Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Telecommunication Engineering

Master thesis

Optimization of Mobile Networks with Flying Base Stations for Mobile Users

Bc. Jakub Nový

Supervisor: Ing. Jan Plachý

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.

I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

Prague, 22 May 2020 …….

Acknowledgement

I wish to thank my supervisor, Mr. Jan Plachý, who introduced me to the topic of UAVs usage in mobile networks and helped me with the thesis whenever it was necessary.

I would also like to express my gratitude to Mr. Ondřej Hudousek, who provided me with valuable comments and insights during the preparations of this work.

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Nový** Jméno: **Jakub** Osobní číslo: **456919**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra telekomunikační techniky**

Studijní program: **Elektronika a komunikace**

Specializace: **Komunikační sítě a internet**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Optimalizace mobilních sítí s létajícími základnovými stanicemi pro mobilní uživatele

Název diplomové práce anglicky:

Optimization of Mobile Networks with Flying Base Stations for Mobile Users

Pokyny pro vypracování:

Seznamte se s využitím bezpilotních letounů jako létajících základnových stanic (FlyBSs) v mobilních sítích. Prostudujte problematiku hledání pozice FlyBS pro mobilní uživatele. Na základně existujících algoritmů navrhněte algoritmus, který na základě požadavků mobilních uživatelů na datovou propustnost nalezne pozici FlyBS a asociaci mobilních uživatelů pro splnění uživatelských požadavků. Navržený algoritmus využije informace, které jsou dostupné v mobilní síti od existujících základnových stanic. Pomocí simulací ověřte funkcionalitu navrženého algoritmu, a porovnejte jej s existujícími algoritmy. Navrhněte jakým způsobem by bylo možné navržený algoritmus implementovat v existujících mobilních sítích.

Seznam doporučené literatury:

[1] J. Plachy, Z. Becvar, P. Mach, R. Marik, M. Vondra, 'Joint Positioning of Flying Base Stations and Association of Users: Evolutionary-Based Approach', IEEE Access, vol. 7, 2019.
[2] A. Fotouhi, M., Ding, and M. Hassan, 'Dynamic base station repositioning to improve spectral efficiency of drone small cells', IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017.
[3] A. Fotouhi, H. Qiang, M., Ding, M., Hassan, L.G., Giordano, A., Garcia-Rodriguez, and J. Yuan, 'Survey on uav cellular communications: Practical aspects, standardization advancements, regulation, and security challenges', IEEE Communications Surveys & Tutorials, 2019.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Jan Plachý, katedra telekomunikační techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **08.01.2020** Termín odevzdání diplomové práce: **22.05.2020**

Platnost zadání diplomové práce: **30.09.2021**

Ing. Jan Plachý podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

. | Podpis studenta

Datum převzetí zadání

Anotace

Jednou z možností jak ulevit přetížené mobilní síti je nasazení dronu jako létající základnové stanice, která poskytuje dočasnou konektivitu uživatelům. Tato práce představuje algoritmus pro umístění dronu a úpravu jeho pozice v reálném čase spolu s návrhem implementace algoritmu do platformy OpenAirInterface.

Samotný algoritmus se skládá ze tří částí – predikce vzdálenosti uživatele od základnové stanice, určení polohy uživatele a klustrování uživatelů. Algoritmus na základě přijetí nových dat upravuje pozici dronu podle aktuálního rozmístění uživatelů.

Algoritmus pro určení pozice dronu využívá pouze základních měření přijímané úrovně signálu a je tudíž nezávislý na konkrétní generaci mobilní sítě. Vyhodnocení a simulace je provedeno na sítích čtvrté generace.

Klíčová slova: mobilní sítě, UAV, určení pozice

Summary

One of the ways to alleviate mobile networks is to exploit a drone as a flying base station, which provides temporal connectivity to mobile users. In this thesis, an algorithm for the drone placement and its real-time repositioning is proposed, together with possible implementation into the OpenAirInterface platform.

The algorithm itself works in three stages – prediction of users' distance to base stations, estimation of users' positions, and clustering of the users. The algorithm runs continuously as new data are received, reacting to the changes in the distribution of users.

The proposed algorithm is designed independently on mobile network technology, utilizing only the basic measurements of the signal strength present in all generations of mobile networks. The evaluation is done for 4G networks.

Key words: mobile networks, UAV, positioning

Contents

1 Introduction

The field of telecommunications technology is undergoing incredibly rapid development. Over the course of several decades, it has moved from analog to digital systems. The development continues by going into 5th generation networks (5G) with radio access technology known as New Radio (NR), while research about future generations is already in progress.

Several factors lie behind the increasing demands on the telecommunications infrastructure. The first aspect is an increasing number of connected users, due to the population's desire to be online and use all the advantages that are brought to them via telecommunications technologies. Another aspect is the change of the transported content, which has seen a significant transformation. Nowadays, the traditional voice calls are on the decline, as the users switch to the exploitation of Internet services, such as music and video streaming or augmented and virtual reality applications. This change in the content of the transmitted data is one of the driving forces behind the increased amount of transferred data.

The telecommunication industry continuously seeks new ways of development to meet all communication requirements. We can see different situations in fixed access networks and radio access networks. In the fixed networks, the progress is limited by the cost of infrastructure – laying optic fibers is expensive and sometimes not possible. Conversely, radio access networks can react faster to the growing capacity and mobility demands due to their larger coverage areas. Improvements are often carried out by replacing the legacy equipment on the site by a more advanced one.

Several organizations are involved in setting the direction of mobile networks evolution. One of the most important ones is the International Telecommunication Union (ITU), which assists in the development of requirements, standards, and frameworks for mobile networks. Another one is the European Telecommunications Standards Institute (ETSI), which publishes regulations and methodologies to guarantee interoperability between all the vendors. The Institute of Electrical and Electronics Engineers (IEEE) is a technical professional organization, which, for example, is responsible for standardization of Wi-Fi (IEEE 802.11) or WiMAX (IEEE 802.16). There are also many alliances focusing on the development of standards for mobile networks. Probably the most important and well-known is the 3rd Generation Partnership Project (3GPP), which has been releasing standards for mobile networks beginning with 3rd generation of mobile networks.

Each standard describes specific aspects of the given technology. The aim is to allow multi-vendor interoperability and to determine the direction of development. However, these standards do not provide specific technical solutions of how to implement standardized technologies. The exact technical solution, together with all proprietary functionalities, is in the hands of the manufacturer.

A great example of the emergence of new approaches in mobile networks without strictly defining them is the use of Unmanned Aerial Vehicle (UAV). UAVs can temporarily extend the system of base stations (BS) by creating additional flying base station (FlyBS) [1]. This area is currently being examined and is not yet fully standardized. However, for further vendor cooperation, the standards are necessary.

The development in this field can be divided according to the type of used UAVs and the altitudes at which they operate. Heavy, hundreds of kilograms weighing aircraft operate mostly at the heights of hundreds to thousands of meters. As a result, they can cover a vast area. An example of this category is the Facebook Aquila project [2] or the Project Loon [3]. On the other side, some UAVs move at the altitudes of tens to hundreds of meters. These UAVs, commonly drones, are small and carry light equipment. Their most significant advantage is the possibility of deployment in a very short time to react to the needs of the mobile network operator. This work targets the second category of UAVs.

There are different ways of FlyBS deployment in mobile networks. The objectives focus on minimization of the outage probability [4], maximization of backhaul network connectivity [5], or maximization of spectral efficiency [6]. The FlyBSs can be deployed to extend the coverage of a selected area, for example, in the event of a BS failure [7]. The FlyBS network can also be used in areas without a functional mobile technology infrastructure – areas affected by natural disasters or at war zones [8]. Another use case is offloading of congested cells during mass gatherings such as sporting events, marches, or demonstrations. In this case, the FlyBS serves a group of devices, commonly known as User Equipment (UE), to provide them with the desired throughput [9].

This work focuses on the last use case, i.e., finding a conglomeration of UEs in the area and increasing the network system throughput by deploying a FlyBS. Therefore, the goal of this thesis is to propose an algorithm, which will find the appropriate position where to place the FlyBS. The FlyBS should be able to independently adjust its positions in order to serve the desired amount of the UEs and simultaneously increase the network throughput.

1.1 Structure of the thesis

The thesis is structured as follows: the mobile network parameters, which are accessible from the mobile network, are summarized in Chapter 2. Based on these parameters, the algorithm for positioning is proposed.

The whole process of FlyBS positioning is formally defined in Chapter 3. In this chapter, we describe the proposed system model, performance metrics, assumptions, and the objective function.

The overview of the proposed algorithm is introduced in Chapter 4. In this chapter, we present the algorithm consisting of three parts:

1. prediction of the signal power for all UEs on a short-term time basis,
2. estimation of the position of the UEs based on the predicted signal power,
3. finding a suitable UE cluster to be served by FlyBS, based on the estimated positions of the UEs.

Various simulations are necessary to evaluate the proposed positioning algorithm and its parts. The common simulation parameters are also defined in this chapter.

Prediction estimates the signal strength from the received measurements. Another important role of the prediction is to provide enough time to collect all measurements and process them. The problem of prediction is outlined in Chapter 5.

In the estimation phase, the UE is localized based on the signal quality from the BSs. The process of UE localization is described in Chapter 6.

Once the UE positions are known, it is necessary to evaluate possible clusters of UEs which may be served by the FlyBS. The clustering presents the core of the thesis, and its details are described in Chapter 7.

All of the algorithms, as described in the previous chapters, are designed to work together. The details of the cooperation of the algorithms and selection of suitable cluster to be served by the FlyBS are presented in Chapter 8.

The possible implementation of the proposed algorithm into the real hardware, with respect to the current state of the art FlyBS at the Faculty of Electrical engineering, is described in Chapter 9. The conclusion of the thesis and a possible future work is presented in Chapter 10.

2 Available parameters of the mobile network

The mobile networks provide various information about the network and UE status for its operation. These parameters range from BS load to UE signal quality and its traffic. In this chapter, parameters that are needed for the proposed algorithm are defined.

FlyBS itself has limited knowledge about the mobile network and needs additional inputs on which it can make a decision. The most important parameters for the deployment of the FlyBS are the reports from UEs connected to their serving BSs and reports from neighbor BSs, i.e., BSs that are close to the UE but provide worse signal quality than the serving BS. Each generation of mobile networks has standardized parameters that are provided. Therefore, we focus on parameters that are common for all generations of mobile networks. In addition to the reports gained from the mobile network, the FlyBS may perform custom measurements which are not standardized in order to gain more precise information about the situation near its location.

We focus on a 4G mobile network, also denoted as Long Term Evolution (LTE), where the target parameter is the Reference Signal Received Power (RSRP). However, measurements of received signal power are present in all generations of mobile networks. The RSRP counterpart in 3G is Received Signal Code Power (RSCP) [10], [11], while in 5G it is Synchronization Signal Reference Signal Received Power (SS-RSRP) [12].

2.1 User equipment reports

In 4G, the UE reports to the serving BS two main indicators: Received Signal Strength Indicator (RSSI) and RSRP [13]. These two indicators are the key elements for the proper decision-making on cell selection and reselection or handover. LTE uses Orthodontal Frequency Division Multiplexing (OFDM) for downlink. The smallest unit is a physical resource block, which consists of 6 or 7 ODFM symbols in the time domain and 12 subcarriers in the frequency domain.

RSSI represents average power for specific OFDM symbols in the whole considered bandwidth. The resulting RSSI value includes not only the serving BS signal but also any interference (power of other BSs, etc.) and noise. This parameter provides complex information about the UE's signal quality condition.

3GPP states that "E-UTRA Carrier Received Signal Strength Indicator (RSSI), comprises the linear average of the total received power (in [W]) observed only in OFDM symbols containing reference symbols for antenna port 0, in the measurement bandwidth, over N number of resource blocks by the UE from all sources, including co-channel serving and non-serving cells, adjacent channel interference, thermal noise etc." [13].

On the contrary to RSSI, RSRP is better for measurements tied to one source of the signal, for example, measurements in one sector of network. RSRP represents the signal strength of a specific BS.

3GPP defines RSRP as "the linear average over the power contributions (in [W]) of the resource elements that carry cell-specific reference signals within the considered measurement frequency bandwidth" [13].

Relative RSRP accuracy, which means RSRP measured from one BS compared to the RSRP measured from the different BS on the same frequency, should not be worse than ±3dB in normal conditions for intra-frequency measurements and ±6dB for inter-frequency measurements as stated in Table 9.1.2.2-1 and Table 9.1.3.2-1 in [14].

3GPP also defines Reference Signal Received Quality (RSRQ), a ratio of RSRP and RSSI multiplied by the number of physical resource blocks over which the RSSI measured. RSRQ is used to indicate the quality of the reference signal received at the UE.

2.2 Reported metrics

For the purposes of this thesis, the most important parameter is RSRP. The RSRP value is usually between ca. -75 dBm at the cell center and -120 dBm at the edge of the cell. The signal strength is mapped on a scale introduced in Table 9.1.4-1 from [14]. In this work, RSRP value means Reported value, i.e., RSRP_30.

Table 2.1 RSRP measurement report mapping (adopted from [14]).

Reported value	Measured quantity value	Unit
RSRP_-17	RSRP< -156	dBm
RSRP_-16	-156 ≤ RSRP< -155	dBm
…	…	…
RSRP_-01	-141 ≤ RSRP< -140	dBm
RSRP_00	RSRP < -140	dBm
RSRP_01	-140 ≤ RSRP < -139	dBm
…	…	…
RSRP_95	-46 ≤ RSRP < -45	dBm
RSRP_96	-45 ≤ RSRP < -44	dBm
RSRP_97	-44 ≤ RSRP	dBm

The report structure of RSRQ is similar to RSRP. The RSRQ reports cover the values between -19.5 dB and -3 dB with 0.5 dB step.

The results of the UE measurements of RSRP and RSRQ are provided to BS through a Dedicated Control Channel. Reports can be sent periodically, after the triggering caused by an event or both. The limitation is the time-period how often the reports can be transmitted – according to [15] the ReportInterval parameter may be set to 120 ms, 240 ms, 480 ms, 640 ms, 1 024 ms, 2 048 ms, 5 120 ms, 10 240 ms, 1 minute, 6 minutes, 12 minutes, 30 minutes, 1 hour.

3 Problem formulation

Deploying a new BS or FlyBS to the existing mobile network changes the overall signal distribution in an area. The changes are caused by adding a transmitter which provides mobile services to the UEs of the newly deployed BS or FlyBS. However, the FlyBS also causes interference to the UEs served by the BSs. In the ideal case, deployment of the new FlyBS presents an advantage – higher signal to noise ratio (SNR) and signal to noise and interference ratio (SINR) to the UEs served by the FlyBS. However, for the UEs connected to other BSs, the FlyBS causes interference and, thus, leads to decreased SNR and SINR of the UEs served by the other BSs.

Usually, deployment of a new BS is carefully planned ahead, and the final location is chosen from a set of predefined possible locations. The placement is influenced by positioning restrictions (such as property owner willingness or suitability of building for the antenna placement). However, in the case of the FlyBS, the situation is different. First of all, the location conditions are different, due to its ability to operate in a 3D space, i.e., significantly increased search space. Also, deployment of the FlyBS must follow national regulations for UAVs. Furthermore, specifics to the FlyBS deployment are considered, such as safety measures, backhaul connectivity, drone endurance and so on. To make the problem even harder, the position of the FlyBS is not fixed and can be adapted in time. Therefore, interference is constantly changing, not only of the UE served by the FlyBS but even of the UE not served by the FlyBS. Factoring all mentioned above, finding a proper position is a challenging task.

