Distributed Resource Allocation for Device-to-Device Communication in LTE/LTE-A Networks

By
Yngve Lågbu

Supervisor
Frank Y. Li

Thesis report for IKT 590 Master thesis, spring 2015
Department of Information and Communication Technology
Faculty of Engineering and Science
University of Agder

May 26, 2015

Abstract

Currently, 3GPP is working on device-to-device (D2D) communications, which is regarded as a part of 5G mobile communication systems. This thesis focuses on allocation of radio resources, based on a scenario where out-of-coverage user equipments (UEs) communicate with a base station (BS), through an user equipment relay (UE-R) located within the cell coverage. Two algorithms have been developed, one focusing on adjusting different parameters to achieve higher performance in the developed system. Whilst the other algorithm targets at battery depended resource allocation including four battery control allocation schemes. The outcome from the first algorithm indicates that the performance is not highly depending on parameter configuration, but leaning towards how spectrum is shared by different users. The results from the second algorithm show that the performance of the studied network relies highly on the battery level of the UE-Rs and therefore they need to have strict battery control policies to exhibit show the benefit for in-band D2D communications.

Keywords: Resource Allocation, Device-to-Device Communications, Long-Term Evolution, Battery Control Schemes, Handshake Protocols.

Preface

This work is the result of the Master thesis in IKT 590 at the University of Agder as part of the Master program in Information and Communication Technology. The thesis is equivalent to 30 ECTS and started on Jan. 2, 2015 and ended on May 26, 2015.

I would like to thank my supervisor Professor Frank. Y. Li for his feedback and guidance with the thesis. I would also like to say thank you to Dr. Benoit Pelletier from InterDigital Canada Ltée for initiating this thesis topic.

Special thanks to my brother, Amund Lågbu. Without his continuous support and encouragement, I never would have been able to achieve my goals.

Yngve Lågbu
Grimstad
May 26, 2015

Contents

List of Tables

List of Figures

Acronyms

3GPP	3rd Generation Partnership Project
5G	Fifth generation wireless communications
BS	Base Station
D2D	Device-to-Device
FDD	Frequency Division Duplex
HTTP	Hypertext Transfer Protocol
LTE	Long-Term Evolution
LTE-A	Long-Term Evolution Advanced
MIMT	Man-in-the-Middle
MSC	Mobile Switching Centre
NMT	Nordic Mobile Telephone
OFDM	Orthogonal Frequency Division Multiplexing
QAM	Quadrature amplitude modulation
QoS	Quality of Service
QPSK	Quadrature phase-shift keying
RB	Resource Block
RE	Resource Element
UE	User Equipment
UE-E	User Equipment End-User
UE-R	User Equipment Relay
VoIP	Voice over Internet Protocol

1 Introduction

This chapter introduce the background and motivation behind the thesis followed by the research questions. Then, we describe the methodology used, before the approach and assumptions are presented, followed by the outline of the thesis.

1.1 Background and Motivation

Currently a standardization of D2D is being established. According to 3rd Generation Partnership Project TR 36.843, D2D is a technology that can benefit the requirements on the way to the fifth generation wireless communications. The possible use cases for D2D communications are numerous. Fig.1, presents five such use cases.

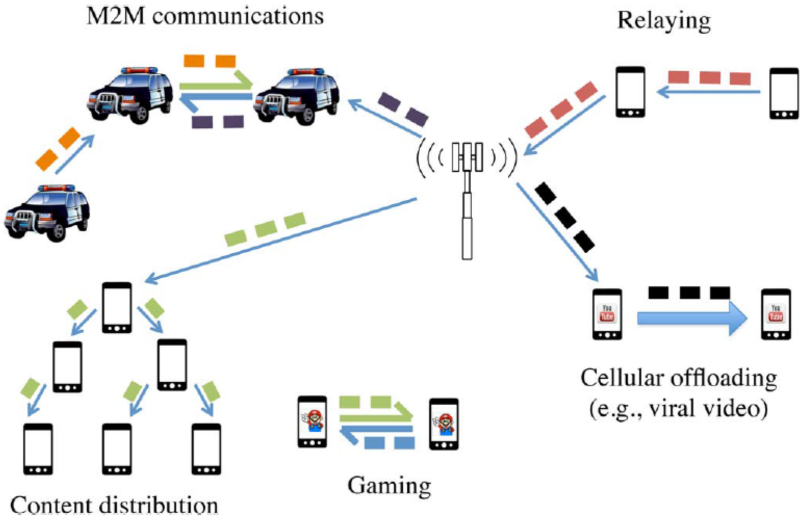


Figure 1: D2D use cases [1]

3GPP TR 36.843 V12.0.1 (2014-03) presents four D2D scenarios, one of these scenarios consists of a UE-R relaying data between the BS and a user equipment end-user (UE-E). The UE-R relays data over a direct link to UE-Es located outside of the cellular network coverage. When surveying how to allocate fair radio resources in this scenario, few published papers were found in the literature. Therefore, this thesis intends to contribute to this interesting research topic and focuses on resource allocation algorithms in this scenario considering one or multiple UE-Rs.

1.2 Research Questions

In this thesis we will develop and analyze two algorithms. The basic algorithm intends to answer the following research questions:

1. When a user is outside the coverage of a cell but can be reached via a relay, how can we allocate resources to the user?
2. Consider D2D end users require different services. How to allocate resources in a reasonable way?
3. How about the performance of the proposed algorithm, considering factors like cellular traffic usage, type of service etc?
4. In which way will different parameter configurations affect the systems performance?

In the advanced algorithm battery level will be considered and the research questions are extended as:

5. Consider UE-Rs have control over their battery level and UE-Es have different requirements for radio resources. How can we select a UE-R which has higher bettery level and how can this UE-R allocate resources to other UE-Es?
6. In which degree does the algorithm perform with fair allocation of resources amongst UE-Rs?

1.3 Methodology

This project will be solved in iterations. An iteration will represent a work period with a short duration - either a week or the time it takes between the supervisor meetings. Each iteration will be solved by using the same approach by following a sort of a recipe containing phases. Starting with a literature review, to obtain or refresh knowledge. Secondly, a design phase - all modules will be created in such phases. Thirdly, an implementation phase. In this phase the created modules will be combined into a system. The design and implementation phases will contain validations of the code, to see that the correct data is obtained. Then the data will be analyzed in the fourth phase. Here the simulation results will be compared with theoretical results, to validate that the system works according to theory. The final phase will contain an evaluation of the acquired results from phase four. After all phases are completed, the iteration is complete. The next cycle starts will use the gained knowledge from the previous iteration, and this information will be used to extend the modules by implementing new more advanced requirements. This cycle will continue until the whole system is complete [4].

1.4 Approach and Assumptions

The approach for the basic algorithm is as follows:

1. Develop a module that consists of a UE-R which receives broadcasts messages from the BS. With that information, the UE-R can decide which UE-E to serve considering the service requirements for the UE-E.
2. Develop an algorithm which handles UE-E requests and makes decision on delaying, blocking, terminating and providing services.
3. Implement a user interface with different configurable parameters, such as number of LTE time frames, maximum number of UE-Es in the network, different UE-E service requests as well as the results after resource allocation.

The approach for the advanced algorithm is as follows:

4. Develop an algorithm which makes resource allocation decision based on battery consumption levels considering battery level of UE-Rs and the service requests from UE-Es.

The limitations and key assumptions for the algorithms are as follows:

- There are several proposals on how to allocate resources using D2D communications, by using either the out-band or the in-band in overlay or underlay mode. Thus, this thesis will focus on resource allocation in a scenario where UE-Rs provides D2D communications to out-of-coverage UE-Es over the in-band in an overlay mode.
- Due to different wireless technologies and channel conditions the simulation results from both algorithms will vary from real-life networks.

- In a deployed network a service can last several minutes, or even hours. Therefore, the duration of the services in this thesis is simplified by drastically shorten the duration of the services. This will lead to a system with unrealistic service durations. However, the output results will be valid, since every component of the system is scaled down to match the simplification.
- The UE-R's own usage is not considered in this thesis, due to the possibility for a high cellular usage by the UE-R at different times in the simulation. Which could lead to a high number of blocked, terminated and delayed sessions during those time intervals.
- Selective connectivity between the UE-Rs and the UE-Es was considered, but this neglects the purpose which D2D communications are to use in proximity services, thus the system have no connectivity restrictions.

The limitations for the basic algorithm are:

- The basic algorithm is developed in Python, and an analysis of an outdoor deployed system cannot be obtained. Therefore, the results in this thesis can vary from a real-life systems.
- The delay values in the system are obtained from delayed D2D transmissions between a UE-R and the UE-Es, and used as a tool to illustrate and provide results that shows which elastic services are delayed at certain times in the simulations.

The assumptions for the advanced algorithm are:

- The advanced algorithm is developed in Java and therefore the results may vary from a deployed outdoor network.
- The UE-Rs use identical batteries and their batteries will decrease at equal rates for each connected UE-E in the system. In a real-life system the UE-Rs will have different battery capacities and their consumptions will decrease with different rates.
- UE-Rs cannot be connected to a battery charger, thus battery power is limited. This algorithm focus on how to allocate resources in a fair way amongst UE-Rs considering their battery levels by energy efficient means. If the UE-Rs were connected to battery chargers, they would be able to

only use a high efficient battery scheme and an algorithm that considers their battery levels would not have been of interest. Nevertheless, a high efficient battery scheme is proposed to illustrate how UE-Rs can allocate a high amount of resources with a high battery power consumption. Also, a linear battery scheme is introduced to illustrate how the UE-Rs would allocate resources if the connectivity follows the battery consumption linearly and an energy efficient scheme is proposed as an approach to enable low battery consumption sharing or selective sharing of cellular resources.

1.5 Thesis Outline

In this Master thesis, we will present two algorithms as an approach to enhance allocation of radio resources in a cellular network. A basic algorithm is developed to observe how parameter configuration in a system affects the systems performance. Firstly, one system is tested to validate the systems output. Secondly, three cases with minor parameter adjustments are compared. Then, the mean value of these cases over 100 simulations are analyzed in a scenario with limited resource allocated to D2D communications. Lastly, we compare how using different modulation techniques affect the system.

Four battery allocation schemes are purposed for allocation of resources based on UE-Rs battery level, and one scheme is implemented into the advanced algorithm. We will perform a test of the system to analyze the fairness of the UE-Rs battery level compared with their allocated resources.

The outline of this thesis is as follows:

- Chap. 2 describes the enabling technologies.
- Chap. 3 presents the chosen scenarios and the designed protocols.
- Chap. 4 illustrates the implementation of the systems.
- Chap. 5 presents the obtained results for the basic algorithm.
- Chap. 6 introduces the results acquired for the advanced algorithm.
- Chap. 7 contains a discussions about the outcome for each algorithm.
- Chap. 8 represents the conclusions and future work.

2 Enabling Technologies

This chapter presents a few relevant technologies which constitute the basics for the work performed in this thesis.

2.1 Mobile Communications

In 1981, Nordic Mobile Telephone (NMT) system was established in Norway and Sweden and one year later in Denmark and Finland. This was the world's first established fully automatic telecommunication network that could operate across country boarders. This network could only provide voice services [5].

In the early 90's the second generation mobile communications emerged. This network was digital and could provide voice and data services such as short message, paging and email. The third generation mobile communications could provide voice, data and multimedia services. This generation mobile communications systems led to a new area in mobile communication called broadband communication. The fourth generation mobile communications called LTE, introduced new technologies for the physical layer like Orthogonal Frequency Division Multiplexing (OFDM) and Beam-forming. Later a new advanced version called LTE-A was introduced, this technology is not a revolution compared to LTE, it's more like an evolution [6]. Recent two years 5G has been a hot topic in both academia and industry which handles mobile communication beyond 2020.

2.2 Device-to-Device Communications

As part of 5G, D2D communication is specified as direct communication amongst two mobile user equipment without crossing the base station[1] or core network. D2D communication is normally translucent to the cellular network and it can take place on the cellular spectrum called in-band, or on the unlicensed spectrum called out-band. In a common cellular network a prerequisite is that all communication has to go through the base station despite that they are in range of D2D communication. This composition appeal to traditional low data rate mobile services like text message and voice call which UEs are not commonly close enough for direct connection. Nevertheless, mobile users today use high data rate services that possibly could be in range for D2D communication, such as video sharing and gaming. In this case D2D communication

---

[1] In this thesis we use BS and enodeB interchangeably.

could greatly enhance the spectral efficiency of the network. Potentially D2D can improve throughput, delay, fairness and energy efficiency [1]. Fig.2 shows that D2D can be either in-band or out-band. For in-band D2D communication both underlay and overlay modes can be adopted. In this thesis we focus on overlay.

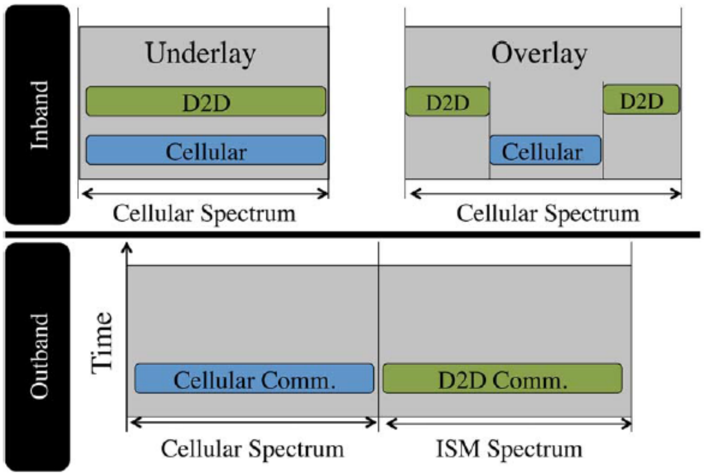


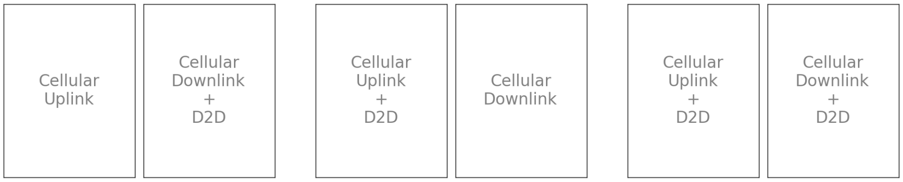
Figure 2: D2D concepts [1]

2.3 Radio Resource Allocation

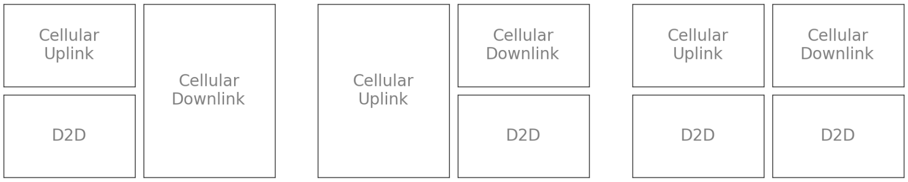
Radio resource includes channels in LTE/LTE-A typically referred to as radio blocks. Resource allocation can be performed by the BS or the MSC. In this thesis we consider that resource allocation decisions are handled by the relay node within the coverage of the cell, and it provides services to other UE-Es outside the network. More information will be provided in Chapter 3.

2.4 Spectrum Sharing for D2D Communication

There are several strategies on how to share the given radio resources. 3GPP assumes that D2D operates in the up-link band when using FDD [2]. However, 3GPP does not specify how to accommodate D2D communications beside the current cellular communication. This can be accomplished by either sharing the resources or allocating dedicated resources to D2D communications [7], the possibilities are presented in Fig. 3.



(a) Shared Down-link (b) Shared Up-link (c) Shared Down and Up-link



(d) Up-link Assigned D2D (e) Down-link Assigned D2D (f) Up and Down-link Assigned D2D

Figure 3: Strategies on how to accommodate D2D communications beside the cellular communication

From Fig. 3a) an unshared cellular up-link with a shared D2D and cellular down-link band. The second scheme in Fig. 3b) has a shared D2D and cellular up-link band with an unshared cellular down-link. Fig. 3c) presents a shared D2D and cellular up-link band and a shared D2D and cellular down-link. From Fig. 3d) D2D is allocated a part of the cellular up-link resources with an unshared cellular down-link. In Fig. 3e) the cellular up-link is unshared and D2D is allocated a part of the down-link band. Lastly, in Fig. 3f) D2D is allocated a part in both the up-link and down-link band.

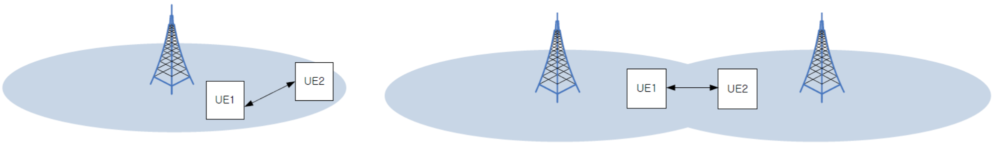
3 Scenarios and Protocol Design

Firstly, the scenarios presented by 3GPP are described in this chapter. Then, the identified scenarios for this thesis are described, followed by the proposed algorithms and an illustration of a protocol sequence diagram.

3.1 3GPP Scenarios

In this section different four different D2D scenarios from 3GPP TR 36.843 V12.0.1 (2014-03) are presented in Fig. 4.

(a) Scenario 1A (b) Scenario 1B



(c) Scenario 1C (d) Scenario 1D

Figure 4: D2D scenarios presented by 3GPP TR 36.843 [2]

In Scenario 1, Fig. 4a), two UEs communicate directly with each other without any cellular interference from the BS. Scenario 2. Fig. 4b), presents a UE-R that is covered by the BS and acts as a relay between an out-of-coverage UE-E and the BS. A BS cover two UEs, that utilize a direct link without any involvement from the BS in Scenario 3, Fig. 4c). Scenario 4, Fig. 4d) shows two UEs are covered by different BSs and the UEs can communicate with each other and act as relays between the two cellular cells.

3.2 Scenarios Identified for this Thesis

The focused scenario in this thesis is Scenario 2. Fig. 4b), using several UE-Es located outside-of-coverage instead of one as presented in the 3GPP scenario. The UE-R[2] is covered by the BS and located at the edge of the cell, and it will broadcast D2D relay invite messages to UE-Es outside the cell. The UE-Es will reply the UE-Rs broadcast messages, and try to connect to the D2D network in order to achieve connectivity. When a UE-E request for D2D communication reach the UE-R, it have to check its requirements before it can either provide or block the D2D request. Two scenarios are identified in this thesis, one targeting to enhance resource allocation performance by parameter configurations, whilst the other is focusing on the UE-Rs battery levels to provide fair resource allocation. The chosen scenarios are two ways to provide outside coverage UE-Es network connectivity, as required in Sec.1.

For the basic scenario, a UE-R and a set N number of UE-Es will be used. This algorithm will analyze parameters such as delay, block, termination and completion of services. Two different services, real-time and elastic will be used. VoIP will represent the real-time services, whilst HTTP denotes the elastic services. This scenario also analyze the use of VoIP priority, to see how this configuration affects the systems performance. The key point of this algorithm is to observe and analyze how parameter adjustments affects the performance of the system when allocating resources. Fig. 5 presents the scenario for the basic algorithm.

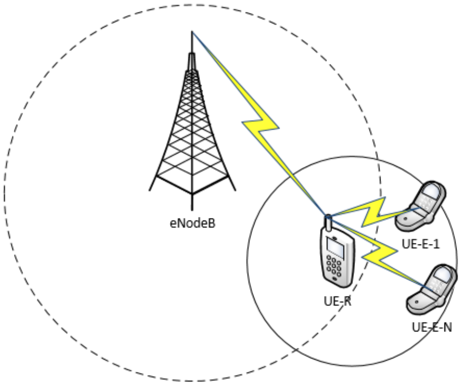


Figure 5: Basic scenario

---

[2] In this thesis we use UE-R and relay UE interchangeably.

The advanced scenario targets UE-Rs battery levels, the system consists of a set N number of UE-Rs and a set number n of UE-Es. The UE-Es can connect at different times in the simulation, and the algorithm will analyze only the UE-Rs battery consumption. The main point of this protocol is to obtain fair resource allocation amongst UE-Rs in a system based on their battery level. The scenario which is considered in the advanced algorithm is illustrated in Fig.6.

