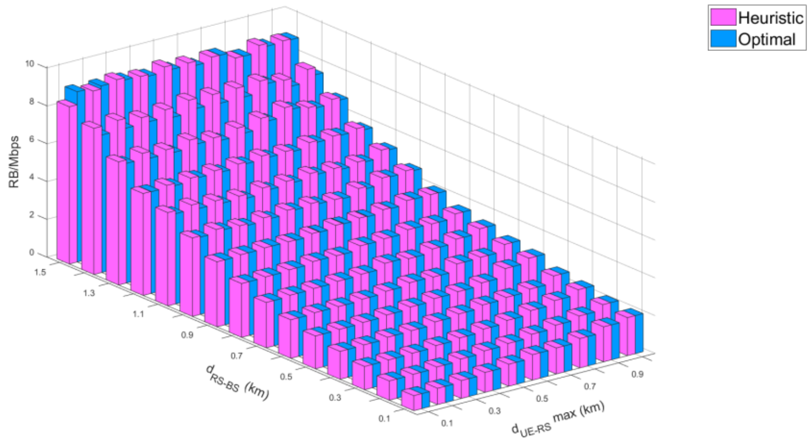

Figure 6-4. Average resource requirement per Mbps for the basic scenario.

The number of UEs that uses the relay station in limited. The heuristic algorithm tends to use the relay more often than the optimal scheduling does. When distances to the relay station and base station increase, usage increases but overall the use remains quite limited. When the distances from the UEs to the base station are short, a direct link is preferred instead of via the relay station.

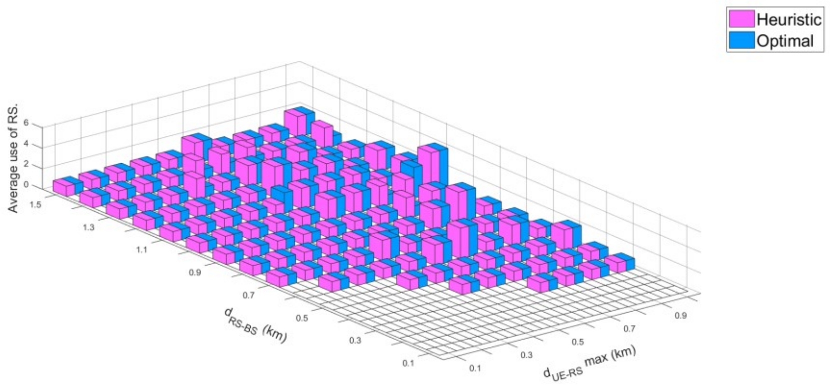

Figure 6-5. Average use of relay station for the basic scenario.

The main reason for differences in the resource allocation of the heuristic algorithm compared to the optimal scheme is the efficiency check. The goal of the efficiency check is to see if resources can be scheduled more efficient by routing UEs direct to the base station instead of via the relay station. When a certain resource allocation scheme does not fit the total available resources and after rescheduling a UE from the relay station to the base station the required amount of resources still exceeds the available resources, the algorithm is unable to determine if this rescheduling is more efficient. The efficiency check is ended and the downgrading process is started to lower the amount of required resources. However it could also have been the case that after a second UE was rescheduled directly to the base station, the amount of required resources would fit the available resources. Thus, no downgrade had to take place.

The reason for this is that calculations are stopped at the maximum number of available resource blocks, in this research 50. If we would let the calculations proceed until we reach the actual number of required resources before and after the rerouting, it can be the case that thousands of calculations are needed to find these values. This would severely increase the calculation time. As a result, the heuristic algorithm would be worthless as it would take too much time to come up with an allocation scheme.

An option would be to let the efficiency check continue when rescheduling a UE from the relay station to the base station still does not fit the available amount of resources. When after rescheduling multiple UEs the required amount of resources does fit the available amount of resources the efficiency check is stopped. If after rescheduling all the UEs to the base station the required amount of resources still does not fit the available resources, than the previous scheduling is put back. As can be seen in Figure 6-6 this way of performing the efficiency check shows a large improvement for the priority UEs for the distances 1.4 and 1.5 kilometer, however at shorter distances the priority UE regularly does not meet its requirement. Also, the non-priority UEs regularly do not meet their requirements at shorter distances as can be seen in Figure 6-7 which were not seen using the unadjusted efficiency check. The reason for this is that often UEs for which it is not beneficial to be rescheduled to the base station are rescheduled. Because there is no insight into the total amount of resources required when it exceeds the available resources, these scheduling errors remain unnoticed, resulting in a worse resource allocation scheme.

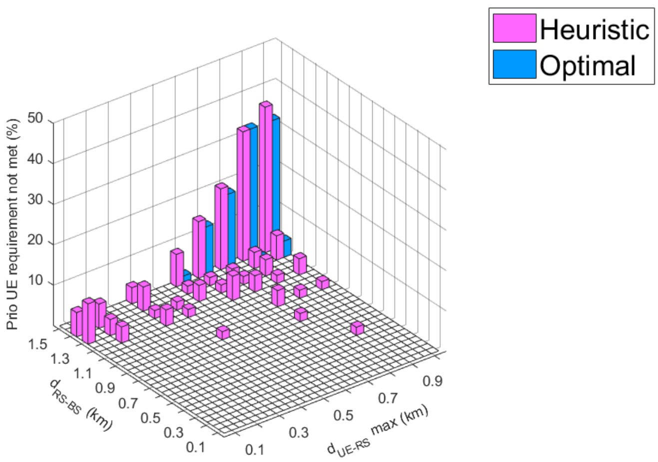

Figure 6-6. Number of times (in percentage) the priority UE is not meeting its requirement when the efficiency check is continued.

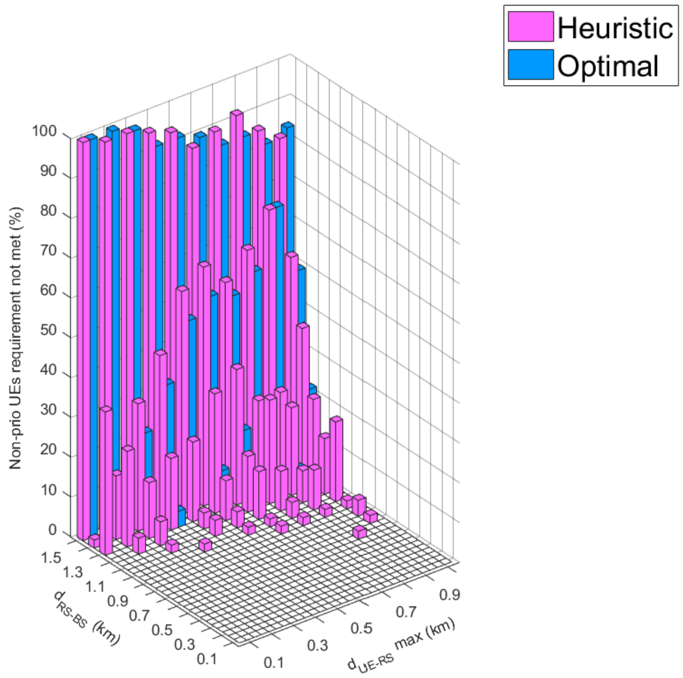

Figure 6-7. Number of times (in percentage) the non-priority UEs are not meeting their requirements when the efficiency check is continued.

The resource usage in Figure 6-8 also shows that this way of performing the efficiency check results in a less efficient resource allocation. Already at short distances differences in resource usage can be seen. UEs are linked directly to the base station while this is not the most efficient path. As can be seen in Figure 6-9 this method of performing the efficiency check makes too little use of the relay station for an optimal resource scheduling. As a result, UEs cannot obtain their maximum throughput. When the distance from the relay station to the base station increases, These errors in route choices have a greater impact on resource usage, which means that, in addition to UEs not meeting their requirements anymore, the average amount of resources needed per Mbps increases faster as well. At $d_{RS-BS} = 1.4\ and\ 1.5\ km$ this effect decreases, as the distance between the UEs and the base station has become so large that a direct link is no longer an option for the higher target rates. The number of UEs that use the relay station is then equal for the algorithm with the adjusted efficiency check and the optimal scheduling.

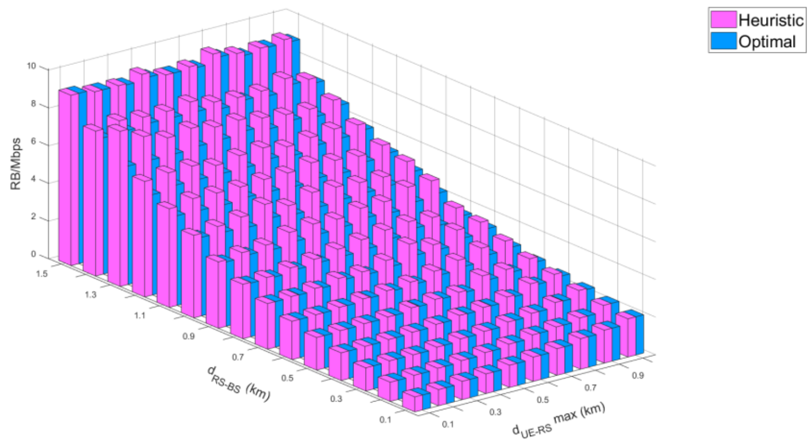

Figure 6-8. Average resource requirement per Mbps when the efficiency check is continued.

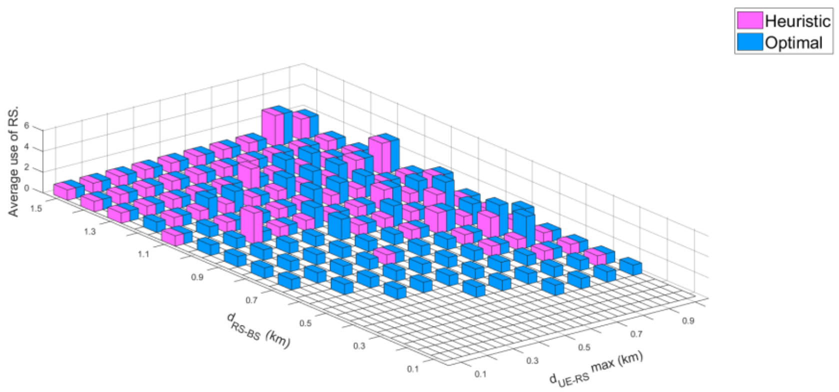

Figure 6-9. Average use of relay when the efficiency check is continued.

6.4.2. Increased transmit power of the relay station.

By increasing the transmit power of the relay station the SINR at the base station will increase, which means that it can be located further from the base station while maintaining the same maximum throughput. Figure 6-10 shows the result for the priority UE when the transmit power of the relay station is doubled from 2 Watt to 4 Watt. This result is quite similar to the result for 2Watt in Figure 6-2, only the maximum distance for the relay station where the priority UE is still meeting its requirement is enlarged to 1.5 kilometer. This increase in distance is very limited compared to the 1.3 kilometer for the 2Watt case. The difference in the maximum distance for the relay station between the heuristic algorithm and the optimal scheduling is also increased from 0.2 to 0.3 km.

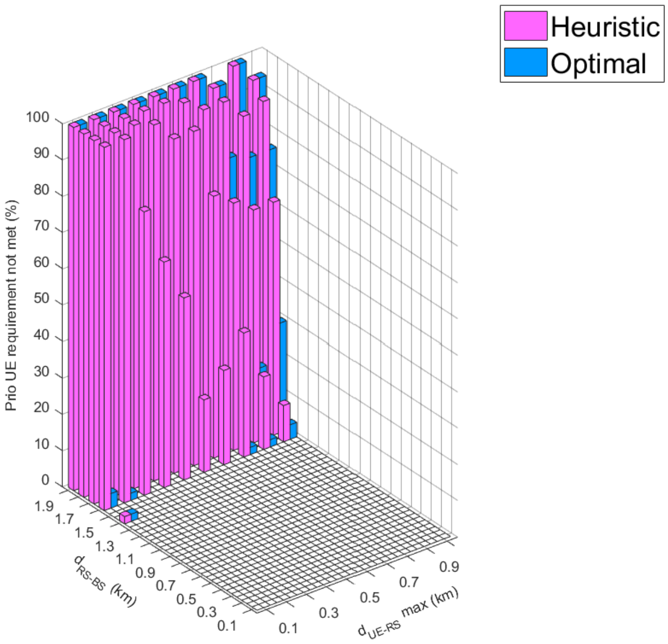

Figure 6-10. Number of times (in percentage) the priority UE is not meeting its requirement when the transmit power of the relay station is increased.

A comparison between Figure 6-3 and Figure 6-11 show a similar increase in maximum distance for the relay station of 0.2 kilometer for the non-priority UEs. However this comparison also shows that the effect of the priority UE not meeting its requirement causing non-priority UEs to meet their requirement decreases. This is well shown at $d_{RS-BS} = 1.6\ km$. The reason that this effect decreases is that for the non-priority UEs the transmit power is not increased and because of this they need more resources to send their video streams to the base station at 1.6 kilometer compared to 1.4 and 1.5 kilometer in Figure 6-3. As a result, the non-priority UEs more often do not meet their requirements anymore.

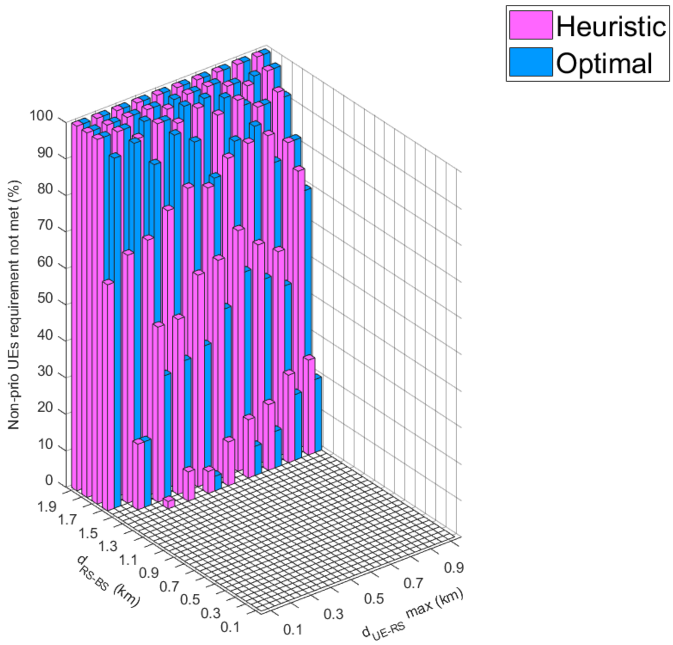

Figure 6-11. Number of times (in percentage) the non-priority UEs do not meet their requirements when the transmit power of the relay station is increased.

As can be seen in Figure 6-12, the heuristic algorithm has a much higher resource usage for the distances d_{RS-BS} = 1.6 till 1.8 kilometer compared to the optimal scheduling. This higher resource usage can be explained by the fact that the priority UE no longer meets its requirement due to the issue with the efficiency check, as described in the previous subsection. This provides additional resources for the non-priority UEs. Since not all UEs can be supported by the relay station, also UEs who are directly connected to the base station are upgraded, which cost much more resources per Mbps, considering the distance.

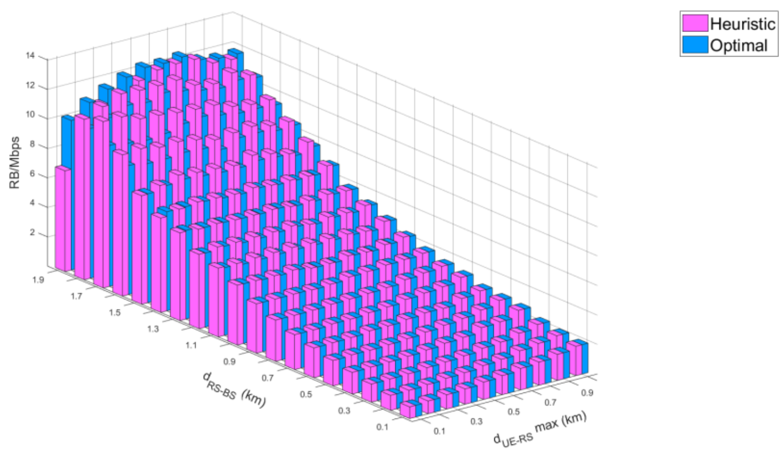

Figure 6-12. Average resource requirement per Mbps when the transmit power of the relay station is increased.

As discussed in Section 4.3 using a higher transmit power will result in a lower critical distance. At shorter distance from the relay station it will be more efficient to send via the relay than directly to the base station. This effect can be seen when we compare Figure 6-13 with Figure 6-5. As distance d_{RS-BS} increases, and thus the average distances from the UEs to the base station, more UEs prefer to send via the relay station. Also when distance $d_{UE-RS}\ max$ increases more use is made of the relay station. What stands out is that for a large distance d_{RS-BS} and small distance $d_{UE-RS}\ max$ the heuristic algorithm has a higher preference for the relay station than the optimal scheme. This is because at the heuristic algorithm the priority UE often has a lower throughput than the optimal scheduling, making it more efficient to have more UEs send their video stream via the relay.