3.1 System model

System model for this thesis is based on [16]. The whole area is permanently served by S SBSs forming a set $\boldsymbol{K}^S = \{k_1^S, k_2^S, \dots, k_S^S\}$, while the F FlyBSs form a set denoted as $\boldsymbol{K}^F = \{k_1^F, k_2^F, \dots, k_F^F\}$. Thus, the set of all $\boldsymbol{K}$ base stations in the area is $\boldsymbol{K} = \boldsymbol{K}^S \cup \boldsymbol{K}^F$. There is also a set $\boldsymbol{U} = \{u_1, u_2, \dots, u_U\}$ of U UEs in the area.

The positions of the SBSs are defined by a set $\boldsymbol{L}^S = \left\{l_{k_1^S}^S, l_{k_2^S}^S, \dots l_{K_S^S}^S\right\}$ of 3D vector coordinates of the SBSs. The positions of SBSs are known in advance, as their location does not change over time. The positions of the FlyBSs form a set $\boldsymbol{L}^F = \left\{l_{k_1^F}^F, l_{k_2^F}^F, \dots l_{K_F^F}^F\right\}$, where $l_{k^F}^F \in \mathbb{R}^3$. It is assumed that the flight altitude may be varying in time. The maximal speed of the FlyBS, respective of the drone carrying the FlyBS is v_{max}. We further define a set of location for all BSs, $\boldsymbol{L}^K = \boldsymbol{L}^S \cup \boldsymbol{L}^F$

The UE u position is given by a 3D vector $l_u^U \in \mathbb{R}^3$ and for all UEs in the area form a set $\boldsymbol{L}^U = \left\{l_{u_1}^U, l_{u_2}^U, \dots, l_{u_U}^U\right\}$. The UEs may move around the area during the time.

Each UE u is allocated a portion of bandwidth from BS k (which may be an SBS or a FlyBS). It is defined as $\beta_{k,u}$. The absolute amount of assigned bandwidth to the u-th UE connected to the k-th BS is then $B_k \cdot \beta_{k,u}$ hertz. The transmission capacity $c_{k,u}$ of k-th BS

to u-th UE is in bits per second and can be calculated using Equation 3.4. Each UE is connected to just one BS at the moment, either one of the SBSs or FlyBSs. The serving SBS or FlyBS is indicated by $\beta_{k,u} > 0$. For the purposes of the thesis, we define a set of numbers of UEs served by the BSs as $\boldsymbol{N} = \{n_{k_1}, n_{k_2}, \ldots, n_K\}$, where n_k represents number of UEs served by the k-th BS, i.e., number of UEs that satisfy $\beta_{k,u} > 0$.

The transmission power level P_k^{tx} of BS k is related to the maximal distance between the BS and UE by the reverse path model, as shown later. This distance in meters is denoted as a radius r_k for BS k.

The overview of notations used in this chapter presents Table 3.1.

Table 3.1 Summary of notations.

Notation	Description
$\boldsymbol{K}, \boldsymbol{K^S}, \boldsymbol{K^F}$	Set of all BSs, set of all SBSs, set of all FlyBSs
$\boldsymbol{U}$	Set of all UEs
$\boldsymbol{N}$	Set of numbers of UEs served by the BSs
K, F, S, U	Number of BSs, FlyBSs, SBSs, and UEs
$\boldsymbol{L^K}, \boldsymbol{L^F}, \boldsymbol{L^S L^U}$	Set of BSs, FlyBSs, SBSs, and UEs positions
l_k^K, l_u^U	Location of k-th BS, location of u-th UE
v_{max}	Maximal drone speed
B_k	Bandwidth of k-th BS
$\beta_{k,u}$	Ratio of assigned bandwidth of k-th BS to u-th UE
P_k^{tx}	Transmission power of k-th BS
P_{min}^{Rx}	Minimal received signal power
r_k	Coverage radius of k-th BS
n_k	Number of UEs served by the k-th BS
$S_{k,u}$	Received signal strength of u-th UE from k-th BS
$N_{k,u}, I_{k,u}$	Noise and Interference between k-th BS and u-th UE
$c_{u,k}$	Transmission capacity between k-th BS and u-th UE
c_f^F	FlyBS f backhaul capacity
$d_{k,u}$	Distance between u-th UE and k-th BS

3.2 Performance metrics

We use several metrics as part of the proposed algorithms. These metrics, namely received signal strength (RSS), signal to noise ratio (SNR), Signal to noise and interference ratio (SINR), Transmission capacity, and the path loss (PL) model, presents fundamental elements of the whole solution.

3.2.1 Received signal strength

Received signal strength (RSS) of the signal of interest can be expressed for downlink as presented in Equation 3.1.

$$S_{k,u} = P_k^{tx} \left| h_{k,u}\left(l_k^K, l_u^U\right) \right|^2 \quad 3.1$$

P_k^{tx} is the transmit power of the k-th BS, $\left| h_{k,u}\left(l_k^K, l_u^U\right) \right|^2$ represents the transfer function between u-th UE and k-th BS. This value is a key parameter for the simulations in this thesis, as it is reported through RSRP reports introduced in Chapter 2.

3.2.2 Signal to noise ratio

One of the most common metrics used to measure signal quality is the signal to noise ratio (SNR). The SNR value can be calculated for each radio channel from the k-th BS to the u-th UE using Equation 3.2. SNR represents the ratio between the level of the desired signal and the level of background noise.

$$SNR_{k,u} = \frac{S_{k,u}}{N_{k,u}} = \frac{P_k^{tx} \left| h_{k,u}\left(l_k^K, l_u^U\right) \right|^2}{N_{k,u}} \quad 3.2$$

$N_{k,u}$ is the noise in Watts influencing the channel between the endpoints.

3.2.3 Signal to noise and interference ratio

Similarly to SNR, the signal to noise and interference ratio (SINR) can be calculated as seen in Equation 3.3. SINR does take into consideration the presence of the interference.

$$SINR_{k,u} = \frac{S_{k,u}}{N_{k,u} + I_{k,u}} = \frac{P_k^{tx} \left| h_{k,u}\left(l_k^K, l_u^U\right) \right|^2}{N_{k,u} + I_{k,u}} \quad 3.3$$

$I_{i,j}$ is interference in Watts caused by other channels to this channel between k-th BS and u-th UE.

3.2.4 Transmission capacity

The transmission capacity of the radio channel for u-th UE served by k-th BS in bits per second is calculated using the Shannon-Hartley theorem in Equation 3.4.

$$c_{k,u} = \beta_{k,u} B_k \log_2(1 + SINR_{k,u}) \quad 3.4$$

where B_k is the bandwidth available for k-th BS and $\beta_{k,u}$ represents a scaling factor, as described above.

3.2.5 Path loss

The signal between a UE and a BS is attenuated by the environment between the two endpoints: the signal received at the UE is decreased by path loss. Estimation of the path

loss in the real environment is a non-trivial problem, and often it is even not possible to calculate the exact value, as it depends on free-space loss, diffraction, refraction, absorption, reflection, and so on. Because of that, path loss models for different scenarios were developed. In this work, we use the Path loss model from [17], which is expressed in Equation 3.5. This model takes into consideration only the Free Space Losses (FSL). It is important that in the used model, as with many other models (for example, the Hata model [18]), the received power depends on the distance logarithmically.

$$PL_{k,u} = 128.1 + 37.6 \cdot \log\left(\frac{d_{k,u}}{1\,000}\right) \quad 3.5$$

$d_{k,u}$ is the distance between the i-th BS and j-th UE. In this work, the path loss is the only parameter of the transfer function $\left|h_{k,u}\left(l_k^K, l_u^U\right)\right|^2$ depending on the distance.

The reverse path loss is introduced in Equation 3.6. to estimate the distance from the know path loss.

$$d_{k,u} = 10^{\frac{PL_{k,u} - 128.1}{37.6}} \cdot 1\,000 \quad 3.6$$

3.3 Assumptions

Due to the complexity of the problem, the proposed algorithm does not take into account the backhaul link - the connection between the FlyBS or the SBS to the core. However, the fact of limited backhaul capacity is represented by the maximal FlyBS f capacity c_f^F, which also covers all other factors constraining the backhaul link capacity. More information about the constraints caused by backhaul is introduced in [19].

The coverage area of the FlyBS is one of the limiting factors. The coverage area defines which UEs can be served. Two components influence the size of the covered area, the FlyBS transmit power (the higher the transmit power level, the greater the range) and the FlyBS distance to the UEs. The area that can be served by the FlyBS is a circle on the ground surface, for which the channel's transfer function between FlyBS and the UE still guarantees a sufficient quality of the received signal P_{min}^{Rx} as expressed in Equation 3.7.

$$P_f^{tx}\left|h_{f,u}\left(l_f^F, l_u^U\right)\right|^2 > P_{min}^{Rx}, \forall f \in \boldsymbol{F} \quad 3.7$$

The demand for the allocated bandwidth for each UE is tightly connected to the quality of the received signal. In this work, we assume that the bandwidth is shared between all UEs served by the same serving BS/FlyBS based on proportional fair scheduling. All UEs served by the same BS achieve the same capacity $c_{k,u}$, but the assigned bandwidth will vary depending on the quality of the received signal.

The proposed algorithm respects the constraint originating from the restriction and limits in the real world. We divide the limitations into sections concerning the following:

- physical attributes of drone - its maximal speed, its maximal fly altitude, maximal distance from the control station, power consumption,
- communication – FlyBS's maximal transmit power, backhaul connectivity, antennas, beamforming, and other advanced techniques,
- legal regulations – restricted airspaces (airports, power plants, etc.), necessity to be supervised by a trained operator,
- positioning algorithm – available inputs and periodicity of updates, computational demands, and precision of inputs.

Some of the limitations may be neglected within the context of this work, e. g. restricted airspaces, as it is not difficult to solve them later. On the other hand, some limitations are strongly impacting the whole design, such as network-related topics or speed of the drone. These cannot be neglected.

The thesis assumes that the transmit power of the SBS is constant and continuous and considers the situation with one FlyBS deployed in the search area. The model allows for further expansion to include the whole set of FlyBSs and implement cooperation between them. In this work, only downlink is taken into consideration.

3.4 Objective function

The objective is to find the best position of FlyBSs at the (future) moment. At this position, the FlyBS should serve as many UEs as possible while satisfying the constraints and increase the overall network system throughput. Thus, the objective is to find $\boldsymbol{L}^{F^*}$, as follows:

$$\boldsymbol{L}^{F^*} = \underset{l_f^F \in \mathbb{R}^3, \forall f \in K^F}{\operatorname{argmax}} \sum_{k \in \boldsymbol{K}^F} n_f \tag{3.8}$$

Subject to:

$$\sum_{u \in \boldsymbol{U}} c_{f,u} \leq c_f^F, \forall f \in \boldsymbol{F} \tag{3.8a}$$

$$\left\| l_f^{F^*} - l_f^F \right\| \leq \Delta t \cdot v_{max}, \forall f \in \boldsymbol{K}^F \tag{3.8b}$$

$$\sum_{k \in \boldsymbol{K}} \mathbb{1}\{\beta_{k,u} > 0\} \leq 1, \forall u \in \boldsymbol{U} \tag{3.8c}$$

$$\sum_{u \in \boldsymbol{U}} \beta_{k,u} \leq 1, \forall k \in \boldsymbol{K} \tag{3.8d}$$

$$\boldsymbol{\beta} \in < 0; 1 > \tag{3.8e}$$

$$\sum_{\substack{k \in \boldsymbol{K} \\ u \in \boldsymbol{U}}} c_{k,u} > \sum_{\substack{k \in \boldsymbol{K} \\ u \in \boldsymbol{U}}} c_{k,u}^{past} \tag{3.8f}$$

$$c_{u,k} > c_u^{req}, \forall u \in \boldsymbol{U} \qquad 3.8g$$

The Constraint 3.9a limits the sum of the UE capacities $c_{f,u}$ not to exceed the FlyBS f capacity c_f^F.

The FlyBS carried by the drone is also limited by its maximal speed, which determinates how far it can move within a given time interval Δt. When repositioning, the target position $l_f^{F^*}$ must be reachable within a given time window. This condition is expressed as Constraint 3.8b.

The UE can be connected to one BS only (one of SBSs or FlyBSs) at the moment, that is, to have resources allocated just from one FlyBS or SBS. The $\beta_{k,u}$ is thus non-zero value just for one BS from the whole set $\boldsymbol{K}$ for each UE. This limit is provided by the Constraint 3.10c The Constraint 3.11d expresses that BS cannot allocate more spectral resources than it has assigned. The Constraint 3.12e expresses the range of the $\beta_{i,j}$ to be between 0 and 1.

The deployed FlyBS has to have a positive impact on the system throughput of the network. The network system throughput (sum of all partial UEs' capacity $c_{k,u}$) after the FlyBS deployment has to be higher than the sum of partial capacity $c_{k,u}^{past}$ of the UEs before the FlyBS deployment. This is captured by the Constraint 3.13f. Finally, the Constraint 3.14g expresses that the UE capacity must be higher than the required one. However, since the traffic is not always predictable, the required traffic is maximized, following [20]

4 Proposed algorithm

This work focuses on the design and implementation of an algorithm for optimizing the position of the FlyBS in real-time with respect to its limitations. The proposed algorithm selects the set of UEs to be served by the FlyBS and finds the FlyBS position in order to maximize the number of the UE served and increase the network throughput.

The resulting algorithm proposed for independent FlyBS movement based on the behavior of UEs relies only on data that can be obtained from the real mobile network. The solution is based on the 4G networks as defined in the standards issued by the relevant standardization bodies, i.e., no change of standard air interface is assumed. Minor modifications to the network are necessary so that the FlyBS has enough information to make decisions, e.g., sharing of UEs' reports between BSs. However, any modifications introduced in the thesis can be put into real operation without having a significant impact on the functioning and performance of the entire network.

Even though the proposed solution targets low computational complexity, the calculations do not necessarily need to happen at the FlyBS. They can be offloaded to a computational node at the edge of the network or in the cloud.

4.1 Algorithm structure

As the positioning of the FlyBS presents a complex task, we divide it into three subtasks, prediction, position estimation, and clustering. Each subtask operates separately and together provide an algorithm to determine FlyBS positions in line with the objective of this thesis, defined in Equation 3.8. The subtask division is illustrated in Figure 4.1. The FlyBS position is being updated based on the received information about the UEs.

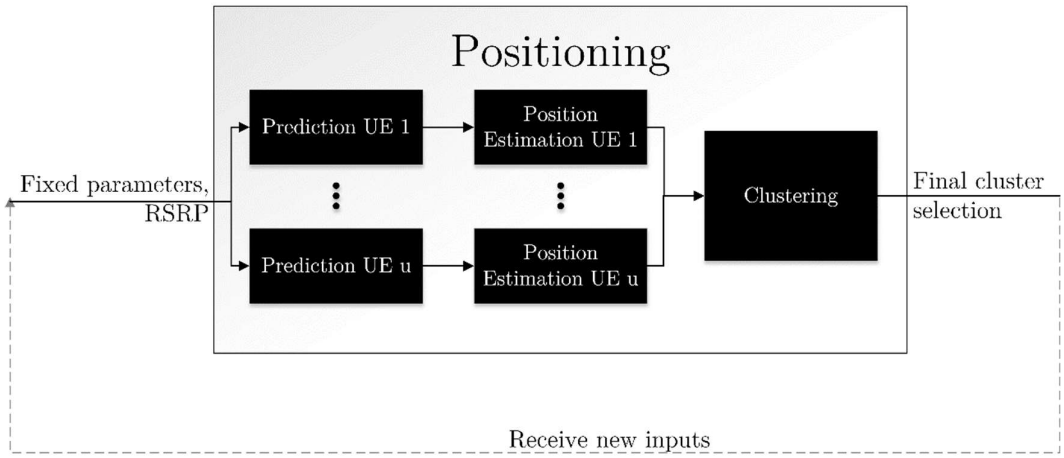

Figure 4.1 Interconnection of the algorithms.