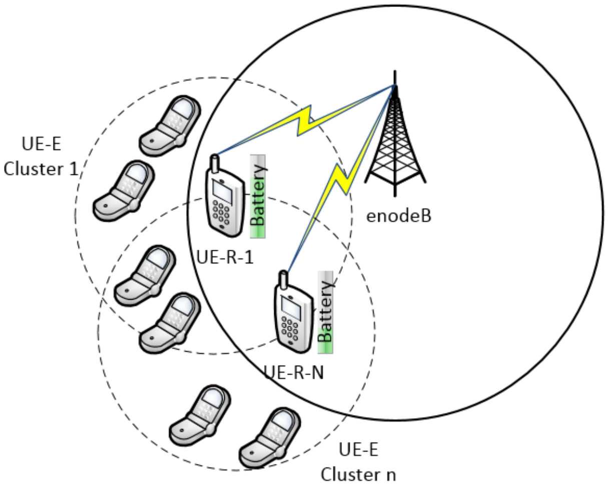


Figure 6: Advanced scenario

3.3 Resource and Service Type Considerations

3.3.1 LTE Time Frames

In Fig.7 a general Long-Term Evolution (LTE) Frequency Division Duplex (FDD) frame is illustrated. Each time frame has a duration of 10 ms and consists of 10 sub-frames each with 1 ms duration. Each sub-frame are then divided into two slots with 0.5 ms duration. A Slot is equal to one resource block (RB) in the time domain and consists of 7 symbols. The Symbol length is affected by the length of the cyclic prefix. This gives a total of 20 RBs, during each LTE time frame [3].

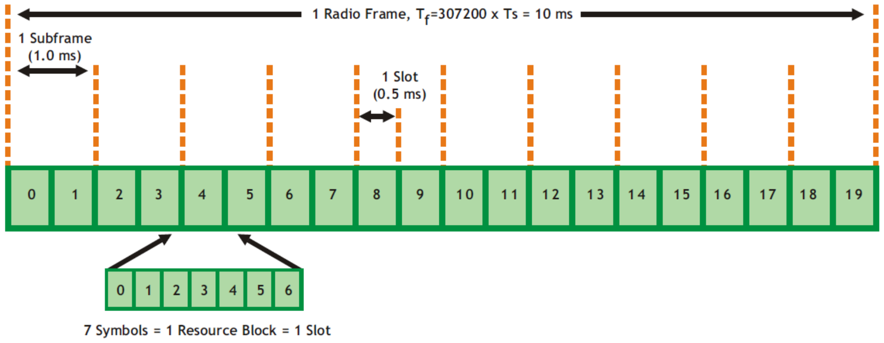


Figure 7: The LTE time frame structure [3]

3.3.2 Resource Blocks and Resource Elements

Fig. 8 provides an illustration of the relationship between symbols, slots and resource blocks. In Fig. 8, a RB consists of a number of resource elements (REs), that is the smallest component of a RB. The number of REs in one RB is provided by multiplying the sub-carriers with the number of symbols. In this figure a normal cyclic prefix is illustrated (7 symbols), but the length of the cyclic prefix can also be 6 symbols (short cyclic prefix). In this thesis we use as illustrated in the figure, 12 sub-carriers and 7 symbols. Hence, each RB consists of 12 sub-carriers x 7 symbols = 84 REs. The transmission bandwidth is depended on the channel bandwidth, the transmission bandwidth is denoted by the number of active transmitting RB in a LTE time frame. Thus, if the bandwidth increases then the number of transmitting RBs also enhance [3].

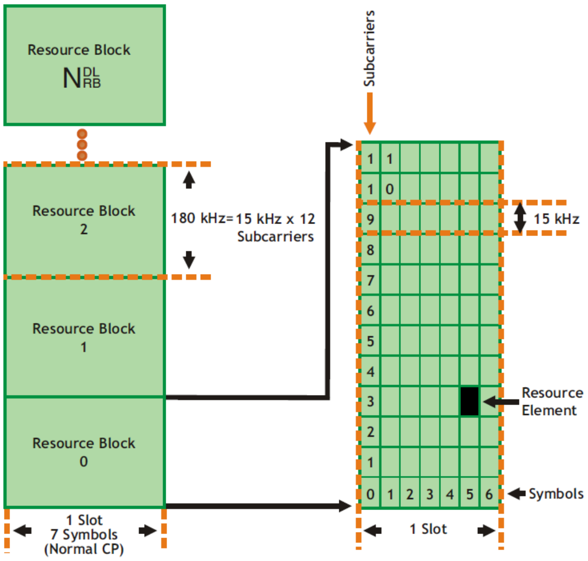


Figure 8: Resource block architecture [3]

3.3.3 Service Types

In this thesis, two different types of services are considered, elastic and real-time services. The duration of an elastic service is affected by the data rate, whilst real-time services are not [8]. There are several protocols that falls under the elastic and real-time service categories. However, in this thesis the Hypertext Transfer Protocol (HTTP) represents the elastic services and Voice over Internet Protocol (VoIP) classifies the real-time division. Tab. 1 presents the required LTE data rate[3] to provide a service with respect to the modulation scheme, which remains constant throughout this thesis. The calculation of the presented values are described in Appendix. A.

---

[3] In this thesis the LTE data rate and the number of REs is used interchangeably.

Table 1: Required number of REs to provide a service type using different modulation schemes

Service Type	Modulation	Number of REs
VoIP	QPSK	28
HTTP	QPSK	60
VoIP	16-QAM	14
HTTP	16-QAM	30
VoIP	64-QAM	10
HTTP	64-QAM	20

3.4 The Designed Handshake Protocol

Fig. 9 presents a proposed protocol that allow UE-Rs the ability to allocate resources to UE-Es based on their available resource pool, as required in Sec.2. A BS broadcasts available resources to be allocated for D2D communication. When the UE-R receives a service request from a UE-E it can either provide the service request or block the request. If the request is granted, the UE-R will reserve the number of sufficient resources to provide the service. Then it will respond the UE-E with a service request granted message. The UE-E will then transmit its data to the BS, with the UE-R acting as a relay between the BS and UE-E. When the session is complete, the UE-E will transmit an end session message to the UE-R, and the UE-R will reallocate the used resources for that session. If the UE-R does not have sufficient resources to provide the service, it will reply the UE-E with a block service request message.

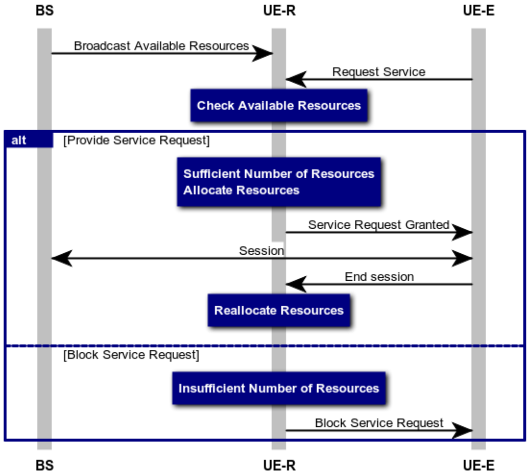


Figure 9: Basic handshake protocol

3.5 Main Modules

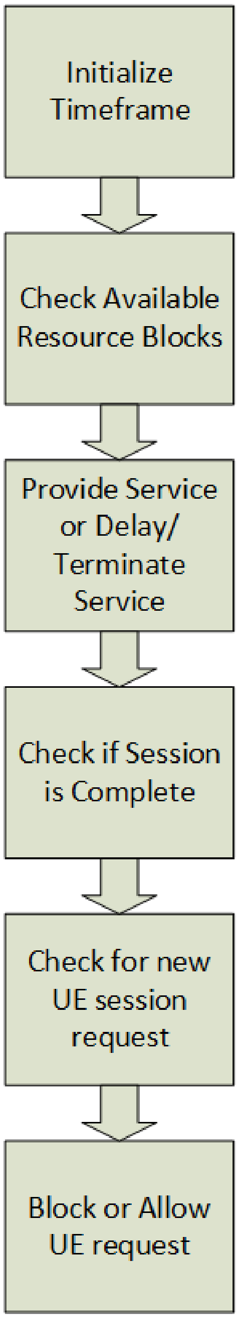


Figure 10: Main modules in the system

Fig. 10 illustrates an LTE time frame iteration. The description is as follows:

- Initialize Time frame: For each iteration the LTE time frame counter is incremented by one, until it reach its range limit. In the initialization of the LTE frame the cellular use in the network is decided by using a set probability parameter.
- Check Available Resource Blocks: The number of available Resource Blocks are different for each LTE time frame. For each time a session receives resource for its desired service the number of resources to be allocated decrease.
- Provide Service or Delay/Terminate Service: If there is a sufficient amount of resources to provide, the UE-R will provide the service for that time frame. If there is an insufficient amount of resources the UE-R will check if the service is an HTTP or VoIP service. The HTTP services will be delayed by incrementing a delay buffer value until it reach a set threshold, then they will be terminated. While, VoIP will be terminated.
- Check if Session is Complete: If all UE-Es have been provided either their service or been blocked/terminated. The UE-R will check the provided UE-Es, to see if their session is complete. If it is, the UE-R will reallocate its resources, such that the other UE-Es in the D2D network can utilize them.
- Check for new UE session request: At the end of the time frame, the UE-R will check if there are any new UE-Es that tries to connect to the D2D network.
- Block or Allow UE request: If there are less than the set UE-E threshold in the D2D network and that all connect UE-Es have been provided their services. The UE-R will allow the new UE-Es to connect to the D2D network, if not it will block the new UE-E requests.

4 Implementation and User Interface

In this chapter, we will present the developed modules that used to answer the research questions in Sec. 1.2 along with the motive of that decision . The modules were designed following the requirements for the approach in Sec.1.4. Then they were combined into a system following the methodology described in Sec. 1.3.

4.1 Implementation of the Basic Algorithm

This section starts with a description of the how the basic modules are combined to form a system, then the system is extended as described in Sec. 1.3, to form a more advanced system. The modules will be presented according to Fig. 10.

4.1.1 Basic Modules

In order to solve Approach 1 in Sec. 1.4, first a basic time frame generator module was created. Based on Sec. 3.3.1, the module was developed for generating LTE time frames to simulate the BSs broadcast messages to the UE-R. Each time frame is generated as an array with a length of 20, where each of the 20 positions representing an RB. The time frame module was designed as an array to make it easy, to set the value of each array value either to 0 or 1 for all 20 array positions. Setting the value to 0, will cause the cellular network to occupy the RB. Whilst, setting the value to 1, will reserve the RB for D2D communication.

The second module created was a UE-E generator module that defined the number of UE-Es in the network as well as an uniform probability distribution function for the type of UE-E requests. The last module developed was sharing the allocated resources. When a UE-E request a service, the number of required REs will be compared with the number of available resources to either provide or deny the service.

Then these modules was combined to form a system. The number of resources available are broadcast from the LTE time frame generator module, then this module allocates available resources to UE-Es generated from the UE-E module. This designed module allocates resources to the UE-Es, starting with the first position in the UE-E generator array. When this UE-E is provided a services, the number of REs for its service will be subtracted from the total number of resources available in the LTE time frame. Then the next UE-E position will be served, until either all UE-Es are provided their required number of REs, or until their are no more available resources in the LTE time frame. If there are an insufficient number of REs for all UE-E requests, the module will print out the UE-Es array position and that the required service has not been provided for that LTE time frame. This system consist of low requirement basic modules with static values and the UE-R can reach UE-Es outside of coverage and make decision about which UE-E to serve or block. Tab. 2 illustrates an overview of the parameter configurations in the system containing only basic modules.

Table 2: Parameter configurations from the developed basic modules

Description	Value
LTE Time Frames	100
Number of UE-Es	10
Cellular Resource Use	50 %
Type of Service	50 % HTTP + 50 % VoIP

4.1.2 Advanced Modules

To fulfill the requirements of Approach 2 in Sec. 1.4. Modules that handles the ongoing requests and services had to be made. A delay buffer was developed, to keep track of the different delays occurring in each flow over the simulation duration. When the system does not have sufficient resources to provide the requested service, this module checks if the service is either an HTTP or VoIP service. If the service is an elastic service, the system will check the status of the UE-E in the delay buffer. If the status value is below a set termination threshold, the module will increment the UE-Es delay flow value. However, if the value extends or equals the threshold value, the status value will be reset, then the UE-Es service will be terminated. When a UE-E with a real-time service enters the module, it will be terminated. All terminations and delays for each UE-E flow will be listed in both a delay and a termination array. When the simulation is complete, these values will be used to plot the UE-Es delay and termination patterns. Using the summation of the delay and termination array will provide the total delay and termination over the simulation and the summation of all UE-Es during a LTE time frame presents the delay over each time frame.

The handle the blocked UE-Es, another module was created. When the time frame has ended and at least one UE-E service is complete, this module checks if any UE-Es have been blocked or terminated. If the available resource requirements have been met for all UE-Es the module will include another UE-E to the system. Thus, if there are any delayed or blocked UE-E sessions, the module will block the new UE-Es attempt to connect to the relay node. For each blocked UE-E a counter is incremented for graphical visualization.

The modules was implemented into the system and when the LTE time frame has insufficient resources for all requests the delay and termination module is initialized. If one UE-E session is complete, the add or block UE-E request module will be initialized at the end of the LTE time frame.

To obtain an graphical outcome for each service completed, a service complete module was developed. Whenever, a session is complete the service complete module is initialized. For this module three arrays have been designed, one for VoIP completed sessions, another for finished HTTP services and the last array for graphical three dimensional mapping. When the module is initialized it will first check if the service is real-time or elastic. Then depending on the service, either the VoIP or HTTP completed array will be incremented by one. Secondly, the graphical mapping array will add 1 for VoIP or 2 for HTTP as value, such that the array values can be separated for plotting purposes.

Lastly, a VoIP Priority module was designed to set service prioritizing in the system. Since VoIP services cannot tolerate any delay, the priority module will sort the sessions such that real-time services always are handled before HTTP sessions by the UE-R.

4.1.3 LTE Time Frame Generator Module

Fig. 11 describes how the sub frames are divided for either cellular use or D2D use in an LTE time frame. Firstly, a parameter p is set in the user interface, see Sub chapter 4.3.1. p denotes the binomial probability for each sub frame. The sub frames are listed in an array with 20 symbols. A value i denotes each sub frame position in the array, and i is incremented by one for each array position until all sub frames has been evaluated.

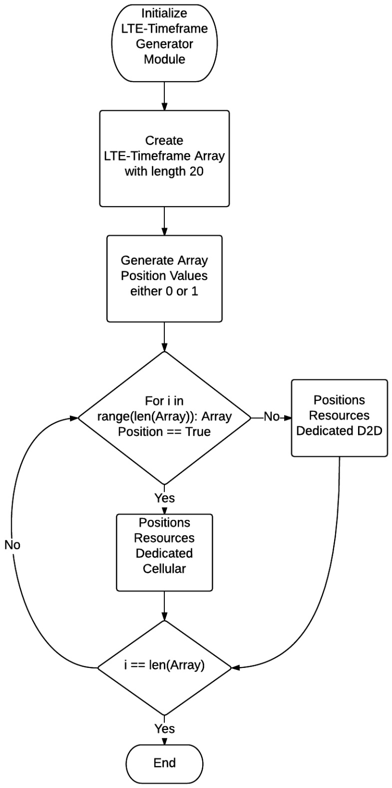


Figure 11: LTE time frame generator module

4.1.4 Service Request Module

Fig. 12 contains a representation of the service request module. First the module initializes the users input values obtained at the start of the simulation. An array will be generated for the UE-R with the number of UE-Es chosen as length. Then the module generates a UE-E for each position in the array, based on the probability parameter p. Next, the module will check each UE-E to see if it is a HTTP or VoIP request and set the array position to either 0 or 1 depending on the selected service request. Then, a new parallel array will be generated with identical length to the UE-R array, this array illustrates the time frame duration of the UE-Es. The integer values generated follows an exponential distribution with a mean value of 10 time frames. Lastly, the exponential array is incremented by one to avoid a zero position in the exponential array.

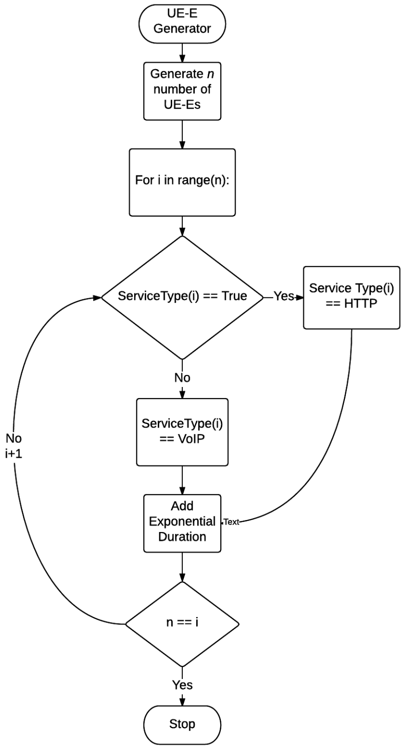


Figure 12: Service request module

4.1.5 Resource Allocation Module

Fig. 13 shows how the resources are allocated in the system. The figure illustrates one sub frame, during one LTE time frame. First the system initialize the requested service from UE-E, either an VoIP or HTTP request. Then the needed REs elements for the request are implemented into a variable, 60 REs for HTTP and 28 for VoIP. The system will then check the resources in the first time frame, if the resources are not sufficient to provide a service, another sub frame is added together with the present sub frame. This is done up to four times when using the up-link band. If the sum of the four sub frames has not sufficient resources to provide a service, the termination and delay module will be initialized. However, if the sum of one of the four sub frames are sufficient the number of REs will be subtracted from the sub frames and the service will be provided. In this system this can occur up to ten times, depending on the number of UE-Es located in the system.

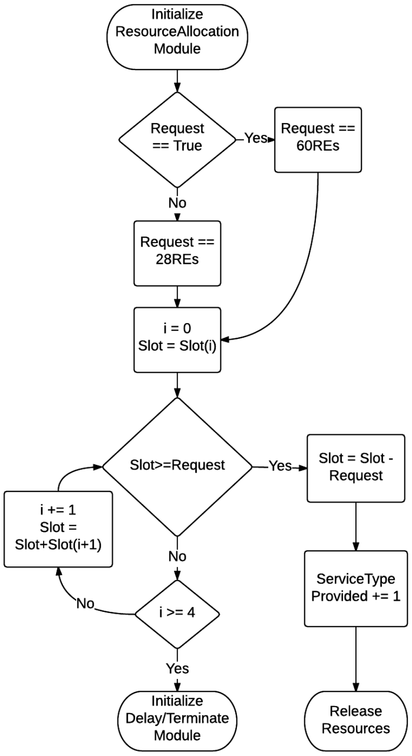


Figure 13: Resource allocation module

4.1.6 Delay and Termination Module

A block scheme on the the delay and termination module is illustrated in Fig. 14. If there are not enough resources to provide all services, the system will enter the delay and termination module. This module will first check if the service is HTTP or VoIP. If the service is HTTP, the module will check the UE-Es data flow delay buffer value. If the value is below a threshold, the delay buffer will be incremented by one and the data flow will be delayed one time frame. If the delay buffer value is equal or larger than the threshold value, the delay buffer will be set to zero and the data flow will be terminated. If the service is VoIP, the UE-Es session will be terminated.