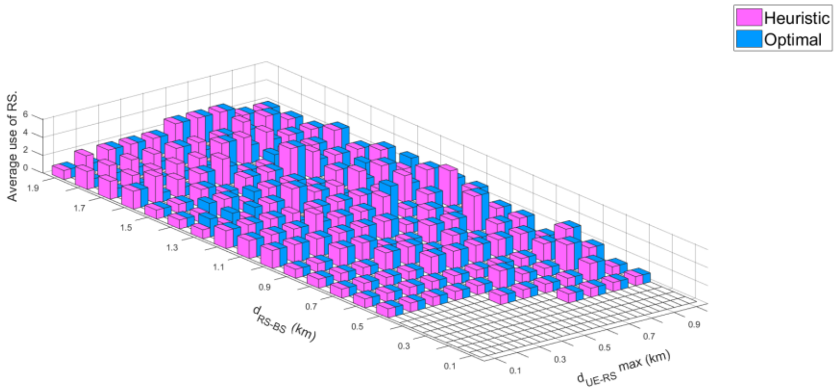

Figure 6-13. Average use of relay station when the transmit power of the relay station is increased.

6.4.3. Increased number of UEs in the area.

As indicated in Section 6.1 the number of UEs for which the optimal scheduling can be determined is limited to 6. Because of this only the results for the heuristic algorithm are available for the scenario with an increased number of UEs in the area. When the number of UEs in the area is increased, the sharp distance d_{RS-BS} at which the priority UE does not meet it requirement is still visible as can be seen in Figure 6-14. Only that distance lies 0.2 kilometers closer to the base station compared to Figure 6-2. A similar observation can be made for the non-priority UEs by comparing Figure 6-15 and Figure 6-3.

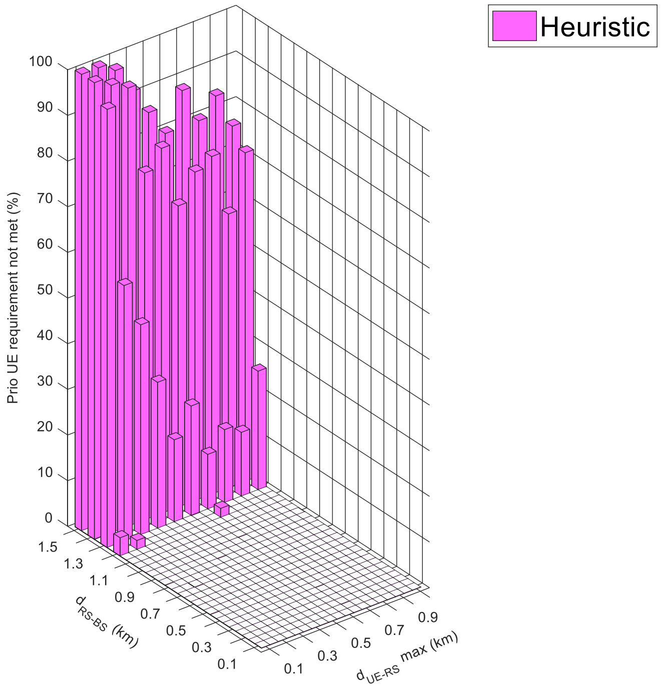

Figure 6-14. Number of times (in percentage) the priority UE is not meeting its requirement when there are 10 UEs in the area.

As distance d_{RS-BS} increases all UEs and the relays station increase their distance to the base station. Therefore, more resources are needed to maintain a certain data rate. Because the available resources have to be distributed over more UEs in comparison to the 6 UE for the basic scenario, the distance from where UEs cannot meet their requirements anymore is shortened.

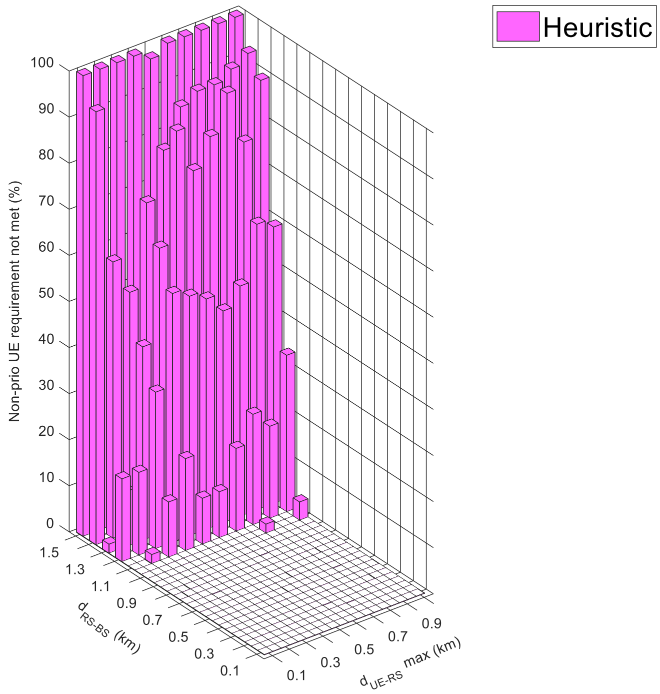

Figure 6-15. Number of times (in percentage) the non-priority UEs do not meet their requirements when there are 10 UEs in the area.

A comparison of Figure 6-4 and Figure 6-16 shows that the average amount of resources needed per Mbps appears to be lower for 10 UEs than for 6 UEs. Especially when the relay station is not too far away from the base station. The reason for this is that with six UEs even at a larger distance it is still possible to give all UEs enough resources to achieve the maximum data rate. With 10 UEs, the resources need to be distributed more and the non-priority UEs no longer can reach the maximum data rate. As described in Section 4.2 and clearly shown in Figure 4-2, the amount of resources needed for a lower data rate does not decrease linear. Which means that for a lower data rate less than a proportional amount of resources are required. This results in a lower average use of resources per Mbps.

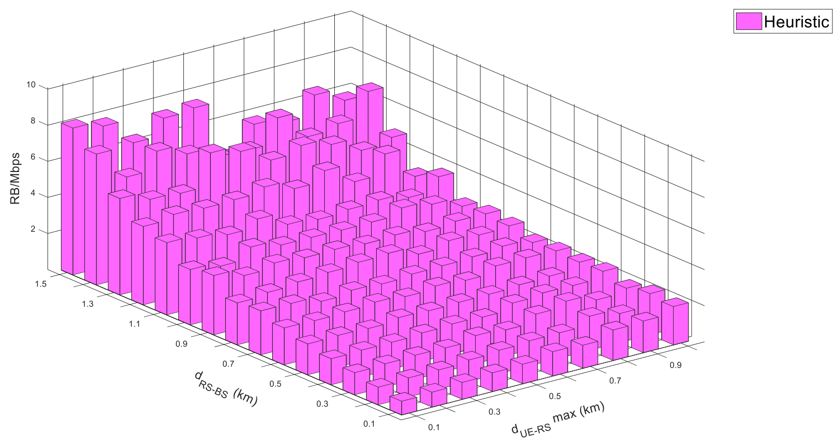

Figure 6-16. Average resource requirement per Mbps when there are 10 UEs in the area.

If we compare Figure 6-17 with Figure 6-5 which shows the average use of the relay station for the 10 UEs scenario and the basic scenario respectively, more use is made of the relay station when there are more UEs in the area. As shown in Figure 6-17, in more places 2 UEs use the relay station simultaneously to send their video stream. What is noticeable here is that the maximum amount of UEs that uses the relay station at the same time has not increased. The reason for this is that the amount of resources needed to support the relay stations target rate is so high that it is more efficient to let the majority of the UEs send their video streams directly to the base station.

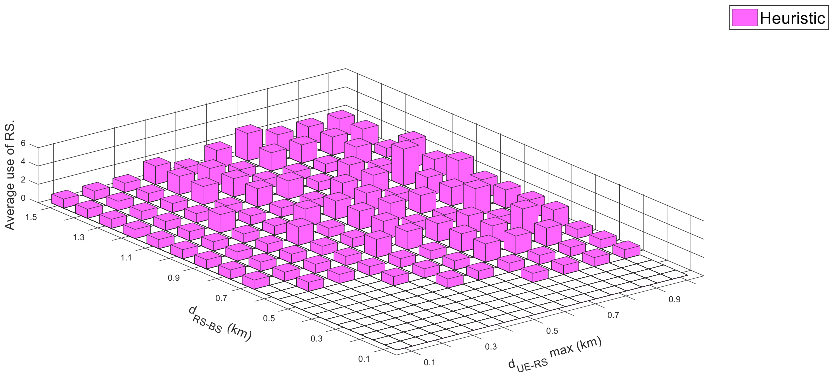

Figure 6-17. Average use of relay station when there are 10 UEs in the area.

6.4.4. Antenna of the relay station at vehicle height

When the antenna of the base station is at vehicle height the quality of the link between the UEs and the relay station deteriorates. The limit of 1.3 kilometer for the priority UE on the d_{RS-BS} axes in Figure 6-18 remains the same compared to the situation were the relay station uses a small antenna mast in Figure 6-2. However, even when the relay is at a shorter distance from the base station, UEs start to fail their requirements when their maximum distance $d_{UE-RS}max$ to the relay station is about 0.5 km. When the relay station is close to the base station, the UEs prefer to communicate to the base station directly.

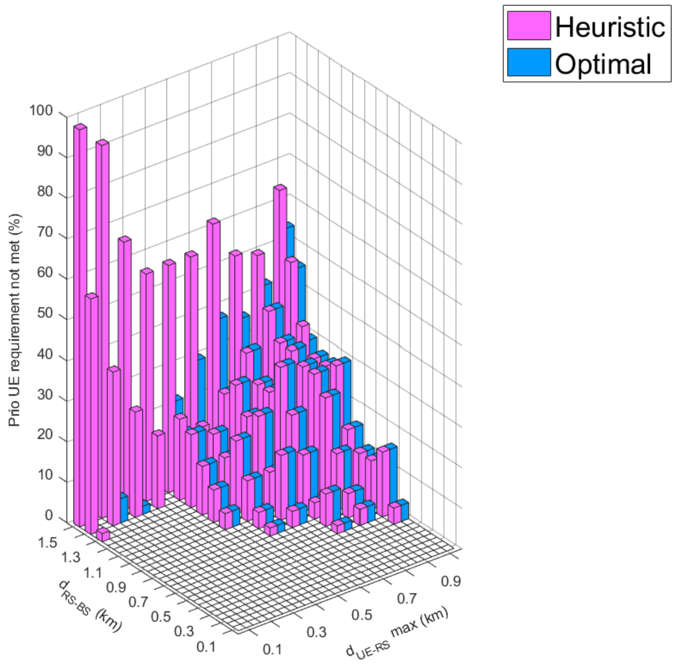

Figure 6-18. Number of times (in percentage) the priority UE is not meeting its requirement when the antenna of the relay station is at vehicle height.

The non-priority UEs show a similar course as the priority UE, as shown in Figure 6-19. The only difference compared to Figure 6-18, is that the non-priority UEs start to fail their requirements when the relay station is at a greater distance. The reason for this is that when the priority UE cannot reach the maximum rate, more resources are available for non-priority UEs, so that they are able to meet their requirements. However, at greater distance resources become more scarce, which means that this compensation does not hold and the non-priority UEs also will not meet their requirements anymore. In addition, the requirement for a non-priority UE is lower than that of the priority UE. This allows the non-priority UEs to use a direct link to the base station at a larger distance.

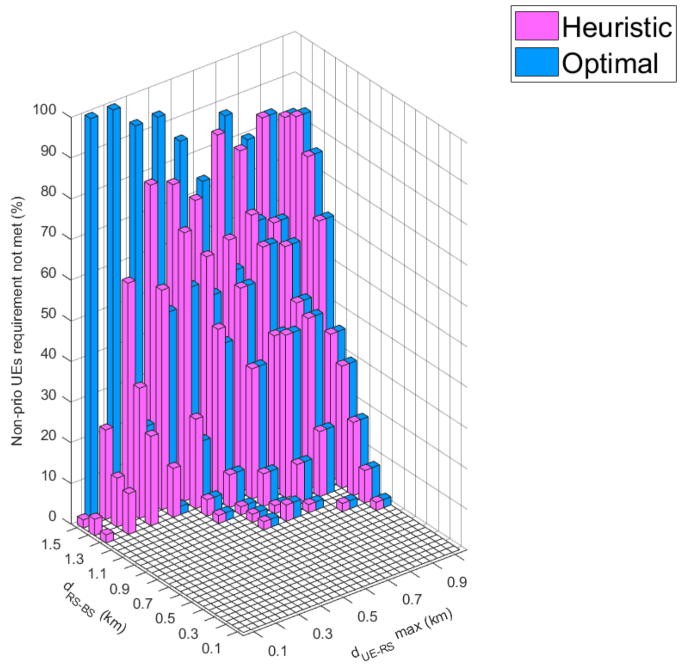

Figure 6-19. Number of times (in percentage) the non-priority UEs do not meet their requirements when the antenna of the relay station is at vehicle height.

A comparison of the resource usage in Figure 6-4 where a small antenna mast is used and Figure 6-20 where the antenna is mounted on the vehicle, show that the resource costs per Mbps is higher for the latter case. This is because, compared to Figure 6-5, also for larger distances a direct link to the base station is preferred as can be seen in Figure 6-21. Only when the relay station and so the UEs are at a greater distance to the base station, the relay station is used. However, this is still less than when a small antenna mast is used. Even with a low link quality between the UE and the relay station the heuristic algorithm tends to use the relay station more often than the optimal scheduling.

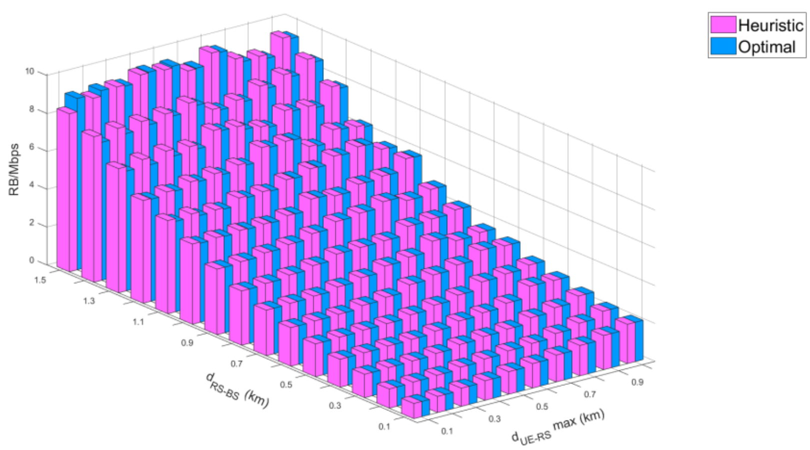

Figure 6-20. Average resource requirement per Mbps when the antenna of the relay station is at vehicle height.

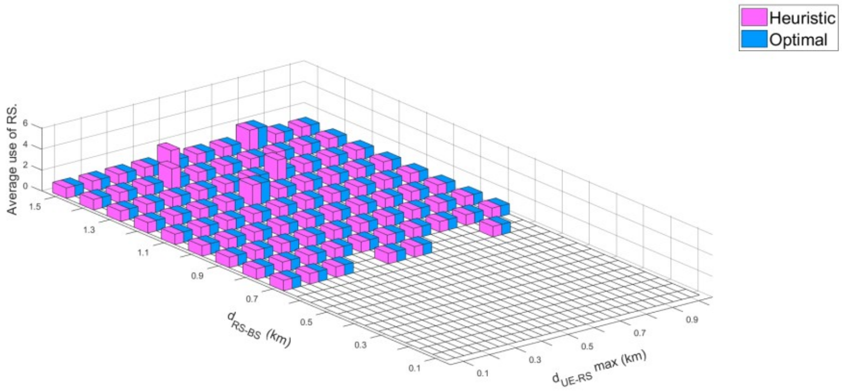

Figure 6-21. Average use of relay station when the antenna of the relay station is at vehicle height.

6.4.5. No relay station

For the results in Figure 6-22 till Figure 6-25 the distance d_{RS-BS} can be seen as the distance from the base station to the center of the area in which the UEs are distributed. A comparison of Figure 6-22 and Figure 6-2 for the Priority UE and Figure 6-23 and Figure 6-3 for the non-priority UEs show the effect of the deployment of a relay station. Especially when distance d_{RS-BS} is greater than 0.6 km, the deployment of a relay station provides a positive contribution to the throughputs of the UE's. The fact that the relay station's contribution is already noticeable at a shorter distance $d_{UE-RS}\ max$ as distance d_{RS-BS} increases is because, when distance d_{RS-BS} increases, the average distance from the UEs to the base station also increases.