The first subtask, prediction, acquires the signal levels of the UEs and predicts its future evolution. The prediction of the UE signal quality is necessary to position the FlyBS proactively, not to only react to past events. It is also necessary even when no time prediction is desired, as it helps to "smoothen" the shape of the discrete RSRP levels to the continuous signal. The continuous signal is beneficial as it allows us to estimate the UE distance from BS better.

The second subtask, position estimation, estimates the UE position based on the predicted RSRP reports. In this subtask, information from several BSs is evaluated to approximate the UE position.

When the position estimation is done, the UEs are divided into clusters in the clustering subtask. This subtask is further divided into two parts. First, possible clusters within the focus area are created. Second, the best candidate cluster is selected. Once the proper cluster is determined, the FlyBS moves to the center of the proper cluster.

The first two algorithms run in parallel for each UE, while the output of the second algorithm is jointly processed by the third algorithm.

4.2 Simulation model

In this thesis, multiple algorithms are proposed and have to be evaluated. Therefore, the simulation model with common parameters for all evaluated algorithms is presented. If we deviate in some parameters in a simulation, this information is given directly for the given simulation.

4.2.1 Simulation area and Base stations

The simulation area is divided into two parts – the focus area and the border area. The focus area is the main monitored and evaluated region, surrounded by the border area. The border area serves to minimalize the impact of the boundary conditions. For example, FlyBS can be placed only in the focus area. However, it may also cover some UEs from the border area, if the FlyBS is close to the edge. Without the border area, the results would be distorted. The FlyBS position would be shifted more to the center of the focus area, as it would cover more UEs than at the border. With the border area, all positions within the focus area have the same conditions. Note that not all simulation needs the border area, e.g., for position prediction only within the focus area. In that case, only the focus area is considered.

The structure of the simulation area is illustrated in Figure 4.2. The focus area is a square of 1 000 meters edge. 4 SBS are positioned in this area, every 250 meters from the border. The inter-site distance is 500 meters.

The border area is a band of 250 meters in width around the focus area. An additional 12 SBSs are placed in this area to complete the grid. The distance of 500 meters between SBSs

is kept. Height of all BS (both in focus and border area) is 30 meters above the ground level.

The transmit power of SBSs is 33 dBm, each SBS have 20 MHz bandwidth. The same bandwidth is available to the FlyBS; the transmit power of FlyBS is 25 dBm. Since the model is idealized, the absolute values are not so important when the proportion between SBS and FlyBS is preserved, and all SBSs have the same transmit power. Carrier frequency is 1 800 MHz and noise level -174 dBm/Hz.

All the simulations were focused on the downlink. This thesis assumes fair scheduling for all UEs.

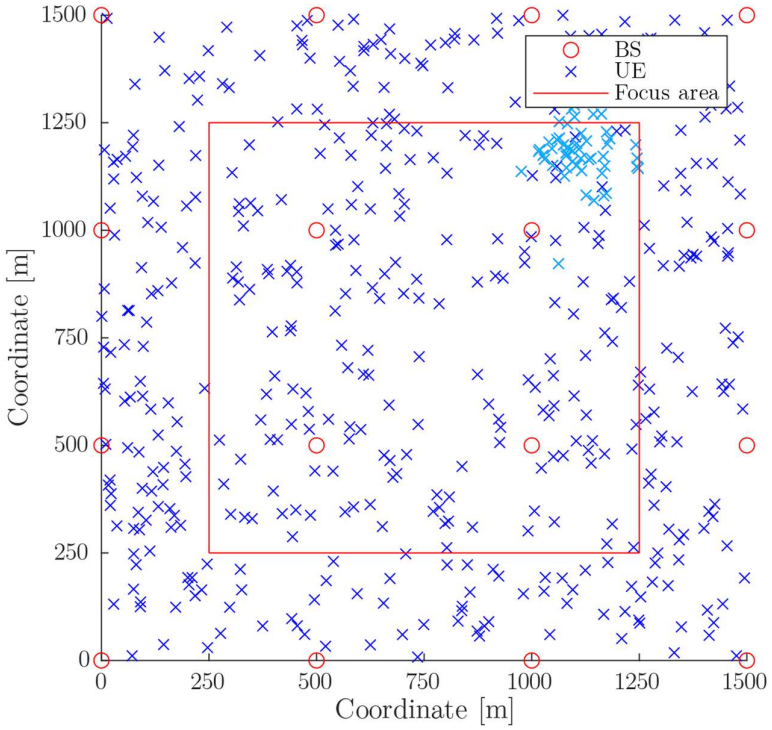

Figure 4.2 Simulation area and UE distribution.

4.2.2 Characteristics of User equipments

UEs are homogeneously distributed across the whole area in the simulations. The number of UEs in the focus area is not kept the same due to the randomness of the UE distribution.

For specific cases, clusters of UEs are deployed. The UE clusters are generated based on a multivariate normal distribution, defined by its center (often a random position within the focus area) and covariance defined by a symmetric matrix $\begin{bmatrix} 3\,000 & 0 \\ 0 & 3\,000 \end{bmatrix}$. The size of the cluster is 50 UEs. The UE cluster is visible in Figure 4.2 close to the upper right SBS in

the focus area (marked in a light blue color). In the simulations, it is assumed that all UEs are close to the ground level; thus, the z-coordinate is neglected (set to zero).

The UEs are assigned to the serving BS based on the highest SNR. All UEs report the quality of the signal by RSRP every 240 ms.

In some simulations, the UEs are moving. The speed is 1 m/s if not stated otherwise. Three possible movement models are used in the thesis:

1. oscillation around a fixed position (for example, to examine the jitter),
2. motion in one direction (for example, to validate the position estimation),
3. following the path from the generator, as described below (for example, to validate the clustering in time).

Many simulations in this thesis require heterogenous speeds of the UEs, with defined average speed. For this purpose, we present a trajectory generator, which works by connecting two random points in the simulation area. These points represent waypoints for the UE. If the cumulative distance between the points is less than it is needed to meet the required average speed, another point is added, and the process repeats.

The resulting movement is then smoothed using interpolation based on Piecewise Cubic Hermite Interpolating Polynomial (Pchip) [21]. As a result, the total trajectory increases slightly, and it is therefore tested whether the resulting speed has not enlarged more than is tolerable. If the speed is acceptable, the algorithm returns an array of the UE motion path samples. The size of the arrray depends on the required length of the simulation. An example of how the algorithm works is shown in Figure 4.3. The circles represent the random points through which the UE passes, the blue dotted line their direct connection in order how the points were generated, and the red line the smoothed trajectory, which is the output of the algorithm.

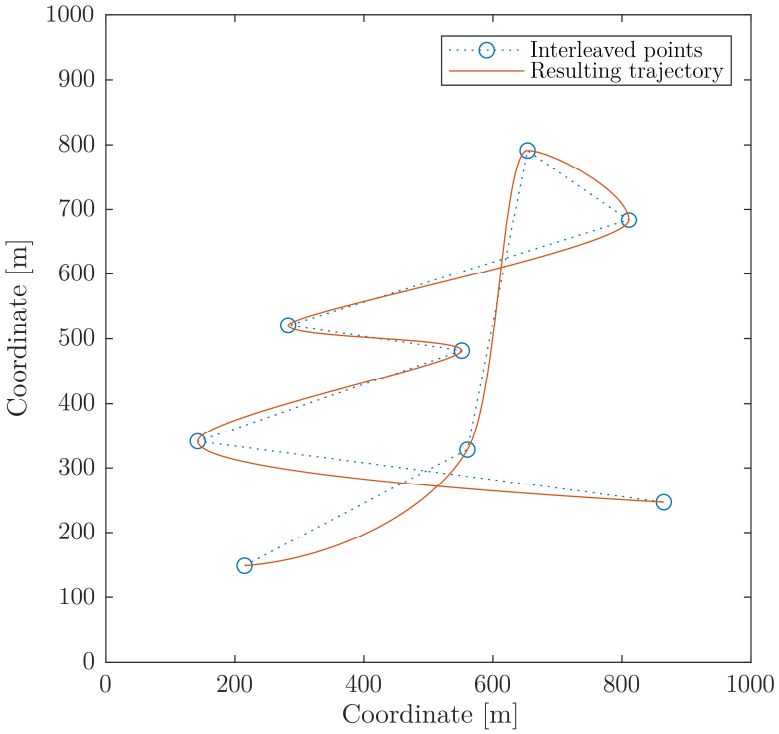

Figure 4.3 UE trajectory with defined average speed.

5 Prediction of the signal power level

The first subtask of the FlyBS positioning consists of the prediction algorithm, as illustrated in Figure 4.1. The prediction algorithm not only predicts a future signal value but converts the discrete RSRP levels into the continuous course. Therefore, a UE location can be estimated more precisely. The increased precision comes from the interpolation of the discrete RSRP values. This way, the impact of discretization is lowered. Distance recovery is also possible from a single RSRP value. However, the distances are then selected from a limited set of discrete values only.

The input data for prediction algorithm (in terms of dynamic information from the network) are RSRP measurements from the UEs. All other metrics, such as distance and absolute position, are derived from RSRP. The prediction time can be in the order of seconds into the future. The precision of prediction depends on the randomness of the UE mobility and the speed of UEs.

5.1 Challenges

It is necessary to find an algorithm that appropriately interpolates the obtained past values with a line or curve and read the received expected RSRP level at the desired point in time. The algorithm should take into account possible large differences in the frequency of RSRP changes and, at the same time, be able to eliminate jitter in the reported RSRP values when the UE is at the boundary of two levels. Both challenges, the jitter elimination and interpolation type are more elaborated in this section.

5.1.1 Jitter elimination

When the UE is slowly moving or is even still, the RSRP report value can change. This is caused by the propagation of the signal and changing environment and leads to jitter in the RSRP values [22], [23].

A basic algorithm used to smooth the waveform and eliminate noise caused by jitter is filtering, specifically the moving-average filter. The moving-average filter output sample $y(n)$ depends on several preceding input samples $x(n), \dots, x(n - windows + 1)$, where window denotes the number of considered points in the past. It is a straightforward and common way to reduce the additional jitter defined as:

$$y(n) = \frac{1}{window}\big(x(n) + x(n-1) + \cdots + x(n - window + 1)\big). \qquad 5.1$$

The application of moving-average filtering on RSRP values with jitter is illustrated in Figure 5.1. In the figure, the RSRP reports as received in time are marked by the blue line. When applied moving-average filter, we receive the course marked by the red line. Finally, when we round the filtered values to the nearest valid RSRP level, we receive the course without jitter, marked by the yellow line. The impact of a single short tremble after filtering

is negligible. However, the moving-average filter does not represent the right approach in this application. When the new RSRP value is received, it has a minimal effect (weight of 1/window) to the result of the filtering. This means that the smoothening is paid by the additional delay. For prediction purposes, it is necessary to consider the new report as soon as it is received. The phenomenon of additional delay is also visible in Figure 5.1.

The moving-average filter also smoothens the regular steps of received reports. Instead of step, it produces a ramp function. Once the ramp is rounded, the delay of the length of one half of the window size shifts the position of the change. This would negatively impact further prediction based on this data.

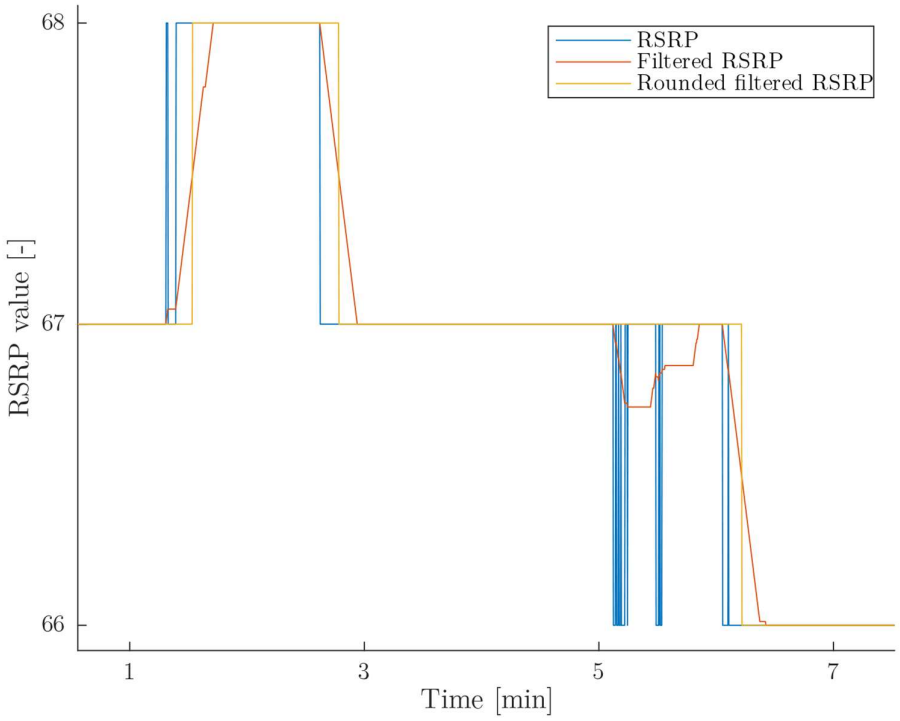

Figure 5.1 Filtering of RSRP measurements using moving-average filter.

It was experimentally determined via the simulation, that the window length needs to be in the order of tenths of samples to eliminate the jitter to an acceptable level. If we assume four reports per second, the minimum window creates delay which has the same or longer duration than the prediction outlook.

5.1.2 Interpolation points

Another challenge is the selection of points for further interpolation [24], [25].

The algorithm has to be working for both slow-moving and stationary UEs (for example, users staying in the same position for a long time, e.g., waiters in the restaurants) as well as for fast-moving UEs (runners, cyclist, etc.). Note that extremely fast-moving UEs (drivers, train passengers) are not in the scope for the FlyBS deployment.

The number of the past RSRP records, which is taken into account for interpolation, directly depends on the frequency of RSRP changes and RSRP changes depend on the behavior of the UEs.

The vast diversity of UEs' average speeds makes it impossible to set the interval or number of records as the fixed value. For example, the interval suitable for an average pedestrian is unnecessarily long for a messenger on the bike and too short for city gardener. The biker's UE will register several times more changes in the RSRP level compared to the pedestrian's UE.

5.2 Proposed algorithm for prediction

With the findings from the previous section, the jitter must be eliminated on-the-go basis. The best interpolation method seems to be a spline interpolation, which interpolates the course by a set of different mathematical functions [24]. The functions connect to each other and, since they are valid only for a short segment, they can be of a lower order [26]. However, we need to restore only the necessarily long last part of the record to estimate the trend. Therefore, the proposed algorithm is based on a lower order polynomial interpolation.

At the same time, the proposed algorithm treats jitter differently. It does not try to eliminate it in advance but evaluates backward whether the last change was regular, or it was a jitter. In the case of jitter, this change is ignored for future predictions. The result is shown in Figure 5.2. We see that the change in the RSRP stream is reflected immediately but ignored once it is evaluated as jitter, and the prediction continues with the same trend as it had before. Note that the impact on the FlyBS position will be smaller, as the precise UE position is calculated from more than one distance from BS.

It may look that even here, the delay is presented. However, this graph compares the real observed values and the predicted value for this point 20 samples in the past, which was done without the knowledge of the last samples. The delay caused by averaging would be added to this prediction error.