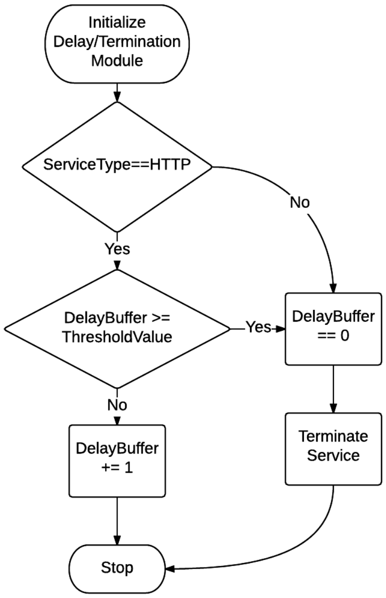


Figure 14: Delay and termination module

4.1.7 Block or Add New UE-Es Module

This module either blocks or adds a new UE-E to the D2D network cluster. From Fig. 15 this module gets initialized when a UE-Es duration reach zero. The UE-R will first check the available D2D resources to see if a new UE-E can be added to the network. Reason for this is that this module is initialized after all UE-Es have been served. If some of the served UE-Es are either terminated or delayed or that threshold number of resources are extended, the UE-R will block the new UE-E request. If not, the UE-R will add the new UE-E to the system. The new UE-Es service request is based on the input value of the probability parameter p, denoted at the start of the program. Then the module will check if the request is HTTP or VoIP and add either 0 or 1 to the UE-R array. An exponential value is then added, and the value is incremented by one to avoid a zero array position.

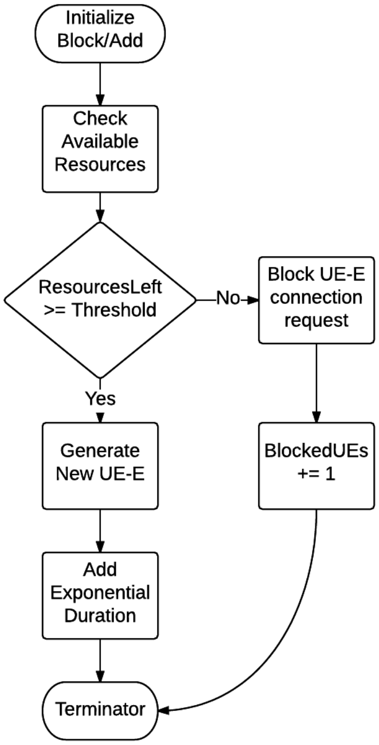


Figure 15: Block or add new UE-Es module

4.1.8 VoIP Priority Module

The VoIP priority setting can be set in the user interface by inserting 1 in the VoIP priority input. Fig. 16 illustrates how the VoIP priority module works.

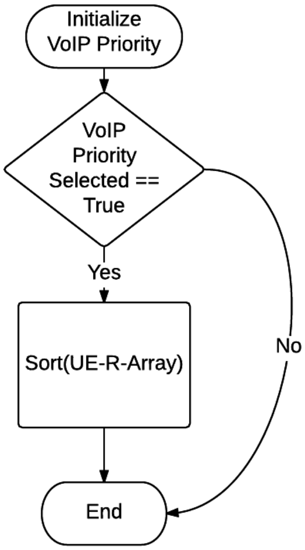


Figure 16: VoIP priority module

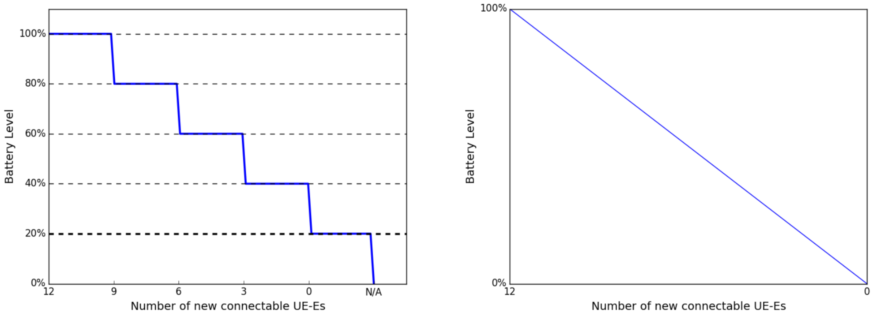
4.2 Implementation of the Advanced Algorithm

In order to utilize in-band D2D communication the UE-Rs requires a battery protocol to monitor and control the UE-Rs battery capacity and efficiency. In this chapter we propose a protocol and different schemes to ensure fair resource allocation in the network.

4.2.1 Battery Control Schemes

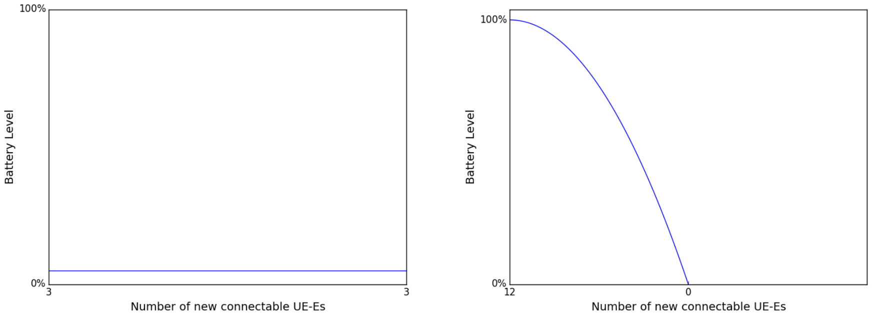
According to Shneiderman's "Eight Golden Rules of Interface Design", users desire the impression that they are in control of the system and that the system reacts to their behavior [9]. In order to provide UE-Rs a sense of control over their UEs when they are sharing their cellular connection, four battery control schemes have been proposed in this subsection. There will be a description of each scheme, along with a brief depiction of its area of use.

To make the figures below more comprehensible a UE-E value range is set. The max. value for the UE-Es connected to a UE-R with 100% batter power is intentionally set to 12. Each UE-R can allocate up to 4 RBs, which leads to the maximum number of VoIP sessions per UE-R using Quadrature phase-shift keying (QPSK) modulation, where one $RB = 84REs$, that gives $(4 * 84REs)/28REs = 12$ connected VoIP sessions.



(a) Hybrid battery control scheme

(b) Linear battery control scheme



(c) Energy efficient battery control scheme

(d) High efficient battery control scheme

Figure 17: Proposed battery control schemes

In Fig. 17a) the number of connectable UE-Es are regulated by a battery thresholds. The idea is that a UE-R with a high battery level close to 100 %, can connect to the maximum number of UE-Es allowed in the system. Then, when the the UE-Rs battery level drop pass a specific threshold value the number of connectable UE-Es will decrease.

A linear battery control scheme is illustrated in Fig. 17b), instead of using a hybrid scheme like in the previous figure, this scheme use a linear battery value. Hence, the UE-Rs efficiency to allow UE-Es to connect to the D2D network decreases linearly with the battery level.

If a UE-R wants to be connectable over a longer period of time, an battery efficiency scheme can be utilized, Fig. 17c) presents such scheme. By using the efficiency scheme, UE-Rs can have few connectable UE-Es while depleting the battery at a constant level.

Fig. 17d) shows a high efficiency battery scheme. This scheme allows the UE-Rs to have a high number of connectable UE-Es, the battery will be depleted at a faster rate without any restrictions to the battery level.

Each scheme in Fig. 17 has its own usage area. The hybrid scheme enables an intelligent use of the UE-Rs battery, by introducing different battery level restrictions on the number of connectable UE-Es. To illustrate an example, if a UE-R with 100% battery level can have maximum 12 UE-Es connected at once, this value will decrease to 9, when the battery level decreases below 80%. Then, 6 when the battery level reach 60 %, 3 UE-Es at 40 % and 0 UE-Es when the battery level is 20 %. The UE-R will stop relaying UE-Es at 20 % so that the UE-R can have enough battery level left to use the UE when the D2D communication relaying is finished. Also, the UE-R will not shut down during the relaying phase.

The linear scheme is only based on the UE-Rs battery level and the number of UE-Es will decrease along with the battery level. This can be a more efficient way to share the resources based on battery compared to the hybrid scheme. However, there is a higher need of monitoring the battery level to connectable UE-Es ratio and a more rapid changes in connectable UE-Es, that could lead to an increase in the UE-Rs battery consumption.

The energy efficient scheme focus more on either one UE-R that provided few UE-Es with connectivity, as well as UE-Rs that want to share some of their connection when they are not using all their allocated resources. An example of this scheme is when UE-Es are located in a building with poor cellular connection. Then the UE with cellular connection, enables UE-R mode such that its resources can be shared with the other UEs without connectivity. Thus, all UE-Es will obtain access to the network and cellular connectivity over a longer period of time.

The high efficiency scheme is based on either a crisis scenario or an user that is charging his UE while providing D2D services. If an earthquake crisis scenario occurs, and victims cannot be located. The help-aid could enable UE-R mode to be able to provide D2D connectivity to locate victims without any battery restrictions. If the victims have a mobile phone and no cellular connections they could obtain broadcast messages from the UE-Rs such that a cellular connection is provided.

Another usage area, is that if a UE-R is charging his battery, the user may want to share more resources without restrictions to the battery level. If the user have 20 % battery left, and starting to charge the UE. There is no need for restrictions for the battery level, since it will only give false information according to the restrictions in the schemes.

4.2.2 Battery Level Handshake Protocol

In this subsection the resource allocation protocol based on battery level is presented. This protocol is an extension from the handshake protocol represented in Fig. 9.

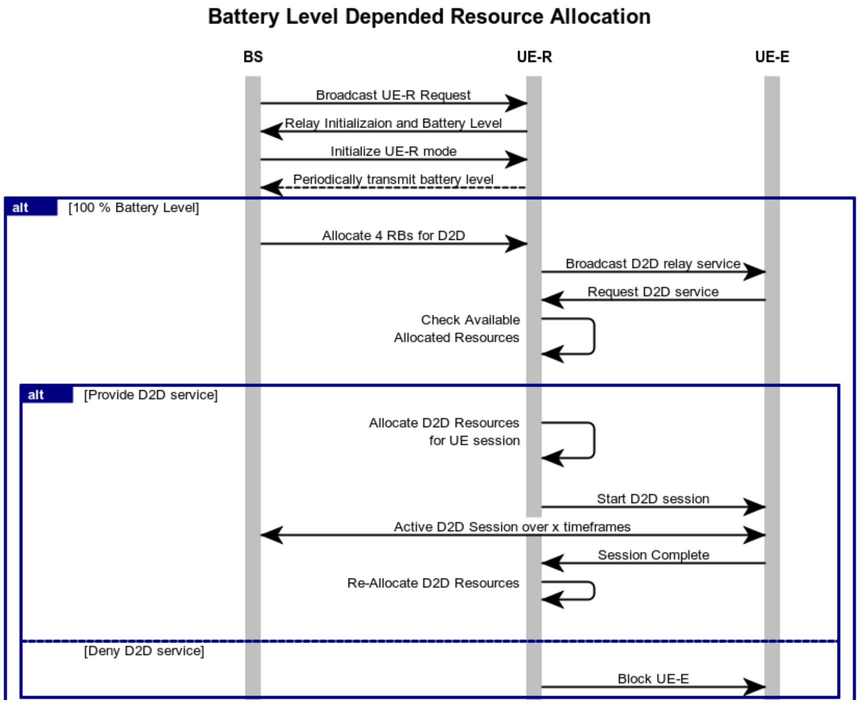


Figure 18: Battery based handshake protocol part 1

Fig.18 presents the negotiation between the BS and the UE-R, and the provide or deny service. The protocol starts with the BS broadcasting UE-R invite messages. The potential UE-Rs will reply with an initializing message together with their battery level. If the UE-R is accepted as a relay UE the BS will send an initialize UE-R mode message and the UE is then linked as a UE-R with the BS. The UE-R can either transmit its battery levels to the BS or have an internal battery scheme. If the battery level of the UE-R is 100 % then the UE-R can be allocated the maximum amount of resources if needed by the BS, in this figure the maximum number is four RBs. Then the UE-R will broadcast an D2D relay service invite message with its position and available resources.

If a UE-E is nearby and outside of cell coverage, the UE-E can reply with an D2D request message. The UE-R will check available resources before it decides to block or provide service to the UE-E. If the UE-R grants provision of relaying the service, it will transmit a start D2D session message. Then it will relay the received transmission both way from the UE-E to the BS and visa verse. When the session is complete, the UE-E will transmit a complete session message to the UE-R. The UE-R will then release the allocated resources. If the UE-R has insufficient resources to provide a service, it will reply with a block message, the UE-E can then retry to obtain D2D relay service.

The procedure to end the UE-Rs relay mode is illustrated in Fig.19, first the UE-R have to send an end relay mode message to the BS. If there are any ongoing UE-E sessions, the BS will reply with a request a handover of UE-Es message. The handover of ongoing UE-Es can either be a success or a failed attempt.

In a successful handover is that the UE-R locates nearby UE-Rs with capacity to take accept UE-Rs sessions. The UE-R will then handover its ongoing sessions, then send a successful handover message to the BS. The BS will reply with an acknowledge message that ends the current UE-R mode, as well as release the UE-Rs RBs.

If the handover is unsuccessful the UE-R is not capable of locating other UE-R with sufficient capacity to accept the ongoing UE-E sessions. The UE-R will then transmit an unsuccessful handover message to the BS. The BS will then reply with a force UE-R mode message. The UE-R will stop broadcasting the D2D service relay message, and finish its ongoing sessions. When all sessions are finished it can reply with a new end message to the BS. The BS will end the session and release the UE-Rs resources.

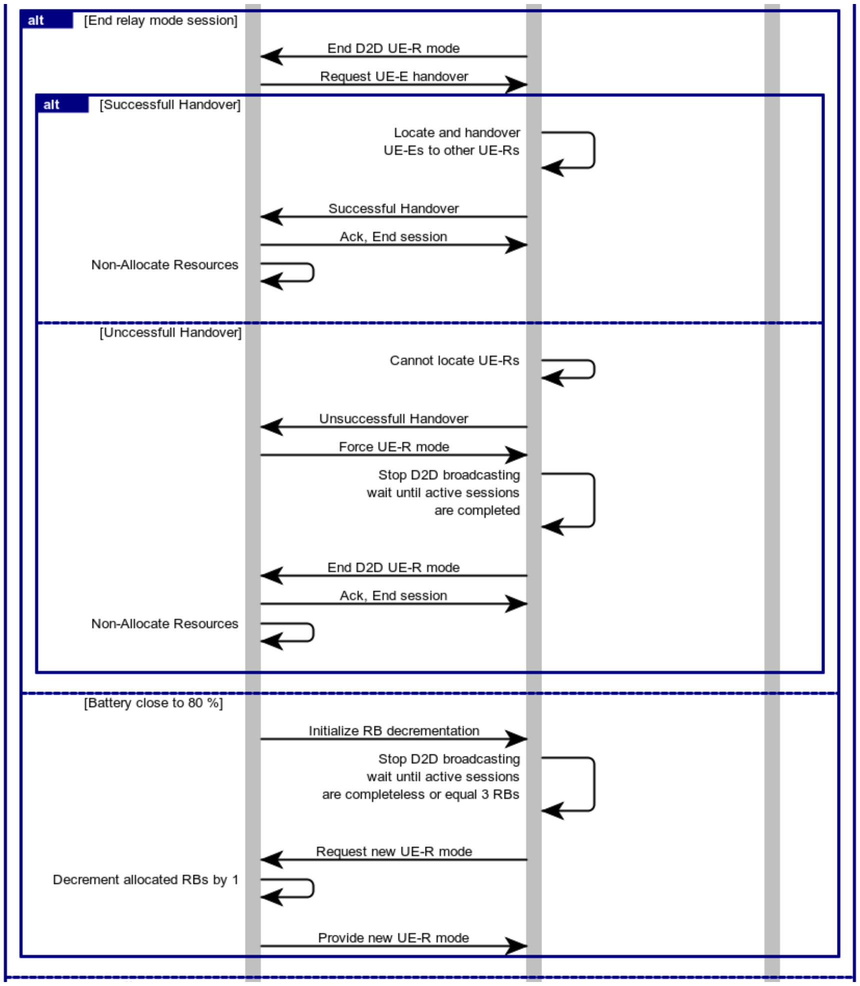


Figure 19: Battery based handshake protocol part 2

From Fig.20 when a UE-Rs battery level decrease beyond a set threshold, for all battery levels in the figure except below 20 %. Either the BS will transmit an increment RB message, if the UE-R is broadcasting its battery level. If the UE-R is not broadcasting, it has to transmit a low battery message. The UE-R will then finish the UE-E sessions until the number of sessions matches the number of allocated RBs with the BS. The UE-R will then transmit a new UE-R mode message, the BS will release resources, based on the UE-Rs battery level. Before, it transmits an provide new UE-R mode message. This figure illustrates an example with four RBs for 100% battery level, 3 RB for 80%, 2 for 60% and 1 for 40%.

When the UE-Rs battery level reach or is getting close to 20 %, the BS will transmit an end UE-R mode message to the UE-R. The UE-R will either handover its ongoing sessions or wait until all ongoing sessions are complete. Then it will reply with its own end UE-R mode message. The BS will reply with an acknowledge and end UE-R mode message, as well as releasing the UE-Rs resources.

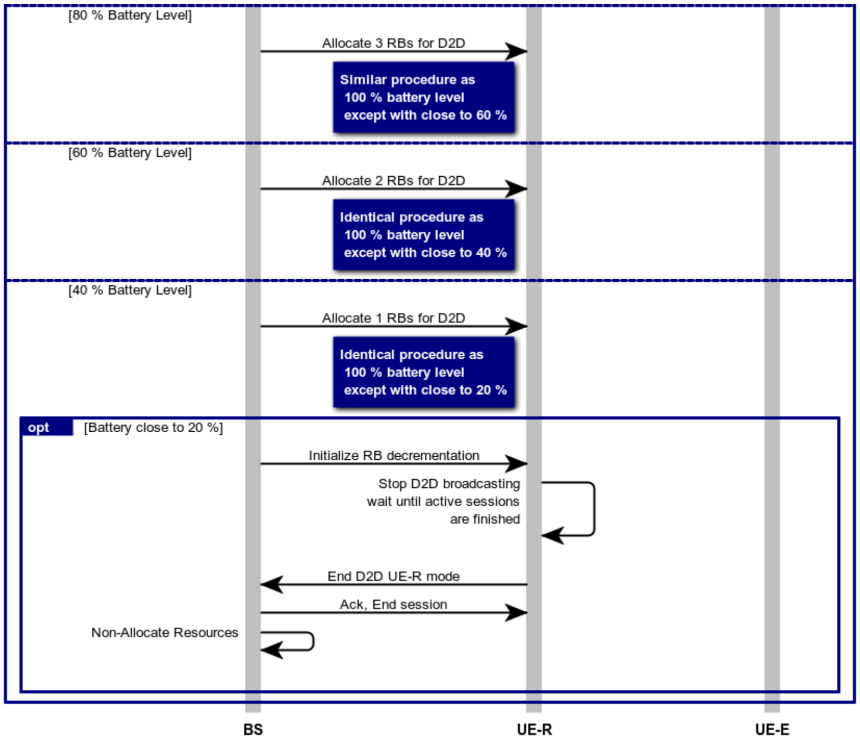


Figure 20: Battery based handshake protocol part 3

4.2.3 UML Diagram for the Advanced Algorithm

Fig. 21, presents an UML diagram of the battery level depended resource protocol. The description of the system and its classes along with the program code in Appendix B.