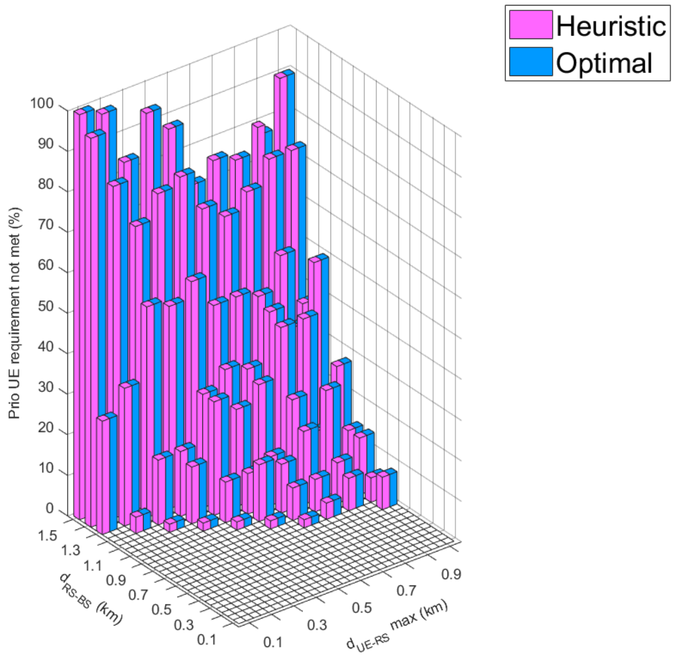

Figure 6-22. Number of times (in percentage) the priority UE is not meeting its requirement when there is no relay station.

The contribution of the relay station to the throughputs of the non-priority UEs is less than to the priority UE. This is because the non-priority UEs may have lower throughput than the priority UE and still meet their requirements. As a result, even at a greater distance to the base station it is more efficient for them to send their video streams directly to the base station. This also illustrates the influence of throughput on the critical distance as discribed in section 4.3.

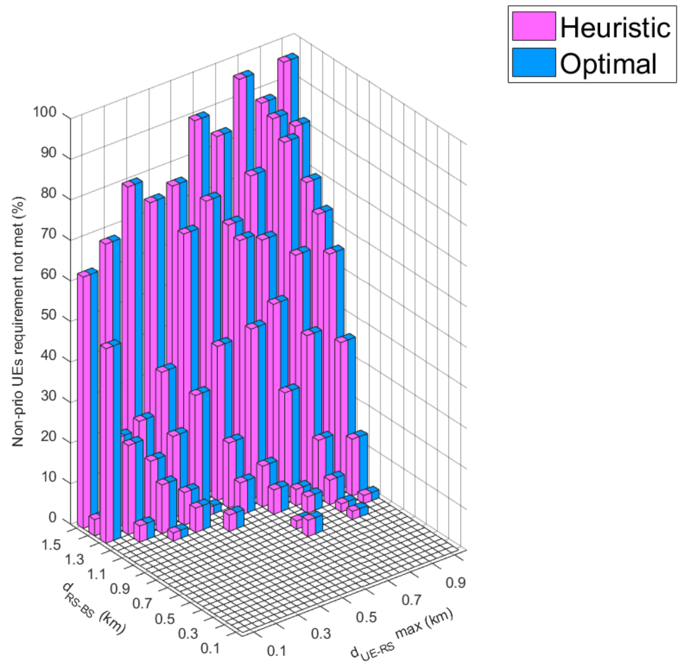

Figure 6-23. Number of times (in percentage) the non-priority UEs do not meet their requirements when there is no relay station.

The average resource usage per Mbps for the scenario without relay station is displayed in Figure 6-24. A comparison with Figure 6-4 for the basic scenario shows that the average resource usage increases faster when no relay station is used. This is because all UEs have to send their video streams directly to the base station without the support of a relay station. The greater the distance to the base station, the more expensive these video streams become compared to the deployment of a relay station.

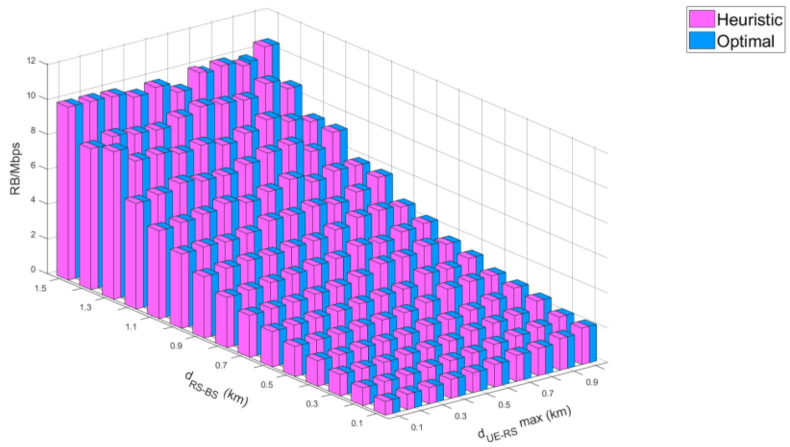

Figure 6-24. Average resource requirement per Mbps when there is no relay station.

Only a few times does the resource allocation, calculated by the heuristic algorithm, differ from the optimal scheduling as shown in Figure 6-25. For the cases $d_{RS-BS} = 0.8\ km$,

$d_{UE-RS} = 1.0\ km$ and $d_{RS-BS} = 1.4\ km$, $d_{UE-RS} = 0.1\ km$ there is a switch of data rates between two UEs. With a reverse allocation, they use just as much resources, which means that despite the difference in resource allocation both the heuristic algorithm and the optimal scheduling are just as efficient at these points. In the remaining 3 differences, the heuristic algorithm did not meet the optimal resource allocation. This also explains the differences in average resource usage visible at row $d_{RS-BS} = 1.5\ km$ in Figure 6-24.

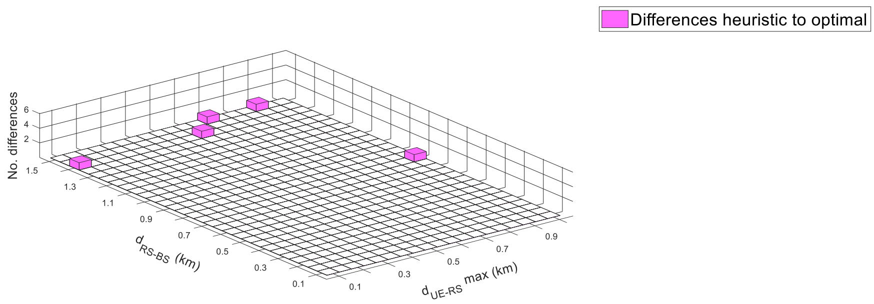

Figure 6-25. Number of differences between the heuristic algorithm and the optimal scheme when there is no relay station.

7. Conclusions and future work

In this research, a heuristic uplink resource allocation algorithm was designed and evaluated. From this research and its results a number of conclusions can be formulated and a number of research directions can be identified that can serve as future work.

7.1. Conclusions

With performing this research the main research question we want to answer is: How to create a suitable low complexity resource allocation algorithm for the distribution of live streaming video in a clustered D2D enabled 5G network supporting Public Safety Services? To formulate the conclusions of this research we will follow the structure of the research questions.

1. Which requirements must be taken into account when developing and evaluating the proposed algorithm?
    a. What are the requirements for communication systems and video used by public safety services?
    b. What are the requirements for streaming different quality type videos?

2. What can be a suitable resource allocation algorithm for meeting the video service requirements and leading to high network efficiency?

3. How does the proposed resource allocation algorithm perform compared to the optimal resource allocation scheme?
    a. To which extent does the proposed resource allocation algorithm deliver the same video quality as the optimal scheme?
    b. What is the resource usage of the proposed resource allocation algorithm, and how does it compare to the optimal scheme?

In the first part of our research we have looked into the requirements for the use of streaming video for public safety services. For public safety services application, network availability is paramount. A first responder must be connected to the network at all times and have the ability to communicate. In the case of this research, this means the ability to send live streaming video to the Central Command Post. A number of requirements regarding delay, resolution and frame rate are given. What is, however, of greater importance is that "more delay and a better picture quality" is preferable to "less delay and a worse picture quality". Which we interpreted as "less frames per second and a better picture quality" is preferable to "more frames per second and a worse picture quality". This has led to an overview of required data rates for different resolution and frame rate combinations in Section 3.2.

Regarding the second research question we have developed an heuristic resources allocation algorithm that works based on calculations made for a single UE in isolation. To this end, we have investigated how to allocate spectral resources for streaming video in a D2D enabled 5G cellular network. Since the formula for calculating the amount of resources needed for a given target rate is very complex, a numerical approach is used. This approach allocates a minimum amount of resources for a longer period of time for each user and the relay station. This ensures a high as possible transmit power per resource block which in turn results in a high as possible throughput. When the Heuristic algorithm has made initial choices for all UEs regarding the amount of resources and the route to the base station, corrections are made in the resource allocation. The algorithm also uses a group of priority UEs because the available resources are very limited and from an operational point of view we want to focus on the most interesting cases.

For answering the third research question we have identified the key parameters which have the most influence on the resulting resource allocation. These key parameters are: Path loss, transmit power, number of users and distances between the UEs, the relay station and the base station. For evaluating the proposed heuristic algorithm a basic scenario is created on which 4 variations are made. In each variant, one of the key parameters is changed in value. In order to evaluate the proposed heuristic algorithm and draw conclusions, it is compared to the optimal scheduling. The performance of the heuristic algorithm is measured in terms of failure rate and efficiency, as described in Section 6.3.

The results, based on simulation, show that the heuristic algorithm is a very promising, efficient and fast method for performing recourse allocation for a clustered D2D enabled 5G network for supporting Public Safety Services. In almost the entire range of the test scenarios, the UEs for both the heuristic algorithm and the optimal scheduling meet their requirements. Only when de distances to the base station become very large, it becomes clear that the heuristic algorithm performs less than the optimal scheduling. As a result, the area where the UEs meet their requirements for the heuristic algorithm is slightly smaller than that of the optimal scheduling. The resource usage of the heuristic algorithm is somewhat higher than optimal scheduling even when both meet the throughput requirements. However, the heuristic algorithm and the optimal scheduling are just as efficient when the relay station is not used. When one of the key parameters is varied, the heuristic algorithm shows similar behaviour.

Despite the promising results, there is definitely room for improvement. As indicated in Subsection 6.4.1, the efficiency check is the part of the heuristic algorithm that reduces the performance of the algorithm. This is confirmed by the fact that in Subsection 6.4.5 it is shown that upgrading and downgrading processes perform similarly to the optimal scheduling. We leave it as future work to create a better performing efficiency check.

7.2. Future Work

This research offers various options for future work. On the one hand, future work can be aimed at improving the developed heuristic algorithm. On the other hand, future work can also focus on extending of what is been achieved with this thesis.

For improving the heuristic resource allocation algorithm the following two proposals are made:

The first option for improving the heuristic algorithm is to improve the efficiency check. The manner in which the efficiency check is performed greatly affects the performance of the algorithm, as shown in Subsection 6.4.1. Due to calculation time, our heuristic algorithm stops its calculations if the required amount of resources exceeds the available resources. Because of this, it is not clear what the effect of reallocating a UE from the relay station to the base station has on the total required amount of resources, when it still exceeds the available resources. As a result, wrong choices are made, which means that resources are distributed in a less efficient way. The efficiency check to be developed should aim at gaining insight to the effect of transferring a UE to the base station on resource usage, while the required amount of resources exceeds the available resources.

The second option for improving the heuristic algorithm is to adjust the upgrading process. In Subsection 6.4.5 we have shown that the upgrading and downgrading processes give similar resource allocation as the optimal scheduling. However, in order to improve the throughput of the priority UE when the relay station is at 1.4 and 1.5 kilometer from the base station, it may be an option to also let the priority UE participate in the upgrading process. Although this is a workaround for the problem of efficiency check this option is definitely worth investigating.

For extending of what is achieved with this thesis we propose the following two options for future work:

The first option for for extending this research is to find a suitable time interval between resource allocations. When all resources have been assigned to the relay station and all UEs, then after a certain period of time, a new snapshot has to be made were it is re-determined how many UEs are present in the area, which UEs have priority and a new resource allocation has to be calculated. This thesis does not indicate what is the most suitable interval for making a new snapshot and perform a new resource allocation. On the one hand, this time interval must not be too short, otherwise the frame rate and/or resolution may change continuously, which will make the videos extremely annoying to look at. Also, for a short time interval, the computational load for the 5G infrastructure will be high. On the other hand the time interval must not be too long as this will degrade the QoS as the allocated resources do not adapt frequently enough to a changing situation.

Our last option for future work is to extend the architecture used in this thesis with more relay stations and so creating more clusters. This means that resources need to be distributed even more. However, if the distance between the clusters is large enough, this also provides the ability to perform reuse of resources.

References

[1] Ericsson, “5G: What is it for?”, White Paper, 2014.

[2] Ericsson, “5G Radio Access”, White Paper, 2016.

[3] R. Ferrus, O. Sallent, G. Baldini and L. Goratti, "LTE: the technology driver for future public safety communications", in IEEE Communications Magazine, vol. 51, no. 10, pp. 154-161, October 2013.

[4] S. Y. Lien, C. C. Chien, F. M. Tseng and T. C. Ho, "3GPP device-to-device communications for beyond 4G cellular networks", in IEEE Communications Magazine, vol. 54, no. 3, pp. 29-35, March 2016.

[5] NGMN Alliance, "Next Generation Mobile Networks", NGMN 5G white paper, 2015.

[6] E. Yaacoub, "On the use of device-to-device communications for QoS and data rate enhancement in LTE public safety networks", 2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Istanbul, 2014.

[7] 5G PPP, "5G empowering vertical industries", brochure, 2016.

[8] J. G. Andrews et al., "What Will 5G Be?", in IEEE Journal on Selected Areas in Communications, vol. 32, no. 6, pp. 1065-1082, June 2014.

[9] Ericsson, “5G systems”, Ericsson White paper, 2015.

[10] Samsung Electronics Co., “5G vision, white paper”, 2015.

[11] METIS Project, “Scenarios, Requirements and KPIs for 5G Mobile and Wireless System”, Deliverable D1.1, 2013.

[12] A. Osseiran et al., "Scenarios for 5G mobile and wireless communications: the vision of the METIS project", in IEEE Communications Magazine, vol. 52, no. 5, pp. 26-35, May 2014.

[13] A. Gupta and R. K. Jha, "A Survey of 5G Network: Architecture and Emerging Technologies", in IEEE Access, vol. 3, no. , pp. 1206-1232, 2015.

[14] NGMN Alliance, "NGMN 5G white paper", version 1.0, 2015.

[15] 5G PPP Architecture Working Group, "View on 5G Architecture", Whitepaper version 1.0, 2016.

[16] I.F. Akyildiz, S. Nie, S.-C. Lin, M. Chandrasekaran, "5G roadmap: 10 key enabling technologies", Computer Networks 106, pp. 17-48, 2016.

[17] C. Beard, W. Stalling, "Wireless Communication Networks and Systems," Pearson, Essex, 2016.

[18] Q. Ye, M. Al-Shalash, C. Caramanis and J. G. Andrews, "Distributed Resource Allocation in Device-to-Device Enhanced Cellular Networks", in IEEE Transactions on Communications, vol. 63, no. 2, pp. 441-454, 2015.

[19] C. Ye, M. C. Gursoy and S. Velipasalar, "Multimedia transmission over device-to-device wireless links", 2016 IEEE International Conference on Multimedia and Expo (ICME), Seattle, WA, pp. 1-6, 2016.

[20] M. N. Tehrani, M. Uysal and H. Yanikomeroglu, "Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions", in IEEE Communications Magazine, vol. 52, no. 5, pp. 86-92, 2014.

[21] A. Asadi, Q. Wang and V. Mancuso, "A Survey on Device-to-Device Communication in Cellular Networks", in IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1801-1819, Fourthquarter 2014.