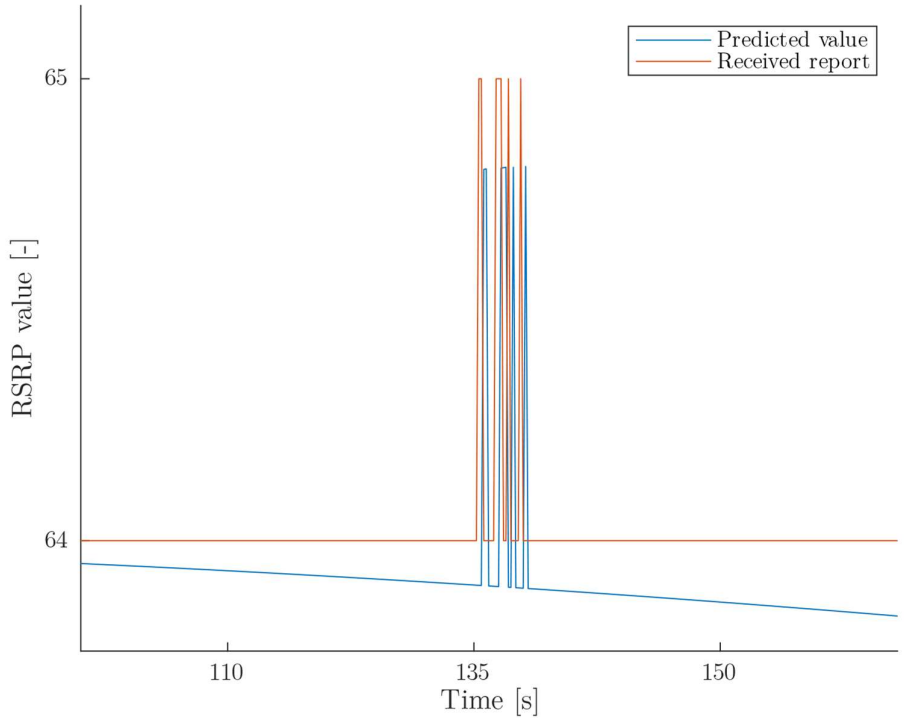

Figure 5.2 Impact of jitter on prediction.

5.3 Algorithm description

Along with selecting the type of interpolation, it is crucial to choose the right points through which the curve will be interleaved. The proposed algorithm selects one point for each continuous segment of values. The algorithm stores individual points in two arrays, which together clearly define the position of the points (RSRP value and timestamp). The array works as a FIFO buffer; the dimension of the array is discussed in more detail below. The analysis of the RSRP values and its duration is the core of the algorithm.

Besides the arrays with points and time stamps, the algorithm uses two counters and two auxiliary variables that maintain the current and previous RSRP values. In the algorithm description, three RSRP values are used:

- current value, which is the RSRP value of currently ending segment,
- past value, which is the RSRP value of the previous segment,
- new value, which represents the first RSRP value that is not the same as for the current segment.

If a new value is received, the duration of the current section ends, and it must be evaluated. In this case, it can be either an unwanted jitter or a valid change. If the new value is different from the past value, the current value is automatically considered valid.

This applies when there is a rapid decline or increase in signal quality. In this case, it is not jitter, and the current value will be taken into account in further estimates.

The information about the current segment (RSRP value and length) is first stored in auxiliary variables, then the RSRP value is updated according to the received report, and the counter resets. Taking into consideration observations from Figure 6.1, simple saving a new report would cause a systematic error; it is necessary to take into account whether it is an increase or a decrease in RSRP. If a decrease occurs, the RSRP value in the value array is set one greater than the actual received value. If there is an increase, the RSRP is stored as received. The auxiliary variable maintaining the current RSRP value is not affected by this modification.

If the newly received value corresponds to the value of the previous section, there is a possibility that the currently completed section is only a jitter. At that moment, the counter for the current section is evaluated. If the section is longer than the set threshold value, it is considered a proper change similarly as in the previous case. The loop starts again; in the next evaluation, the newly started section will be analyzed again to see if it was a jitter or not.

If the current segment was not long enough, the section would not be saved. In this case, the variables are modified as follows: the RSRP value is reset to the previous one (read from the auxiliary variable, not directly from the acquired measurement), the counter of the previous and current values are summed. In this way, the jitter section is added to the previous segment, and the process continues. However, it is necessary to adjust the points in the array for interpolation. Since the last section was evaluated as invalid, the arrays are shortened by removing the last element at each of them.

Once there are enough points to obtain a prediction, the polynomial curve fitting algorithm acquires the coefficients of the polynomial, which is then evaluated at the desired point in the future. Until enough points for prediction are collected, the proposed algorithm returns the current value as a prediction. This is not optimal but can be easily eliminated by starting the collecting of RSRP reports ahead.

During the testing of the algorithm, it was found that it achieves very good results for situations where there are frequent changes in the RSRP. Segments, where the RSRP does not change much, are a problem if a more significant increase or decrease precedes them. The algorithm still assumes a similar intensity of changes and therefore predicts change, even if there is none in real operation. For this reason, the limit of the predicted value was set to a maximum of ± one level from the current report. In the case of frequent reporting (several times per second), relatively fluent movement of users and a prediction horizon in the order of seconds, this limitation does not present a problem.

The predicted signal level can be easily converted to the distance using a reverse path loss model described by 3.6.

Algorithm 5.1 Prediction of the signal power levels.

input: RSRP threshold value *thres*,
stream of RSRP values from one BS *rsrp*,
timestamps of received RSRP value *time,*
order of interpolation curve *order,*
distance from the current timestamp when the prediction should be evaluated *horizon,*
number of points which should be used for prediction *amount, amount* > *order.*

output: Predicted RSRP value *predict.*

variables: Last received RSRP report $rsrp^c$,
RSRP level preceding the current one $rsrp^p$,
counter of how many same RSRP values were received since the last change $counter^c$,
length of previous segment $counter^p$,
array of timestamps used for interpolation (x-axis) $points^t$,
array of RSRP values used for interpolation (y-axis) $points^v$,
array of coefficients of data fitting curve *coef.*

Note: initialization is done when the first value is received
$rsrp^c = rsrp$
$counter^c = 1$
$rsrp^p = 0$
$counter^p = 0$
$points_1^t = 0$
$points_2^t = 0$
$points_1^v = rsrp$
$points_2^v = rsrp$

While received rsrp **do**
    if rsrp == $rsrp^c$
        $counter^c$++
    else if $counter^c$ > thres & rsrp != $rsrp^p$
        $counter^p = counter^c$
        $rsrp^p = rsrp^c$
        $rsrp^c = rsrp$
        $counter^c = 1$
        $points^t = points^t \cup time$
        if rsrp > $rsrp^p$
            $points^v = points^v \cup rsrp$
        else
            $points^v = points^v \cup (rsrp + 1)$
            if $||points^t|| >$ amount
                $points^t = points^t \setminus points_1^t$
                $points^v = points^v \setminus points_1^v$

```
                end if
            end if
        else
            counter^c = counter^c + counter^p + 1
            rsrp^c = rsrp
            points^t = points^t \ points^t_end
            points^v = points^v \ points^v_end
        end if
        if ||points^t|| > order
            coef = polynomialCurveFitting(points^t, points^v, order)
            predict = polynomialEvaluation(coef, points^t_end + horizon)
            if predict – rsrp < -1
                predict = rsrp – 1
            else if predict – rsrp > 1
                predict = rsrp + 1
            end if
        else
            predict = rsrp
        end if
    return predict
    end while
```

Note: the algorithm returns the correct values only after the initial learning phase

5.4 Evaluation

It is necessary to have a moving UE to verify the operation of the algorithm. Depending on the movement direction, the speed of approaching the BS or moving away from it varies. This enables us to test the prediction algorithm in changing conditions. Its position was obtained using the trajectory generator introduced in Section 4.2.1.

Figure 5.3 shows the outputs of the prediction algorithm. The figure compares the predicted value (2 seconds ahead) of the distance from the nearest BS for a given point in time obtained by the algorithm described above (red line) together with the actual value that occurred at that point (blue line). The graph is also supplemented by the direct conversion of RSRP to distance. This value is given here to make it clear how each new RSRP change improves the prediction.

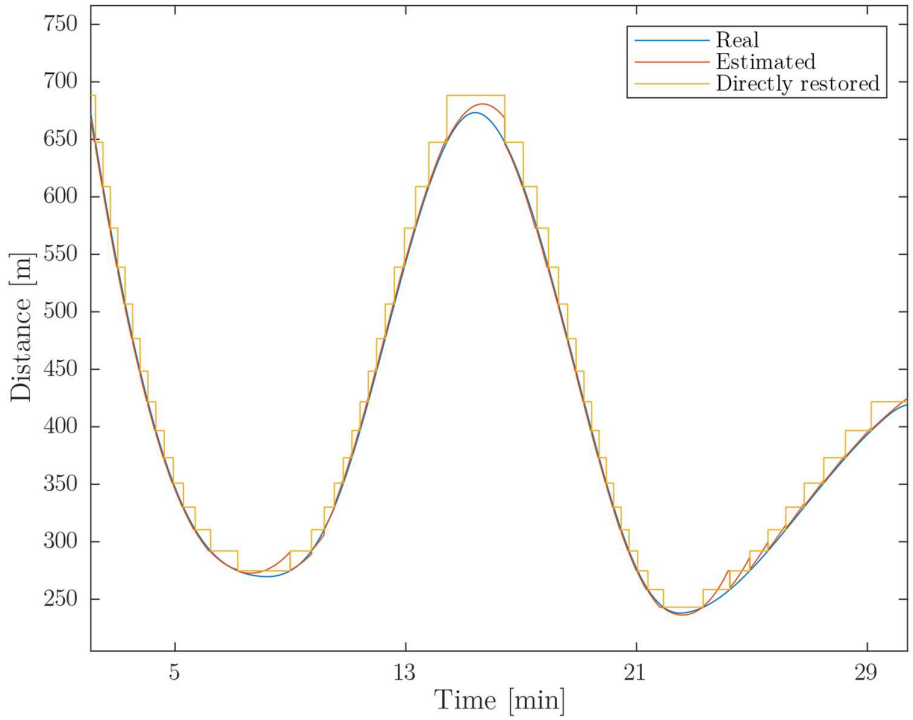

Figure 5.3 Comparison of real, predicted, and directly restored distances.

The accuracy of the prediction depends on three factors:

- the number of past points taken into account,
- the level of the interlaced polynomial,
- time horizon, how many samples ahead we make the prediction – higher value leads to decreased prediction accuracy.

The simulation to obtain the most suitable degree of the polynomial and the number of samples consists of analyzing 345 600 samples of a continually moving UE. This corresponds to approximately 4 reports per second for 24 hours. The speed of the device is 1 m/s all the time, which is a speed of slow walk. During the simulation, polynomials of the order 2, 3, 4, 6, and 8 are tested; the exanimated length of the points array is 3, 5, 7, 9, 13, and 17 samples. Note that not all the lengths are applicable to all orders of polynomials. For the interpolation by a polynomial of order n to be unambiguous, it is necessary to have at least n + 1 points available. Therefore, the number of considered points is always greater than the level of the polynomial. The position of the UE is predicted 20 samples ahead.

The measurement results in Table 5.1 present that for the given dataset and distribution, the lowest tested order of the polynomial curve, i.e. the quadratic curve, is the most suitable one. At the same time, the lower the number of interpolation points, the closer the result

is to the real value. This finding is in line with the fact that the interpolation algorithm tries to minimize the overall error. Therefore, for a large number of points and low order of the polynomial, the interpolation through the last known points will not be so precise, as the curve needs to fit also the previous points.

Table 5.1 Prediction error distances – polynomial interpolation.

Order of polynomial	2	3	4	6	8
3 input points	2.41 m	-	-	-	-
5 input points	3.32 m	4.47 m	4.61 m	-	-
7 input points	4.66 m	5.82 m	4.39 m	7.27 m	-
9 input points	5.34 m	7.10 m	5.47 m	6.35 m	8.94 m
13 input points	7.23 m	9.14 m	6.83 m	5.57 m	7.26 m
17 input points	8.82 m	11.35 m	6.87 m	6.52 m	7.65 m

On the same setup, we also test the prediction using the linear interpolation (first order polynomial). The results are significantly worse than for polynomial interpolation: when using 2 input points, the average error distance was 7.42 meters and worsen up to 22.31 meters when 17 input points are used.

Table 5.2 Prediction error distances – linear interpolation.

Linear interpolation	
2 input points	7.42 m
3 input points	9.86 m
5 input points	15.15 m
7 input points	19.05 m
9 input points	21.41 m
13 input points	23.08 m
17 input points	22.31 m

Along with the examination of the ideal setting of the interpolation curve for prediction, the accuracy of the prediction is investigated at different speeds and horizons. For this simulation, the best combination of polynomial order and number of points from the previous simulation was used – quadratic curve with 3 points to interleave. The metric is the distance error (an absolute difference between the actual and estimated position). We evaluate the relation between distance error and the prediction horizon for five different speeds of UE: 0.5, 1, 2, 4, and 8 m/s.

The dependency of a median of distance error on the prediction horizon is displayed in Figure 5.4. It is clear from the figure that for prediction close to the present time, the error is relatively small but increases rapidly. For more distant estimates, the error rate change is less significant, as the error rate is already relatively large. We also see that the prediction precision is better for lower UE speeds.

The algorithm encountered the limitations caused by a condition on one level differentiation when predicting more than a minute in ahead. Therefore, the results for such a long estimate are not meaningful.

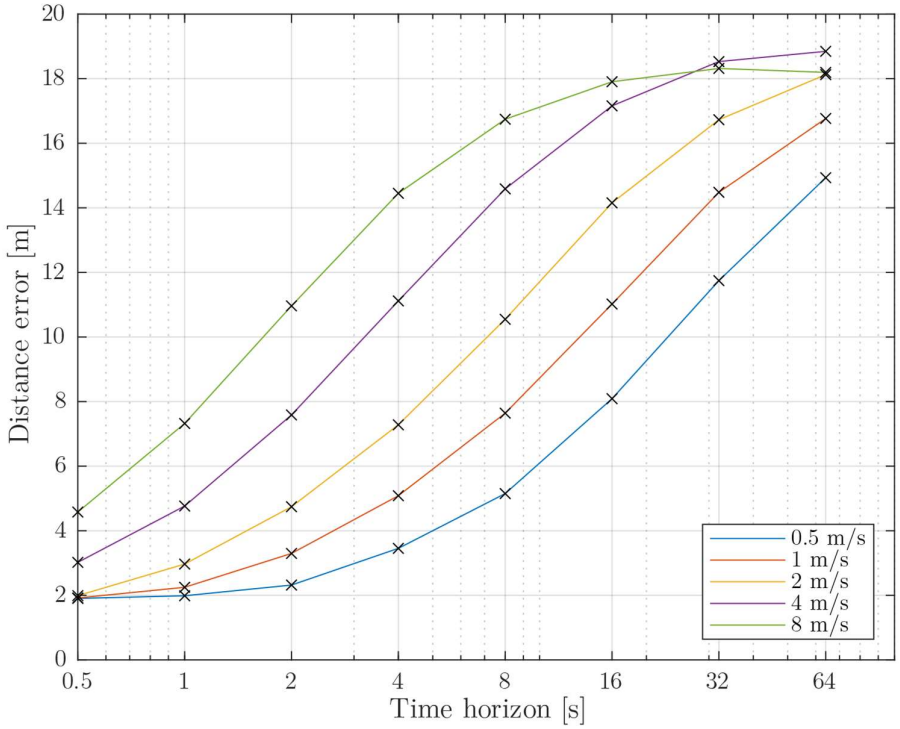

Figure 5.4 Distance estimation error for different UE speeds – median.

Looking at the average value of the error in Figure 5.5, it is possible to observe that the slope of the average error rate depends on the speed. If the UE moves slower, the inaccuracy of the estimate increases more slowly over time than for higher speeds. Interestingly, the average error first decreases with increasing speed and then increases from a certain limit. This fact may be due to the type of interpolated curve.