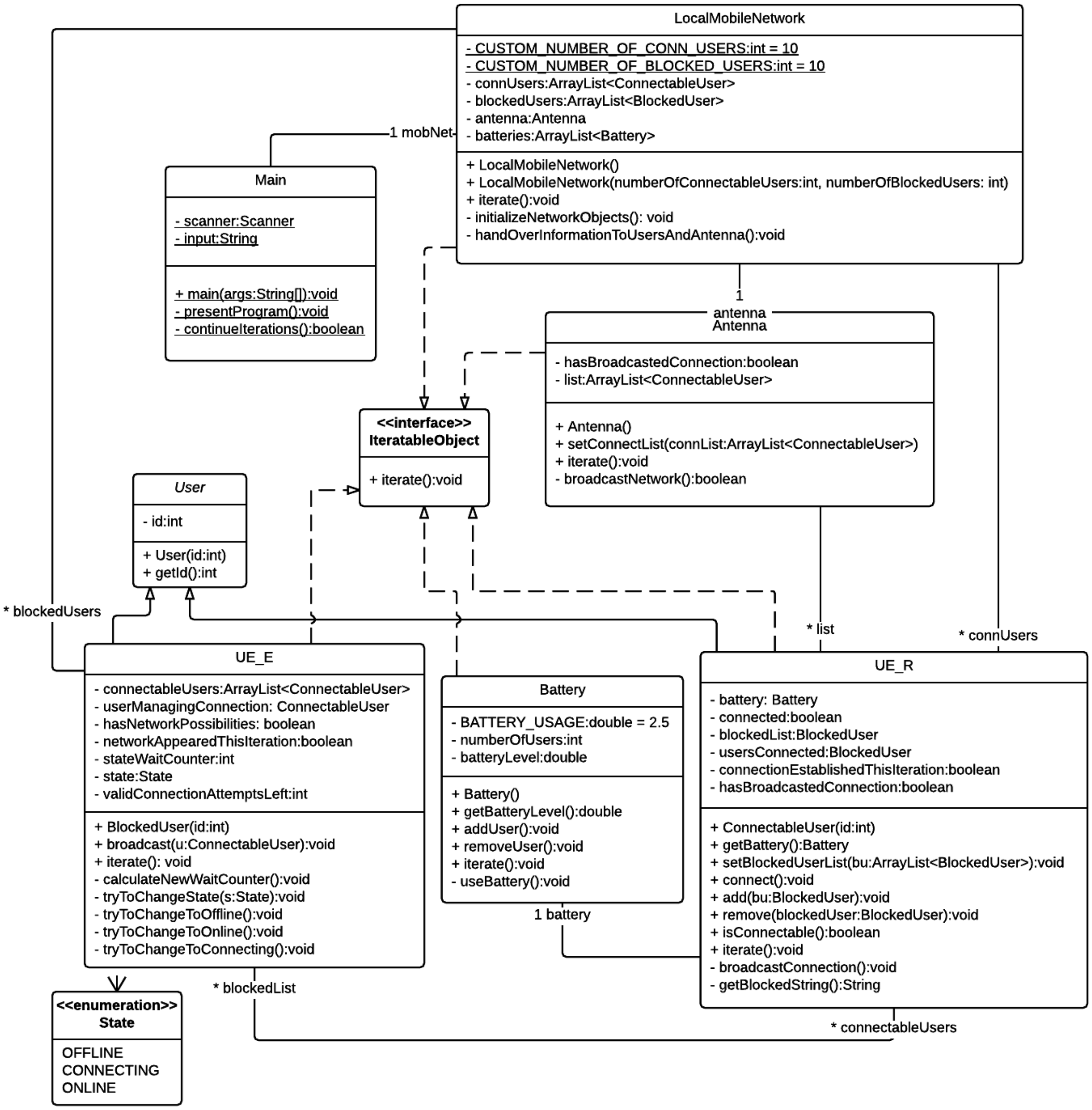


Figure 21: UML diagram for the advanced algorithm

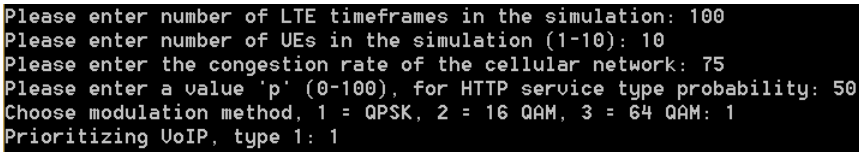
4.3 User Interfaces

4.3.1 User Interface for Basic Algorithm

In Approach 3, Sec. 1.4 the requirement is a more dynamic system, by letting the user set some key parameters before the system is initialized. To fulfill the requirements set, the parameter values had to be changed from static to dynamic. Also, a higher level of user freedom had to introduced to the system. Firstly, the number of LTE time frames and UE-Es in the system had to be made dynamic. This was done by setting their values to undeclared variables that is declared by an input function initialized by the user at the start of the program. If the user does not set any value, a default value will be set.

The probability that whether a RB is occupied by cellular resources or D2D communications and the probability for which type of service a UE-E request is set in input functions by the user. The user can input a value from 0 - 100 in each case, this value is then used in a Binomial Probability Distribution function for each input. Instead of setting either 0 or 1 to as the array value in the LTE time frame generator module, the Binomial probability is taken for each array position. For the type of service possibility, the uniform distribution was replaced with an Binomial distribution, using the input value as the probability indicator. Both initialized UE-Es at the start of the program and the new UE-Es that connect to the UE-R during the simulation follows the Binomial distribution.

In Fig. 22 an example of the user interface is presented with following parameter values, 100 LTE time frames, 10 UE-Es, 50% chance that each RB is occupied by the cellular network, 50% HTTP and 50% VoIP probability, QPSK modulation and VoIP priority enabled.

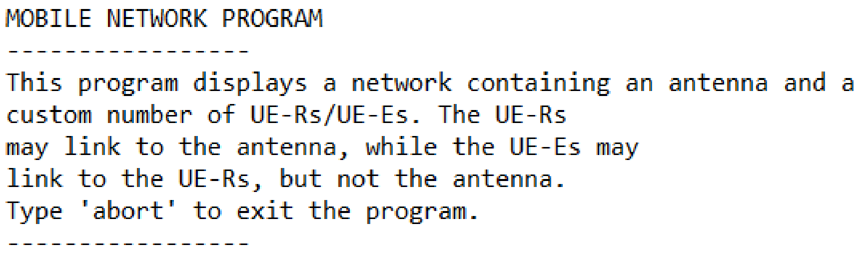


```
Please enter number of LTE timeframes in the simulation: 100
Please enter number of UEs in the simulation (1-10): 10
Please enter the congestion rate of the cellular network: 75
Please enter a value 'p' (0-100), for HTTP service type probability: 50
Choose modulation method, 1 = QPSK, 2 = 16 QAM, 3 = 64 QAM: 1
Prioritizing VoIP, type 1: 1
```

Figure 22: User interface for the basic protocol

4.3.2 User Interface for Advanced Algorithm

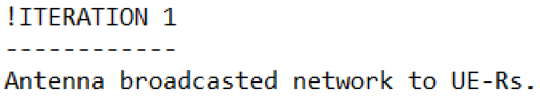
When launching the console, first some information about the program is presented as in Fig.23.



```
MOBILE NETWORK PROGRAM
-----------------
This program displays a network containing an antenna and a
custom number of UE-Rs/UE-Es. The UE-Rs
may link to the antenna, while the UE-Es may
link to the UE-Rs, but not the antenna.
Type 'abort' to exit the program.
-----------------
```

Figure 23: System description at start up

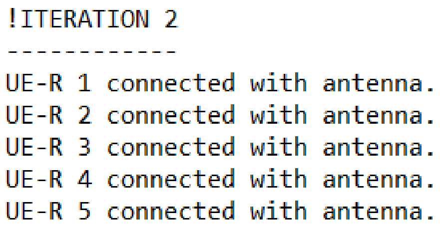
Then by pressing Enter, the program will start as illustrated in Fig.24 , by initializing the first iteration.



```
!ITERATION 1
------------
Antenna broadcasted network to UE-Rs.
```

Figure 24: Console iteration 1

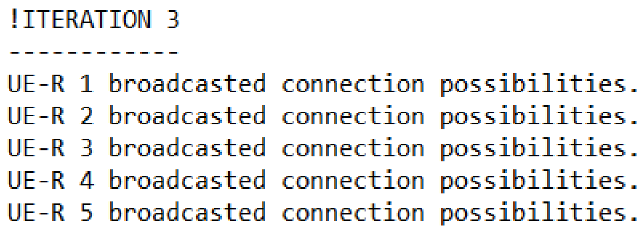
The system will iterate for each time the user press Enter, while the system is running. In Iteration 1, the BS broadcast its resources to the UE-Rs. Then, the UE-Rs will try to connect to the antenna as in Iteration 2, see Fig.25.



```
!ITERATION 2
------------
UE-R 1 connected with antenna.
UE-R 2 connected with antenna.
UE-R 3 connected with antenna.
UE-R 4 connected with antenna.
UE-R 5 connected with antenna.
```

Figure 25: Console iteration 2

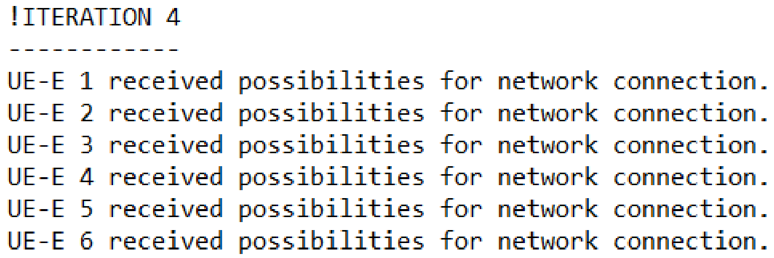
The UE-Rs will broadcast their connection possibilities to the UE-Es as shown in Fig.26.



```
!ITERATION 3
------------
UE-R 1 broadcasted connection possibilities.
UE-R 2 broadcasted connection possibilities.
UE-R 3 broadcasted connection possibilities.
UE-R 4 broadcasted connection possibilities.
UE-R 5 broadcasted connection possibilities.
```

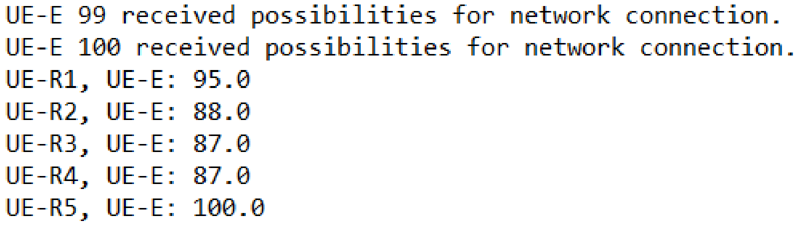
Figure 26: Console iteration 3

Next, the UE-Es will receive the connections broadcast by the UE-Rs. The UE-Es id will be listed in the console output illustrated in Fig. 27a), together with the UE-Rs id and battery level, presented in Fig. 27b).



```
!ITERATION 4
------------
UE-E 1 received possibilities for network connection.
UE-E 2 received possibilities for network connection.
UE-E 3 received possibilities for network connection.
UE-E 4 received possibilities for network connection.
UE-E 5 received possibilities for network connection.
UE-E 6 received possibilities for network connection.
```

(a) Console iteration 4a

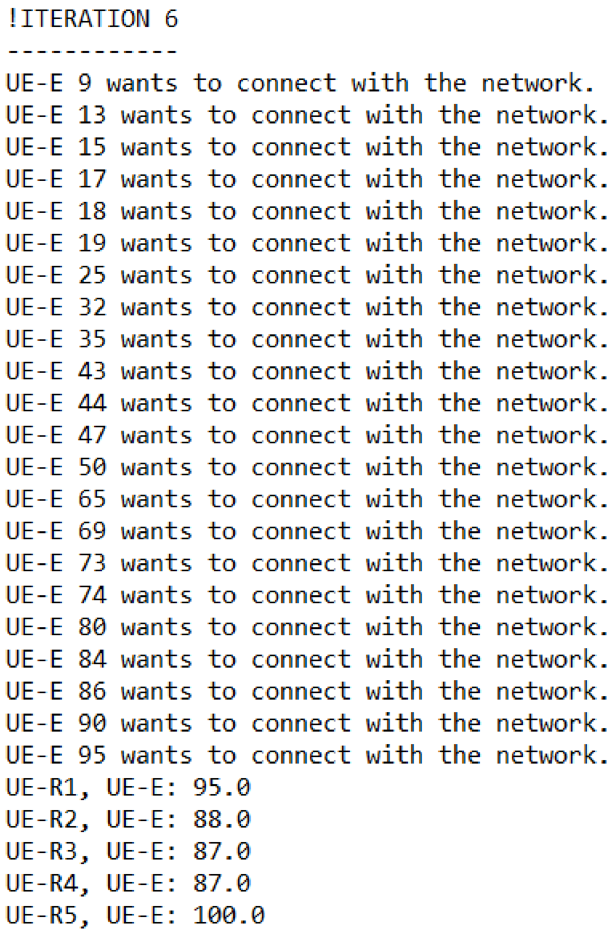


```
UE-E 99 received possibilities for network connection.
UE-E 100 received possibilities for network connection.
UE-R1, UE-E: 95.0
UE-R2, UE-E: 88.0
UE-R3, UE-E: 87.0
UE-R4, UE-E: 87.0
UE-R5, UE-E: 100.0
```

(b) Console iteration 4b

Figure 27: Console iteration 4

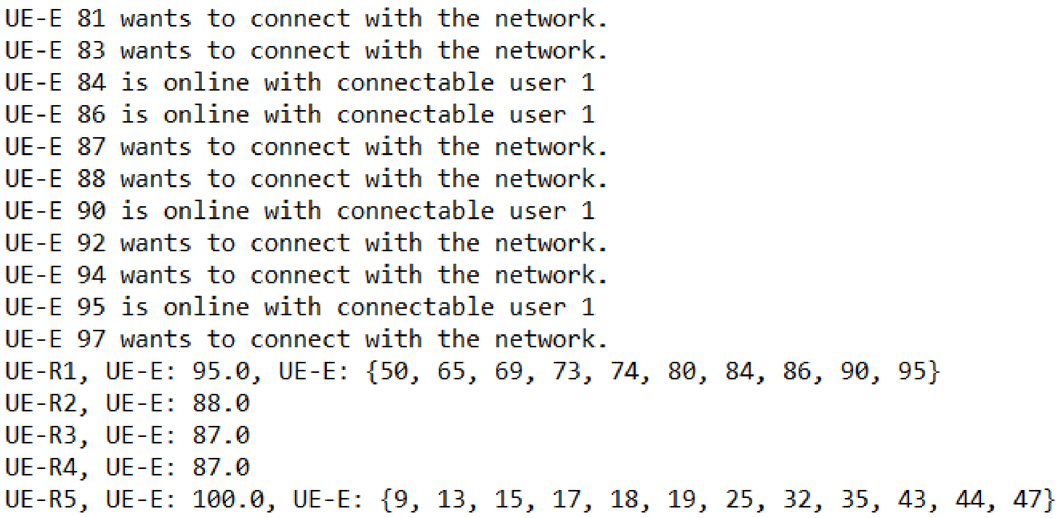
In Iteration 5, no events occur due to the delay UE-Es needs to figure out what to do. However, in Iteration 6, UE-Es transmits connection requests shown in Fig.28 .



```
!ITERATION 6
------------
UE-E 9 wants to connect with the network.
UE-E 13 wants to connect with the network.
UE-E 15 wants to connect with the network.
UE-E 17 wants to connect with the network.
UE-E 18 wants to connect with the network.
UE-E 19 wants to connect with the network.
UE-E 25 wants to connect with the network.
UE-E 32 wants to connect with the network.
UE-E 35 wants to connect with the network.
UE-E 43 wants to connect with the network.
UE-E 44 wants to connect with the network.
UE-E 47 wants to connect with the network.
UE-E 50 wants to connect with the network.
UE-E 65 wants to connect with the network.
UE-E 69 wants to connect with the network.
UE-E 73 wants to connect with the network.
UE-E 74 wants to connect with the network.
UE-E 80 wants to connect with the network.
UE-E 84 wants to connect with the network.
UE-E 86 wants to connect with the network.
UE-E 90 wants to connect with the network.
UE-E 95 wants to connect with the network.
UE-R1, UE-E: 95.0
UE-R2, UE-E: 88.0
UE-R3, UE-E: 87.0
UE-R4, UE-E: 87.0
UE-R5, UE-E: 100.0
```

Figure 28: Console iteration 6

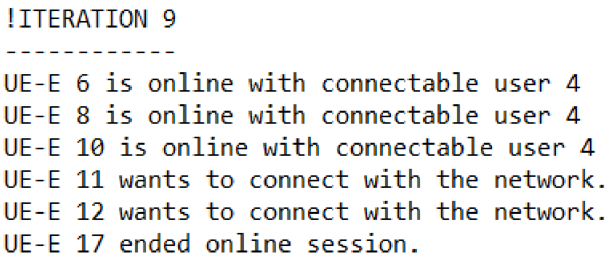
The UE-Es that are accepted by the UE-Rs will be online in Iteration 7, whilst the UE-Es that did not receive connections will retry connection requests. Also, new UE-Es will try to connect to the network, as in Fig.29.



```
UE-E 81 wants to connect with the network.
UE-E 83 wants to connect with the network.
UE-E 84 is online with connectable user 1
UE-E 86 is online with connectable user 1
UE-E 87 wants to connect with the network.
UE-E 88 wants to connect with the network.
UE-E 90 is online with connectable user 1
UE-E 92 wants to connect with the network.
UE-E 94 wants to connect with the network.
UE-E 95 is online with connectable user 1
UE-E 97 wants to connect with the network.
UE-R1, UE-E: 95.0, UE-E: {50, 65, 69, 73, 74, 80, 84, 86, 90, 95}
UE-R2, UE-E: 88.0
UE-R3, UE-E: 87.0
UE-R4, UE-E: 87.0
UE-R5, UE-E: 100.0, UE-E: {9, 13, 15, 17, 18, 19, 25, 32, 35, 43, 44, 47}
```

Figure 29: Console iteration 7

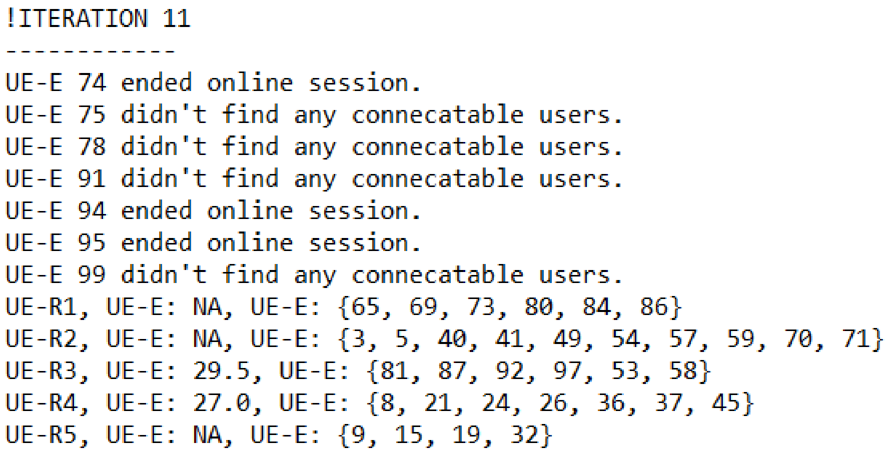
When a UE-E ends its session, the console will output the UE-Es id together with the text, ended online session as in Fig.30.



```
!ITERATION 9
------------
UE-E 6 is online with connectable user 4
UE-E 8 is online with connectable user 4
UE-E 10 is online with connectable user 4
UE-E 11 wants to connect with the network.
UE-E 12 wants to connect with the network.
UE-E 17 ended online session.
```

Figure 30: Console iteration 9

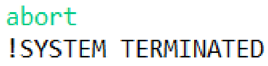
Presented in Fig.31. When a UE-E cannot connect to the network over 3 iterations, the console will output the UE-Es id along with the text, didn't find any connectable users. UE-R1, UE-R2 and UE-R5 are below the battery threshold level. Thus, their status has changed to N/A.



```
!ITERATION 11
------------
UE-E 74 ended online session.
UE-E 75 didn't find any connecatable users.
UE-E 78 didn't find any connecatable users.
UE-E 91 didn't find any connecatable users.
UE-E 94 ended online session.
UE-E 95 ended online session.
UE-E 99 didn't find any connecatable users.
UE-R1, UE-E: NA, UE-E: {65, 69, 73, 80, 84, 86}
UE-R2, UE-E: NA, UE-E: {3, 5, 40, 41, 49, 54, 57, 59, 70, 71}
UE-R3, UE-E: 29.5, UE-E: {81, 87, 92, 97, 53, 58}
UE-R4, UE-E: 27.0, UE-E: {8, 21, 24, 26, 36, 37, 45}
UE-R5, UE-E: NA, UE-E: {9, 15, 19, 32}
```

Figure 31: Console iteration 11

The system will continue to run, until abort is entered into the console input bar or if the system reach 10.000 iterations. By typing abort, the system will output !SYSTEM TERMINATED, as in Fig.32.



```
abort
!SYSTEM TERMINATED
```

Figure 32: Console abort

5 Numerical Results from the Basic Algorithm

To answer the research questions in Sec. 1.2, different test cases are analyzed. In the first part of this chapter a description on the different static parameter configurations are listed. Then, the obtained outcomes from four systems using different parameter configurations are presented to verify and observe the systems nature by using graphical visualized results for all occurred events during the simulations. Next, three cases are simulated 100 times and the average results from the tests are compared and evaluated. Lastly, two different modulations techniques are analyzed to see how the modulation schemes affects the systems performance.