[22] Motorola, "Video Solutions: Understanding Video Quality", White Paper, 2008.
[23] M. H. Pinson, S. Wolf and R. B. Stafford, "Video Performance Requirements for Tactical Video Applications," 2007 IEEE Conference on Technologies for Homeland Security, Woburn, MA, 2007.
[24] I. E. G. Richardson, “H.264 and MPEG-4 video compression: Video coding for next-generation multimedia”, Wiley InterScience, Chichester, 2003.
[25] Ericsson, “5G systems”, Ericsson White paper, 2015.
[26] [Online]. Available: http://celtic-san.com/2014/07/29/mac-and-scheduling/. [Accessed 15 12 2016].
[27] D. Dimitrova, “Analysing uplink scheduling in mobile networks : a flow-level perspective, Phd. Thesis, University of Twente,” Enschede, 2010.
[28] H. Holma and A. Toskala. WCDMA for UMTS. John Wiley & Sons Ltd, 2001.
[29] 3GPP TS 36.942., "LTE; Evolved Universal Terrestrial Radio Access (EUTRA);Radio Frequency (RF) system scenarios", 2010.
[30] 3GPP TR 30.03U., "Selection procedures for the choice of radio transmission technologies of the UMTS", version 3.2.0, 1998.
[31] S. Hakola, T. Chen, J. Lehtomaki and T. Koskela, "Device-To-Device (D2D) Communication in Cellular Network - Performance Analysis of Optimum and Practical Communication Mode Selection," 2010 IEEE Wireless Communication and Networking Conference, Sydney, 2010.
[32] Y. Wang and G. de Veciana, "Improving user perceived QoS in D2D networks via binary quantile opportunistic scheduling", 2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Tempe, AZ, 2016.
[33] N. Odeh, M. Abolhasan and F. Safaei, "Low Complexity Interference Aware Distributed Resource Allocation for Multi-Cell OFDMA Cooperative Relay Networks", 2010 IEEE Wireless Communication and Networking Conference, Sydney, NSW, 2010.
[34] S. K. Agrawal and K. Sharma, "5G millimeter wave (mmWave) communications", 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 3630-3634, New Delhi, 2016.
[35] E. G. Larsson, O. Edfors, F. Tufvesson and T. L. Marzetta, "Massive MIMO for next generation wireless systems", in IEEE Communications Magazine, vol. 52, no. 2, pp. 186-195, February 2014.
[36] J. Qiao, X. S. Shen, J. W. Mark, Q. Shen, Y. He and L. Lei, "Enabling device-to-device communications in millimeter-wave 5G cellular networks", in IEEE Communications Magazine, vol. 53, no. 1, pp. 209-215, 2015.
[37] ETSI, "NFV White paper", SDN and OpenFlow World Congress, Darmstadt, 2012.
[38] I.F. Akyildiz, S.-C. Lin, P. Wang, "Wireless software-defined networks (WSDNs) and network function virtualization (NFV) for 5G cellular systems: an overview and qualitative evaluation", Comput. Netw. 93, Part 1 pp 66–79, 2015.
[39] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher and V. Young, "Mobile Edge Computing A key technology towards 5G", ETSI White Paper No. 11, ETSI, 2015.
[40] J. G. Andrews, "Seven ways that HetNets are a cellular paradigm shift," in IEEE Communications Magazine, vol. 51, no. 3, pp. 136-144, March 2013.
[41] M. Ji, G. Caire and A. F. Molisch, "Wireless Device-to-Device Caching Networks: Basic Principles and System Performance", in IEEE Journal on Selected Areas in Communications, vol. 34, no. 1, pp. 176-189, 2016.

Appendix A: Technological innovations supporting 5G

This appendix discusses briefly the concepts of the most Ground-breaking technological innovations which are needed for meeting the ambitious requirements set for 5G. For meeting the requirements set for 5G all these technologies will have to be integrated.

Millimeter wave spectrum
Millimeter wave (mmWave) technology is aimed at using the frequency band from 5 GHz to 300 GHz and is considered the most promising technology to boost the network throughput and capacity. 5G mmWave communication systems offer a low latency 100 Gbps data connection for >100 billion devices [34]. To give an idea, the mmWave band from 20 GHz to 50 GHz already provides ten times more bandwidth than the entire 4G cellular band [10]. Frequency in the range of millimeter waves suffer significantly from energy absorption caused by atmosphere, rain, and snow. This causes limitation in signal transmission distance making mmWave only suitable for short range communication [8], [16]. On the one hand this can be seen as a limitation, but the great advantage is that frequencies can be reused at a small distance increasing network capacity.

Massive MIMO
In a Massive MIMO antenna system multiple antennas work together, unlike traditional antennas, to focus energy on to a small area in a 3D environment. By setting up multiple beams to different counterparts simultaneously huge improvements can be made in frequency efficiency, throughput and radiated energy efficiency. Other benefits of massive MIMO include extensive use of inexpensive low-power components, reduced latency, simplification of the MAC layer, and robustness against intentional jamming [35], [14].

Beamforming
Beamforming is a combination of high frequency transmission (mmWave) combined with Massive MIMO. This combination results in narrow high data rate beams that can be steered towards an intended receiver. In a beamforming enabled 5G network all UEs will have their own high data rate beam from the base station. This does not only apply to links between the base station and UEs, but can also be used for wireless backhauls connecting wireless base stations and helper nodes [36].

Wireless Software Defined Networking
Up to now commercial wireless networks are hardware-based and rely on closed and inflexible architectural designs. Such inflexible hardware-based architectures typically lead to a 10-year cycle for a new generation of wireless networks to be standardized and deployed. A technology that has been in use for years in wired networks is Software Defined Networking (SDN). SDN is often defined as the decoupling of the control and the data planes. This is done by removing control decisions from the hardware, e.g., switches. Doing so, paths can be

defined by a remote centralized management entity or by a group of such entities, rather than by using a distributed routing protocol or by examining a packets source and destination addresses locally in an interconnected device [9]. The main benefits of this architecture are the logical decoupling of the network intelligence to separate software based controllers, exposing the network capabilities through an application program interface, and enabling the application to request and manipulate services provided by the network [8].

Network Function Virtualisation

In literature Network Function Virtualisation (NFV) and WSDN are often mentioned together because they complement each other. However, since WSDN and NFV do not depend on each other and can be implemented independently, we want to separately discuss these two technologies.

NFV can be seen as the separation of the functionality of a device, and the hardware on which it operates. NFV has a great impact on network architecture given that functions become virtualized entities decoupled from hardware, as can be seen in Figure A-1. Hardware can be shared and possibly non-specialized COTS equipment (like standard processors, switches, storage) can be implemented.

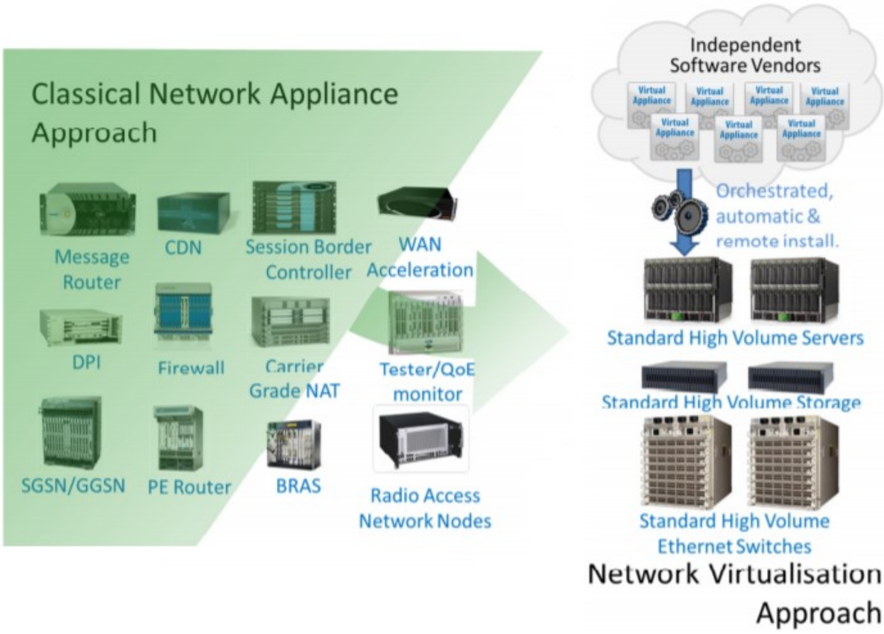

Figure A-1. Vision for Network Functions Virtualisation [37].

NFV enables network functions that were traditionally tied to hardware appliances to run on cloud computing infrastructure in a data center. This does not imply that the NFV infrastructure will be equivalent to commercial cloud or enterprise cloud. What is expected is that there will be a high degree of reuse of what commercial cloud offers. It is also expected that some requirements of mobile networks such as the separation of the data plane, control plane and management plane, will not be feasible within the commercial cloud [8].

The benefit of NFV is not immediately obvious. Simply virtualizing existing network nodes (e.g., gateways, MMEs) may make them cheaper to implement but will not reduce the network complexity or provide the needed adaptability to specific use cases [14]. However, NFV brings many potential benefits from cost reduction to great variety of system openness. NFV reduces CAPEX, OPEX, and power consumption through consolidating equipment and exploiting the economies of scale of the IT industry. Moreover, it increases speed of time to market by minimizing the typical network operator cycle of innovation. Also, NFV provides the availability of network appliance multi-version and multi-tenancy, which allows uses of a single platform for different applications, users and tenants, which enables a wide variety of eco-systems and encourages openness [38].

Mobile edge Computing

Mobile Edge Computing (MEC) provides cloud-computing capabilities at the edge of the mobile network, within the Radio Access Network (RAN) and in close proximity to mobile users (see Figure A-2). The aim is to reduce latency, ensure highly efficient network operation and service delivery. The MEC environment is characterized by low latency, proximity, high bandwidth, and real-time insight into radio network information and location awareness. All of this contributes to enhancing mobile broadband experience.

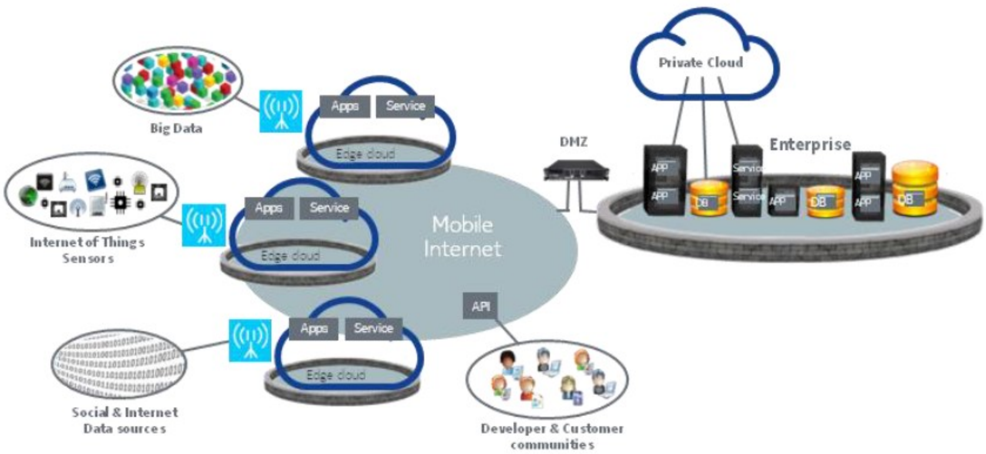

Figure A-2. Improved QoE with Mobile Edge Computing in close proximity to end users [39].

MEC is based on a virtualized platform, with an approach complementary to NFV. NFV focuses on network functions, MEC enables applications to run at the edge of the network. The infrastructure necessary for hosting MEC and NFV are quite similar, so from efficiency considerations both can be hosted on the same platform [39].

Radio Access Techniques

Bandwidth in the frequency domain is a scarce and costly resource, therefore efficient spectral resource management is a key factor to increase network capacity and to enable high data rates for an ever growing number of users. A lot of research is performed on how to assign available resources as efficiently as possible. More about this topic is discussed in chapter 5 “recent studies on D2D solutions for video applications”.

In 4G LTE, which is considered as the basis for 5G [9], E-UTRA uses OFDMA for the downlink channel and SC-FDMA for the uplink channel. The latter is used to overcome the high peak-to-average power ratio and thereby safe power at the UE [17], [16]. In LTE resources are divided over frequency and time. A resource block is the smallest resource allocation unit which can be assigned to a specific device. In the frequency domain LTE can work with bandwidths of 1.4 MHz, 3 MHz, 5 MHZ, 10 MHz, 15 MHz or 20 MHz containing 6, 15, 25, 50, 75 and 100 resource blocks respectively. In the time domain LTE uses radio frames of 10ms which include 10 sub-frames of 1ms. Each sub-frame contains two resource blocks of 0.5ms duration [17]. Figure A-3 shows an example of assigned resource blocks using a 1.4 MHz frequency band.

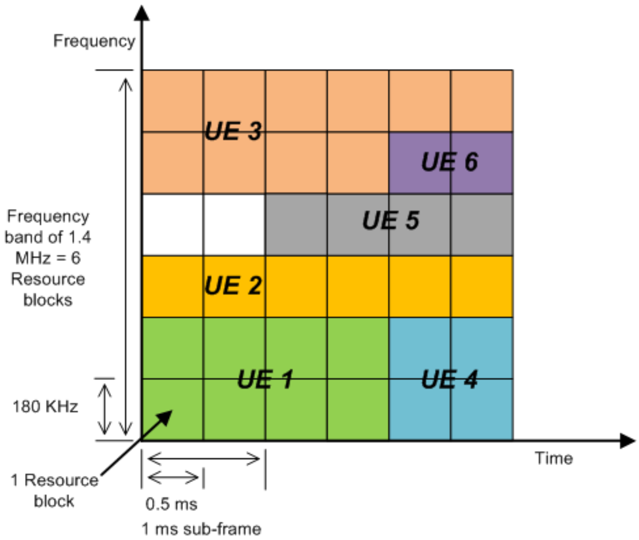

Figure A-3. Allocation of resource blocks in LTE [26].

Although OFDMA and SC-FDMA work well for 4G LTE networks, they do not support several scenarios foreseen for 5G networks. For example tactile Internet, requires a very low latency of the order of 1 ms. Such a low latency cannot be achieved using OFDM. Other applications like IoT have scenarios where the devices are not connected to the base station at all time. The power constraints in some of these devices prevent the devices from having full synchronization with the base station, which is needed for OFDM. According to [16] the multiple access technique for 5G networks should:

- Have a low latency;
- Allow a loose or more preferably no synchronization;
- Introduce low interference which will increase the effective spectral efficiency;
- Work efficiently using multiple antennas;
- Be power efficient, so that it does not drain out the power of low power devices.

There are several multiple access techniques that are currently being considered for 5G systems. Besides OFDM these are FBMC, UFMC, GFDM and NOMA. Most of them support several use cases that are envisioned to be supported by 5G networks [16].

Network densification

The traditional macrocell network architecture, that has been the basic network architecture in former cellular generations, will not be able to keep up with the tremendous growth in connected devices and demand for ever increasing bandwidth [40]. A key solution to this is network densification. Network densification uses cells with different sizes, creating multiple layers of cells throughout the network also called a Heterogeneous network or Hetnet, as shown in Figure A-4. When using smaller cells the frequency spectrum can be reused more often resulting in a higher network capacity. In addition, the base station has to serve less UE's. From a user's point of view, when multiple layers of cells are covering the same area, a UE can be assigned to the cell that provides the best data throughput. Also, when using smaller cells UE's may need less power to reach a network node [16]. The downside to network densification is that a particular area needs much more base stations and backhaul connections increasing the costs for the deployment and operational phase [8], [40], [41].

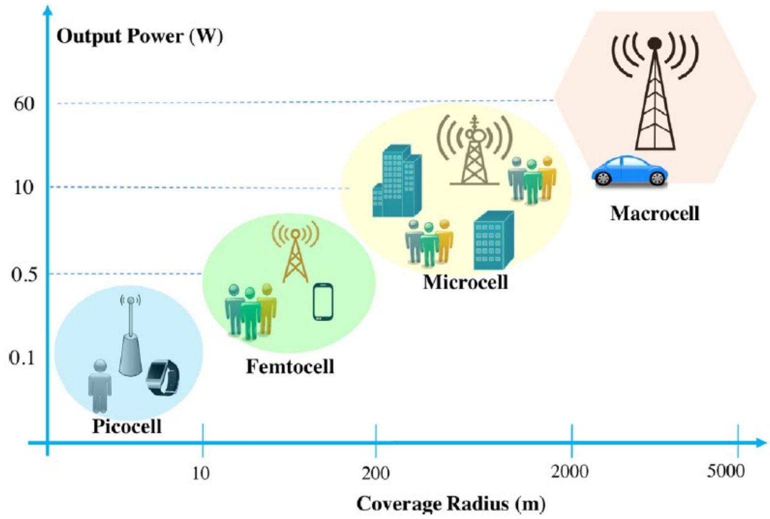

Figure A-4. The different layers of network densification [16].

Appendix B: Matlab codes

This appendix contains the Matlab code created during this research. By adding this code to the report, we want to offer others the opportunity to use this code or parts of it to improve the proposed heuristic algorithm or for their own research. To limit the size of the appendix, a small font size is used.