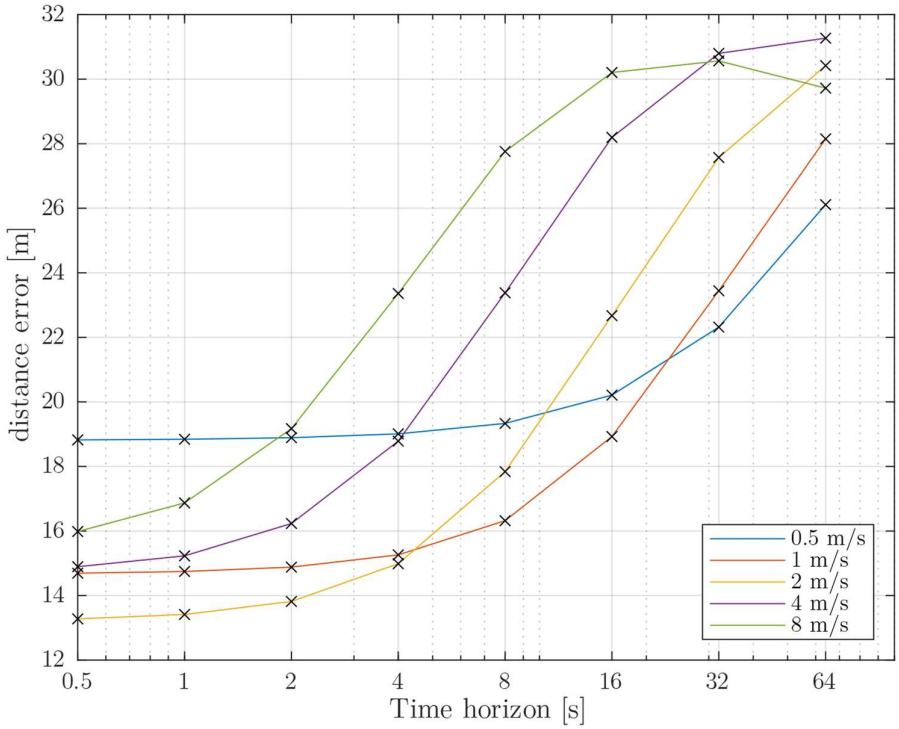

Figure 5.5 Distance estimation error for different UE speeds – average.

6 Position estimation

In order for the FlyBS to correctly evaluate its movement, it is necessary to know the location of the UEs. This knowledge is also crucial for the correct selection of the initial deployment position and operation of the FlyBS.

The inputs for the position estimation algorithm are distances obtained from the prediction algorithm. For an unequivocal estimation of the position, it is necessary to have distances at least from three BSs in order to calculate the position that corresponds to the reported values.

Assuming the logarithmic dependence of RSRP level on distance as defined in Equation 3.5, the best resolution is in the proximity of the BS and worsens with increasing distance. The difference in recovered distance precision is significant – for situations when the UE is close to the BS and thus has a high RSRP level, the precision is in order of meters. For low reported values, one RSRP level difference means distance difference in the order of hundreds of meters.

Figure 6.1 shows the movement of a UE around the simulation area during the time. This movement is represented in terms of distance to the nearest BS (blue line). The directly recovered distance from the RSRP values using Equation 3.6 and Table 2.1 (red line) represents the non-linear mapping of distance RSRP levels. When the UE is around 450 meters away from the BS, the difference of 1 RSRP level represents ca. 22 meters. For the distance around 750 meters, the 1 RSRP level difference equals to more than 30 meters.

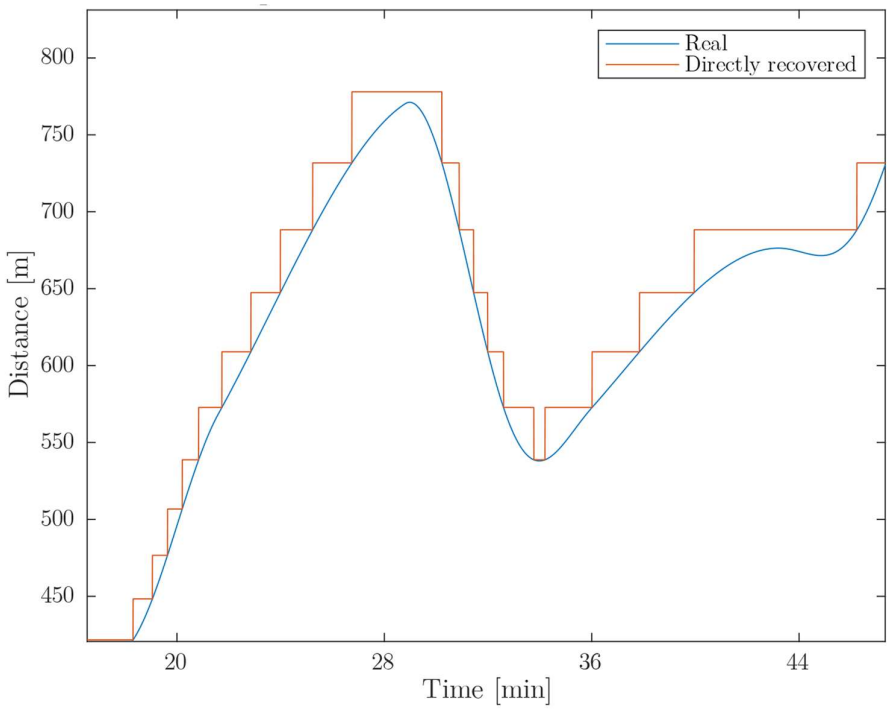

Figure 6.1 Comparison of real and calculated distance.

6.1 Proposed algorithm for position estimation

Using the distance as a radius, we can draw a series of circles with SBSs in the centers. The circle represents an infinite number of possible positions of the UE from the perspective of the SBS to which the circle belongs.

In the ideal case, all the circles belonging to different SBSs would cross in one exact point. However, this situation is unlikely due to the distance restoration from degraded information caused by reporting discrete RSRP values. When the distance obtained from RSRP is plotted, the circle represents the upper bound of real position due to RSRP mapping scheme. In the real environment, the situation will also be influenced by signal propagation inhomogeneity. This is seen in Figure 6.2., where the estimated UE position is the point with a minimal sum of the squares of the errors from all circles.

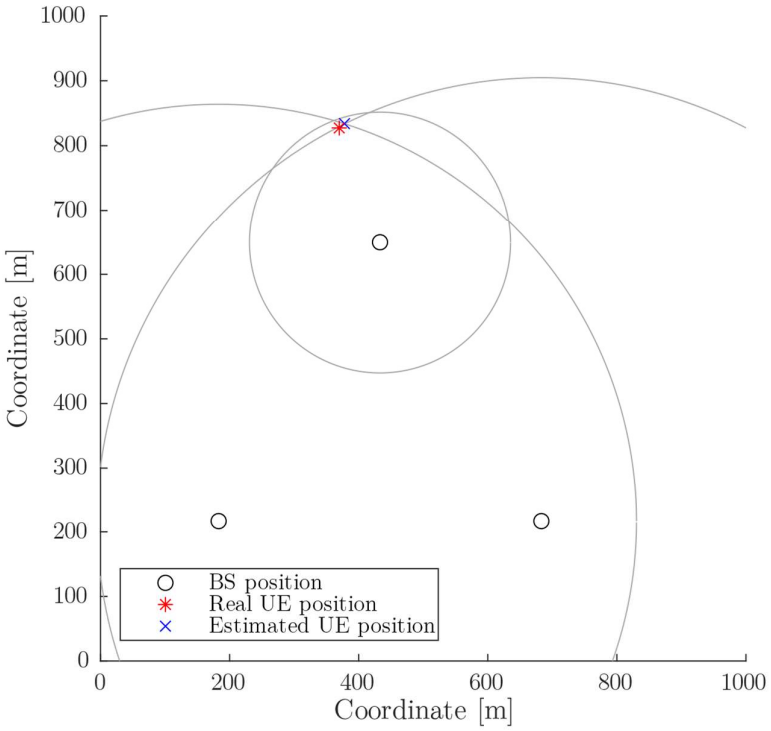

Figure 6.2 Estimation of the UE location.

The estimated UE position is the point defined by the vector of coordinates, which has the lowest sum of the squares of the errors, as expressed in Equation 6.1.

$$l_u^{U*} = \underset{l_u^U \in \mathbb{R}^3, \forall u \in \boldsymbol{U}}{\operatorname{argmin}} \sum_i \left(\mathrm{d}_{u,k}^2 - \left(l_k^K - l_u^U \right)^2 \right)^2 \qquad 6.1$$

The function above can be minimalized using gradient so that Equation 6.1 is met.

$$\nabla \sum_{i} \left(\mathrm{d}_{u,k}^2 - \left(l_k^K - l_u^U \right)^2 \right)^2 = 0 \qquad 6.2$$

The mentioned system of Equations for finding the position of UEs is possible to solve using mathematical solvers.

6.2 Evaluation

We can enhance the algorithm by selecting the proper inputs. We may set how many BSs and which one should be taken into consideration and how they should be included (whether some form of weighting will improve the results). We carry out a set of simulations, focusing on these open questions.

The simulation was done for three, four, five, and seven SBS. In each scenario, the SBS were deployed evenly, so they created a shape of triangle, square, pentagon, and hexagon with one SBS in the center. The distance between the two closest SBS remained the same during all the scenarios, as well as the area served by one SBS. This setup results in a larger overall area while deploying more SBS while preserving the SBS density the same. This is essential to get comparable results for each scenario.

The simulation is performed on a sample of 1 000 randomly generated UE positions. The direct distance conversion from RSRP value is used, without prediction.

The first simulation focuses on the number of considered SBSs and weighting based on the distance between SBS and UE. The positions of UEs are calculated without weighting, using linear weighting, and exponential weighting. The information from nearby base stations is in case of weighting stressed the most. The same set of input data (UEs positions) was used for each approach.

In this simulation, the focus area was changing depending on the number of SBS. This ensures a similar distance for UE to the nearest SBS in all scenarios, as well as the same average served area by one SBS.

The results in terms of average error distance for all four described scenarios are presented in Table 6.1. The table shows that the best results were obtained with as few BS as possible. This may be striking at first sight; more measurements do not mean better results. In the case of three BS, the whole area was the smallest. This means that also the distance to the most distant BS was the smallest, ensuring a high RSRP level and better resolution.

Table 6.1 Error distance dependencies on weighting and number of SBSs.

Number of SBSs	3	4	5	7
Without weighting	20.69 m	22.92 m	25.65 m	24.16 m
Linear weighting	20.68 m	22.89 m	24.42 m	23.61 m
Exponential weighting	20.78 m	21.54 m	23.13 m	22.04 m

Table 6.1 also shows that weighting does not have a significant effect if the number of BSs taken into account is small. Its importance increases with an increasing number of BS. Overall, the best results are achieved by exponential weighting, which is, however, computationally demanding.

The position is estimated in the 5 BS scenario based on different sets of selected BS to verify the hypothesis that the accuracy depends on the selected SBSs. The results are presented in Table 6.2. This allows us to compare the impact of which BS serves as the input for the algorithm. If the position is calculated from a set of three closest BS reports, the average distance error is 37.13 meters. On the other hand, when calculating using the data from the three most distinct ones, the error is significantly larger, 60.34 meters. The simulation confirms the observation that it does not matter how many reports the UE can collect, what is important is to use just the essential minimum of reports from the closest SBS.

Table 6.2 Error distance dependencies on SBSs selection.

Considered SBSs	3 closest	All 5	3 most remote
Error distance [m]	37.12	47.77	60.34

6.3 Enhancement for a single RSRP report

Figure 6.3 shows real and estimated positions for a set of users in the 3 SBS scenario. From the figure is visible that the estimated positions are further from the center of the area compared to the real positions. This phenomenon is caused by the way distance is contained in the RSRP.

If we simply restore the distance from received RSRP values, the real distance will always be smaller, as illustrated on Figure 6.1. With this knowledge, we can adjust the RSRP (respective radius, which depends on RSRP) before entering the algorithm to estimate the position.

Simple increasing the RSRP by 0.5 minimizes the error distance. In the original proposal, the error distance was between 0 and a positive value m. In the case of modified RSRP,

the error distance is between -m/2 and +m/2. The absolute maximal error distance is thus halved.

Simulation proved that the error distance decreased by 47.09 % to 11.10 meters for the same test set randomly placed UEs.

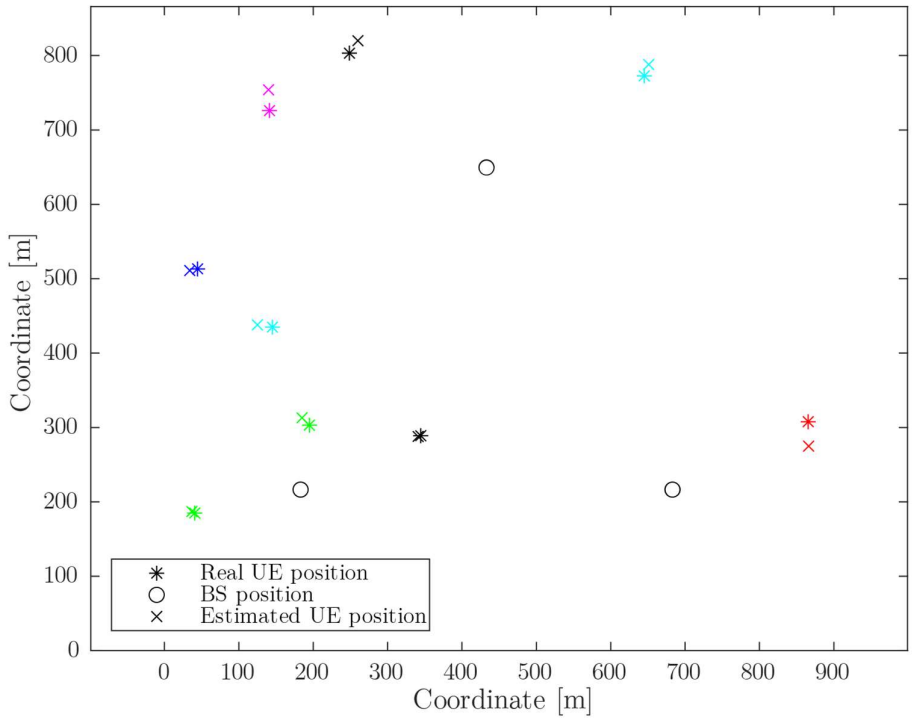

Figure 6.3 Comparison of estimated and real UE positions.

The improvement mentioned above by statically shifting RSRP is applicable when one measurement report of RSRP values is available. The positioning based on the prediction presents a better way how to adjust RSRP value. Prediction can be fully used only when a sufficient number of reports for is available.

6.4 Connection to the prediction algorithm

A more complex simulation is done to demonstrate the benefits of prediction prior to position estimation. 165 UEs are presented in the area, 30 of them joined into the cluster with significantly smaller mutual distances. The UEs within the group follow a predefined trajectory by the path generator with an average speed of 1 m/s. Remaining UEs move with speed equally distributed between 0 to 2 meters per second in one random direction. During the simulation, 2 400 samples per each UE is evaluated. The prediction horizon is set to 12 samples, i.e., 3 seconds.

The real trajectory of UEs is recorded by gradient color change in Figure 6.4. The first sample is represented by the light blue, while the last sample by the dark blue color. The exact location of UEs is not accessible directly to the evaluation backend, so the system recalculates the position from reported measurement, as introduced in Section 5.2.