5.1 Parameter Configuration

In Tab. 3 the constant parameters that will appear in all cases are presented.

Table 3: Parameters for the basic algorithm

Parameters	Value	Type
Bandwidth	5 MHz	
Modulation		QPSK, 16-QAM, 64-QAM
Subcarriers per RB	12	
Cyclic Prefix length	7	Normal Cyclic Prefix
RB per LTE time frame	20	
LTE time frame duration	10 ms	
RB time duration	0.5 ms	
Type of services		HTTP, VoIP
Number of UE-Rs	1	
Number of UE-Es	10	

Tab. 4 represents the parameters used in each of the four cases analyzed.

Table 4: Parameter configurations for all cases

Case	Cellular Resource Use	Type of Service	VoIP Priority
1	0	50 % HTTP + 50 % VoIP	No
2	75 %	50 % HTTP + 50 % VoIP	Yes
3	75 %	75 % HTTP + 25 % VoIP	Yes
4	75 %	25 % HTTP + 75 % VoIP	Yes

5.2 Case 1

In Case 1 the simulation parameters are described in Tab. 5.

Table 5: Parameter configurations for Case 1

LTE Time Frames	Cellular Resource Use	Type of Service	Modulation	VoIP Priority
100	0	50 % HTTP + 50 % VoIP	QPSK	No

In Fig. 33 the provided services over the duration of the simulation are illustrated. The 3D plot have one axis called UE[4] number and illustrates the data flows for each UE-E. The Timeframes axis, represents the number of time frames over the simulation. The vertical axis called Required Resource Elements, denotes the provided service values such as HTTP, VoIP and N/A[5]. When a UE-E is terminated, the N/A value will portray a black color. In this figure, no terminations has occurred. However, white columns represents completed services and they are represented at several locations in this plot.

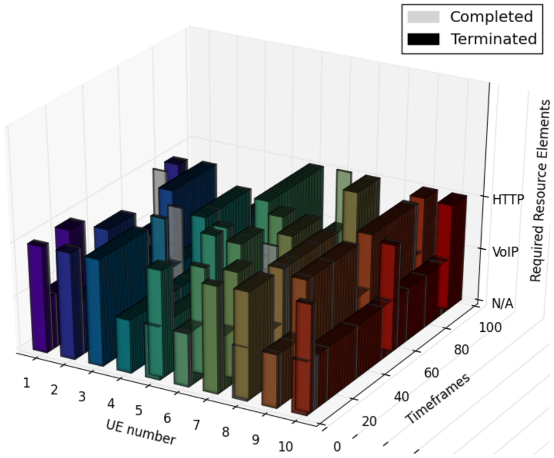


Figure 33: Services provided during the simulation of Case 1

Fig. 34 is correlated with Fig. 33, by an 3D depiction of the occurrences over the time frames for all delayed requests, terminated, blocked services, completed real-time and elastic sessions. By comparing these figures, the completed VoIP and HTTP sessions occurs at identical places in time and space. The completed HTTP and VoIP sessions are as predicted, since the probability that a service request is either HTTP or VoIP is uniform. Hence, these results are valid.

---

[4] In this chapter we use the UE number, UE-flow and UE-E interchangeably.

[5] The N/A value is not called 0, since it can either contain a blocked, terminated- or delayed service.

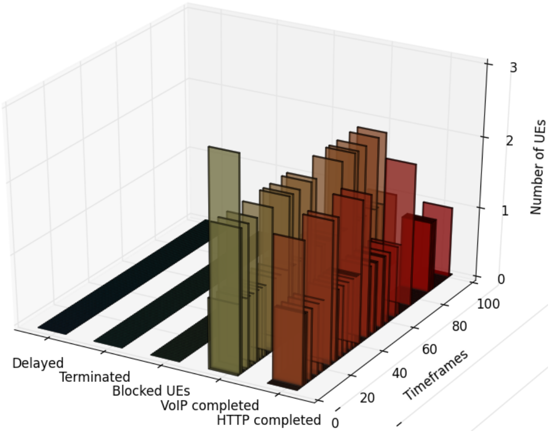


Figure 34: Delayed, terminated, blocked and completed service occurrences in Case 1

The outcomes in Fig. 35 image the summation of the occurrences from Fig. 34. In this figure there are 63 VoIP and 38 HTTP services completed over the simulation. According to Tab. the service probability in this case is uniform. Thus, the result may seem strange or even not valid, since there are a higher amount of real-time services completed than elastic. However, the presented outcome is acquired from running the simulation once and therefore the results can differ compared with results obtained from running the simulation one hundred times.

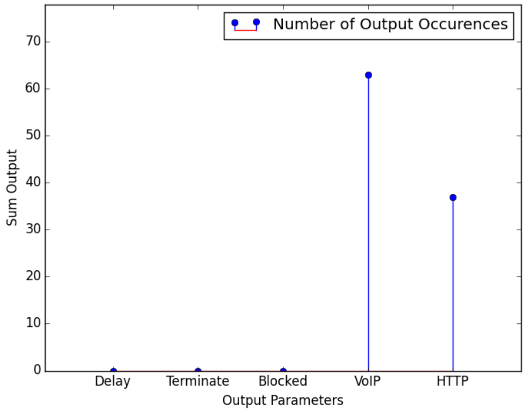


Figure 35: Output occurrences over the simulation of Case 1

The delay incidents for Case 1 for each UE flow at a specific LTE time frame is presented in Fig. 36. There are no occurrences of delayed requests in Fig. 35. Hence, this result is as anticipated.

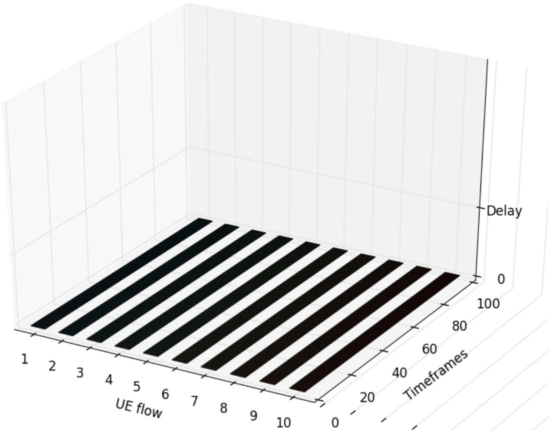
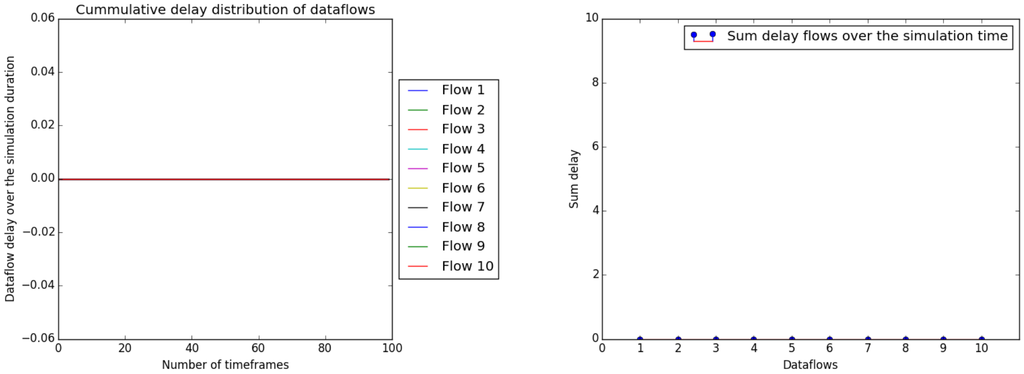


Figure 36: Delayed occurrences for each UE-E flow over the simulation of Case 1

Fig. 37a) and Fig. 37b) contains the same outcome, only the imaging is different. In this case, the results from both figures are identical. Therefore, this result is valid.



(a) Sum of the delay for each flow over the simulation

(b) Delay over each data flow

Figure 37: Data flow illustration of delayed occurrences during the simulation of Case 1

5.3 Case 2

Case 2 simulation parameters are described in Tab. 6.

Table 6: Parameter configurations for Case 2

LTE Time Frames	Cellular Resource Use	Service Probability	Modulation	VoIP Priority
100	75 %	50 % HTTP + 50 % VoIP	QPSK	Yes

In Fig. 38 the number of allocated resources over the simulation duration is illustrated. By enabling VoIP priority, real-time services will be allocated resources before elastic services. In the figure, UE-E-1 will have highest priority. Then, UE-E-2 will have the second highest, and the rest of the UE-Es follows respectively. As presented in the figure, UE-E-9 and UE-E-10 have a higher probability to obtain delayed requests and terminated services. Since, the available resource pool can be empty when its their time to obtain resources. The outcome from the figure is as anticipated, since the VoIP sessions are mostly located around the first half of the simulation, whilst the elastic services are place on the second half.

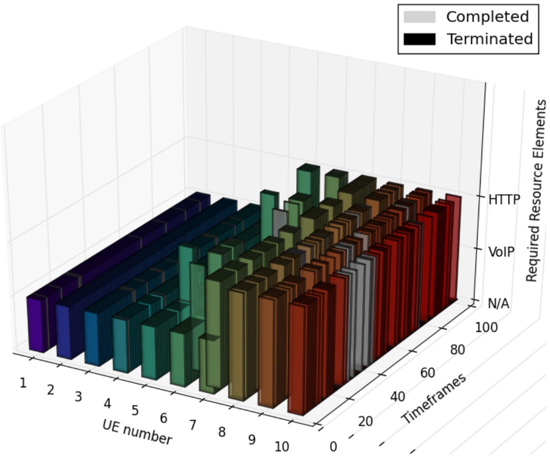


Figure 38: Services provided during the simulation of Case 2

From Fig. 39 the occurrences of delayed requests are high. This is as anticipated, since the elastic services are not prioritized. Thus, its a higher probability that they will be delayed or terminated. The delay is spread throughout the simulation, whilst the terminated and blocked services are centered around time frames 20-80. One possible reason could be that the available resources for D2D communication are low at during this time interval.

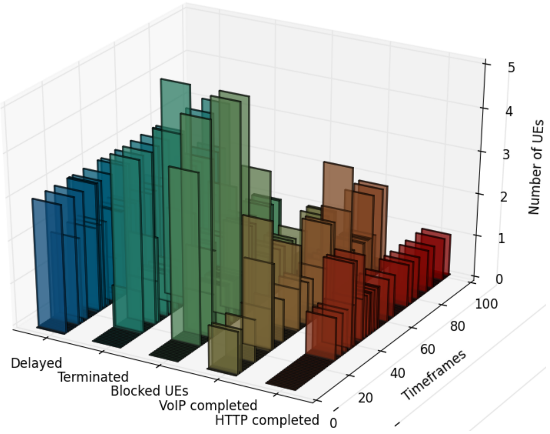


Figure 39: Delayed, terminated, blocked and completed service occurrences in Case 2

The number of output occurrences are presented in Fig. 40. The value of the delayed requests are similar to the sum of the terminated and blocked services. The number of completed VoIP and HTTP services are as expected.

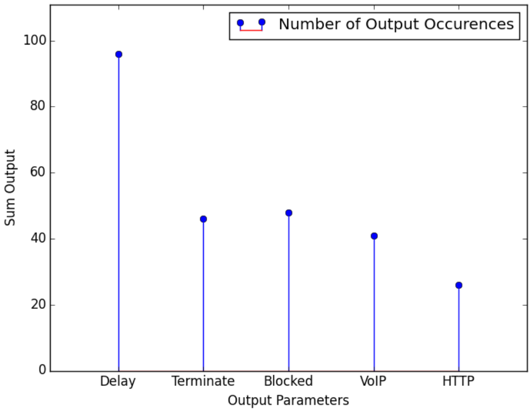


Figure 40: Output occurrences over the simulation of Case 2

Fig. 41 shows in which UE-E data flow and time frame the delay occurred. Also, the delays are mostly located between UE-E data flow 8-10, as expected. Comparing this figure with Fig. 39 we can see that they are correlated based on the delay.

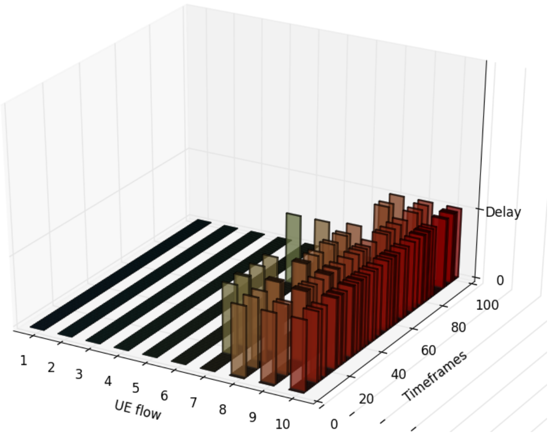
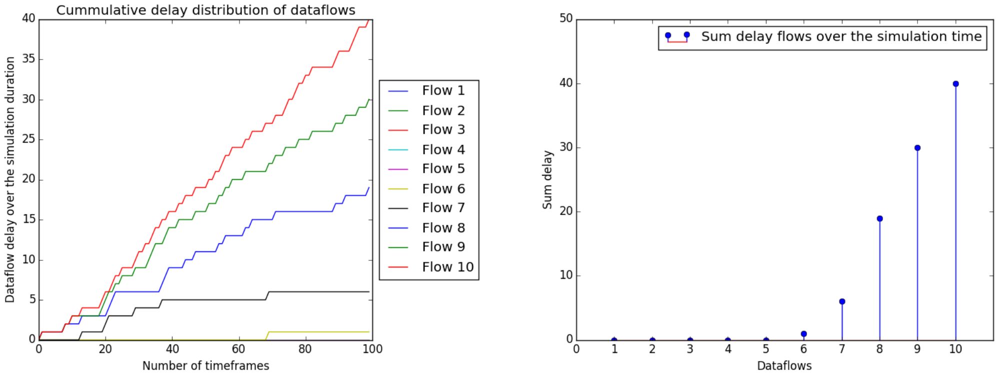


Figure 41: Delay occurrences for each UE-E flow over the simulation of Case 2

In Fig. 42a) the occurrences of the delay is presented, as well as which flows are affected by delay. Fig. 42b) illustrate each UE-E flow delay sum and as anticipated both figures shows that data flows 8 to 10 are most affected by congestion delay.



(a) Sum of the delay for each flow over the simulation

(b) Delay over each data flow

Figure 42: Data flow illustration of delayed occurrences during the simulation of Case 2

5.4 Case 3

The parameters for Case 3 are described in Tab. 7.

Table 7: Parameter configurations for Case 3

LTE Time Frames	Cellular Resource Use	Type of Service	Modulation	VoIP Priority
100	75 %	75 % HTTP + 25 % VoIP	QPSK	Yes

Presented in Fig. 43 is the number of allocated resources for UE-E services. The elastic services are most present in the figure, since the service type probability is 75 % for HTTP. Therefore, the output is as anticipated.

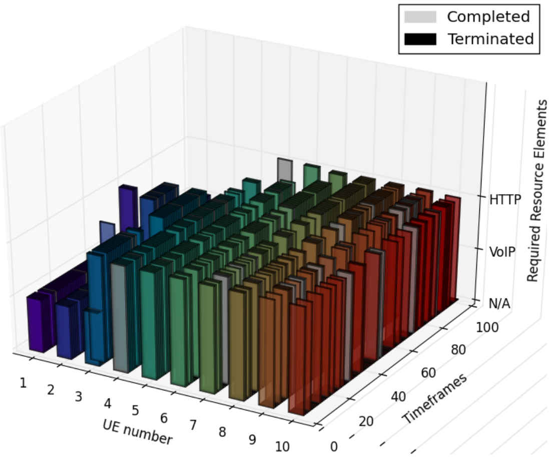


Figure 43: Services provided during the simulation of Case 3

The different service modes are illustrated in Fig. 44. This figure is as expected on behalf of Fig. 43. The number of terminated and blocked UE-Es are spread throughout the simulation. The number of delayed UE-Es are high, this is expected since it is 75% chance that the request is an elastic service request. Hence, there are more HTTP services completed than VoIP as expected.

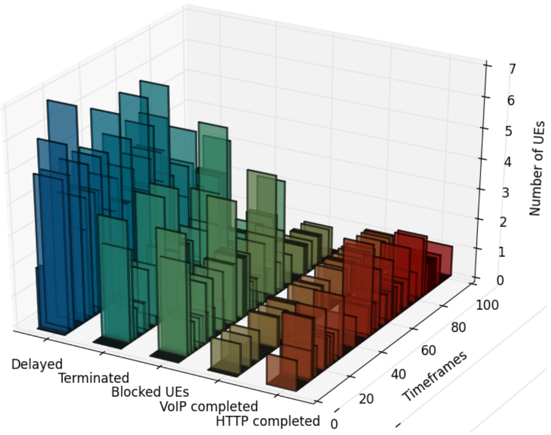


Figure 44: Delayed, terminated, blocked and completed service occurrences in Case 3

Fig. 45 represents the results from the simulation. The delay rate is high compared to the blocked and terminated rate, as anticipated. If the number of elastic services are dominant and a time frame contain an insufficient amount of resources to provide all services, there is a higher probability that sessions are delayed than blocked or terminated.

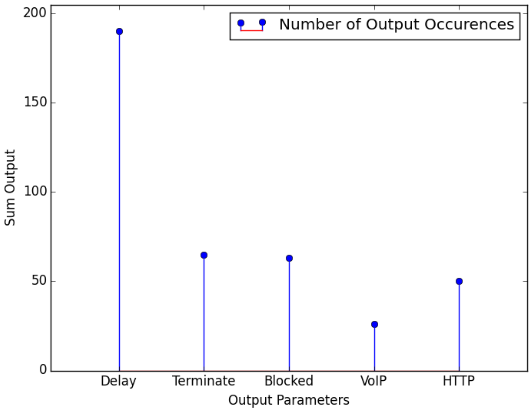


Figure 45: Output occurrences over the simulation of Case 3

The delay represented in Fig. 46 is according to the predicted outcome. As presumed, UE flow 1-2 are free from delay. Since the VoIP priority is enabled, the UE flows 1 - 2 will have a high probability to only consist of VoIP sessions.

Hence, the figure shows that UE-E flows 8 - 10 will accommodate most of the delay in the simulation. The session duration of these flows will then be extended. Such, that their service requests will not change, and their delay sum is correlated with the number of LTE time frames. Hence, an increase in the number of time frames, will result in an increase in their delay sum.

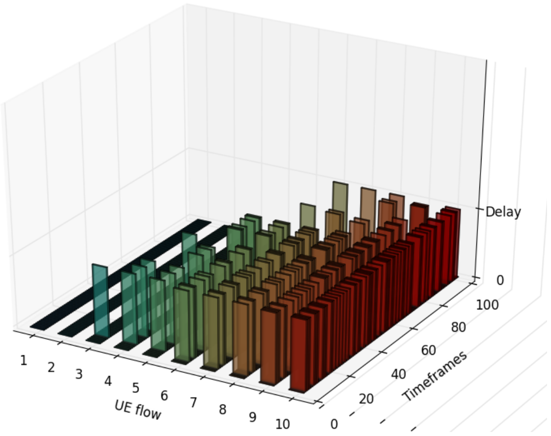
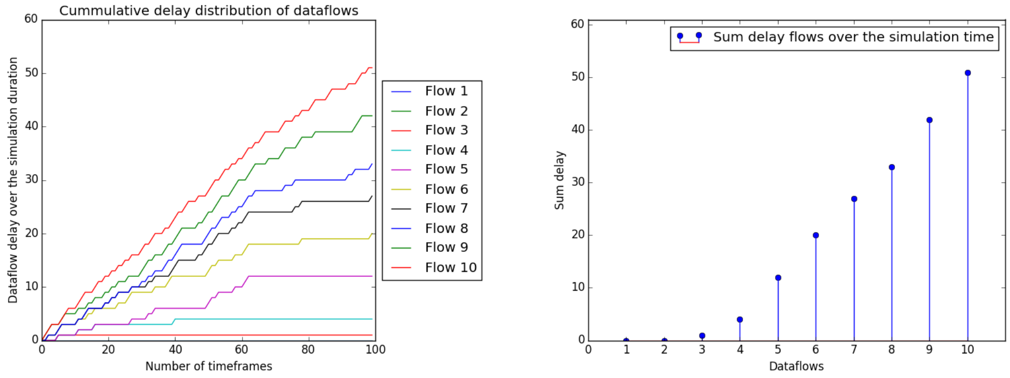


Figure 46: Delay occurrences for each UE-E flow over the simulation of Case 3

Fig. illustrates the delay over each data flow. In Fig. 47a) the increase in delay in Flow 7 - 10 between time frames 0 - 60 can resemble a linear distribution. A similar correlation can bee seen in Fig. 47b) between the data flows 3 - 10.