Main body

```
clear;
clc;

BW = 10;
RB = (BW*0.9)/0.18;

Prsw = 2; % relay power in watt
Prs = 10*log10(Prsw); % relay power in dB
Puew = 1; % relay power in watt
Pue = 10*log10(Puew); % relay power in dB
users = 6;

TR = [0.1 0.2 0.3 0.4 0.8 1.2 2]; %Target rates
TR_l = length(TR);
prio_gp = [1]; % UEs in the priority group
prio_gp_size = length(prio_gp);

user_table = [];
dir_rel = [];
priority_table = [];
connected_table = [];
work_table = zeros(users,12);
rate_assigned = zeros(9,4);
alg_lap_sum = zeros(9,4);
teller_rate = 0;
teller_res = 0;
opt_prio_req_met = 0;
opt_non_prio_req_met = 0;
prio_req_not_met = 0;
non_prio_req_not_met = 0;
data_points_rate = [];
data_points_res = [];
rs_axes = [];
ue_axes = [];
RS_dist = 0;
UE_dist_max = 0;

row = 1;
volgende = 1;
flag = 0;
for step_rs = 15:1:15 % distance RS-BS in steps of 0.1 km starting at 0.1km
for step_ue = 1:1:10 % distance UE-RS in steps of 0.1 km starting at 0.1km
    for lap = 1:1:50 % number of random droppings of UEs

        % Give UEs and RS new location

        [ RS_dist, RS_angle, UE_dist, UE_angle, UE_dist_RS, UE_dist_max ] = Place_devices( users, step_ue,
step_rs )

        ue_dist_rs = UE_dist_RS'

        UE_dist = [1.5840    1.5153    1.4963    1.5334    1.4460    1.4944] % 11 1
        UE_angle = [225.4810  224.1536  224.9451  225.2424  224.9253  225.3615]
        UE_dist_RS = [0.0850    0.0270    0.0040    0.0340    0.0540    0.0110]


        % setup tables
        [ user_table, dir_rel, priority_table, connected_table ] = Cluster_tables( Prs, Pue, users,
RS_dist, RS_angle, UE_dist, UE_angle, UE_dist_RS, TR, TR_l, prio_gp, prio_gp_size );

         %%% try minimal setup non-prio UEs 0,8 and prio UEs 2 %%%
         minimum = find(abs(TR-0.8) < 0.001);
         desired = find(abs(TR-2) < 0.001);
         not_possible = 0;

         for f = 1:1:users
             if (priority_table(f,2) == 1) && (connected_table(f,1+desired) == 1)&& (dir_rel(f,1+desired)
~= 1) %prio, connected, direct
                 work_table(f,1) =  user_table(f,4+desired);
             elseif (priority_table(f,2) == 1) && (connected_table(f,1+desired) == 1)&&
(dir_rel(f,1+desired) == 1) %prio, connected, relay
                  work_table(f,2) =  user_table(f,4+(2*TR_l)+desired);
                  work_table(f,2+desired) =  1;
```

```
            elseif (priority_table(f,2) ~= 1) && (connected_table(f,1+minimum) == 1)&&
(dir_rel(f,1+minimum) ~= 1) %niet prio, connected, direct
               work_table(f,1) =  user_table(f,4+minimum);
            elseif (priority_table(f,2) ~= 1) && (connected_table(f,1+minimum) == 1)&&
(dir_rel(f,1+minimum) == 1) %niet prio, connected, relay
                work_table(f,2) =  user_table(f,4+(2*TR_l)+minimum);
                work_table(f,2+minimum) =  1;
            end

        end

        work_table_safe_1 = work_table
        % check if all users are connected
        for f = 1:1:users
           fit = work_table(f,1) + work_table(f,2);
           if fit == 0
              not_possible = not_possible +1;
           end
        end

        % Is there a more efficient resource allocation?
        [ work_table_check1, dir_rel_check1, check_count_1 ] = eff_check( user_table, dir_rel, work_table,
users, TR, TR_l, RB, Prs, RS_dist );

        if check_count_1 > 0
             work_table = work_table_check1;
             dir_rel = dir_rel_check1;
        end

        % Does the basic setup fit?
        [ work_table_sum_init, work_table_calc_init ] = work_table_calc( work_table, users, TR, TR_l, RB,
Prs, RS_dist);

        % If basic setup does fit upgrade non-priority UEs
        if (work_table_calc_init(4,1) > 0) && (not_possible == 0)
            work_table_safe = work_table;

           [ work_table ] = Cluster_upgrade(  user_table, priority_table, connected_table, dir_rel,
work_table, users, TR, TR_l, RB, Prs, RS_dist);

           [ work_table_check2, dir_rel_check2, check_count_2 ] = eff_check( user_table, dir_rel,
work_table, users, TR, TR_l, RB, Prs, RS_dist );

            if check_count_2 > 0
                work_table = work_table_check2;
                dir_rel = dir_rel_check2;
            end

            [ ~, work_table_calc_plot ] = work_table_calc( work_table, users, TR, TR_l, RB, Prs, RS_dist);
%geeft enkel inzicht in gebruikte RBs

        else  % If basic setup does not fit downgrade all UEs until it does.
            work_table_safe = work_table;

            [ work_table ] = Cluster_downgrade(   user_table, priority_table, connected_table, dir_rel,
work_table, users, TR, TR_l, RB, Prs, RS_dist   );

            [ ~, work_table_calc_plot ] = work_table_calc( work_table, users, TR, TR_l, RB, Prs, RS_dist);
        end

        % Plot and show results

        [ ~, final_rate, final_resources ] = Cluster_draw(user_table, work_table_calc_plot, work_table,
users, prio_gp, UE_angle, UE_dist, RS_angle, RS_dist )
       rate_assigned(9,4)= 0;

        % determine optimal scheduling
        [ opt_rates, opt_resources ] = Cluster_optimal_scalelable_6_function_v07( user_table, users, Prs,
RS_dist, TR);


        alg_rates(lap,:) = final_rate;
        optimal_rates(lap,:) = opt_rates;

        alg_resources(lap,:) = final_resources;
        optimal_resources(lap,:) = opt_resources;

        tot_alg_rates(row,:) = final_rate;
        tot_optimal_rates(row,:) = opt_rates;

        tot_alg_resources(row,:) = final_resources;
        tot_optimal_resources(row,:) = opt_resources;

        row = row + 1;

        work_table = zeros(users,12);
        end
    end

end
```

Random placement of UEs

```
function [ RS_dist_new, RS_angle, UE_dist_n, UE_angle_n, UE_dist, UE_dist_max_new] = Place_devices( users,
count_ue, count_rs )

%This function updates the locations of the UEs and the relay station

UE_dist_max_initial = 0.1;
RS_dist_initial = 0.1;
RS_angle = 225;

UE_dist_max_new = (UE_dist_max_initial - 0.1) + (0.1 * count_ue);
RS_dist_new = (RS_dist_initial - 0.1) + (0.1 * count_rs);

UE_dist = round( 1000*(0 + (UE_dist_max_new - 0)*rand(users,1)))/1000;
UE_angle = round((0 + (360-0)*rand(users,1)));

% Set distances UEs from BS to RS location
for f = 1:1:users
    [ UE_dist_n(f), UE_angle_n(f)] = Loc_random_UE( UE_dist(f), UE_angle(f), RS_dist_new, RS_angle);
end

end

function [ rho_n, theta_n] = Loc_random_UE( d_ue, UE_angle, d_rs, RS_angle )

% moving UEs located around center in polarplot to area around relay station center
% and calculate distance to base station.

[x_rs,y_rs] = pol2cart(deg2rad(RS_angle),d_rs);
[x_ue,y_ue] = pol2cart(deg2rad(UE_angle),d_ue);

 x_n = x_ue + x_rs;
 y_n = y_ue + y_rs;

 [theta,rho_n] = cart2pol(x_n,y_n);

 theta_n = rad2deg(theta);
 if theta_n < 0
    theta_n = theta_n + 360;
 end

end
```

Step 1. Snapshot

```
function [ user_table, dir_rel, priority_table, connected_table ] = Cluster_tables( Prs, Pue, users,
RS_dist, RS_angle, UE_dist, UE_angle, UE_dist_RS,TR, TR_l, prio_gp, prio_gp_size )

% This function creates the user_table, dir-rel_table, Priority_table and connected_table.
% based on these tables the Heuristic algorithm makes choices.

% Opzetten resources table
 for d = 1:1:users
     user_table(d,1) = d; % User ID
     user_table(d,2) = UE_dist(d); % distance to BS
     user_table(d,3) = UE_angle(d);  % angle
     user_table(d,4) = UE_dist_RS(d); % distance to RS for compare

%      user_table(d,4) = Dist_UE_RS(UE_dist(d), UE_angle(d), RS_dist, RS_angle);  % distance to RS for
cluster_main_v06

     dir_rel(d,1) = d; % setting up direct-relay table.
     priority_table(d,1) = d; % setting up priority table
     connected_table(d,1) = d; % setting up connected table

     for f = 1:1:TR_l
         % resources table
         user_table(d,4+f) = UE_direct(TR(f), Pue, UE_dist(d));
         [user_table(d,4+TR_l+f),user_table(d,4+(2*TR_l)+f)] = UE_via_RS(TR(f), Prs, Pue, RS_dist,
user_table(d,4));

         % direct vs relay table
         if ((user_table(d,4+f) >= user_table(d,4+TR_l+f)) && (user_table(d,4+f) > 0)...
            && (user_table(d,4+TR_l+f) > 0)) || ((user_table(d,4+f) == 0) && (user_table(d,4+TR_l+f)~= 0))
             dir_rel(d,1+f) = 1;
         else
             dir_rel(d,1+f) = 0;
         end
         if (user_table(d,4+f) == 0) && (user_table(d,4+TR_l+f) == 0)
             connected_table(d,1+f) = 0;
         else
             connected_table(d,1+f) = 1;
         end
     end

          % priority table
```

```
     if any(d==prio_gp)
           priority_table(d,2) = 1; % set priority flag
     else
           priority_table(d,2) = 0; % set priority flag
     end
 end

end
```

determine amount of resources via relay station

```
function [ RBs_total, RBs_ue ] = UE_via_RS( Target_rate, Prs, Pue, drs, due )

% This function calculates the resources required for a given target rate when using
% the relay station.

%Ldue = 128.1+(37.6*log10(due)); %loss due to distance UE-RS (dB). Antenna at height BS
Ldue = 131.3+(38.4*log10(due)); %loss due to distance UE-RS (dB). Antenna at 10m height
%Ldue = 148+(40*log10(due)); %loss due to distance UE-RS (dB). antenna at vehicle height
RBue_max = floor (10^((Pue-(-10+Ldue+(-146.45)))/10)); % determine max RBs UE-RS

%Ldrs = 10000000+(100000*log10(drs)); % impossible to use relay station
Ldrs = 128.1+(37.6*log10(drs)); %loss due to distance RS-BS (dB)
RBrs_max = floor (10^((Prs-(-10+Ldrs+(-146.45)))/10)); % determine max RBs RS_BS

% calculates the BRs needed to transmit via the RS

Rate_min = 0;
time = 0;
for BRs_users = 1:1:50 %Increasing the number of RBs for the UE

    if Rate_min == 0
        % determine rate UE
        RBmin = BRs_users-1;
        SINRue = 10^((Pue-(10*log10(BRs_users))-Ldue-(-146.45))/10);
        Rue = (BRs_users*0.18)*0.4*log2(1+SINRue);

        % determine SINRi + rate rate RBue-1
        SINRue_1 = 10^((Pue-(10*log10(RBmin))-Ldue-(-146.45))/10);
        Rue_M1 = (RBmin*0.18)*0.4*log2(1+SINRue_1);

        if (Rue >=Target_rate) && (RBmin ~= 0)
            diff = Rue-Rue_M1;
            short = Target_rate-Rue_M1;
            time =ceil((short/diff)*1000)/1000;
            UErate =((1-time)*Rue_M1)+(time*Rue);
            Rate_min = 1;
        elseif (Rue >=Target_rate) && (RBmin == 0)
             time = ceil((Target_rate/Rue)*1000)/1000;
             UErate = time*Rue;
             Rate_min = 1;
        end

        % Detemine if relay can support the UE
        RB_relay =0;
        if Rue >=Target_rate;
            RB_UE = RBmin+time;
            x =0;
            RSrate =0;
            time_M1 =0;
            time_P1 = 0;
            time_P11 =0;
            for RBrs = 1:1:50 % aantal RBs
                if x==0
                    % determining Rate relay
                    SINRrs = 10^((Prs-(10*log10(RBrs))-Ldrs-(-146.45))/10);
                    Rrs = (RBrs*0.18)*0.4*log2(1+SINRrs);

                    if (Rrs >= UErate) && (x==0)
                        x =1;
                        RB_RSmin1 = RBrs-1;
                        SINR_1 = 10^((Prs-(10*log10(RB_RSmin1))-Ldrs-(-146.45))/10);
                        Rrs_1 = (RB_RSmin1*0.18)*0.4*log2(1+SINR_1);

                        if RB_RSmin1 ~= 0
                            diff = Rrs-Rrs_1;
                            short = (UErate)-Rrs_1;
                            time_M1 =ceil((short/diff)*1000)/1000;
                            RSrate =((1-time_M1)*Rrs_1)+(time_M1*Rrs);
                        elseif RB_RSmin1 == 0
                             time_M1 = ceil((UErate/Rrs)*1000)/1000;
                             RSrate = time_M1*Rrs;
                        end
                         RB_rs = RB_RSmin1 + time_M1;
                         RBs_subtotal = RB_UE+RB_rs; % total RBs used

                    end
                end
            end
```

```
        end
    end
end

if (Rate_min == 0) || (x == 0) || (RB_rs > RBrs_max) || (RB_UE > RBue_max)
    RBs_total = 0;
    RBs_ue = 0;
else
    RBs_total = RBs_subtotal;
    RBs_ue = RB_UE;
end

end
```

determine amount of resources direct to base station

```
function [ RB_direct ] = UE_direct( Target_rate, Pue, due_dir  )

% This function calculates the RBs needed for direct transmission to the BS

Ldue_dir = 128.1+(37.6*log10(due_dir)); %loss due to distance UE-BS dB
RBue_max = floor (10^((Pue-(-10+Ldue_dir+(-146.45)))/10)); % determine max RBs


Rate_min_RBs = 0;
time_dir = 0;
for BRs_users = 1:1:50 %Increase the number of RBs
    if Rate_min_RBs == 0
        RBmin_dir = BRs_users-1;

        % determine SINRi + rate for RBue
        SINRue_dir = 10^((Pue-(10*log10(BRs_users))-Ldue_dir-(-146.45))/10);
        Rue_dir = (BRs_users*0.18)*0.4*log2(1+SINRue_dir);

        % detemine SINRi + rate voor RBue-1
        SINRue_1_dir = 10^((Pue-(10*log10(RBmin_dir))-Ldue_dir-(-146.45))/10);
        Rue_M1_dir = (RBmin_dir*0.18)*0.4*log2(1+SINRue_1_dir);

        if (Rue_dir >=Target_rate) && (RBmin_dir ~= 0)
            diff_dir = Rue_dir-Rue_M1_dir;
            short_dir = Target_rate-Rue_M1_dir;
            time_dir =ceil((short_dir/diff_dir)*1000)/1000;
            UErate_dir =((1-time_dir)*Rue_M1_dir)+(time_dir*Rue_dir);
            Rate_min_RBs = 1;
        elseif (Rue_dir >=Target_rate) && (RBmin_dir == 0)
             time_dir = ceil((Target_rate/Rue_dir)*1000)/1000;
             UErate_dir = time_dir*Rue_dir;
             Rate_min_RBs = 1;
        end
    end
end
% if the amount of required resources fit the available resources give back
% value, else return 0.
if (Rate_min_RBs == 1) && ((RBmin_dir+time_dir) <= RBue_max)
    RB_direct = RBmin_dir+time_dir;
else
    RB_direct = 0;
end

end
```