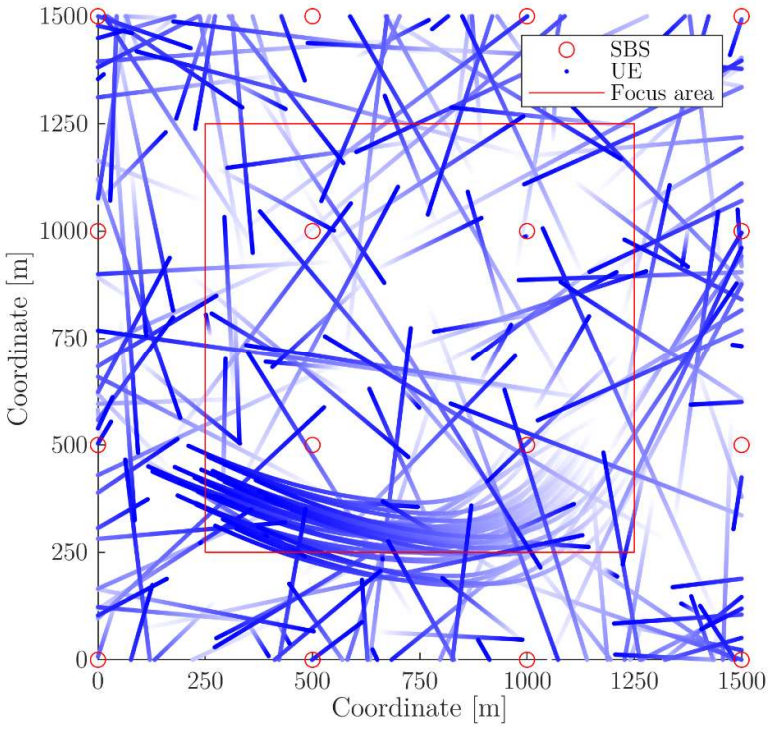

Figure 6.4 UE movement in time – real positions.

Estimation of the positions using current RSRP values (without prediction or enhancement from Section 6.3) is shown in Figure 6.5. From this figure, it is clear that the estimated trajectory is heavily impacted by the distance to the closest SBS and does not follow the original straight line – it is also not continuous. The median of the error distance (absolute difference between real and estimated position) is in this case 15.55 meters and average 16.06 meters.

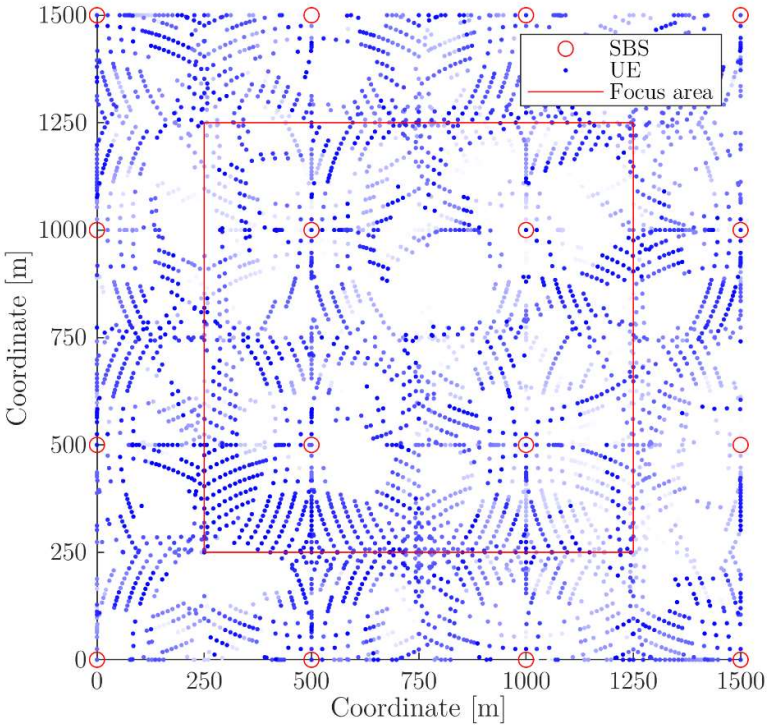

Figure 6.5 UE movement in time – directly estimated positions.

The situation is different in the case of estimation using predicted values, as depicted in Figure 6.6. The figure shows some discontinuities, but the overall movement record is clear and faithful. The error distance median decreased to 5.94 meters, the average is 11.88 meters. This shows one important fact – the prediction, which creates "smoothening" of received signal levels, provides significantly better results. In addition, the gained time present a valuable resource.

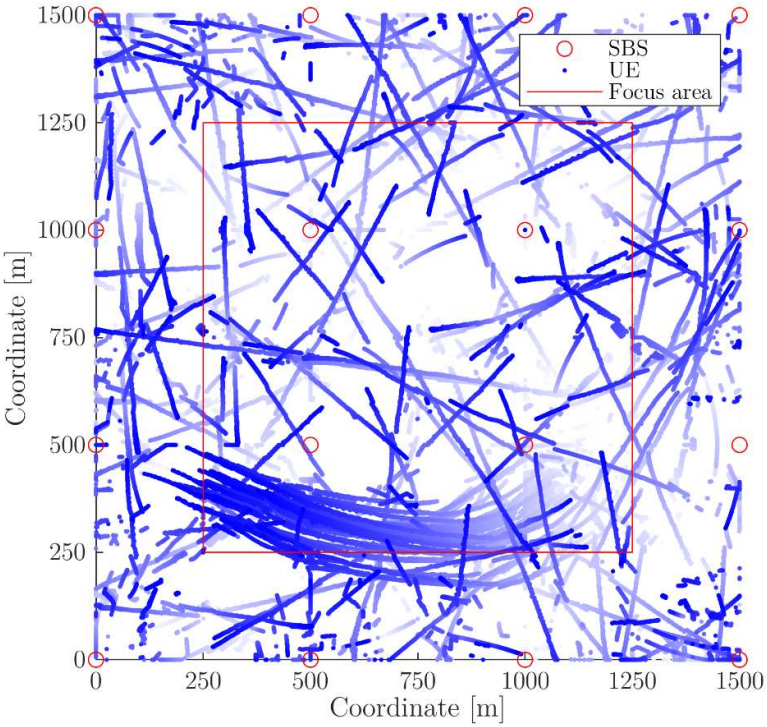

Figure 6.6 UE movement in time – predicted and estimated positions.

7 Clustering

The FlyBS provides mobile connectivity to the UEs the same way as the SBS, i.e. serves a set of UEs within its coverage area. Therefore, to determine UEs that are to be served by a FlyBS or BS clustering algorithms can be exploited.

Finding possible UE clusters is the goal for the third subtask of the positioning algorithm, as shown in Figure 4.1. It is the most challenging part of the work, as the solution space is very vast. There can be more possible UE clusters, so the result of the clustering algorithm is a set of several possible clusters. The UE positions are always the input to the clustering algorithm, as well as the physical capabilities of the drone carrying FlyBS and the FlyBS itself.

7.1 Proposed algorithm for clustering

The algorithm chooses the set of users to be served based on the FlyBS communication capacities, UE distribution, and the quality of the channel between UEs and SBS. The selection is made in a way that the FlyBS serves rather UEs with poor signal quality far from the SBS than the large UE group with sufficient signal strength form SBS. This leads to better resource utilization if proportional fair scheduling is applied in the network. The network chooses the serving cell by traditional procedures as defined for the given mobile network, based on the signal quality. Even though the set of UEs served by the FlyBS may be controlled by the network elements, the proposed approach prefers to avoid it and use only the FlyBS position as a way how to affect which UEs will be served.

The proposed algorithm thus targets users with the worst SNR ratio given by the signal quality condition:

$$SNR_{k,u} < SNR_{set}, \forall u \in \boldsymbol{U}, \tag{7.1}$$

UE u will be evaluated as a candidate for association with FlyBS and taken into account in the selection if the UE meets the requirements in Equation 7.1. The threshold value SNR_{set} can change depending on the situation and the covered area.

The proposed clustering algorithm for the FlyBS placement needs for its proper functioning:

1. the (estimated) position of all users in the focus area,
2. the signal strength from serving SBS in terms of RSRP value for each UE,
3. FlyBS radius r_{FlyBS} which can be obtained by putting the maximal transmit power of FlyBS into Equation 3.6,
4. RSRP threshold value from which the signal fulfills the signal quality condition.

The algorithm itself works in 3 phases, each reducing the proportions of data to be evaluated. In the first phase, the UEs fulfilling the signal quality condition are selected. In the second phase, the clusters from UEs fulfilling the signal quality condition are created.

In the third phase, number of all UEs in most promising clusters is calculated. The scheme of operation is shown in Figure 7.1, a more detailed description follows.

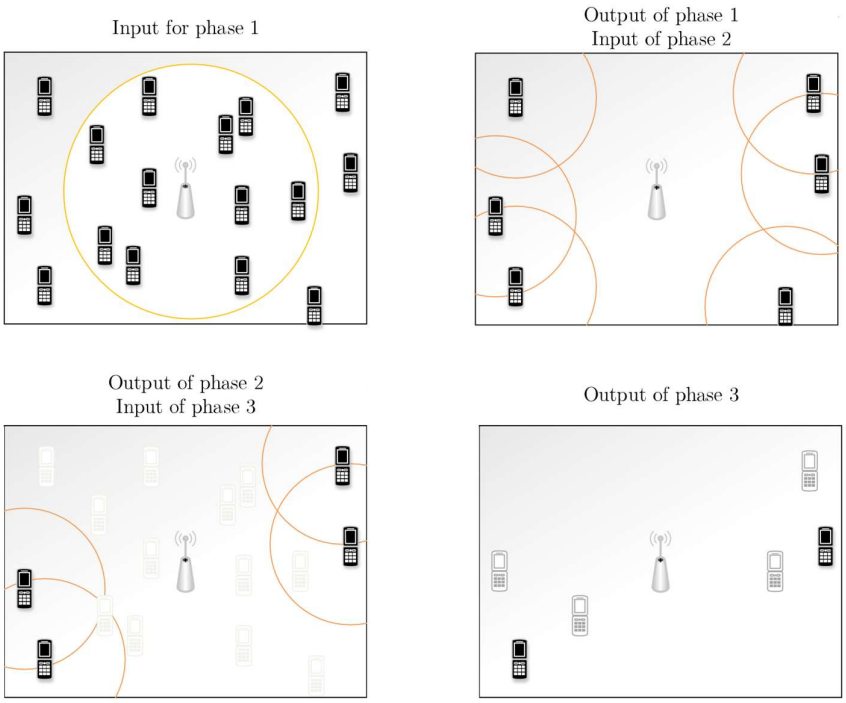

Figure 7.1 Clustering algorithm – overview.

7.2 Algorithm description

As a first step, the algorithm evaluates which users meet the signal quality condition, i.e. findsd candidates for FlyBS coverage. This step excludes users near the BSs that achieve high RSRP values. In the simulations, the decision RSRP value is fixed at 70 due to lower signal attenuation compared to the actual deployment. The condition with selected threshold fulfills around half of all UEs in the area. Note that as the RSRP value we understand the reported value directly as presented in 3GPP standard and introduced in Section 2.2. The value can also be determined dynamically (for example, as the median RSRP from all UEs). The set of users who meet the condition in the simulations can be seen in Figure 7.2.

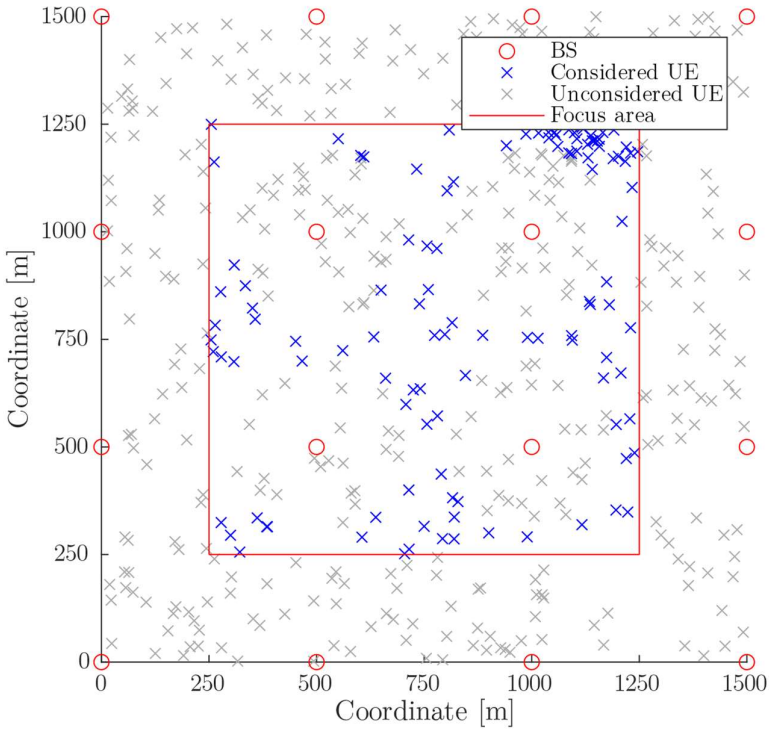

Figure 7.2 Example of candidate UEs for the clustering algorithm.

The positions of users who meet the condition for RSRP also represent the possible positions where to place the FlyBS. This reduction is beneficial in the way of reduced complexity while having a minor impact on the resulting cluster size. Simulation with 255 UEs comparing the optimal position (which was found using the brute-force approach) and the best position after the reduction showed that the difference of biggest cluster size was maximum of 2 UEs in ten runs of simulation (on average 7.44 %). With respect to uncertainty proceed from the position estimation, this simplification is justifiable.

One by one, we evaluate each possible cluster center, based on how many UEs with a signal worse than the threshold value are in a circle defined by the FlyBS radius. The number of tested positions can be reduced even more, which is further described in Chapter 10.

Once all possible centers have been evaluated, the results are sorted in descending order based on the number of UEs in the cluster, meaning that in the first place is the position that has the most users around it, who are simultaneously fulfilling the signal quality condition. However, it would not be right to declare results obtained only based on users fulfilling the signal quality condition. Therefore, for a relatively small number of the first clusters in the list, it is evaluated how many of all users are in range. From this new list, the final cluster to be served is selected (as later discussed in Chapter 8).

Due to the sequential approach, the number of iterations of each loop (phase) is shortening. During the first loop, it is necessary to go through all users, but only a simple comparison is made. The second loop (evaluation of users in range) will have significantly fewer iterations – the second loop will have approximately half of the iterations compared to the first one in the case of the RSRP threshold set to the median of RSRP values from all UEs. In the last loop, when the total number of UEs in the range is evaluated, an even smaller number of points of interest (cluster centers) is evaluated (in the order of percentages to the total number of points).

Algorithm 7.1 Clustering.

input: Position of UEs within the area defined by its location *location*,
maximal radius the transit power of FlyBS allows to serve r_{FlyBs},
RSRP threshold value $rsrp^{thres}$,
RSRP value of each UE $rsrp_i$,
number of clusters to return n.

output: Possible cluster centers *centers*, ||center|| <= ||location||,
UEs in these clusters *members*.

variables: Matrix of cluster members for each center$_i$ *list*.

Phase 1: eliminating the UEs with high SNR
centre = {}
for each j **in** location **do**
    if rsrp$_i$ < rsrp^{thres}
        center = center ∪ location$_j$
    end if
end for each

Phase 2: creating possible clusters
list = {}
for each i **in** list
    for each k **in** center
        if | center$_i$, center$_k$| < r$_{FlyBs}$
            list$_i$ = list$_i$ ∪ centre$_k$
        end if
    end for each
end for each
sort center, **such that** center$_i$ < center$_{i+1}$ & |list$_i$| ≥ |list$_{i+1}$|

Phase 3: evaluating only the biggest clusters
for first n **in** center
    for each i **in** location
        if ||center$_n$ - location$_j$|| < r$_{FlyBs}$

```
                    members_n = members_n ∪ location_j
                end if
            end for
        end for each
        sort members, such that members_n < memebers_{n+1} & |members_n| ≥ |members_{n+1}|
        return center, members
```

7.3 Comparison with other clustering algorithms

To show the performance of the proposed clustering algorithm, we compare it with two existing ones. The first approach is the clustering based on the Gaussian Mixture Model (GMM) [27], commonly used in machine learning for clustering based on unsupervised learning. The second algorithm used for clustering is Mean Shift (MS) [28], which finds the location with the highest density of data points in a given area. Clustering based on the MS does not need to know the number of clusters in advance in comparison to GMM.