(a) Sum of the delay for each flow over the simulation

(b) Delay over each data flow

Figure 47: Data flow illustration of delayed occurrences during the simulation of Case 3

5.5 Case 4

In Tab. 8 the simulation parameters are presented for Case 4.

Table 8: Parameter configurations for Case 4

LTE Time Frames	Cellular Resource Use	Type of Service	Modulation	VoIP Priority
100	75 %	25 % HTTP + 75 % VoIP	QPSK	Yes

Fig. 48 presents the provided required REs in the simulation. Since, the VoIP probability is high, the figure contains a high number of VoIP sessions as anticipated.

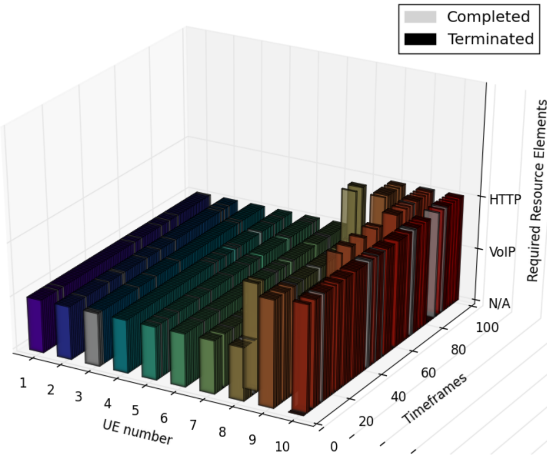


Figure 48: Services provided during the simulation of Case 4

In Fig. 49 the number of occurred termination of sessions are low. However, when a termination occurs the number of terminated UE-Es are high, due to the large amount of VoIP sessions in the simulation. Hence, when there are few resources to allocate, the termination will have a high probability to affect several UE-Es.

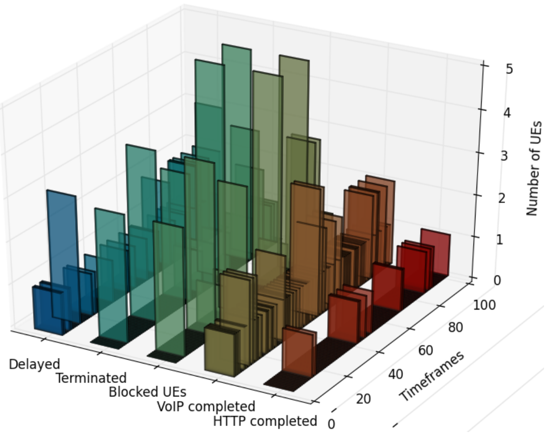


Figure 49: Delayed, terminated, blocked and completed service occurrences in Case 4

In Fig. 50 the number of occurrences are presented. The amount of delay, termination and blocking events are low. In this simulation the number of required resources for real-time services are less than half of what elastic services requires. This gives the system a higher probability to have a sufficient resource pool for all sessions, than with using a higher elastic service probability. Thus, the number of finished VoIP sessions are high as anticipated.

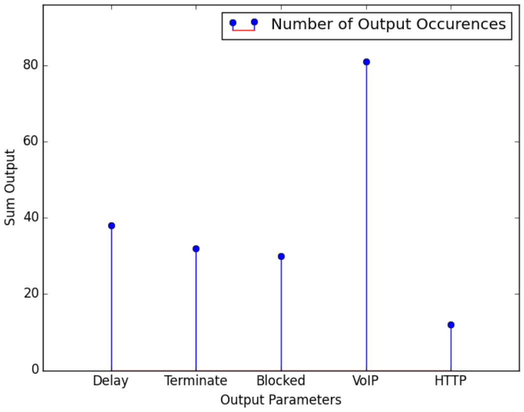


Figure 50: Output occurrences over the simulation of Case 4

Presented in Fig. 51 the VoIP priority and high number of VoIP sessions cause the delay to be spread from UE-E flow 8 - 10. The result in this figure is as expected.

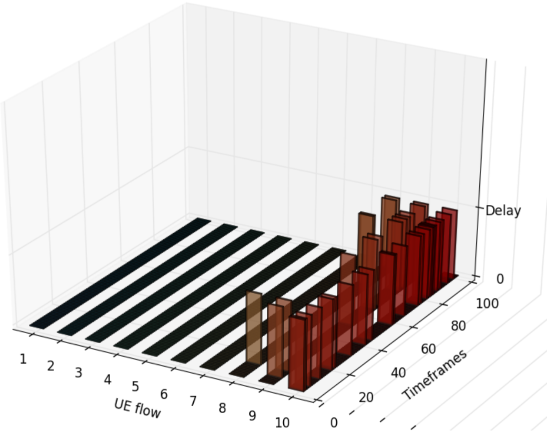
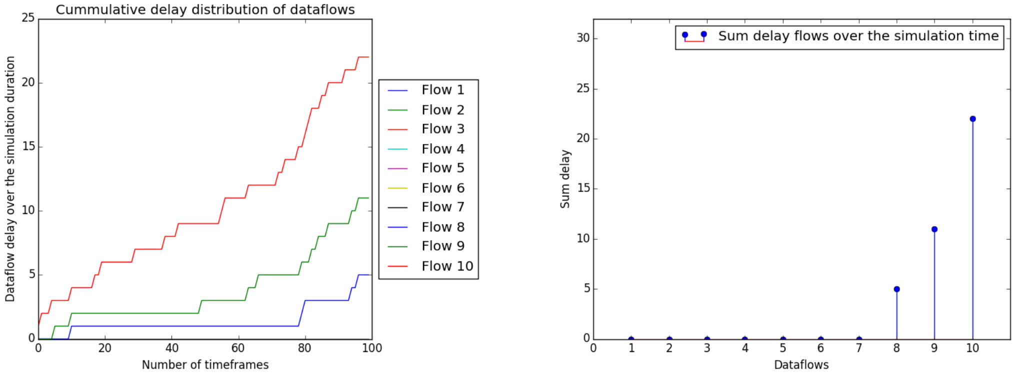


Figure 51: Delay occurrences for each UE-E flow over the simulation of Case 4

Fig. 52a) shows the addition of delay over the data flows over the simulation duration. In Fig. 52b) the sum of the delay for each UE-flow is illustrated. These figures are as anticipated, based on the results in Fig. 49.



(a) Sum of the delay for each flow over the simulation

(b) Delay over each data flow

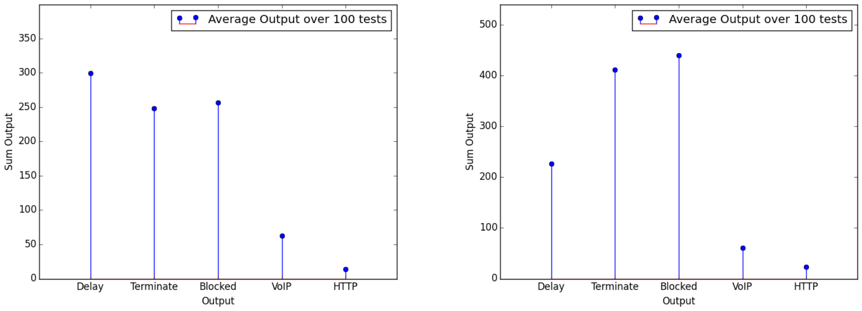
Figure 52: Data flow illustration of delayed occurrences during the simulation of Case 4

5.6 Performance Evaluation with/without VoIP Priority

In this section the cases from Subsec. 5.3, 5.4 and 5.5 are compared by taking the the mean value obtained from each system over a 100 simulations. Each case consist of two systems, one with VoIP priority enabled and another without utilizing the VoIP priority configuration. Also instead of using a cellular probability of 75 %, the cellular traffic is 90 % instead. Case 1 from Sec. 5.2 will not be analyzed in this section, since the change of the systems cellular usage will make it identical to Case 2.

5.6.1 Case 2

Fig. 53 presents the mean value results achieved by running the simulation 100 times. In Fig. 53a) the enabled VoIP priority setting is enabled and in Fig. 53b) the VoIP priority is disabled. For an easier overview of the obtained result the are listed in Tab. 9.



(a) VoIP priority enabled (b) VoIP priority disabled

Figure 53: Case 2 with/without VoIP priority

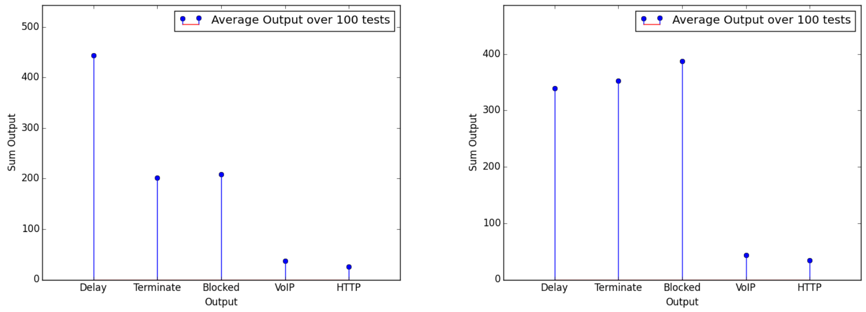
Table 9: 50 % HTTP + 50 % VoIP system with/without VoIP priority

Test Results	VoIP Priority Enabled	VoIP Priority Disabled
Average of delayed requests	299,57	226,32
Average of terminated services	248,41	410,65
Average of blocked services	256,5	439,86
Average of completed VoIP services	62,38	60,75
Average of completed HTTP services	13,62	23,13
Average of completed total services	76	83,88

In this case we use an uniform probability that a service type is either HTTP or VoIP. From Tab. 9 we can observe that enabling VoIP priority will cause the number of occurred terminations and blocked UE-Es to go down, compared to a system that has disabled VoIP priority. Also, the number of completed VoIP sessions is slightly higher in the priority enabled system. However, the delay is higher along with the number of completed HTTP sessions, and the total number of completed services are lower in an enabled VoIP priority system.

5.6.2 Case 3

The graphical outcome from testing VoIP enabled vs VoIP disabled priority systems are denoted in Fig. 54. Fig. 54a) represents the enabled VoIP system, while Fig. 54b) presents the disabled priority system.



(a) VoIP priority enabled (b) VoIP priority disabled

Figure 54: Case 3 with/without VoIP priority

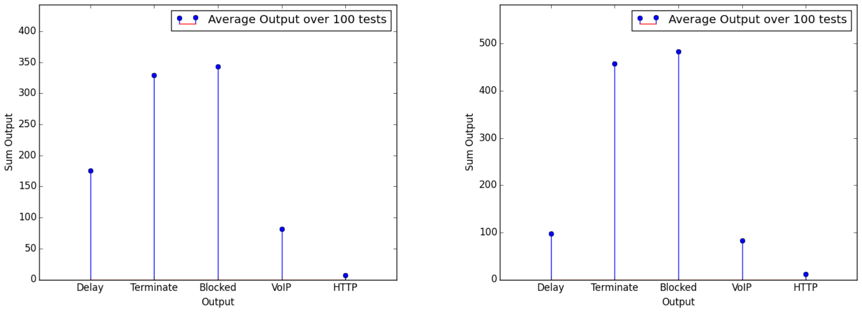
Table 10: 75 % HTTP + 25 % VoIP system with/without VoIP priority

Test Results	VoIP Priority Enabled	VoIP Priority Disabled
Average of delayed requests	443,42	338,82
Average of terminated services	201,34	352,2
Average of blocked services	208,7	387,0
Average of completed VoIP services	37,25	43,25
Average of completed HTTP services	25,12	34,12
Average of completed total services	62,37	77,37

In Tab. 10 the data from Fig.54 are represented. From the table we can see that the delay is lower and that the completion rate for VoIP and HTTP services are higher when not using VoIP priority in the system. Nevertheless, the termination and blocking rates are much higher.

5.6.3 Case 4

Fig.55 presents the mean values obtained from running Case 4 with VoIP priority enabled, see Fig. 55a), and VoIP priority disabled, see Fig. 55b).



(a) VoIP priority enabled

(b) VoIP priority disabled

Figure 55: Case 4 with/without VoIP priority

Table 11: 25 % HTTP + 75 % VoIP system with/without VoIP priority

Test Results	VoIP Priority Enabled	VoIP Priority Disabled
Average of delayed requests	175,25	97,73
Average of terminated services	329,0	457,87
Average of blocked services	343,17	483,17
Average of completed VoIP services	81,25	82,83
Average of completed HTTP services	7,38	11,09
Average of completed total services	88,63	93,92

Tab. 11 represents the results from both of the figures in Fig.55. This table shows that the termination and blocking probability is lower in a VoIP enabled priority system. However, the number of completed sessions are also lower both for real-time and elastic services.

5.7 Performance Evaluation using Different Modulation Schemes

To see if there are any difference using another modulation technique we compare the results from Sec. 5.6.2, where QPSK modulation was used. With simulations only changing the modulation to 16-QAM, all other premises are identical. Tab. 12 compares the obtained results for two systems with enabled VoIP priority with different modulation schemes, one utilizing QPSK modulation whilst another use 16-QAM modulation. In Tab. 13, we can observe that the number of VoIP sessions completed is higher using QPSK than 16-QAM. However, the overall the performance of the system is enhanced when using 16-QAM modulation. Both tables shows that all systems achieve enhanced performance using 16-QAM modulation, compared with using QPSK modulation. However, using 16-QAM modulation requires better channel quality and Quality of Service (QoS) parameters than QPSK. In this thesis we are not focusing on physical layer communication, so the obtained results in this section can only be used to verify the simulations.

Table 12: Modulation scheme comparison with VoIP priority

Test Results	QPSK modulation	16-QAM modulation
Average of delayed requests	443,42	285,41
Average of terminated services	201,34	134,01
Average of blocked services	208,7	136,2
Average of completed VoIP services	37,25	27,87
Average of completed HTTP services	25,12	46,65
Average of completed total services	62,37	74,52

Table 13: Modulation scheme comparison without VoIP priority

Test Results	QPSK modulation	16-QAM modulation
Average of delayed requests	338,82	258,42
Average of terminated services	352,2	184,02
Average of blocked services	387,0	199,63
Average of completed VoIP services	43,25	34,05
Average of completed HTTP services	34,12	52,47
Average of completed total services	77,37	86,52

6 Numerical Results from the Advanced Algorithm

6.1 Parameter Configuration

The configurations in the advanced algorithm are presented in Tab. 14.

Table 14: Parameter configurations for the advanced algorithm

Parameters	Values
UE-R	5 units
UE-E	100 units
UE-E Clusters	1
VoIP	28 REs
Battery Consumption per UE-E	1%
UE-R start up Battery Level	x+80%

From the table above the number of UE-E clusters is set to 1 for simplicity, and therefore this simulation differs from Fig.6 that illustrates this system. The UE-Rs battery level can range between 80% and 100%. To enable UE-Es to select UE-Rs based on their battery level, a random integer is introduced, x and it denotes an uniform random distributed integer with a value between 0% and 20%.

The implemented algorithm utilize the hybrid battery scheme presented in Fig. 17a), and the number of UE-Es that can be served are decreasing along with the battery level of the UE-Rs. There are 4 different modes in this scheme denoted as n, with 4 modes ranging from 0% to 100%, the UE-R will initialize a new mode when the battery level reach 80%, 60%, 40% and 20%. Each time a UE-Rs battery level decrease below one of these values, the ongoing sessions are finished and the UE-R enters a new mode. The min. threshold value for this system is 20%, when the battery percentage is below 20% the UE-R will stop broadcasting D2D invite messages and finish its current ongoing sessions and end UE-R mode. Tab. 15 shows the different UE-R modes with the number of max. UE-Es connectable for each mode.

Table 15: Threshold values for each UE-R mode

UE-R Modes	Max. UE-E sessions
80% < UE-R Battery =< 100%	Max. 12 UE-Es
60% < UE-R Battery =< 80%	Max. 9 UE-Es
40% < UE-R Battery =< 60%	Max. 6 UE-Es
20% < UE-R Battery =< 40%	Max. 3 UE-Es
0% < UE-R Battery =< 20%	Threshold

6.2 Numerical Results

The hybrid scheme is proposed to achieve fair resource allocation based on UE-Rs battery level, such that each UE-Rs battery is depleted at a correlated rate. Fig.56 presents the battery depletion rate of each UE-R through the system iterations. Every time the console receives a manual input the system will iterate, therefore the figures in this section use the number of iterations instead of LTE time frames.

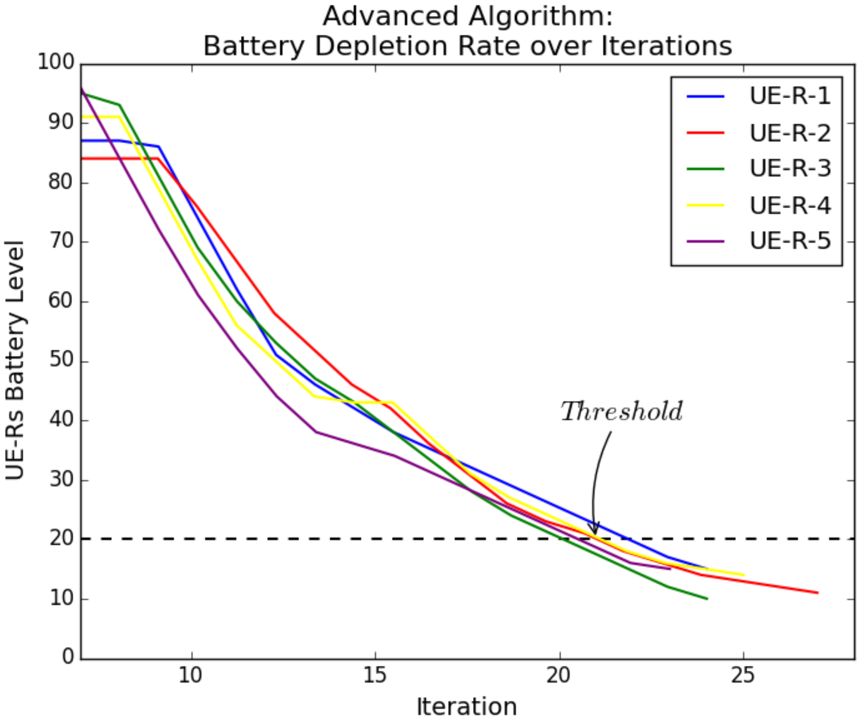


Figure 56: UE-Rs battery depletion rate using a hybrid battery scheme

The figure starts at iteration 7, since iteration 1 to 6 are dedicated to connection setup between the UE-Rs and the BS as well as obtaining the battery schemes and D2D message broadcasting. The starting percentage values of the UE-R batteries varies from 96 to 84. UE-R-5 have the highest battery level when UE-Es tries to connect and are therefore prioritized by UE-Es as relay for D2D communication. Therefore, its battery level decrease faster than the other UE-Rs between iterations 7 and 13, due to its high amount of connected UE-Es. At iteration 13, UE-R-5s battery level is below 40% and the max. number of UE-Es to serve is reduced to 3 UE-Es. Hence, it battery level is declining at a slower rate than the other UE-Rs, that will have a higher battery and priority level, until their levels reach UE-R-5 levels. UE-R-1 and UE-R-2 have a delayed start at serving UE-Es, compared with the other UE-Rs. Their D2D connections does not start before iteration 8 and 9. At iteration 13, UE-R-4s battery level is 44%, during the next two iterations it only serve one UE-E and its decrease with a rate of 1 %. From a viewers perspective this can cause the graph to appear as the battery level increases, however this is not the case. The annotation of the threshold value and its dashed line is inserted to make the figure more self-descriptive and it shows that each UE-R reach or pass the thresholds dashed line between iteration 19 and 21. When the UE-Rs reach the threshold they finish ongoing sessions, therefore the duration of the UE-Rs varies.