Step 3b. downgrade UEs

```
function [ work_table ] = Cluster_downgrade( user_table, priority_table, connected_table, dir_rel,
work_table, users, TR, TR_l, RB, Prs, RS_dist    )


% This function downgrades all UEs according to table 5.1.

% decision matrix for downgrading
    decision_matrix(:,1) = [5 4 4 4 3 2 1 1 1 1]; % for non-prio UEs
    decision_matrix(:,2) = [1 0 0 1 1 1 0 0 0 0];
    decision_matrix(:,4) = [7 7 6 5 5 5 5 4 3 2]; % for prio UEs
    decision_matrix(:,5) = [0 1 1 0 0 0 1 1 1 1];

    % If a UE does not meet the requirement find its maximum rate
    for f = 1:1:users
       if (work_table(f,1) == 0) && (work_table(f,2) == 0)
           [~, c_col] = find(abs(connected_table(f,2:8)-1) < 0.001);
           tmp_c = length(c_col);
           if tmp_c > 0
               if dir_rel(f,1+tmp_c) == 0
                   work_table(f,1) = user_table(f,4+tmp_c);
               elseif dir_rel(f,1+tmp_c) == 1
                   work_table(f,2) = user_table(f,4+(2*TR_l)+tmp_c)
                   work_table(f,2+tmp_c) = 1;
               end
```

```
           end
       end
    end
         work_table_safe_2a = work_table
         [ work_table_down_1, dir_rel_down_1, down_count_1 ] = eff_check...
             ( user_table, dir_rel, work_table, users, TR, TR_l, RB, Prs, RS_dist );

         if down_count_1 > 0
             work_table = work_table_down_1;
             dir_rel = dir_rel_down_1;
         end
           work_table_safe_2b = work_table


    if (sum(work_table(:,1)) + sum(work_table(:,2))) > 0
        for down = 1:1:10 % go through decision matrix (10 steps)
           % determine downgrade values
           for f = 1:1:users
               if (priority_table(f,2) ~= 1) && ((work_table(f,1)+ work_table(f,2)) > 0)  % non prio UEs
and connected
                    [~, d_col] = find(abs(work_table(f,3:9)-1) < 0.001);
                    if d_col == decision_matrix(down,1)% when using relay
                        if dir_rel(f,1+d_col-1) == 1
                            work_table(f,10) = user_table(f,4+TR_l+d_col)-user_table(f,4+TR_l+d_col-
decision_matrix(down,2));
                        elseif dir_rel(f,d_col) ~= 1
                            work_table(f,10) = user_table(f,4+TR_l+d_col)-user_table(f,4+d_col-
decision_matrix(down,2));
                        end
                    elseif (sum(work_table(f,3:2+TR_l)) == 0) && (work_table(f,1) ~= 0) % when direct to BS
                        if find(abs(user_table(f,5:11)-work_table(f,1)) < 0.001) == decision_matrix(down,1)
                            [f_user, f_col] = find(abs(user_table(f,5:4+TR_l)-work_table(f,1)) < 0.001);
                             work_table(f,10) = work_table(f,1)-user_table(f,4+f_col-
decision_matrix(down,2));
                        end
                    end
               elseif (priority_table(f,2) == 1)% prio UEs, when via relay
                    [~, e_col] = find(abs(work_table(f,3:9)-1) < 0.001);
                    if e_col == decision_matrix(down,4);
                        if dir_rel(f,1+e_col-1) == 1
                            work_table(f,11) = user_table(f,4+TR_l+e_col)-user_table(f,4+TR_l+e_col-
decision_matrix(down,5));
                        elseif dir_rel(f,e_col) ~= 1
                            work_table(f,11) = user_table(f,4+TR_l+e_col)-user_table(f,4+e_col-
decision_matrix(down,5));
                        end
                     elseif (sum(work_table(f,3:2+TR_l)) == 0) && (work_table(f,1) ~= 0) % when direct to
BS
                        if find(abs(user_table(f,5:11)-work_table(f,1)) < 0.001) == decision_matrix(down,4)
                            [~, g_col] = find(abs(user_table(f,5:4+TR_l)-work_table(f,1)) < 0.001)
                            work_table(f,11) = work_table(f,1)-user_table(f,4+g_col-
decision_matrix(down,5))
                        end
                    end

               end
           end

work_table_safe_2c = work_table
           [ ~, work_table_calc_down ] = work_table_calc( work_table, users, TR, TR_l, RB, Prs, RS_dist)
           if decision_matrix(down,2) == 1

               while (max(work_table(:,10)) ~= 0) && (work_table_calc_down(4,1) <= 0) % loop for non-prio
UEs
                % determine downgrade values and downgrade
                   [~,h_ue] = max(work_table(:,10));
                   if work_table(h_ue,1) == 0 % UE connected via RS
                       [h_user, ue_col] = find(abs(work_table(h_ue,3:9)-1) < 0.001);
                       if dir_rel(h_ue,1+ue_col-decision_matrix(down,2)) == 1  % stays via relay
                           work_table(h_ue,2) = user_table(h_ue,4+(2*TR_l)+ue_col-decision_matrix(down,2));
                           work_table(h_ue,2+ue_col) = 0;
                           work_table(h_ue,2+ue_col-decision_matrix(down,2)) = 1;
                           work_table(h_ue,3+TR_l) = 0;
                       elseif dir_rel(h_ue,1+ue_col-decision_matrix(down,2)) ~= 1  % becomes direct to BS
                           work_table(h_ue,1) = user_table(h_ue,4+ue_col-decision_matrix(down,2));
                           work_table(h_ue,2) = 0;
                           work_table(h_ue,2+ue_col) = 0;
                           work_table(h_ue,3+TR_l) = 0;
                           dir_rel(h_ue,1+ue_col-1) = 0;
                       end

                   elseif work_table(h_ue,1) ~= 0 % cennected directly to BS
                           [~, ue_col] = find(abs(user_table(h_ue,5:11)-work_table(h_ue,1)) < 0.001)
                           h_ue
                           work_table(h_ue,1) = user_table(h_ue,4+ue_col-decision_matrix(down,2));
                           work_table(h_ue,3+TR_l) = 0
                           dir_rel(h_ue,1+ue_col-1) = 0
                   end
```

```
                     work_table_safe_3a = work_table
                     [ work_table_down_2, dir_rel_down_2, down_count_2 ] = eff_check...
                         ( user_table, dir_rel, work_table, users, TR, TR_l, RB, Prs, RS_dist );

                     if down_count_2 > 0
                         work_table = work_table_down_2;
                         dir_rel = dir_rel_down_2;
                     end
                       work_table_safe_3b = work_table

                   % check if the downgrade fits the resources available
                   work_table_down = work_table(:,1:2+TR_l);
                   [ ~, work_table_calc_down ] = work_table_calc( work_table_down, users, TR, TR_l, RB,
Prs, RS_dist)
               end

           elseif decision_matrix(down,2) ~= 1
               while (max(work_table(:,11)) ~= 0) && (work_table_calc_down(4,1) <= 0) % loop for prio UEs
                  [~,h_prio] = max(work_table(:,11))
                  [~, p_col] = find(abs(work_table(h_prio,3:9)-1) < 0.001)
                  if work_table(h_prio,1) == 0
                   if dir_rel(h_prio,1+p_col-decision_matrix(down,5)) == 1  % stays via relay
                       work_table(h_prio,2) = user_table(h_prio,4+(2*TR_l)+p_col-decision_matrix(down,5));
                       work_table(h_prio,2+p_col) = 0;
                       work_table(h_prio,2+p_col-decision_matrix(down,5)) = 1;
                       work_table(h_prio,4+TR_l) = 0;
                   elseif dir_rel(h_prio,1+p_col-decision_matrix(down,5)) ~= 1  % becomes direct
                       work_table(h_prio,1) = user_table(h_prio, 4+p_col-1);
                       work_table(h_prio,2) = 0;
                       work_table(h_prio,2+p_col) = 0;
                       work_table(h_prio,4+TR_l) = 0;
                   end

                elseif work_table(h_prio,1) ~= 0 % direct
                       work_table(h_prio,1) = user_table(h_prio,4+g_col-decision_matrix(down,5));
                       work_table(h_prio,4+TR_l) = 0;
                  end

                     work_table_safe_4a = work_table
                     [ work_table_down_3, dir_rel_down_3, down_count_3 ] = eff_check( user_table, dir_rel,
work_table, users, TR, TR_l, RB, Prs, RS_dist );

                     if down_count_3 > 0
                         work_table = work_table_down_3;
                         dir_rel = dir_rel_down_3;
                     end
                       work_table_safe_4b = work_table

               % check if the downgrade fits the resources available
               work_table_down = work_table(:,1:2+TR_l);
               [ ~, work_table_calc_down ] = work_table_calc( work_table_down, users, TR, TR_l, RB, Prs,
RS_dist);
               end

           end

           if work_table_calc_down(4,1) > 0
               work_table = work_table(:,1:2+TR_l);
               break
           end

        end

    [ work_table ] = Cluster_upgrade( user_table, priority_table, connected_table, dir_rel, work_table,
users, TR, TR_l, RB, Prs, RS_dist);

    end

end
```

Step 3a. Upgrade non-priority UEs

```
function [ work_table ] = Cluster_upgrade(  user_table, priority_table, connected_table, dir_rel,
work_table, users, TR, TR_l, RB, Prs, RS_dist  )

% this function upgrades non-prio UEs

for step = 1:1:7
    for f = 1:1:users % determine upgrade values
        if work_table(f,1) ~= 0
            waarde = work_table(f,1);
            [~, col_a] = find(abs(user_table(f,5:4+TR_l)-waarde) < 0.001)
             if isempty(col_a)
                work_table(f,3+TR_l) = 0;
                work_table(f,4+TR_l) = 0;
            else
```

```
            value_loc = col_a

             if (priority_table(f,2) ~= 1) && (connected_table(f,value_loc+1) == 1) &&
             (dir_rel(f,value_loc+1) ~= 1) && (value_loc <= 6)%non-prio, connected, direct
                 if (connected_table(f,value_loc+2) == 1) && (dir_rel(f,value_loc+2) ~= 1) % 1 up rate,
                 connected, direct
                     work_table(f,3+TR_l) = user_table(f,value_loc+5)-user_table(f,value_loc+4);
                     work_table(f,4+TR_l) = 0;
                 elseif (connected_table(f,value_loc+2) == 1) && (dir_rel(f,value_loc+2) == 1) % 1 up
                 rate, connected, direct becomes relay
                    work_table(f,3+TR_l) = user_table(f,TR_l+value_loc+5)-user_table(f,value_loc+4);
                    work_table(f,4+TR_l) = 1;
                 else
                    work_table(f,3+TR_l) = 0;
                    work_table(f,4+TR_l) = 0;
                 end
            end
        end

    elseif work_table(f,2) ~= 0
     value_loc = find(abs(work_table(f,3:2+TR_l)-1) < 0.001);
      if (priority_table(f,2) ~= 1) && (connected_table(f,value_loc+1) == 1)&&
     (dir_rel(f,value_loc+1)  == 1) && (value_loc <= 6)%niet prio, connected, relay
            if (connected_table(f,value_loc+2) == 1)&& (dir_rel(f,value_loc+2) == 1) % 1 up rate,
            connected, direct
                 work_table(f,3+TR_l) = user_table(f,TR_l+value_loc+5)-user_table(f,TR_l+value_loc+4);
                 work_table(f,4+TR_l) = 0;
            else
                work_table(f,3+TR_l) = 0;
                work_table(f,4+TR_l) = 0;
             end
     end
    end
    if (length(work_table(f,:)) > 9)
        if (work_table(f,10) < 0.0001)
            work_table(f,10) = round(0);
        end
    end
end
work_table(users,13)= 0;

% search UE that can be upgrades using a minimum of extra resources.
    for f = 1:1:users
        work_table_safe_up = work_table
        if isempty(work_table(:,3+TR_l))
            break
        else
            if (work_table(f,10) < 0.0001) && (work_table(f,10) > -0.0001)
                work_table(f,10) = 0;
            end

            tmp_a = unique(work_table(:,3+TR_l))
            if (tmp_a(1) == 0) && (length(tmp_a) > 1)
                min_RBs = tmp_a(2)
            elseif (tmp_a(1) == 0) && (length(tmp_a) == 1)
                break
            else
                min_RBs = tmp_a(1)
            end
        end
        [a_user, a_col] = find(abs(work_table(:,3+TR_l)-min_RBs) < 0.00001)
        a_user_lenght = length(a_user);
        if a_user_lenght > 1
            a_user = a_user(1)
            a_col = a_col(1)
        end
        tmp_b = work_table(a_user,1)+work_table(a_user,2);
        if work_table(a_user,1) ~= 0
            [b_user,b_col] = find(abs(user_table(a_user,5:4+TR_l)-tmp_b) < 0.00001)
            base = 4
        else
            [b_user,b_col] = find(abs(user_table(a_user,5+(2*TR_l):4+(3*TR_l))-tmp_b) < 0.00001)
            base = 4+(2*TR_l)
        end

        b_user_lenght = length(b_user);
        if b_user_lenght > 1
            b_row = b_user(1);
            b_col = b_col(1);
        end


       if (base > 4) && (connected_table(a_user,1+b_col+1) ~= 0) % next is relay en connected
            work_table(a_user,1) = 0;
            work_table(a_user,2) = user_table(a_user,4+(2*TR_l)+b_col+1);
            work_table(a_user,2+b_col) = 0;
            work_table(a_user,3+b_col) = 1;
            work_table(a_user,3+TR_l) = 0;
        elseif (base > 4) && (connected_table(a_user,1+b_col+1) == 0) % relay and next is not connected
```

```
            work_table(a_user,1) = 0;
            work_table(a_user,2) = 0;
            work_table(a_user,2+b_col) = 0;
            work_table(a_user,3+b_col) = 0;
            work_table(a_user,3+TR_l) = 0;
         elseif (base == 4) && (dir_rel(a_user,1+b_col+1) == 0) && (connected_table(a_user,1+b_col+1) ~= 0)
         % direct, next direct and connected
            work_table(a_user,1) = user_table(a_user,4+b_col+1);
            work_table(a_user,2) = 0;
            work_table(a_user,3+TR_l) = 0;
         elseif (base == 4) && (dir_rel(a_user,1+b_col+1) ~= 0) && (connected_table(a_user,1+b_col+1) ~= 0)
         % direct, next relay en connected
            work_table(a_user,1) = 0;
            work_table(a_user,2) = user_table(a_user,4+(2*TR_l)+b_col+1);
            work_table(a_user,3+b_col) = 1;
            work_table(a_user,3+TR_l) = 0;
        elseif (base == 4) && (connected_table(a_user,1+b_col+1) == 0) % direct and next not connected
            work_table(a_user,1) = 0;
            work_table(a_user,2) = 0;
            work_table(a_user,3+TR_l) = 0;
        end

         [ work_table_up_1, dir_rel_up, up_count_1 ] = eff_check( user_table, dir_rel, work_table, users,
TR, TR_l, RB, Prs, RS_dist )

        if up_count_1 > 0
             work_table = work_table_up_1;
             dir_rel = dir_rel_up;
        end

      % does the upgrade fit the total amount of resources?
      work_table_up_1 = work_table(:,1:2+TR_l)
      [ ~, work_table_calc_up_1 ] = work_table_calc( work_table_up_1, users, TR, TR_l, RB, Prs, RS_dist)
      if (work_table_calc_up_1(4,1) <= 0) &&(work_table(a_user,2) == 0)

          [ work_table_up_2, dir_rel_up, up_count_2 ] = eff_check( user_table, dir_rel, work_table, users,
TR, TR_l, RB, Prs, RS_dist )

        if up_count_2 > 0
             work_table = work_table_up_2;
             dir_rel = dir_rel_up;
        end

        [ ~, work_table_calc_up_2 ] = work_table_calc( work_table, users, TR, TR_l, RB, Prs, RS_dist)

          if (work_table_calc_up_2(4,1) <= 0) && (work_table(a_user,2) == 0)
           work_table = work_table_safe_up
           break
          end
      elseif (work_table_calc_up_1(4,1) <= 0) && (work_table(a_user,2) ~= 0)
          work_table = work_table_safe_up;
          work_table(a_user,10) = 0;
      end

    end
    work_table_up_3 = work_table
      [ ~, work_table_calc_up_2 ] = work_table_calc( work_table_up_3, users, TR, TR_l, RB, Prs, RS_dist)
      if work_table_calc_up_2(4,1) <= 0;
        work_table = work_table_up_3
       break

      end

      work_table1 = work_table

end
```

Efficiency check

```
function [ work_table, dir_rel, check_count ] = eff_check( user_table, dir_rel, work_table, users, TR,
TR_l, RB, Prs, RS_dist )

% checks if resources can be scheduled more efficient
count_rel = 0;
count_dir = 0;
check_count = 0;
[ ~, work_table_calc_check ] = work_table_calc( work_table, users, TR, TR_l, RB, Prs, RS_dist);
work_table_rel(:,1) = work_table(:,2);
for f = 1:1:users
    if work_table_rel(f,1) > 0
        [~, c_col] = find(abs(user_table(f,19:25)-work_table_rel(f,1)) < 0.001);
        if user_table(f,4+c_col) > 0
            count_rel = count_rel + 1;
            work_table_rel(f,2) = abs(user_table(f,4+TR_l+c_col) - user_table(f,4+c_col));
        end
    end
end

% check if UEs can switch from relay station to base station
```

```
if sum(work_table_rel(:,1)) > 0
   for f = 1:1:count_rel
        work_table_tmp = work_table
        dir_rel_tmp = dir_rel;
        [ ~, work_table_calc_check ] = work_table_calc( work_table, users, TR, TR_l, RB, Prs, RS_dist);
        tmp_a = unique(work_table_rel(:,2))
        if tmp_a(1) > 0
            tmp_b = tmp_a(1)
        elseif (tmp_a(1) == 0) && (length(tmp_a) > 1)
            tmp_b = tmp_a(2)
        else
            work_table = work_table_tmp;
            dir_rel = dir_rel_tmp;
            break
        end
        [a_user, a_col] = find(abs(work_table_rel-tmp_b) < 0.001);
        a_user_lenght = length(a_user);
        if a_user_lenght > 1
            a_user = a_user(1);
            a_col = a_col(1);
        end
        work_table_rel(a_user,2) = 0;
        value_loc = find(abs(work_table(a_user,3:9)-1) < 0.001);
        new_value = user_table(a_user,4+value_loc) %RBs direct to BS instead of via RS
        if new_value > 0
            work_table(a_user,1) = new_value;
            work_table(a_user,2) = 0;
            work_table(a_user,2+value_loc) = 0
            dir_rel(a_user,1+value_loc) = 0;
        end
        [ ~, work_table_calc_eff ] = work_table_calc( work_table, users, TR, TR_l, RB, Prs, RS_dist);
        if work_table_calc_eff(4,1) < work_table_calc_check(4,1)
            work_table = work_table_tmp;
            dir_rel = dir_rel_tmp;
            break
        end
        check_count = check_count +1;
   end
end

end
```