7.3.1 Gaussian Mixture Model

Mixture models are generally used for unsupervised learning. The GMM is commonly used for evaluating a set of points distributed in space, where it is assumed that the points follow Gaussian distribution. The Gaussian distribution is one of the most common distributions to model real-world situations. The GMM is often used for audio and video analysis, but it was proven in [29] that also the UEs could be divided into clusters based on this model.

The GMM is a probabilistic heuristic approach to statistically analyze the set of data $\boldsymbol{x}$:

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} \omega_i \, g(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \qquad 7.2$$

Equation 7.2 presents a one-dimensional model, which is parametrized by the weight of i-th component (mixing probability) ω_i, its mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$. The mean represents the center of the cluster, the covariance shapes the cluster. λ represents the latent (hidden) parameters of the GMM model, and M is the total number of components.

An iterative method called Expectation - Maximization (E-M) algorithm is exploited to estimate the GMM parameters. E-M algorithm represents an iterative approach to minimize the error function. The algorithm alternates between two steps:

1. expectation step, where the function for the log-likelihood that the i-th point is generated by the j-th Gaussianis is created,
2. maximization step, where the log-likelihood is evaluated and the parameters updated.

In other words, during the expectation step, we find the weights which denote the probability that the point belongs to the j-th cluster. In maximization step, we evaluate the clusters based on the weights of all points.

Each iteration increases the accuracy of the log-likelihood function, and the algorithm will converge [27].

7.3.2 Mean Shift Clustering

MS is an analysis technique [28] based on examining the density function. MS algorithm is sometimes referred also as a mode-seeking algorithm, as it shifts the points in the direction of the highest density of data points. It is used for Mean Shift Clustering (MSC) [30] and image processing.

The algorithm is based on the Kernel Density Estimation (KDE) – a statistical method of estimating the probability density function. The algorithm assigns the weight (in a nonparametric statistic called kernel) to each point in the data set. By this, it creates the probability surface. A common kernel is Gaussian one:

$$k(x) = e^{-\frac{x^2}{2\sigma^2}} \tag{7.3}$$

The Gaussian kernel is an exponential function, where the standard deviation σ is used as the bandwidth parameter for the MSC algorithm. The bandwidth affects the resulting shape of the density function, as it governs the number of peaks and overall smoothness of the KDE.

MSC algorithm locates the maxima of a density function for a given set of data. Similarly to the GMM, MSC is an iterative method. On the contrary to other popular clustering algorithms, such as K-means [31], MSC does not require to have an anticipated number of clusters as an input; it only depends on the bandwidth σ of the kernel.

The MSC starts with an initial estimate x. The estimate is replaced by so-called mean shift m(x) in each iteration until it reaches the KDE surface peak:

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)} \tag{7.4}$$

The mean shift depends on the kernel function K, which denotes the weight of nearby points. $N(x)$ is the neighborhood of x.

7.4 Evaluation

In the case of GMM, we follow the scenario introduced in [29]. Several clusters in each SBS serving radius are first calculated using the E-M algorithm. As the radius of the FlyBS is

a circle, only used outputs from the E-M algorithm are the means representing the cluster centers. The radius is defined by the transmit power of the FlyBS.

The necessary number of iterations of the E-M algorithm to converge to the optimum depends on the initial selection of points. In the simulations, the iteration counter limit is set to 100. Another limitation is that with the given set of data, a maximum of 4 clusters per BS can be created.

Once the cluster centers are known, we calculate how many UEs belong to the given cluster. The clusters are then ordered based on the number of users. Note that this approach does not take into consideration different channel conditions and uses only the positions of users.

In the case of MSC, we choose a slightly different approach. We introduce the same signal quality condition as for the proposed algorithm in Section 7.1: only users with poor signal quality are considered into the clusters. The bandwidth of the kernel function is set static. The ideal bandwidth depends on the distribution of the UEs and represents the key parameter denoting the number of clusters that will be created. Finding a proper bandwidth would require developing an additional algorithm, which would increase the overall complexity.

For all GMM, MSC, and proposed algorithm, the cluster which is covering most users is chosen as the final one.

The clustering algorithms are tested in two scenarios. In the first one, all 450 UEs are distributed uniformly over the whole area, around half of them in the focus area. In the second scenario, an additional group of users forming a close group is added to the uniformly distributed UEs.

Especially in the second scenario, the first FlyBS is placed above the UE group independently on the clustering algorithm. Because of that, more FlyBS were successively added to the same focus area in order to see the behavior of the clustering algorithms. The previous FlyBS are for the estimation of the new one considered as static BS, which differs from the SBS only by the transmission power (the transmission power level remains on the level set for FlyBS).

To evaluate the results, we use two parameters. First is the average throughput for UEs, calculated over all UEs. This ensures that the results are not distorted by the different number of UEs connected to each BS. If the average would be calculated only from the UE throughput for each BS (which is due to the proportional fair scheduling the same for all connected UEs), the lower number of UEs connected to the FlyBS would mean an increase in the average throughput. This would corrupt the results.

The summary of the comparison based on scenario 1 is presented in Table 7.1. The graphical deployment of all three FlyBSs by three different algorithms shows Figure 7.3. Each algorithm is represented by one color; each round of FlyBS deployment is represented by one type of line.

The positions calculated by the proposed algorithm and MSC algorithm are often close to each other. However, the proposed algorithm covere more UEs in almost all simulations. Due to the close location of the clusters, also the differences in throughput are lower compared to the GMM algorithm. In some cases, the MSC performs better than the proposed algorithm, even though the results are never significantly distinct.

The placement of the FlyBS based on the GMM algorithm clearly shows how important it is to take into account the position of SBS[1]. The FlyBS is quite often deployed close to the existing SBS, which results in strong interference and low utilization of the FlyBS.

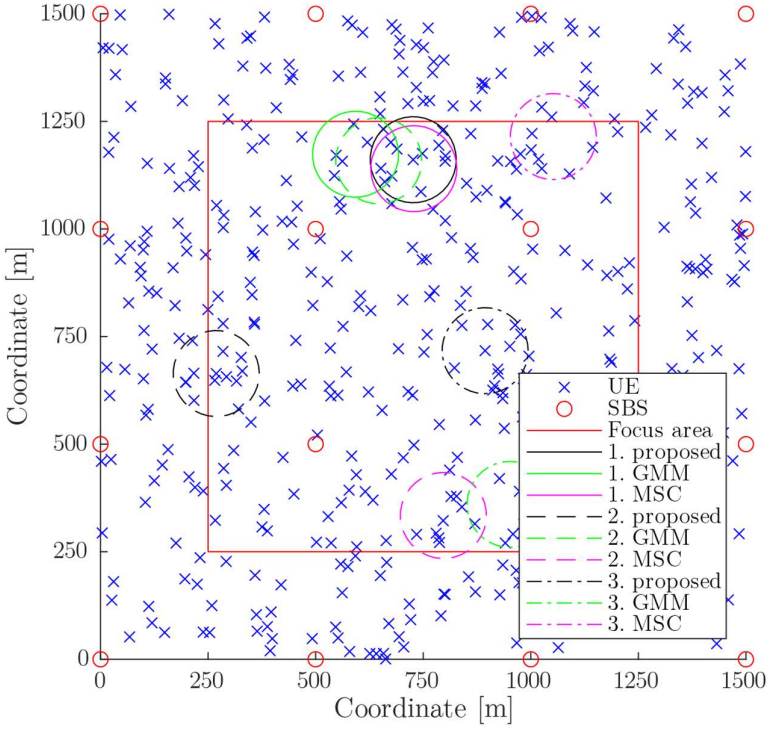

Figure 7.3 Locations of FlyBSs based on different clustering algorithms, scenario 1.

[1] Both proposed and MSC algorithm reflects the SBS positions though RSRP reports.

Table 7.1 Comparison of clustering algorithms, scenario 1.

	1 FlyBS utilised		2 FlyBSs utilised		3 FlyBSs utilised	
	Overall average throughput	**Served UEs**	**Overall average throughput**	**Served UEs**	**Overall average throughput**	**Served UEs**
Proposed Algorithm	highest	22	highest	18	highest	14
GMM algorithm	lowest	16	lowest	17	lowest	12
MSC algorithm	medium	21	medium	11	medium	11

If a group of 50 UEs close to each other is added (scenario 2), we achieve different results, as shown in Table 7.2 and Figure 7.4.

Again, the fact that the proposed and MSC algorithms consider the position of SBS sometimes prevents them from being close to the UEs conglomeration, as it was near the SBS. If at least part of the added group were fulfilling the signal quality condition, the FlyBS is deployed there no matter which algorithm was used.

The first FlyBS is placed in the same location no matter what clustering algorithm is used. More significant differences between the recommended FlyBS position are visible starting by the deployment of the second FlyBS. The GMM algorithm selects the same location as for the first cluster because the UEs are gathered there. The proposed and MSC algorithm choose different locations, as the location with the highest UEs density is already served by the first FlyBS. Even though the location for second and third FlyBS varies depending on the used algorithm, the FlyBSs serve a similar number of UEs.

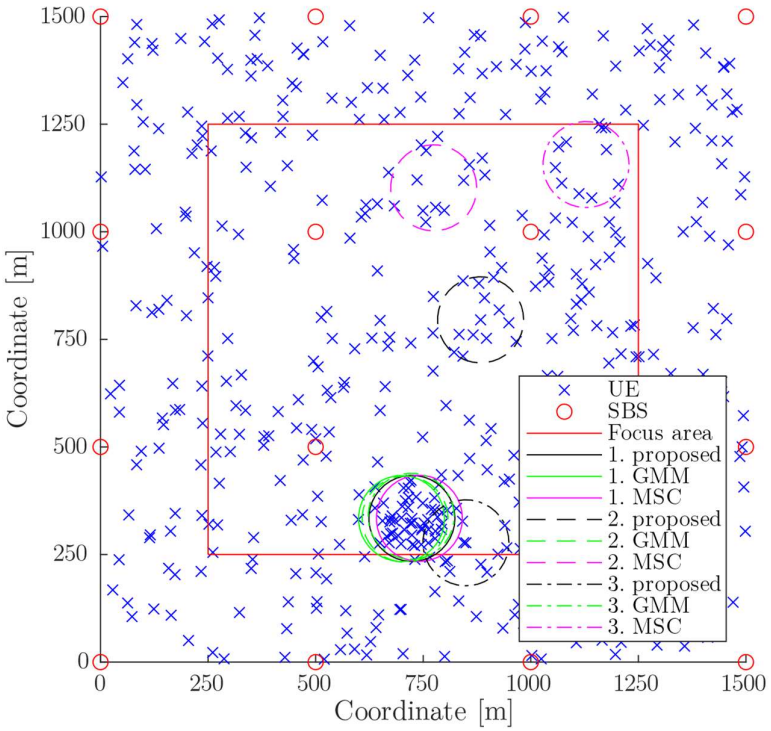

Figure 7.4 Locations of FlyBSs based on different clustering algorithms, scenario 2.

Table 7.2 Comparison of clustering algorithms, scenario 2.

	1 FlyBS utilised		2 FlyBSs utilised		3 FlyBSs utilised	
	Overall average throughput	**Served UEs**	**Overall average throughput**	**Served UEs**	**Overall average throughput**	**Served UEs**
Proposed Algorithm	similar	61	highest	16	highest	17
GMM algorithm	similar	61	lowest	42	lowest	38
MSC algorithm	similar	61	medium	14	medium	14

The simulation proves that if more FlyBS are deployed, their mutual optimization is necessary and iterative approach is not the best approach to select a suitable location. Also, the current deployment of SBS has to be taken into account, and the area where the FlyBS can be deployed evaluated as a whole.

8 Final positioning

Once the prediction, position estimation, and clustering algorithms are done, the final step is to select one suitable cluster, which will be served in the next moment. The optimal cluster has a center within reach of the FlyBS and the highest number of covered UEs. In the process of selecting the suitable cluster constraint given by the FlyBS speed is considered. The FlyBS speed defines maximal distance that the FlyBS is capable of overcoming between two clustering intervals. This condition is expressed in Equation 3.8b. For this purpose, we propose a simple selection algorithm, which evaluates the feasibility of the cluster center and the number of covered UEs.

The sequence of selected cluster centers in time presents the path for the FlyBS. It is desired to find the cluster in the way that the path is uninterrupted by any hops longer than the drone can overcome within a given time period.

8.1 Algorithm description

The proposed algorithm works as follows: from the whole set of possible clusters, received from the clustering algorithm, the one with the highest number of UEs is selected. Then the condition whether this cluster is reachable is tested. If yes, this cluster is selected as suitable, and the algorithm ends. However, if the cluster is unreachable, the number of UEs in the largest cluster is decreased and stored for further comparison. Then, all clusters with at least the same number of UEs as in the largest one are tested. The process repeats until the solution is found, or none of the clusters is suitable. In the case that no solution is found, the distance constraint is relaxed, and the largest cluster returned.

Algorithm 5.1 Suitable cluster selection.

```
input:      Possible cluster centers Centers,
            number of UEs within each cluster UEs,
            maximal overcome distance Dist.

output:     Next position for the FlyBS Position.

variables:  Number of UEs in the cluster Maximum.

Maximum = max(UEs)
While true
    for each i in Centres
        if ||Prev – Centres(i)|| < Dist & UEs(i) >= Maximum
            Position = Centres(i)
            return Position
        end if
    end for
```

```
        Maximum = Maximum - 1
        if Maximum == 0
                Maximum = max(UEs)
                Position = Centres(Maximum)
        end if
end while
```

8.2 Evaluation

As introduced in Section 6.2, the best estimation of the position is achieved when three closest BSs are considered. In the simulation, we use the reports from the SBSs. No reports from FlyBSs are used, as it may reach the maximal RSRP value due to the proximity to the UE. With a given path loss model (expressed in Equation 3.5), distances between UE and BS below 44 meters are reported as the highest RSRP level, as the signal is very strong.

The proposed selection algorithm (considering both maximal drone speed and the number of UEs in the cluster) is tested against two more straightforward ways of selection: first considering only maximal speed, and second considering only maximal number of UEs in the cluster. The overall simulation assesses 2 400 samples – the time thus equals circa to ten minutes, if we receive the four measurements every second. The simulation length corresponds roughly to the time today's drones are capable of staying in the air. The UE distribution and other simulation parameters are similar as in the simulation in Section 6.4.

Each selection method evaluates two metrics: number of discontinuities in FlyBS's path (indicating in how many cases the FlyBS is not able to find a suitable cluster), and the relative increase in the average throughput per UE compared to the situation without deployed FlyBS. We set the initial drone speed for this simulation ten times higher than the average speed of the UEs. However, the impact of maximal speed is discussed in detail later.

To show the importance of the position estimation, outputs from the clustering algorithm using the precise UE positions are compared with the outputs from the clustering based on the predicted and estimated values, as introduced in Chapter 6. The results are shown in Table 8.1 for the situation when we select the reachable cluster with the maximal number of UEs, a cluster with a maximal number of UEs, and a cluster considering only its availability.

The results confirm that the throughput increase is highest if the only reflected factor is the size of a cluster. However, this also brings a high number of discontinuities in the FlyBS's path, especially in the case when the inputs are predicted and estimated positions.

Table 8.1 Comparison of different criteria for cluster selection on predicted values.