Tab. 16 presents each UE-Rs battery level value for each iteration in Fig. 56. Each column except the first from the left represents an iteration, and the double horizontal line separates the iterations and the UE-Rs battery levels.

Table 16: UE-Rs battery levels over the iterations

	Battery Level over Iterations in Percentage																			
UE-R	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	87	87	86	74	62	51	46	42	38	35	32	29	26	23	20	17	15	NA	NA	NA
2	84	84	84	76	67	58	52	46	42	36	31	26	23	21	18	16	14	13	12	11
3	95	93	81	69	60	53	47	43	38	33	28	24	21	18	15	12	10	NA	NA	NA
4	91	91	79	67	56	50	44	43	43	37	31	27	24	21	18	16	15	14	13	NA
5	96	84	72	61	52	44	38	36	34	31	28	25	22	19	16	15	NA	NA	NA	NA

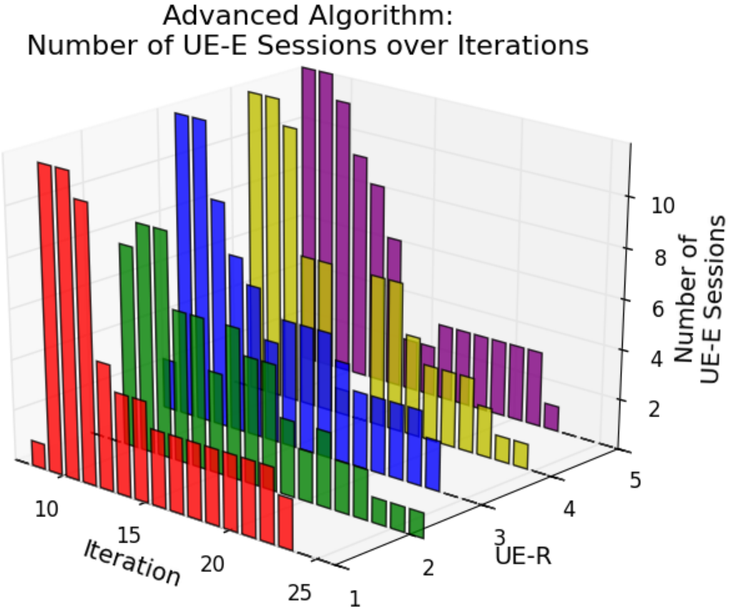


Figure 57: Number of provided sessions over each iteration

The number of UE-E sessions at each iteration is shown in Fig. 57. In the two first iterations of the simulation the UE-Es are replying the broadcast messages from the relays, and the replies arrive at different times to the UE-Rs. The amount of ongoing sessions increase and peak around iteration 10, since more ongoing UE-E sessions leads to a higher depletion rate of the UE-Rs battery levels. Thus, the number of provided sessions starts decrease and when a UE-R reach a battery level threshold, described in Tab. 15. It adapts to that threshold mode's requirements, and the max. number of ongoing service requests decrease. There are some exceptions such as UE-R-4 between iteration 11 and 14. The ongoing sessions drop from 6 to 1, then from 1 to 0 and then it climbs back to 6 sessions again at iteration 14. This event occurs when a UE-R is finished with its ongoing sessions, but its battery level is not highly prioritized by the UE-Es.

7 Discussions

To answer the Research Questions in Sec. 1.2, we will discuss the obtained results from each algorithm in this chapter. Starting with the numerical results from the basic algorithm, then the obtained outcomes from the advanced algorithm.

7.1 Basic Algorithm Discussions

For Research Question 1, the UE-Es in this thesis are reached by a UE-R that makes its own decisions on which UE-Es to serve, block, terminate or delay based on its resource pool. This allocation method allows the BS to reserve an amount of the available resources for the UE-R, and flood broadcast messages containing information on the number of feasible resources for D2D. The available resources are flooded by the BS each time frame, however in a real-life network this could cause high overhead in the system. Another approach is to allocate resources for UE-Es by letting the BS handle all decisions. When UE-Es are allocated resources from the BS, the experiences of terminations and delays could be neglected. Since, the BS would occupy a part of its resource pool to that each UE-E. However a much higher overlay would be expected in such system, due to the resource negotiation between UE-Rs and BS.

The proposal for Research Question 2, is to use two different types of service such as real-time and elastic services. Each UE-R can utilize up to four resource blocks from the up-link band for each service request. This amount may be unrealistic and not valid compared to the amount of resources a deployed network allocates. However this basic algorithm is used as a stepping stone to identify new methods to enhance performance using cellular D2D relay links, thus the algorithm distinguish itself from real-life systems in some areas.

The real-time services like VoIP have a much lower delay threshold than HTTP services, and the service cannot deal with delay in this algorithm. If there are an insufficient amount of resources to provide a VoIP service, the service will be terminated. The intention behind this termination decision is to make an explicit difference between the two services. A VoIP priority configuration was developed to meet the delay requirements for VoIP services, and by enabling this parameter the UE-Rs achieve the ability to prioritize VoIP sessions over HTTP sessions.

To fulfill Research Question 3, different cases were created and analyzed.

From Sec. 5.1, a delay pattern is revealed with the exception of Case 1. This case is developed to observe how the system performs when the number of available resources always are sufficient. In that case the occurrences of delayed, blocked and terminated services were not present as expected, since all the resources are dedicated to D2D communications in the system.

When observing the other cases, each simulation presents a similar delay pattern that is correlated with the number of UE-Es. By introducing more UE-Es in the system than the UE-R can handle with VoIP priority enabled, the delayed UE-Es will follow a resembling linear pattern, starting with the first delayed UE-E flow. This is to a degree expected, since when a delay occur the subsequent flows will also be delayed. Therefore, there should be a restriction on how many UE-E flows that could be added to the system when using VoIP priority. A proposal could be to introduce a dynamic selector on the number of UE-Es, that is based on the mean value of available resources over a period of time.

From the obtained results, the number of services provided are higher when the probability for VoIP as service type is high. The results shows that less resources required by the services, improves the performance of the system. Thus, the services provided over D2D communication using the in-band, should be restricted to low data rate services. However, in this thesis interference is not considered, and a higher number of ongoing services may cause higher interference in the network.

From the results obtained in Sec. 5.6, we can see that in each case, that the number of completed VoIP sessions are approximately the same for both systems with or without VoIP priority enabled. Also, using VoIP priority reduces the number of terminated and blocked UE-Es during a simulation. Thus, UE-Es requesting VoIP services in a system utilizing real-time service priority are less likely to be terminated or blocked. However, UE-Es located in such networks are in general more exposed to delay.

This introduce a new issue, if real-time services requires less resources than elastic services. The number of resources left after all VoIP requests are handled will less likely be sufficient to provide the need from the HTTP services requests. Then the HTTP requests will be either delayed for a longer duration of time or terminated. If the number of sufficient resources for HTTP services occurs every third time frame, for an elastic service with a long duration. The delay buffer for that UE-E will reset every third time frame, and the time it will take to provide the service will be extended by three times. If this issue occurs and the exponential probability for the elastic service request duration, exceed 33 time frames over a simulation of 100 LTE time frames. Then the HTTP service request will cause delay over the whole duration of the simulation and the session will not be completed. If this occurs in a real-time system, the delayed UE-E flows could cause congestion in the network. There is a higher probability that an elastic service request can cause a decline in system performance using VoIP priority when the number of D2D allocated resources are limited, than not using VoIP priority.

For HTTP service requests, the disabled VoIP priority systems are more advantageous in all cases, by completing more elastic services and by having reduced delays compared with its VoIP enabled counterparts. However, this network lacks robustness when it comes to its blocking and termination rates. The lowest combined number of blocked and terminated UE-Es during a simulation for VoIP priority disabled systems are approximately 387 + 352,2 = 691,2. This means that the probability that a UE-E is blocked or terminated are almost 9 times higher than to complete a service (691,2 / 77,37 = 8,93), see Tab. 10.

Comparing the output from using different modulation, the 16-QAM has a performance advantage over QPSK as expected. However, this configuration was developed to validate the system and other parameters can be added to extend the parameters into a more realistic physical layer module.

To answer Research Question 4 in Sec. 1.2, the results obtained are highly depended on the parameter settings, when cellular usage is high. By using a different binomial probability that a request is either HTTP or VoIP, will manifest in a difference in completed sessions ranging from 62,37 to 93,92. If the D2D network resources are limited, increasing the probability that a UE-E will request a VoIP session will enhance the network performance. VoIP priority could be utilized to minimize the number of blocked and terminated UE-Es. However, enabling this setting will affect the general system performance with respect to total completed sessions and network delay. From Case 1, Sec. 5.2, we can observe that the probability for delay, blocking and termination of UE-Es are zero. Thus, an important factor for the network performance is the variations of occupied resources used for the cellular network. If the cellular network and the D2D network share resources, the number of completed sessions will depend on how much band D2D communications are allocated. Implying, that if the cellular network have priority over the radio spectrum resources and a high probability to occupy RBs. Leading to a high risk that the D2D network will be unstable for real-time service usage. Hence, this will lead to a high termination and block rate of services and a poor network performance. Therefore, the results indicates that adjusting the basic algorithms parameters have a high influence when the cellular network use a high amount of the spectrum. Although when D2D communication is dedicated as a part of the up-link band larger than the UE-R requirements, the parameter configurations can be neglected as illustrated in Sec. 5.2.

7.2 Advanced Algorithm Discussions

The advanced algorithm was developed as a way to introduce a UE-R selection amongst UE-Es. Since no channel characteristics are applied, UE-Rs with different battery levels were proposed as a solution to make each UE-R distinctive to the UE-Es decision of D2D connection requests. How to manage a UE-Rs battery power in an energy efficient way, is arguably as important as the performance of the radio channel or the optimization on how to allocate network resource for D2D communications. Without any battery power control schemes, the UE-Rs battery will have a high risk of fast depletion that will cause the UE-Rs to shut down. Hence, the network will not be able to provide stable and robust D2D links through UE-Rs, to out-side-of coverage UE-Es.

To solve Research Question 5, four battery allocation schemes were proposed in Sec.(4.2.1), to illustrate new ways to utilize the battery consumption to provide UE-Rs control over their battery power. From the schemes UE-Rs can obtain the ability to choose which battery scheme to put in use, based on its own requirements. The hybrid scheme was proposed and implemented in the algorithm to adjust the UE-Rs connectivity possibilities according to its battery level. The energy efficient scheme was designed to handle high allocation of resources, without any battery restrictions. While the energy efficient scheme was created such that UE-Rs can either share resources over a long time interval or share resources in a selective manner. The linear scheme was proposed to provide an easy battery control with a only a linear decrease in battery level and connectivity amongst UE-Rs. With these designed schemes UE-Rs can have the control of both their battery consumption and sharing possibilities.

The proposed solution for Research Question 6, is an algorithm utilizing the hybrid scheme for fair battery depended resource allocation. This scheme use four battery threshold values that decides the number of max. connected UE-Es at a given time for the UE-Rs. The results obtained in Sec. 6, shows promising results on behalf of fair battery consumption amongst UE-Rs. The UE-Es will attempt to connect to the UE-Rs with the highest battery level. Hence, the fairness of battery consumption amongst UE-Rs are high, since the UE-Es will not try to connect to the UE-Rs with a low battery level if other high battery level UE-Rs are within reach. After a while the battery level amongst UE-Rs will flatten, and this causes the UE-Rs battery levels to decrease at a correlated rate. This battery scheme allows UE-Rs with poor battery conditions, to have a longer lifetime in the network.

By intention the delay and terminate parameters were not considered when developing the advanced algorithm. Since, the basic algorithm have already analyzed these parameters. Hence, the algorithm targets only battery dependent allocation, and using all parameters analyzed would not have given any new relevant information on allocation based on fair battery levels. The console only consists of a constant number of UE-Rs and UE-Es, such that the analysis of the battery is in focus. When doing a litterateur review on battery levels for in-band D2D communications with overlay mode enabled, no proposals could be found on battery control schemes focusing on this specific scenario.

The console output in Sec. (4.3.2), shows that battery dependency is a important factor in in-band D2D communications. The strategies on how to allocate resources can not be fully utilized without any battery level schemes. The obtained results without battery dependency are not valid in real-life, since the battery consumption will cause UE-Rs to shutdown at an exponential rate. Since, the UE-Rs lifetime depends on the following formula from the advanced algorithm, $BatteryPercentage * ConnectedUsers = RelayNodeLifetime$.

From the results we can observe that benefits from using battery depended allocation is a larger cellular coverage area by providing longer lifetimes amongst UE-Rs. If the UE-Rs can be online for a longer duration of time this will provide the network coverage to be extended. Without a battery allocation scheme, UE-Rs could shut down at a rapid rate, and the extended relayed area of the network would also be affected by these events. Therefore, an improved battery scheme will results in a larger cellular coverage area over time with UE-Rs shutting down at a controlled rate.

An issue with the proposed algorithm is when all UE-Rs are online for a long duration of time. This could cause the battery levels to decrease below the end session threshold. Thus, all UE-Rs could shut down during a short time interval and no D2D communication can be provided in the network. However, this event is not likely to occur, since the UE-Rs positions are not constant. In a real-life network, UE-Rs will be replaced at a dynamic rate and prevent such event to happen. In this thesis mobility is not considered, therefore this event will happen frequently in the simulation.

The battery consumption rate set in the console code, and plays a crucial part on the system performance. In the simulation the rate is predefined and static, causing the UE-Rs battery levels to decrease at a constant rate based on connected UE-Es. In reality this rate would be dynamic, and the difference in battery capacity and power consumption varies amongst UE-Rs. Some UE-Rs will have better battery capacity than others, however with the constant rate of the battery level consumption amongst UE-Rs and their lifetime would be almost identical in the network. Although, the UE-Rs with high battery capacity would more likely have a greater impact on the performance of the system, compared with the UE-Rs with low battery capacity.

One an important field that is not considered in this thesis, that needs to be discussed is the security aspect of D2D communication. By allowing UE-Rs to broadcast and provide D2D communications by sharing their cellular resources their Authentication Keys could be vulnerable to exploits. One solution can be to provide UE-Es their own unique identifier, such that the BS can separate UE-Es from UE-Rs.

From the UE-Es perspective, by allowing UE-Rs the ability to relay traffic, issues such as the man-in-the-middle attacks can occur. This form of attack is executed by relaying traffic and alter the communication between two entities. A solution to prevent MIMT attacks could be to develop advanced public key infrastructure routines with a third party validating all transmitted data, which would lead to a larger overhead in the system [10].

Another issue is the encryption of data between the UE-E and the BS, since the UE-R should not be able to access the relayed data. A solution can be to use only use encrypted D2D communications between the UE-Rs and UE-Es. The UEs in the network would be more secure using this protocol, however this can lead to more delay and overhead in the network [10].

8 Conclusions and Future Work

This chapter presents the conclusions based on the discussions in Chap. 7, followed by the contributions and suggestions for future work.

8.1 Conclusions

The outcomes shows that the performance of the basic algorithm is not highly depending on parameter configuration, but very much related to how spectrum is shared by cellular and D2D users. The results indicates that allocating dedicated resources to D2D communications enhances the performance of the system, compared to sharing D2D resources.

The advanced algorithm propose new challenges when it comes to resource allocation in a network. The UE-Rs battery usage has a higher influence on the system than expected, and has to be considered as a major part when standardizing the D2D technology as a part of 5G. Introducing battery level control schemes focusing on fair battery consumption amongst UE-Rs, benefits the cell coverage by increasing the duration UE-Rs can be online and therefore enhance the connectivity area of the network.

8.2 Contributions

During this thesis we have designed two handshake protocols that has been implemented into two algorithms. One is focusing on parameter configurations to enhance resource allocation, whilst the other targets UE-Rs battery levels to provide fair battery depended resource allocation.

The contributions for this thesis are as follows:

1. Proposed to a solution which targets a 3GPP scenario.
2. Designed two handshake protocols in Sec.3.4 and Sec. 4.2.2.
3. Developed a basic algorithm.
4. Designed four battery control schemes.
5. Developed an advanced algorithm.
6. Implementation and numerical results for each algorithm.

8.3 Future Works

In this Master thesis two algorithms has been developed and analyzed, the obtained results has contributed to interesting suggestions for new research topics. The proposals for future works are:

- Introduce selective connectivity between the UE-Rs and the UE-Es based on restrictions set and handled by the UE-Rs.
- Develop an algorithm which allows UE-Rs to have full control over their own ability to share resources, such that the allocation of resources depends on their current battery levels.
- Implement security procedures to provide secure transmissions using D2D communications.
- Extend or propose advanced battery control schemes to enhance the UE-Rs control over their resource sharing to meet their requirements, using battery depended sharing.
- Extend the proposed battery depended handshake algorithm, to enhance the performance of resource allocation based on the UE-R's battery level.

References

[1] V. M. Arash Asadi, Qing Wang, "A survey on device-to-device communication in cellular networks," Tech. Rep. VOL. 16, NO. 4, IEEE COMMUNICATION SURVEYS & TUTORIALS, 2014.

[2] "Tr 36.843," tech. rep., 3GPP, 2014.

[3] A. Corporation, "Lte resource guide," 2009.

[4] L. Williams, "A survey of agile development methodologies," 2007.

[5] J. Schiller, *Mobile Communications, 2nd Edition*. PEARSON EDUCATION LIMITED, 2003.

[6] Y. Ma, *Broadbandwireless communication systems: Channel modeling and system performance analysis*. PhD thesis, University of Agder, 2011.

[7] M. Meibergen, "Device-to-device communications underlaying a cellular network," Master's thesis, Delft University of Technology, 2011.

[8] L. Jiao, "Modeling and performance analysis of channel assembling in multichannel cognitive radio networks with spectrum adaptation." IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 61, NO. 6, JULY 2012.

[9] B. Shneiderman and C. Plaisant, *Designing the User Interface*. Boston: Pearson Higher Education, 2010.

[10] M. Stamp, *Information Security: Principles and Practice, 2nd Edition*. Wiley, 2011.

Appendix

A Data Rate Calculation

The data rate required for each service is presented in Tab. 17.

Table 17: Data rate required for the service types

Service type	Data rate
VoIP services	55 Kbps
HTTP services	120 Kbps

The data rate for each service needs to be converted from bits per second to LTE data rate. Therefore modulation techniques are required in order to denote the number of bits represented for each symbol. Tab. 18 presents the modulation techniques used in this thesis.

Table 18: Overview of the modulation techniques

Modulation	Number of bits	Bits per symbol
QPSK	2^2	2
16-QAM	2^4	4
64-QAM	2^6	6

The following formula is used to calculate the required REs to provide a service depending on the modulation, $\frac{Data\ Rate*Time}{Modulation} = Resource\ Elements$. The calculation of the LTE data rate is presented in Tab. 19.