Optimal resource alocation

```
function [ optimal_rates, optimal_resources ] = Cluster_optimal_scalelable_6_function_v07( user_table,
users, Prs, RS_dist, TR)

% setup baseline table
bin_table= abs(dec2bin(0:(2^users-1)) - '1');
[rows_a,users] = size(bin_table);

for i = 1:1:rows_a
   for k = 1:1:users
      if bin_table(i,k) == 0
          bin_table(i,k) = 8;
      end
   end
end

 input_a = {[0:4],[0:6]};
  sub_combi_a = allcomb(input_a{:});
  [length_a,~] = size(sub_combi_a);

   input_b = {[5:6],[3:6]};
  sub_combi_b = allcomb(input_b{:});
  [length_b,~] = size(sub_combi_b);

  decision_matrix = sub_combi_a;
  decision_matrix(length_a+1:length_a+length_b,:) = sub_combi_b(1:length_b,:);
  [length_b,~] = size(decision_matrix);

% search for best baseline
y = 1;
for i = 1:1:rows_a
    for j = 1:1:length_b
        for k = 1:1:users
            if (k == 1)
            target(y,k) = bin_table(i,k) + decision_matrix(j,1);
           else
            target(y,k) = bin_table(i,k) + decision_matrix(j,2);
           end

        end
      y = y + 1;
    end
end
target_safe_a = target;
```

```
[length_a,~] = size(target);
target(y-1,(2*users)+4) = 0;

for j = 1:1:y-1
   for k = 1:1:users
        if (target(j,k) == 1)
            if user_table(k,4+1) ~= 0
                target(j,users+1+k) = 0.1;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,5);
        elseif (target(j,k) == 2)
            if user_table(k,4+2) ~= 0
                target(j,users+1+k) = 0.2;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,6);
        elseif (target(j,k) == 3)
            if user_table(k,4+3) ~= 0
                target(j,users+1+k) = 0.3;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,7);
        elseif (target(j,k) == 4)
            if user_table(k,4+4) ~= 0
                target(j,users+1+k) = 0.4;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,8);
        elseif (target(j,k) == 5)
            if user_table(k,4+5) ~= 0
                target(j,users+1+k) = 0.8;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,9);
        elseif (target(j,k) == 6)
            if user_table(k,4+6) ~= 0
                target(j,users+1+k) = 1.2;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,10);
        elseif (target(j,k) == 7)
            if user_table(k,4+7) ~= 0
                target(j,users+1+k) = 2;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,11);
        elseif (target(j,k) == 8)
            if user_table(k,4+8) ~= 0
                target(j,users+1+k) = 0.1;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,19);
        elseif (target(j,k) == 9)
            if user_table(k,4+9) ~= 0
                target(j,users+1+k) = 0.2;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,20);
        elseif (target(j,k) == 10)
            if user_table(k,4+10) ~= 0
                target(j,users+1+k) = 0.3;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,21);
        elseif (target(j,k) ==11)
            if user_table(k,4+11) ~= 0
                target(j,users+1+k) = 0.4;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,22);
        elseif (target(j,k) == 12)
            if user_table(k,4+12) ~= 0
                target(j,users+1+k) = 0.8;
            else
                target(j,users+1+k) = 0;
            end
```

```
            resources(j,k) = user_table(k,23);
        elseif (target(j,k) == 13)
            if user_table(k,4+13) ~= 0
                target(j,users+1+k) = 1.2;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,24);
        elseif (target(j,k) == 14)
            if user_table(k,4+14) ~= 0
                target(j,users+1+k) = 2;
            else
                target(j,users+1+k) = 0;
            end
            resources(j,k) = user_table(k,25);
    end
    if target(j,k) >= 8
        target(j,(2*users)+3) = target(j,(2*users)+3) + target(j,users+1+k); % determine TR for RS
    end
   end
      target(j,(2*users)+4) = sum(target(j,users+2:(2*users)+1));
   [ resources(j,users+2) ] = UE_direct(target(j,(2*users)+3), Prs, RS_dist); % resources relay
   if (target(j,(2*users)+3) > 0) && (resources(j,users+2) == 0)
       resources(j,users+2) = 100 ;
   end
   resources(j,users+3) = sum(resources(j,1:users+2));
   if (sum(resources(j,users+3)) > 50) || (any(resources(j,1:users) == 0))
        resources(j,users+3) = 0;
   end
    if j > (length_a/2) % keep cheapest: UE-BS or UE-RS-BS
       if  (resources(j,users+3) >= resources(j-(length_a/2),users+3)) && (resources(j-
(length_a/2),users+3) > 0)
            resources(j,users+3) = 0;
       elseif (resources(j,users+3) < resources(j-(length_a/2),users+3)) && (resources(j,users+3) > 0)
            resources(j-(length_a/2),users+3) = 0;
       end
    end
end

resources_safe_a = resources;

% remove all rows from tables which do not fit in RB
loc_a = 1;
for j = 1:1:length_a
    if (resources(j,users+3) == 0)
        to_del_a(loc_a) = j;
        loc_a = loc_a + 1;
    end
end
target(to_del_a,:) = [];
resources(to_del_a,:) = [];
[length_b,~] = size(target);

% find maximum rate for prio user and delete all rows with lower rates
for prio_max = 7:-1:1
  if (any(target(:,1) == prio_max)) || (any(target(:,1) == prio_max+7))
      maximum = prio_max;
      break
  end
end

target_safe_b = target;
resources_safe_b = resources;

loc_b = 1;
for f = 1:1:length_b
      if (target(f,1) < maximum) || ((target(f,1) > maximum) && (target(f,1) < (maximum+7)))
        to_del_b(loc_b) = f;
        loc_b = loc_b + 1;
    end
end
target(to_del_b,:) = [];
resources(to_del_b,:) = [];

target_safe_c1 = target;
resources_safe_c1 = resources;


% find maximum base rate for non prio UEs
max_val_a = max(target(:,(2*users)+4));
[row_max, ~] = ind2sub(size(target),find(target(:,(2*users)+4)==max_val_a));
[length_c,~] = size(row_max);
for f = 1:1:length_c
    target_base_1(f,:) = target(row_max(f,1),:);
    resources_base_1(f,:) = resources(row_max(f,1),:);
end

target_safe_c2 = target_base_1;
```

```
resources_safe_c2 = resources_base_1;

PUE_min = min(target_base_1(:,1));
PUE_max = max(target_base_1(:,1));

[length_c,~] = size(target_base_1);

target_safe_d1 = target_base_1;
resources_safe_d1 = resources_base_1;

% if more options available take cheapest in RS RBs
if length_c > 1
    min_val_a = min(resources_base_1(:,users+2));
    loc_b2 = 1;
    for f = 1:1:length_c
          if resources_base_1(f,users+2) > min_val_a
            to_del_b2(loc_b2) = f;
            loc_b2 = loc_b2 + 1;
        end
    end
    if loc_b2 > 1
        target_base_1(to_del_b2,:) = [];
        resources_base_1(to_del_b2,:) = [];
    end

    target_safe_d = target_base_1;
    resources_safe_d = resources_base_1;

    [length_cc,~] = size(target_base_1);

    if length_cc > 1
        min_val_aa = min(resources_base_1(:,users+3));
        [row_min_a,~] = ind2sub(size(resources_base_1),find(resources_base_1==min_val_aa));
        [length_d,~] = size(row_min_a);

        for f = 1:1:length_d
            target_base(f,:) = target_base_1(row_min_a(f),:);
            resources_base(f,:) = resources_base_1(row_min_a(f),:);
        end
    else
        target_base = target_base_1;
        resources_base = resources_base_1;
    end
else
    target_base = target_base_1;
    resources_base = resources_base_1;
end

% check if a non prio ue is at maximum and if so, upgrade prio ue if possible.

if (target_base(1,users+2) < 2) && target_base(1,users+3) < 2
    flag_a = 0;
    for f = 2:1:users
        if target_base(1,f) > 7
           np_base_value =  target_base(1,f)-7
        else
           np_base_value =  target_base(1,f)
        end
       if  (user_table(f,5+np_base_value) < 0.0001) && (user_table(f,12+np_base_value) < 0.0001) % if a np
UE is at limit, first upgrade prio UE
        flag_a = 1
        break
       end
    end
    if flag_a == 1
        for f = 1:1:users
            x(f) = target_base(1,f);
            if x(f) < 8
                b(f) = x(f) + 7;
                a(f) = x(f);
            else
                b(f) = x(f);
                a(f) = x(f) - 7;
            end
            aa(f) = 7;
            bb(f) = 14;

        end
        input_a = {[a(1):aa(1) b(1):bb(1)],[a(2) b(2)],[a(3) b(3)],[a(4) b(4)],[a(5) b(5)],[a(6) b(6)]};
        sub_combi_2 = allcomb(input_a{:});
        [length_dd,~] = size(sub_combi_2);

        rates_pu(:,1:users) = sub_combi_2(:,1:users);

       for j = 1:1:length_dd
           for k = 1:1:users
                if (rates_pu(j,k) == 1)
                    if user_table(k,4+1) ~= 0
```

```
            rates_pu(j,users+1+k) = 0.1;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,5);
    elseif (rates_pu(j,k) == 2)
        if user_table(k,4+2) ~= 0
            rates_pu(j,users+1+k) = 0.2;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,6);
    elseif (rates_pu(j,k) == 3)
        if user_table(k,4+3) ~= 0
            rates_pu(j,users+1+k) = 0.3;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,7);
    elseif (rates_pu(j,k) == 4)
        if user_table(k,4+4) ~= 0
            rates_pu(j,users+1+k) = 0.4;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,8);
    elseif (rates_pu(j,k) == 5)
        if user_table(k,4+5) ~= 0
            rates_pu(j,users+1+k) = 0.8;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,9);
    elseif (rates_pu(j,k) == 6)
        if user_table(k,4+6) ~= 0
            rates_pu(j,users+1+k) = 1.2;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,10);
    elseif (rates_pu(j,k) == 7)
        if user_table(k,4+7) ~= 0
            rates_pu(j,users+1+k) = 2;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,11);
    elseif (rates_pu(j,k) == 8)
        if user_table(k,4+8) ~= 0
            rates_pu(j,users+1+k) = 0.1;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,19);
    elseif (rates_pu(j,k) == 9)
        if user_table(k,4+9) ~= 0
            rates_pu(j,users+1+k) = 0.2;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,20);
    elseif (rates_pu(j,k) == 10)
        if user_table(k,4+10) ~= 0
            rates_pu(j,users+1+k) = 0.3;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,21);
    elseif (rates_pu(j,k) ==11)
        if user_table(k,4+11) ~= 0
            rates_pu(j,users+1+k) = 0.4;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,22);
    elseif (rates_pu(j,k) == 12)
        if user_table(k,4+12) ~= 0
            rates_pu(j,users+1+k) = 0.8;
        else
            rates_pu(j,users+1+k) = 0;
        end
        resources_pu(j,k) = user_table(k,23);
    elseif (rates_pu(j,k) == 13)
        if user_table(k,4+13) ~= 0
            rates_pu(j,users+1+k) = 1.2;
        else
            rates_pu(j,users+1+k) = 0;
        end
```

```
                resources_pu(j,k) = user_table(k,24);
            elseif (rates_pu(j,k) == 14)
                if user_table(k,4+14) ~= 0
                    rates_pu(j,users+1+k) = 2;
                else
                    rates_pu(j,users+1+k) = 0;
                end
                resources_pu(j,k) = user_table(k,25);
            end
            if rates_pu(j,k) >= 8
            rates_pu(j,(2*users)+3) = rates_pu(j,(2*users)+3) + rates_pu(j,users+1+k); % determine TR
for RS
            end
        end

         rates_pu(j,(2*users)+4) = sum(rates_pu(j,users+2: (2*users)+1));

         [ resources_pu(j,users+2) ] = UE_direct(rates_pu(j,(2*users)+3), Prs, RS_dist); % resources
relay
         if (rates_pu(j,(2*users)+3) > 0) && (resources_pu(j,users+2) == 0)
             resources_pu(j,users+2) = 100 ;
         end
         resources_pu(j,users+3) = sum(resources_pu(j,1:users+2));
         if (sum(resources_pu(j,users+3)) > 50) || (any(resources_pu(j,1:users) == 0))
             resources_pu(j,users+3) = 0;
         end
         if j > (length_dd/2)
            if  (resources_pu(j,users+3) >= resources_pu(j-(length_dd/2),users+3)) && (resources_pu(j-
(length_dd/2),users+3) > 0)
                resources_pu(j,users+3) = 0;
            elseif (resources_pu(j,users+3) < resources_pu(j-(length_dd/2),users+3)) &&
(resources_pu(j,users+3) > 0)
                resources_pu(j-(length_dd/2),users+3) = 0;
            end
         end
     end

     loc_dd = 1;
     for j = 1:1:length_dd
         if resources_pu(j,users+3) == 0
             to_del_dd(loc_dd) = j;
             loc_dd = loc_dd + 1;
         end
     end
     if loc_dd > 1
         rates_pu(to_del_dd,:) = [];
         resources_pu(to_del_dd,:) = [];
     end
     [length_ddd,~] = size(rates_pu);

     resources_pu_safe = resources_pu;
     rates_pu_safe = rates_pu;

    % find maximum rate for prio user and delete all rows with lower rates
     for f = 7:-1:1
       if (any(rates_pu(:,1) == f)) || (any(rates_pu(:,1) == f+7))
           maximum = f;
           break
       end
     end

     loc_bb = 1;
     for f = 1:1:length_ddd
           if (rates_pu(f,1) < maximum) || ((rates_pu(f,1) > maximum) && (rates_pu(f,1) < (maximum+7)))
             to_del_bb(loc_bb) = f;
             loc_bb = loc_bb + 1;
         end
     end
     if loc_bb > 1
         rates_pu(to_del_bb,:) = [];
         resources_pu(to_del_bb,:) = [];
     end
     [length_dddd,~] = size(rates_pu);

     PUE_min = min(rates_pu(:,1));
     PUE_max = max(rates_pu(:,1));

     % choose cheapest combination
     if length_dddd > 1
         min_val_b = min(resources_pu(:,users+3));
         [row_min_b, ~] = ind2sub(size(resources_pu),find(resources_pu==min_val_b));
         [length_d5,~] = size(row_min_b);


         for f = 1:1:length_d5
             target_base(f,:) = rates_pu(row_min_b(f),:);
             resources_base(f,:) = resources_pu(row_min_b(f),:);
         end
```

```
        else
            target_base = rates_pu;
            resources_base = resources_pu;
        end

    end
end


% Check if non prio UEs can be upgraded

% determine optimal combination for non prio UEs
% setup short combination table
for f = 2:1:users
    x(f) = target_base(1,f);
    if x(f) < 8
        b(f) = x(f) + 7;
        a(f) = x(f);
    else
        b(f) = x(f);
        a(f) = x(f) - 7;
    end
    aa(f) = a(f)+3;
    if aa(f) > 7
        aa(f) = 7;
    end
    bb(f) = b(f)+3;
    if bb(f) > 14
        bb(f) = 14;
    end
end


input = {[a(2):aa(2) b(2):bb(2)],[a(3):aa(3) b(3):bb(3)],[a(4):aa(4) b(4):bb(4)],[a(5):aa(5)
b(5):bb(5)],[a(6):aa(6) b(6):bb(6)]};
sub_combi = allcomb(input{:});
[length_e,~] = size(sub_combi);

sub_combi_safe_1 = sub_combi;

% delete invalid combinations
to_del_c = [];
loc_c = 1;
for f = 1:1:length_e
      if ((~any(sub_combi(f,:) == a(2))) && (~any(sub_combi(f,:) == b(2))))
          to_del_c(loc_c) = f;
        loc_c = loc_c + 1;
    end
end
sub_combi(to_del_c,:) = [];

 sub_combi_safe_a = sub_combi;
[length_f,columns_f] = size(sub_combi);

if PUE_min == PUE_max% setup data rate table (rates_last) and resources table
    for i = 1:1:length_f
        rates_last(i,1) = target_base(1,1); % rate of prio UE
    end
    rates_last(:,2:1+columns_f) = sub_combi(:,1:columns_f);
    rates_last(length_f,(2*users)+4) = 0;
else
    for i = 1:1:length_f
        rates_last(i,1) = maximum; % rate of prio UE
    end
    for i = length_f+1:1:2*length_f
        rates_last(i,1) = maximum+7; % rate of prio UE
    end
    rates_last(1:length_f,2:1+columns_f) = sub_combi(1:length_f,1:columns_f);
    rates_last(length_f+1:2*length_f,2:1+columns_f) = sub_combi(1:length_f,1:columns_f)
    rates_last(2*length_f,(2*users)+4) = 0;
    length_f = length_f *2
end

rates_last_safe_a = rates_last;

for j = 1:1:length_f
   for k = 1:1:users
        if (rates_last(j,k) == 1)
            if user_table(k,4+1) ~= 0
                rates_last(j,users+1+k) = 0.1;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,5);
        elseif (rates_last(j,k) == 2)
            if user_table(k,4+2) ~= 0
                rates_last(j,users+1+k) = 0.2;
            else
```

```
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,6);
    elseif (rates_last(j,k) == 3)
        if user_table(k,4+3) ~= 0
            rates_last(j,users+1+k) = 0.3;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,7);
    elseif (rates_last(j,k) == 4)
        if user_table(k,4+4) ~= 0
            rates_last(j,users+1+k) = 0.4;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,8);
    elseif (rates_last(j,k) == 5)
        if user_table(k,4+5) ~= 0
            rates_last(j,users+1+k) = 0.8;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,9);