Table 19: LTE data rate calculations for each service type

Service Type	Modulation	Formula	Number of required REs
VoIP	QPSK	$\frac{55Kbps*1ms}{2^2} = 27.5REs$	28
HTTP	QPSK	$\frac{120Kbps*1ms}{2^2} = 60REs$	60
VoIP	16-QAM	$\frac{55Kbps*1ms}{2^4} = 13.75REs$	14
HTTP	16-QAM	$\frac{120Kbps*1ms}{2^4} = 30REs$	30
VoIP	64-QAM	$\frac{55Kbps*1ms}{2^6} = 9.16REs$	10
HTTP	64-QAM	$\frac{120Kbps*1ms}{2^6} = 20REs$	20

B Advanced Algorithm Code

```java
/**
 *
 * @author Yngve Lågbu
 *
 */
import java.util.Scanner;

public class Main
{
   private static Scanner scanner;
   private static String input;

  /**
   *
   * @desc This is the start of the program
   *
   * @param args Not relevant
   *
   */

   public static void main(String[] args)
   {
      LocalMobileNetwork mobNet = new LocalMobileNetwork(5, 100);

      presentProgram();

      int counter = 1;

      while(counter < 10000)
      {
         System.out.println("!ITERATION␣" +  counter + "\n------------");

         mobNet.iterate();

        if(!continueIterations())
           break;

        counter++;
      }

      System.out.println("!SYSTEM␣TERMINATED");
   }

   /**
    * This method presents the output of the program to the user
    */
   private static void presentProgram()
   {

      String message = "MOBILE␣NETWORK␣PROGRAM␣\n";
      message += "-----------------␣\n";
      message += "This␣program␣displays␣a␣network␣containing␣an␣antenna␣and␣a␣\n";
      message += "custom␣number␣of␣UE-Rs/UE-Es.␣The␣UE-Rs␣\n";
      message += "may␣link␣to␣the␣antenna,␣while␣the␣UE-Es␣may␣\n";
      message += "link␣to␣the␣UE-Rs,␣but␣not␣the␣antenna.␣\n";
      message += "Type␣'abort'␣to␣exit␣the␣program.␣\n";
      message += "-----------------";
      System.out.println(message);
   }

   /**
    *
    * @return Continue iteration initialize the scanner, it returns true if abort is
          not inserted
    */

   private static boolean continueIterations()
   {
      if(scanner == null)
```

```
          scanner = new Scanner(System.in);

         input = scanner.nextLine();

         if(input.toString().contains("abort"))
            return false;

         return true;
   }
}
```

Listing 1: MainMethod

```
/**
 *
 * @author Yngve Lågbu
 * The Antenna will implement the iteratableObject, such that the class iterates.
 *
 */
import java.util.ArrayList;

public class Antenna implements IteratableObject
{
   private boolean hasBroadcastedConnection;
   private ArrayList<UE_R> list;
   /**
    * At the program startup the antenna class has no broadcast connection,
         therefore the
    * hasBroadcastedConnection variable has to be set to false. The antenna
         initialize an array-list for
    * the connectableUsers
    */
   public Antenna()
   {
      hasBroadcastedConnection = false;
      list = new ArrayList<UE_R>();
   }
   /***
    *
    * @param connList This method stores the connectableUsers in an array-list
    */
   public void setConnectList(ArrayList<UE_R> connList)
   {
      this.list = connList;
   }
   /**
    *
    * @return The broadcastNetwork method returns false if there are no available
         connectable users.
    * If there are connectableUsers, the class will iterate through the list of
         connectableUsers.
    * Then the hasBroadcastedConnection is set to true and the system will print out
          the print statement.
    */
   private boolean broadcastNetwork()
   {
      if(list.size() == 0)
         return false;

      for(int i = 0 ; i < list.size() ; i++)
            list.get(i).connect();

      this.hasBroadcastedConnection = true;

      System.out.println("Antenna␣broadcasted␣network␣to␣UE-Rs.");

      return true;
   }

   /**
```

```
    * The iterate class makes the iterations in the system. If the
        hasBroadcastedConnection is false,
    * the iterate class will initialize the broadcastNetwork method
    */

   public void iterate()
   {
      if(!hasBroadcastedConnection)
         broadcastNetwork();
   }

}
```

Listing 2: Antenna Method

```
/**
 *
 * @author Yngve Lågbu
 *
 */
public class Battery implements IteratableObject
{
   //0.0-10.0
   /**
    * The BATTERY_USAGE sets the battery level usage of the UE-Rs in the network.
        Depending on connected
    * UE-Es
    */
   private final double BATTERY_USAGE = 1.0;

   private int numberOfUsers;
   private double batteryLevel;

   /**
    * The Battery method sets the start values for the UE-Rs battery level. In this
        system the start values
    * can range from 80 % to 100 %.
    */
   public Battery()
   {
      batteryLevel = 80 + Math.floor(Math.random()*21);
      numberOfUsers = 0;
   }
   /**
    *
    * @return getBatteryLevel returns the battery level of the UE-Rs.
    */
   public double getBatteryLevel()
   {
      return batteryLevel;
   }
   /**
    * addUser increments the numberOfUsers by one each time its called.
    */
   public void addUser()
   {
      numberOfUsers++;
   }
   /**
    * removeUser will subtract one UE-E each time it's call, when the numberOfUser >
         0.
    */
   public void removeUser()
   {
      if(numberOfUsers > 0)
         numberOfUsers--;
   }

   /**
    * The iterate class is initialized to iterate the battery level by executing the
         useBattery method.
```

```
     */
    public void iterate()
    {
        useBattery();
    }

    /**
     * useBattery subtracts the BATTERY_USAGE * numberOfUsers for each UE-R.
     * If the UE-Rs battery level is subtracted below 0, the battery level is set to
          0.
     */
    private void useBattery()
    {
        batteryLevel -= (numberOfUsers * BATTERY_USAGE);

        if(batteryLevel < 0)
            batteryLevel = 0;
    }
}
```

Listing 3: Battery Method

```
/**
 *
 * @author Yngve Lågbu
 *
 */

import java.util.ArrayList;
/**
 *
 * State is an enumerator for three different states, OFFLINE, CONNECTING and ONLINE
       .
 *
 */
enum State{
    OFFLINE,
    CONNECTING,
    ONLINE;
}

/**
 *
 * The class BlockedUser is for UE-Es that cannot connect directly to the antenna
 *
 */

public class UE_E extends User implements IteratableObject
{
    private ArrayList<UE_R> connectableUsers;
    private UE_R userManagingConnection;
    private boolean hasNetworkPossibilities;
    private boolean networkAppearedThisIteration;

    private int stateWaitCounter;
    private State state;
    private int validConnectionAttemptsLeft;

    /**
     *
     * @param id
     * The id value is set for each UE-E. The starting state is set to OFFLINE.
     * Then a wait counter is called.
     * The UE-E has no network possibilities at the start, therefore it is set to
          false.
     * The UE-E will then check for connectable UE-Rs. By checking the array-list.
     * Managing connection is set to null since it has no users to connect with yet.
     * networkAppearedThisIteration is false, since no networks have appeard yet.
     */
    public UE_E(int id)
    {
```

```
    super(id);

    this.state = State.OFFLINE;
    this.stateWaitCounter = calculateNewWaitCounter();

    hasNetworkPossibilities = false;
    connectableUsers = new ArrayList<UE_R>();
    userManagingConnection = null;
    networkAppearedThisIteration = false;
}

/**
 *
 * @return An integer that returns the waiting time, ceiling is used so the
        integer can never be zero.
 */
private int calculateNewWaitCounter()
{
    return (int) Math.ceil(Math.random()*5);
}

/**
 *
 * @param u
 * This method is called by the UE-Rs, if the UE-Es has an
        networkAppearedThisIteration and hasNetworkPossibilities
 * the UE-R will add the UE-E to its arraylist.
 */
public void broadcast(UE_R u)
{
    if(!this.networkAppearedThisIteration)
        this.networkAppearedThisIteration = true;

    if(!this.hasNetworkPossibilities)
        this.hasNetworkPossibilities = true;

    this.connectableUsers.add(u);
}

/**
 * Iterate is called by the LocalMobileNetwork, and the first time
        networkAppearedThisIteration occur
 * it will print out some information and set networkAppeardThisIterations to
        false.
 * After that it will try to change state.
 */
public void iterate()
{
    if(hasNetworkPossibilities)
    {
        if(this.networkAppearedThisIteration)
        {
            System.out.println("UE-E␣" + this.getId() + "␣received␣possibilities␣for
                ␣network␣connection.");
            this.networkAppearedThisIteration = false;
        }

        else
        {
             tryToChangeState(state);
        }
    }
}

/**
 *
 * @param s
 * tryToChangeState can be either be OFFLINE, CONNECTING or ONLINE. When a method
         is called, then break is called.
 */
private void tryToChangeState(State s)
{
    switch(s){
```

```
        case OFFLINE:
            tryTochangeToConnecting();
            break;
        case CONNECTING:
            tryTochangeToOnline();
            break;
        case ONLINE:
            tryToChangeToOffline();
            break;
    }
}

/**
 * The UE−E is connected, if the waitCounter > 0, it will decrement the online
        time by 1
 * If the session is finished then the system will print out some information and
         the UE−E enters offline mode
 * and the UE−R removes the UE−E from its online UE−E array.
 */
private void tryToChangeToOffline()
{
    if(this.stateWaitCounter == 0)
    {
        System.out.println("UE−E␣" + this.getId() + "␣ended␣online␣session.");
        this.stateWaitCounter = this.calculateNewWaitCounter();
        this.state = State.OFFLINE;
        this.userManagingConnection.remove(this);
        this.userManagingConnection = null;
    }

    else
        this.stateWaitCounter--;
}

/**
 * If the UE−E has 0 connection attempts left, a new wait counter will be
        provided and the UE−E state will change to
 * OFFLINE.
 * The key point of this method is to find UE−Rs that has capacity to provide
        services, then it will find the
 * most suitable UE−R by compairing battery levels
 */

private void tryTochangeToOnline()
{
    if(this.validConnectionAttemptsLeft == 0)
    {
        this.state = State.OFFLINE;
        this.stateWaitCounter = this.calculateNewWaitCounter();
        System.out.println("UE−E␣" + this.getId() + "␣didn't␣find␣any␣connecatable␣
              users.");
    }

    else
    {
        ArrayList<UE_R> possibleAccessPoints = new ArrayList<UE_R>();

        for(int i = 0 ; i < this.connectableUsers.size(); i++)
            if(this.connectableUsers.get(i).isConnectable())
                possibleAccessPoints.add(this.connectableUsers.get(i));

        if(possibleAccessPoints.size() > 0)
        {
            UE_R bestProposal = new UE_R(−1);

            for(int i = 0 ; i < possibleAccessPoints.size() ; i++)
            {
                if(i == 0)
                    bestProposal = possibleAccessPoints.get(0);

                else
                    if(bestProposal.getBattery().getBatteryLevel() <
                          possibleAccessPoints.get(i).getBattery().getBatteryLevel())
```

```
                     bestProposal = possibleAccessPoints.get(i);
            }

            this.userManagingConnection = bestProposal;
            this.userManagingConnection.add(this);

            this.state = State.ONLINE;
            this.stateWaitCounter = this.calculateNewWaitCounter();

            System.out.println("UE-E␣" + this.getId() + "␣is␣online␣with␣connectable
                 ␣user␣" + this.userManagingConnection.getId());
         }

         validConnectionAttemptsLeft --;
      }
   }

   /**
    * This is the delay method before an UE-E connects with 3 connection attempts.
    */
   private void tryTochangeToConnecting()
   {
      if(this.stateWaitCounter <= 0)
      {
         stateWaitCounter = this.calculateNewWaitCounter();
         this.state = State.CONNECTING;

         this.validConnectionAttemptsLeft = 2;

         System.out.println("UE-E␣" + this.getId() + "␣wants␣to␣connect␣with␣the␣
              network.");
      }

      else
         stateWaitCounter--;
   }
}
```

Listing 4: UE-E Method

```
/**
 * @author Yngve Lågbu
 *
 */
import java.util.ArrayList;
/**
 *
 * This is a user that has network connection to the antenna.
 *
 */
public class UE_R extends User implements IteratableObject
{
   private Battery battery;
   private boolean connected;
   private ArrayList<UE_E> blockedList;
   private ArrayList<UE_E> usersConnected;
   private boolean connectionEstablishedThisIteration;
   private boolean hasBroadcastedConnection;
   /**
    *
    * @param id
    * This is the constructor for the UE-Rs, the LocalMobileNetwork gives each UE-R
         object an id.
    * When an object of the UE_R class is initialized.
    * It receives battery, and a set UE-Es
    */
   public UE_R(int id)
   {
      super(id);

      connectionEstablishedThisIteration = false;
```

```
   hasBroadcastedConnection = false;

   battery = new Battery();
   connected = false;

   blockedList = new ArrayList<UE_E>();
   usersConnected = new ArrayList<UE_E>();
}
/**
 *
 * @return
 * If UE-Es or LocalMobileNetworks wants to information about the UE-Rs battery,
      this method is called.
 */
public Battery getBattery()
{
   return this.battery;
}
/**
 *
 * @param bu
 * The UE-R have a list of UE-Es, they are initialized as blocked users, when
      this method is called, the UE-R obtains
 * a list of UE-Es.
 */
public void setBlockedUserList(ArrayList<UE_E> bu)
{
   this.blockedList = bu;
}

/**
 * The antenna calls connect, and then the UE-Rs will broadcast their connections
       .
 */
public void connect()
{
   connected = true;
   broadcastConnection();
}
/**
 * When broadcastConnection is called, connectionEstablishedThisIteration is set
      to true.
 */
private void broadcastConnection()
{
   connectionEstablishedThisIteration = true;
}
/**
 *
 * @param bu
 * UE-E know that the UE-R has capacity, when this method is called the UE-E adds
       itself to the UE-R.
 */
public void add(UE_E bu)
{
   this.usersConnected.add(bu);
   this.battery.addUser();
}
/**
 *
 * @return
 * The method returns true if the battery level is capable to support one or more
       UE-Es. If not false is returned.
 */
public boolean isConnectable()
{
   double batteryLevel = battery.getBatteryLevel();

   if(batteryLevel > 80)
   {
      if(usersConnected.size() < 12)
         return true;
```

```
        return false;
    }

    else if(batteryLevel > 60)
    {
        if(usersConnected.size() < 9)
            return true;

        return false;
    }

    else if(batteryLevel > 40)
    {
        if(usersConnected.size() < 6)
            return true;

        return false;
    }

    else if(batteryLevel > 20)
    {
        if(usersConnected.size() < 3)
            return true;

        return false;
    }

    else
        return false;
}

/**
 * If the UE-R is connected to the antenna, the antenna print statement is
        printed out.
 * Else if the UE-R has battery level higher than 20 %, it will call the
        broadcast method for each UE-E.
 * Else the UE-Rs battery level is printed out as NA.
 */
public void iterate()
{
    if(connected)
    {
        if(this.connectionEstablishedThisIteration)
        {
            this.connectionEstablishedThisIteration = false;
            System.out.println("UE-R␣" + this.getId() + "␣connected␣with␣antenna.");
        }

        else if(!this.hasBroadcastedConnection)
        {
            if(this.battery.getBatteryLevel() > 20)
            {
                for(int i = 0 ; i < this.blockedList.size(); i++)
                    this.blockedList.get(i).broadcast(this);

                System.out.println("UE-R␣" + this.getId() + "␣broadcasted␣connection␣
                        possibilities.");
            }

            this.hasBroadcastedConnection = true;
        }

        else
        {
            if(this.getBattery().getBatteryLevel() > 20)
            {
                System.out.println("UE-R" + this.getId() + ",␣Battery:␣" + this.
                        battery.getBatteryLevel() + this.getBlockedString());
            }
            else
                System.out.println("UE-R" + this.getId() + ",␣Battery:␣NA" + this.
                        getBlockedString());
        }
```

```
        }
    }
    /**
     *
     * @return
     * Creates a string with UE-Es that are currently connected to UE-R and returns
         the string.
     */
    private String getBlockedString()
    {
        String blockedIds = "";

        if(this.usersConnected.size() > 0)
        {
            blockedIds = ",␣UE-E:␣{";

            for(int i = 0 ; i < this.usersConnected.size(); i++)
            {
                blockedIds += this.usersConnected.get(i).getId();

                if(i != (this.usersConnected.size() - 1))
                    blockedIds += ",␣";

                else
                    blockedIds += "}";
            }
        }

        return blockedIds;
    }

    /**
     *
     * @param blockedUser
     * Called by the UE_E, when the UE-E change its state to OFFLINE. The UE-E is
         then removed from the list of
     * UE-Es connected to the UE-R.
     */
    public void remove(UE_E blockedUser)
    {
        this.usersConnected.remove(blockedUser);
        this.battery.removeUser();
    }
}
```

Listing 5: UE-R Method

```
/**
 *
 * @author Yngve Lågbu
 * @desc This interface is implemented by objects which has to implement the iterate
     method
 *
 */
public interface IteratableObject
{
    public void iterate();
}
```

Listing 6: Iterable Object Method

```
/**
 *
 * @author Yngve Lågbu
 *
 */
import java.util.ArrayList;

public class LocalMobileNetwork implements IteratableObject
```

```
{
   private final static int CUSTOM_NUMBER_OF_CONN_USERS = 10;
   private final static int CUSTOM_NUMBER_OF_BLOCKED_USERS = 10;

   private ArrayList<UE_R> connUsers;
   private ArrayList<UE_E> blockedUsers;

   private Antenna antenna;

   private ArrayList<Battery> batteries;

   /**
    * Default constructor
    */
   public LocalMobileNetwork()
   {
      this(CUSTOM_NUMBER_OF_CONN_USERS, CUSTOM_NUMBER_OF_BLOCKED_USERS);
   }

   /**
    *
    * @param    numberOfConnectableUsers The number of users which may
    *           connect to an antenna.
    * @param    numberOfBlockedUsers The number of users which may not
    *           connect to an antenna.
    */
   public LocalMobileNetwork(int numberOfConnectableUsers, int numberOfBlockedUsers)
   {
      this.initializeNetworkObjects();

      if(numberOfConnectableUsers < 1 || numberOfConnectableUsers > 1000000)
      {
         System.out.println("!NUMBER␣OF␣UE-Rs␣CHANGED␣TO␣DEFAULT");

         for(int i = 0 ; i < LocalMobileNetwork.CUSTOM_NUMBER_OF_CONN_USERS ; i++)
            connUsers.add(new UE_R(i + 1));
      }

      else
         for(int i = 0 ; i < numberOfConnectableUsers ; i++)
            connUsers.add(new UE_R(i + 1));

      if(numberOfBlockedUsers < 1 || numberOfBlockedUsers > 1000000)
      {
         System.out.println("!NUMBER␣OF␣UE-Es␣CHANGED␣TO␣DEFAULT");

         for(int i = 0 ; i < LocalMobileNetwork.CUSTOM_NUMBER_OF_BLOCKED_USERS ; i
              ++)
            blockedUsers.add(new UE_E(i + 1));
      }

      else
         for(int i = 0 ; i < numberOfBlockedUsers ; i++)
            blockedUsers.add(new UE_E(i + 1));

      handOverInformationToUsersAndAntenna();
   }

   /**
    * @desc Creates objects needed - connectable users,
    * blocked users, antenna and batteries.
    */

   private void initializeNetworkObjects()
   {
      this.connUsers = new ArrayList<UE_R>();
      this.blockedUsers = new ArrayList<UE_E>();

      this.antenna = new Antenna();

      this.batteries = new ArrayList<Battery>();
   }
```

```
    /**
     * @desc Iterates through connectableUsers, adds blockedUsers to each
         connectableUser,
     * adds the connectableUsers to a battery list in LocalMobileNetwork
     */

    private void handOverInformationToUsersAndAntenna()
    {
        this.antenna.setConnectList(connUsers);

        for(int i = 0 ; i < connUsers.size() ; i++)
        {
            connUsers.get(i).setBlockedUserList(blockedUsers);
            batteries.add(connUsers.get(i).getBattery());
        }

    }

    /**
     * @desc Calls the iterate method for each object
     */

    public void iterate()
    {
        for(int i = 0 ; i < this.batteries.size(); i++)
            this.batteries.get(i).iterate();

        for(int i = 0 ; i < this.blockedUsers.size(); i++)
            this.blockedUsers.get(i).iterate();

        for(int i = 0 ; i < this.connUsers.size(); i++)
            this.connUsers.get(i).iterate();

        antenna.iterate();
    }
}
```

Listing 7: Local Phone Network Method

```
/**
 *
 * @author Yngve Lågbu
 *
 * @desc User class which is a super class for both ConnectableUser
 * and BlockedUser. Since this class is abstract no objects may be made by this
     class.
 * BlockedUser and ConnectableUser has an unique ID, this is common for both user-
     classes.
 */
public abstract class User
{
    private int id;

    public User(int id)
    {
        this.id = id;
    }

    public int getId()
    {
        return id;
    }
}
```

Listing 8: User Method