    elseif (rates_last(j,k) == 6)
        if user_table(k,4+6) ~= 0
            rates_last(j,users+1+k) = 1.2;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,10);
    elseif (rates_last(j,k) == 7)
        if user_table(k,4+7) ~= 0
            rates_last(j,users+1+k) = 2;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,11);
    elseif (rates_last(j,k) == 8)
        if user_table(k,4+8) ~= 0
            rates_last(j,users+1+k) = 0.1;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,19);
    elseif (rates_last(j,k) == 9)
        if user_table(k,4+9) ~= 0
            rates_last(j,users+1+k) = 0.2;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,20);
    elseif (rates_last(j,k) == 10)
        if user_table(k,4+10) ~= 0
            rates_last(j,users+1+k) = 0.3;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,21);
    elseif (rates_last(j,k) ==11)
        if user_table(k,4+11) ~= 0
            rates_last(j,users+1+k) = 0.4;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,22);
    elseif (rates_last(j,k) == 12)
        if user_table(k,4+12) ~= 0
            rates_last(j,users+1+k) = 0.8;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,23);
    elseif (rates_last(j,k) == 13)
        if user_table(k,4+13) ~= 0
            rates_last(j,users+1+k) = 1.2;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,24);
    elseif (rates_last(j,k) == 14)
        if user_table(k,4+14) ~= 0
            rates_last(j,users+1+k) = 2;
        else
            rates_last(j,users+1+k) = 0;
        end
        resources_last(j,k) = user_table(k,25);
    end
```

```
        if rates_last(j,k) >= 8
        rates_last(j,(2*users)+3) = rates_last(j,(2*users)+3) + rates_last(j,users+1+k); % determine TR for
RS
        end
   end

    rates_last(j,(2*users)+4) = sum(rates_last(j,users+2: (2*users)+1));

    [ resources_last(j,users+2) ] = UE_direct(rates_last(j,(2*users)+3), Prs, RS_dist); % resources relay
    if (rates_last(j,(2*users)+3) > 0) && (resources_last(j,users+2) == 0)
        resources_last(j,users+2) = 100 ;
    end
    resources_last(j,users+3) = sum(resources_last(j,1:users+2));
    if (sum(resources_last(j,users+3)) > 50) || (any(resources_last(j,1:users) == 0))
        resources_last(j,users+3) = 0;
    end
    if j > (length_f/2)
       if  (resources_last(j,users+3) >= resources_last(j-(length_f/2),users+3)) && (resources_last(j-
(length_f/2),users+3) > 0)
            resources_last(j,users+3) = 0;
       elseif (resources_last(j,users+3) < resources_last(j-(length_f/2),users+3)) &&
(resources_last(j,users+3) > 0)
            resources_last(j-(length_f/2),users+3) = 0;
       end
    end
end


rates_last_safe_b = rates_last;
resources_last_safe_b = resources_last;

% throw away combinations that do not fit the available resources
[length_g,~] = size(rates_last);
[length_gg,~] = size(resources_last);
if length_g ~= length_gg
    resources_last(length_g,9) = 0;
end
loc_d = 1;
for j = 1:1:length_g
    if resources_last(j,users+3) == 0
        to_del_d(loc_d) = j;
        loc_d = loc_d + 1;
    end
end
if loc_d > 1
    rates_last(to_del_d,:) = [];
    resources_last(to_del_d,:) = [];
end
[length_h,~] = size(rates_last);

rates_last_safe_c = rates_last;
resources_last_safe_c = resources_last;

optimal_resources = [];
optimal_rates = [];

% If multiple combinations remain find most optimal

if length_h > 1
    %determine for all combinations the number of upgrades per user from the start rate
    for d = 1:1:users-1
        [~, start_val(1,d)] = find(abs(TR-target_base(1,users+2+d)) < 0.001);
    end

    n = 0; % Rows
    m = 1; % count
    up_val = [];
    loc_dd = 1;
    for f = 1:1:length_h
        flag = 0;
        up_val(f,users) = 0;
        for d = 1:1:users-1
            if (rates_last(f,users+2+d) > target_base(1,users+2+d))
               up_val(f,d) = find(abs(TR-rates_last(f,users+2+d)) < 0.001) - start_val(1,d);
               n = n + 1;
            elseif (rates_last(f,users+2+d) < target_base(1,users+2+d))
                   up_val(f,d) = -10;
                   n = n + 1;
            end
            up_val(f,users) = sum (up_val(f,1:users-1));
        end
        most_imp(m,1) = n;
        m = m+1;
        n = 0;
    end

    if loc_dd > 1
    rates_last(to_del_dd,:) = [];
```

```
    resources_last(to_del_dd,:) = [];
    up_val(to_del_dd,:) = [];
    end
    [length_h,~] = size(rates_last);

    rates_last_safe_c1 = rates_last;
    resources_last_safe_c1 = resources_last;
    up_val_safe_c1 = up_val;

    % Determine the largest increase in data rate. if more than 2 steps
    % setup long combination table.
    most_imp_max_a = max(max(up_val(:,1:users-1))) % largest improvement compared to base value (a and b)

    if (most_imp_max_a > 2) && (aa(2) < 7) % if improvement is larger then 2 and room for improvememt left
       [ rates_last, resources_last, up_val, most_imp ] = Optimal_6_long_run_v02(a, aa, b, bb,user_table,
target_base, TR, users, Prs, RS_dist, PUE_min, PUE_max, maximum)
       [length_h,~] = size(rates_last);
    end

    rates_last_safe_cc = rates_last;
    resources_last_safe_cc = resources_last;
    up_val_safe_cc = up_val;

    % keep rows that show the most individual
    % increases(say 3 out of 5 UEs for example).

    most_imp_UEs_b = max(most_imp(:,1));
    up_val_max = max(up_val(:,users));
    if up_val_max > 0
        loc_e = 1;
        for j = 1:1:length_h
            if sum(up_val(j,1:users-1)~=0) < most_imp_UEs_b
                to_del_e(loc_e) = j;
                loc_e = loc_e + 1;
            end
        end
        if loc_e > 1
            rates_last(to_del_e,:) = [];
            resources_last(to_del_e,:) = [];
            up_val(to_del_e,:) = [];
        end
        [length_i,~] = size(rates_last);


        rates_last_safe_d = rates_last;
        resources_last_safe_d = resources_last;
        up_val_safe_d = up_val;

        % of remaining list chooce the best allocation.

        candidate_1 = rates_last(1,:);
        candidate_1_res = resources_last(1,:);

        if length_i > 1
            for f = 1:1:length_i-1
                can_1(f,:) = candidate_1;
                a = 1;
                diff_can_1 = [0]
                diff_can_2 = [0]
                candidate_2 = rates_last_safe_d(f+1,:)
                candidate_2_res = resources_last_safe_d(f+1,:);
               for d = users+2:1:(2*users)+1
                   if candidate_1(1,d) ~= candidate_2(1,d)
                        diff_can_1(a) = candidate_1(1,d)
                        d_can_1(f,a) = candidate_1(1,d)
                        diff_can_2(a) = candidate_2(1,d)
                        d_can_2(f,a) = candidate_2(1,d)
                        a = a + 1
                   end
               end
               if a > 1
                    count_d = 0
                    [length_i] = length(diff_can_1)
                    for g = 1:1:length_i
                        if diff_can_1(g) < diff_can_2(g)
                            count_d = count_d + 1
                        elseif diff_can_1(g) > diff_can_2(g)
                            count_d = count_d - 1
                        end
                    end
                    if length_i == 1
                        if count_d == 1 % if candidate 2 is better replace
                            candidate_1 = candidate_2
                            candidate_1_res = candidate_2_res;
                        end
                    elseif length_i == 2
                        if (count_d == 2) || ((count_d == 0) && (min(diff_can_1) < min(diff_can_2)))% if
both UEs candidate 2 are beter replace
```

```
                            candidate_1 = candidate_2
                            candidate_1_res = candidate_2_res;
                        elseif ((count_d == 0) && (min(diff_can_1) == min(diff_can_2)) && (max(diff_can_1)
== max(diff_can_2)))
                            if candidate_1_res(1,users+3) > candidate_2_res(1,users+3)
                               candidate_1 = candidate_2
                               candidate_1_res = candidate_2_res;
                            end
                        end
                    elseif length_i == 3
                        if (count_d == 3) || ((count_d == -1) && (min(diff_can_1) < min(diff_can_2))) ||
((count_d == -1) && (min(diff_can_1) == min(diff_can_2)) && (sum(diff_can_1(:)== max(diff_can_1(:)))) <
(sum(diff_can_2(:)== max(diff_can_2(:)))) && (min(diff_can_2) == max(diff_can_2)) )...
                           ((count_d == 1) && (min(diff_can_1) == min(diff_can_2)) && (max(diff_can_1) <
max(diff_can_2)) )
                            candidate_1 = candidate_2
                            candidate_1_res = candidate_2_res;
                        end
                    end
               else
                    if candidate_1_res(1,users+3) > candidate_2_res(1,users+3)
                       candidate_1 = candidate_2;
                       candidate_1_res = candidate_2_res;
                    end
               end
            end
        end
        % this is the optimal combination
        optimal_rates  = candidate_1;
        optimal_resources = candidate_1_res;
    else

        % this is the optimal combination.... except
        optimal_resources  = resources_last(1,:);
        optimal_rates = rates_last(1,:);
        [length_k2,~] = size(rates_last);
        if length_k2 > 1

            for f = 1:1:length_k2 % if there is a cheaper one
                if resources_last(f,users+3) < optimal_resources(1,users+3)
                    optimal_resources  = resources_last(f,:);
                    optimal_rates = rates_last(f,:);
                end
            end
        end

    end

else
    optimal_rates = rates_last;
    optimal_resources = resources_last;
end


end


function [ rates_last, resources_last, up_val, most_imp ] = Optimal_6_long_run(a, aa, b, bb,user_table,
target_base, TR, users, Prs, RS_dist, PUE_min, PUE_max, maximum)

% this function is used for the optimal scheme if the improvement in data
% rate is 3 steps or higher compared to the base value. This saves a lot
% of time calculating the optimal scheme.

  input = {[a(2):7 b(2):14],[a(3):7 b(3):14],[a(4):7 b(4):14],[a(5):7 b(5):14],[a(6):7 b(6):14]};%,[a(7):7
b(7):14]}; %,[a(8):7 b(8):14],[a(9):7 b(9):14]};
  sub_combi = allcomb(input{:});
  [length_e,~] = size(sub_combi);



  to_del_c = [];
  loc_ = 1;
for f = 1:1:length_e
      if ((~any(sub_combi(f,:) == a(2))) && (~any(sub_combi(f,:) == b(2))))
          to_del_c(loc_) = f;
        loc_ = loc_ + 1;
    end
end
sub_combi(to_del_c,:) = [];
[length_f,columns_f] = size(sub_combi);

if PUE_min == PUE_max% setup data rate table (rates_last) and resources table
    for i = 1:1:length_f
        rates_last(i,1) = target_base(1,1); % rate of prio UE
    end
    rates_last(:,2:1+columns_f) = sub_combi(:,1:columns_f);
    rates_last(length_f,(2*users)+4) = 0;
```

```
else
    for i = 1:1:length_f
        rates_last(i,1) = maximum; % rate of prio UE
    end
    for i = length_f+1:1:2*length_f
        rates_last(i,1) = maximum+7; % rate of prio UE
    end
    rates_last(1:length_f,2:1+columns_f) = sub_combi(1:length_f,1:columns_f);
    rates_last(length_f+1:2*length_f,2:1+columns_f) = sub_combi(1:length_f,1:columns_f);
    rates_last(2*length_f,(2*users)+4) = 0;
    length_f = length_f *2;
end

rates_last_safe_c = rates_last;

for j = 1:1:length_f
   for k = 1:1:users
        if (rates_last(j,k) == 1)
            if user_table(k,4+1) ~= 0
                rates_last(j,users+1+k) = 0.1;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,5);
        elseif (rates_last(j,k) == 2)
            if user_table(k,4+2) ~= 0
                rates_last(j,users+1+k) = 0.2;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,6);
        elseif (rates_last(j,k) == 3)
            if user_table(k,4+3) ~= 0
                rates_last(j,users+1+k) = 0.3;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,7);
        elseif (rates_last(j,k) == 4)
            if user_table(k,4+4) ~= 0
                rates_last(j,users+1+k) = 0.4;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,8);
        elseif (rates_last(j,k) == 5)
            if user_table(k,4+5) ~= 0
                rates_last(j,users+1+k) = 0.8;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,9);
        elseif (rates_last(j,k) == 6)
            if user_table(k,4+6) ~= 0
                rates_last(j,users+1+k) = 1.2;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,10);
        elseif (rates_last(j,k) == 7)
            if user_table(k,4+7) ~= 0
                rates_last(j,users+1+k) = 2;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,11);
        elseif (rates_last(j,k) == 8)
            if user_table(k,4+8) ~= 0
                rates_last(j,users+1+k) = 0.1;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,19);
        elseif (rates_last(j,k) == 9)
            if user_table(k,4+9) ~= 0
                rates_last(j,users+1+k) = 0.2;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,20);
        elseif (rates_last(j,k) == 10)
            if user_table(k,4+10) ~= 0
                rates_last(j,users+1+k) = 0.3;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,21);
        elseif (rates_last(j,k) ==11)
```

```
            if user_table(k,4+11) ~= 0
                rates_last(j,users+1+k) = 0.4;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,22);
        elseif (rates_last(j,k) == 12)
            if user_table(k,4+12) ~= 0
                rates_last(j,users+1+k) = 0.8;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,23);
        elseif (rates_last(j,k) == 13)
            if user_table(k,4+13) ~= 0
                rates_last(j,users+1+k) = 1.2;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,24);
        elseif (rates_last(j,k) == 14)
            if user_table(k,4+14) ~= 0
                rates_last(j,users+1+k) = 2;
            else
                rates_last(j,users+1+k) = 0;
            end
            resources_last(j,k) = user_table(k,25);
        end
        if rates_last(j,k) >= 8
        rates_last(j,(2*users)+3) = rates_last(j,(2*users)+3) + rates_last(j,users+1+k); % determine TR for
RS
        end
    end

    rates_last(j,(2*users)+4) = sum(rates_last(j,users+2: (2*users)+1));

    [ resources_last(j,users+2) ] = UE_direct(rates_last(j,(2*users)+3), Prs, RS_dist); % resources relay
    if (rates_last(j,(2*users)+3) > 0) && (resources_last(j,users+2) == 0)
        resources_last(j,users+2) = 100 ;
    end
    resources_last(j,users+3) = sum(resources_last(j,1:users+2));
    if (sum(resources_last(j,users+3)) > 50) || (any(resources_last(j,1:users) == 0))
        resources_last(j,users+3) = 0;
    end
    if j > (length_f/2)
       if  (resources_last(j,users+3) >= resources_last(j-(length_f/2),users+3)) && (resources_last(j-
(length_f/2),users+3) > 0)
            resources_last(j,users+3) = 0;
       elseif (resources_last(j,users+3) < resources_last(j-(length_f/2),users+3)) &&
(resources_last(j,users+3) > 0)
            resources_last(j-(length_f/2),users+3) = 0;
       end
    end
end


rates_last_safe_b = rates_last;
resources_last_safe_b = resources_last;

[length_g,~] = size(rates_last);
loc_d = 1;
for j = 1:1:length_g
    if resources_last(j,users+3) == 0
        to_del_d(loc_d) = j;
        loc_d = loc_d + 1;
    end
end
if loc_d > 1
    rates_last(to_del_d,:) = [];
    resources_last(to_del_d,:) = [];
end
[length_h,~] = size(rates_last);

rates_last_safe_c = rates_last;
resources_last_safe_c = resources_last;

for d = 1:1:users-1
    [~, start_val(1,d)] = find(abs(TR-rates_last(1,users+2+d)) < 0.001);
    start_rate(1,d) = rates_last(1,users+2+d);
end

n = 0; % Rows
m = 1; % count
up_val = [];
Best_impovement = [];
for f = 1:1:length_h
    up_val(f,users) = 0;
    for d = 1:1:users-1
```

```
        if (rates_last(f,users+2+d) > rates_last(1,users+2+d))
           up_val(f,d) = find(abs(TR-rates_last(f,users+2+d)) < 0.001) - start_val(1,d);
           n = n + 1;
        elseif (rates_last(f,users+2+d) < rates_last(1,users+2+d))
           up_val(f,d) = -10;
           n = n + 1;
        end
        up_val(f,users) = sum (up_val(f,1:users-1));
    end
    most_imp(m,1) = n;
    m = m+1;
    n = 0;
end

% largest improvement compared to base value (a and b)
most_imp_max_a = max(max(up_val(:,1:users-1)));

end
```