		Reachable and maximal	Only maximal	Only reachable
Discontinuities [-]	**Predicted**	44	381	44
	Real	9	48	8
Throughput increase [%]	**Predicted**	4.33	6.33	4.34
	Real	5.95	6.35	5.99

When the drone speed is set to 10 times the speed of UE, the results are similar as if there was no speed limitation at all. A set of simulations with different maximal drone speeds are conducted to examine this deeper. The results are summarized in Table 8.2 both for predicted and estimated values and real values. We can observe the difference between results based on which set of UE position is used. When real positions are used, the drone speed can be lower to achieve a similar number of discontinuities in its path compared to the predicted and estimated positions. Also, if the drone is not capable of shifting quickly enough, it causes overall throughput degradation. In that situation, it is not favorable to deploy the FlyBS.

Table 8.2 Impact of maximal drone speed.

Drone speed/average UE speed		**1**	**2**	**4**	**8**
Discontinuities [-]	**Predicted**	760	760	760	53
	Real	-27.95	-27.95	-27.95	3.94
Throughput increase [%]	**Predicted**	828	226	9	9
	Real	-30.72	-3.77	5.95	5.95

The snapshot of the FlyBS's position is shown in Figure 8.1 and Figure 8.2. These figures show the determined cluster and the FlyBS path for two cases. In the first case (Figure 8.1), we use known UE positions. In the second case (Figure 8.2), we use the predicted and estimated UE positions 12 samples ahead. Both figures show the same moment, it is thus possible to also compare the real and predicted UEs position, which does not vary significantly. We also see that the FlyBS path is more efficient when the exact positions are known. However, both paths are passing roughly the same positions in the focus area.

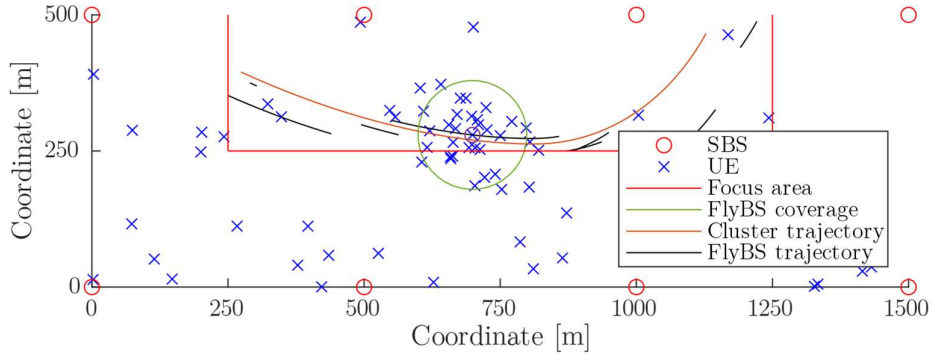

Figure 8.1 FlyBS trajectory for real UE positions.

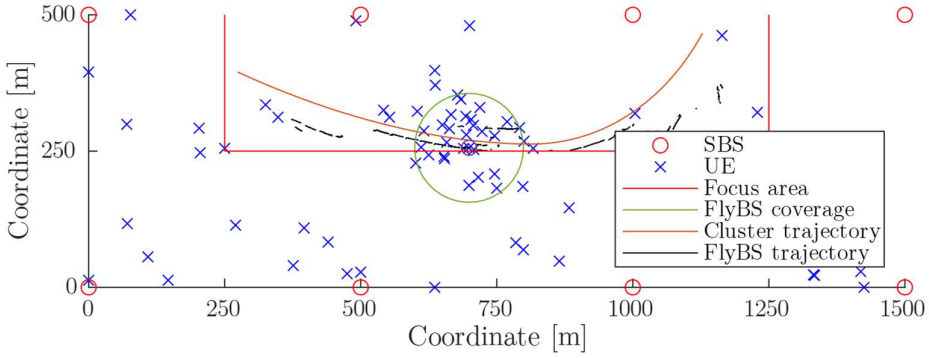

Figure 8.2 FlyBS trajectory for predicted and estimated UE positions.

9 Implementation

The proposed solution for the determination of the FlyBS is designed to require information that can be found in the real mobile networks. Therefore, it allows implementation into the existing mobile networks. In this thesis, it is assumed that the 4G mobile network is used, i.e., LTE. Due to the constraints of the real mobile network, an emulated mobile network is used for implementation. The emulated mobile network provides the same functionalities as the real hardware, but with significantly lower price and ease of modifications. The implementation into the emulated system allows us to prepare a variety of test scenarios to evaluate whether the algorithm operates in the same manner as during the simulations in MATLAB.

The project utilizing a UAV (more precisely a hexacopter) as a FlyBS has been under development in the 5gmobile research lab at the Department of Telecommunication Engineering of Faculty of Electrical Engineering for several years [32]. Currently, the start and landing of hexacopter is operated by a human operator. However, the FlyBS is capable to position itself to the selected GSP coordinates that are determined in this thesis. This project can be extended to include the proposed positioning algorithm.

9.1 OpenAirInterface

In the FlyBS setup at the Department of Telecommunication Engineering, OpenAirInterface (OAI) platform [33] is exploited for emulation of the 4G mobile network. The OAI platform implements 3GPP standards for radio access network (RAN) as well as core network (CN). This platform also enables emulation of the UE. The benefit of the OAI is its ability to run in both simulation and emulation mode. The simulation option significantly eases the process of development and debugging due to the ability to set the whole scenario exactly as it is needed and guarantee reproducibility.

The OAI platform runs on general purpose processors and supports software-defined radio cards, such as Universal Software Radio Peripheral (USRP). The whole OAI platform consists of software written in a mix of C, C++, and Python. Due to being a completely software solution, it allows for Network Function Virtualization of all elements – the whole emulated 4G network is possible to run on a single Linux-based machine. Because of that, OAI software is ideal for proof-of-concepts, testing, or prototyping.

Since the platform is open to the public and easily accessible, it has many users and contributors to the OAI repositories. This enables the platform to evolve continuously. Main features from LTE standard are already included, and nowadays, the developers implement NR [34].

The OAI is developed within OpenAirInterface Software Alliance (OSA). The OSA is a French non-profit organization, established by Eurecom in 2014. The purpose of the

alliance is to provide tools and open-source software for 4G and 5G wireless research. The global reach of the OAI platform is shown by its strategic members of the alliance consisting of companies such as Orange, Qualcomm, Fujitsu, or Nokia Bell labs. One of the community members is also the Czech Technical University in Prague.

For purposes of positioning algorithm implementation, the main focus is on RAN part of the network. As the OAI implementation is in line with the 3GPP standards, we can identify separate layers of the LTE control plane stack [35] in the source code. The stack for the UE and the BS, in LTE called eNodeB, consists of Physical Layer (PHY), Medium Access Layer (MAC), Radio Link Control (RLC), Packet Data Convergence Control (PDCP) and Radio Resource Control (RRC). UE also uses Non-Access Stratum (NAS) protocols to exchange control plane messages with Mobility Management Entity (MME). The LTE control plane stack schema is drawn in Figure 9.1.

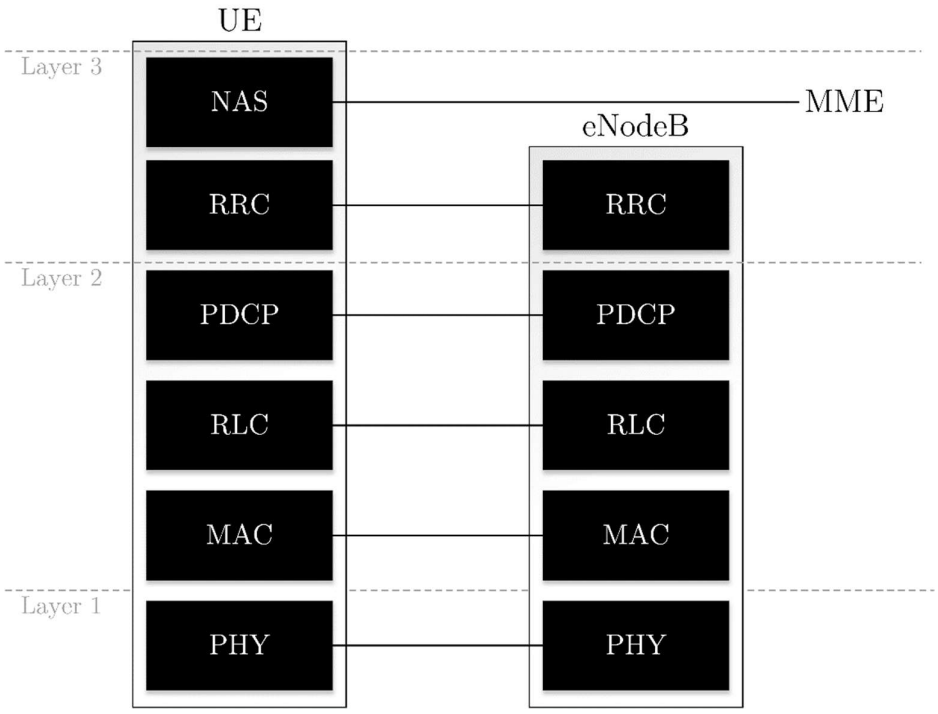

Figure 9.1 LTE stack for UE.

OAI uses an internal messaging client called ITTI to communicate between layers of the protocol stack (represented by the separate threads). ITTI also manages the live cycle of threads.

In 2016, the OAI platform incorporated FlexRAN [36] – software-defined RAN platform. With the use of FlexRAN, it is possible to separate data and control planes of the eNodeB. The platform consists of Master Controller (representing Control Plane) and several FlexRAN Agents (one for each eNodeB). The Agent can interact with other Agents or act

as a simple local executor of Master Controller instructions. Configurable Real-time capable south-bound API created by FlexRAN presents an ideal approach to receive the necessary information for the positioning algorithm from the network. FlexRAN agent has access to information from all layers from Figure 9.1, where the PHY and MAC layers are further splitted into the lower and upper part.

FlexRAN Agent collects the RSRP reports at the RRC layer by calling function `flexran_get_rrc_pcell_rsrp` and `flexran_get_rrc_neigh_rsrp`. FlexRAN is also able to use ITTI to interact with another task. This allows us to gain also parameters, which are not included in FleRAN platform code but are present in OAI. The messages transferred over the ITTI can be captured and examined using the ITTI Analyzer tool. The analyzer comes together with the source-code of OAI RAN.

9.2 O-RAN

The transformation of legacy RAN into open, virtualized, and intelligent RAN is the goal of O-RAN Alliance [37]. O-RAN introduces the concept of standardized interfaces within the RAN to allow easier network virtualization and interoperation.

O-RAN alliance was founded in 2018 by five large telecommunication operators (AT&T, China Mobile, Deutsche Telekom, NTT DOCOMO, and Orange). Nowadays, O-RAN connects over 160 leading companies ranking from hardware vendors through network operators to academia institutions.

O-RAN architecture [38] splits the functionalities of BS into the Radio Unit (RU), which includes Radio-frequency (RF) elements and lower PHY layer, Distributed Unit (DU), taking care of higher PHY, MAC, and RLC layers, and Central Unit (CU), covering RRC and PDCP layers. The architecture is further complemented by the Near real-time RAN Intelligent Controller (near-RT RIC) – an entity that manages, for example, radio connection, mobility, Quality of Service, or interference. Near-RT RIC can also host 3rd party applications. The last architecture block is the Orchestration and Automation platform, e.g., ONAP. The simplified O-RAN architecture is presented in Figure 9.2, where also some of the defined interfaces are shown.

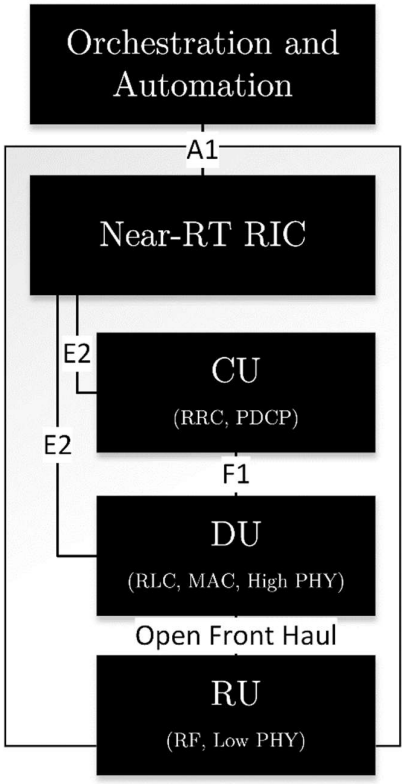

Figure 9.2 O-RAN architecture.

From the perspective of this thesis, the open E2 interface connecting the near-RT RIC and CU, DU, and eNodeB is interesting. E2 should be understood more like a platform rather than a simple interface, as its possibilities are vast. The proposed algorithm, together with other necessary UAV management components, can be then implemented as a xApp (microservice running at the Near-RT RIC).

As the impact of O-RAN is growing, the proposed O-RAN architecture will likely be available on the market in the future. That would simplify the incorporation of FlyBSs into mobile networks and speed the process up.

10 Conclusion

Deployment of a UAV, such as a drone, with communication hardware of BS (denoted as FlyBS) presents a possible way how to offload overloaded mobile cells, e.g., in case of mass events. The FlyBSs, in comparison to the SBSs, are inherently mobile and can change their location during operation. This thesis aims at proposing an algorithm to determine FlyBS placement based on the information from the mobile network and its possible implementation.

The positioning algorithm for the FlyBS is split into three subtasks. The first subtask consists of the estimation of UE distances to the BSs based on the signal quality reports from the UEs. In this phase, the discrete signal quality reports are interpolated to obtain distance as a continuous function of the UE movement.

The second subtask estimates the UE position based on the distances from the BSs. The algorithm searches for a location with a minimal sum of square errors of distances. Simulations show that the best accuracy is achieved with the necessary minimum of three closest BSs. This is due to the relation between the RSRP measurement levels and distances, which provide the best resolution for short distances.

The last subtask creates several clusters of UEs, based on their estimated positions. The clusters' positions determine possible areas to be served by the FlyBS. The proposed solution introduces a signal quality condition, which limits the set of possible clusters' locations. The position which allows serving the highest number of UEs is then selected. The proposed clustering algorithm achieves similar or better results in means of throughput and number of served UEs than compared Gaussian Mixture Model and Mean Shift clustering algorithm.

Determination of the possible clusters is not the final step, as the constraints of the FlyBS must be taken into account. Therefore, an algorithm to select a suitable cluster from the set of possible clusters is proposed. The proposed solution works by finding a cluster, which covers the highest number of UEs and is reachable at the same time. From the performance evaluation, it is shown that the overall network throughput is highest if the largest cluster is preferred. However, this leads to a high number of discontinuities[2] in the FlyBS path.

The overall efficiency of FlyBS repositioning strongly depends on the precision of UE position estimation, as shown in comparison to the case where the real UE positions are known. When tested on the set of real UE positions, the FlyBS experiences fewer discontinuities and increased the network throughput even with low maximal drone speed. On the contrary, when the estimated UE positions are used, the drone path is inconsistent, and the drone has to be able to move faster in order to increase the network throughput.

[2] Situation, when the UAV is not able to reach the desired cluster center in time due to its hardware limitations.

In the last part of the thesis, a possible implementation into mobile network is proposed. The implementation is based on the OAI platform, which enables us to run an emulated mobile network with USRP. FlexRAN platform within OAI is exploited to obtain the necessary mobile network information. FlexRAN provides access to BS to collect the required network measurements, mainly RSRP, for the positioning of the FlyBS.

In this thesis, we have proposed a positioning algorithm for the FlyBS positioning based only on the real mobile network data and designed its implementation into the mobile network. The problem of FlyBS positioning is a highly researched topic and the proposed solution goes beyond the state of the art solution by avoiding the necessity of additional measurements in the mobile network.

The proposed positioning algorithm can be in the future enhanced by several functionalities, which are described as the future work. The prediction of signal power level struggles when the distance between UE and BS is high due to insufficient accuracy of RSRP reports. Therefore, a Timing Advance [35] besides the RSRP reports can be exploited to improve the algorithm accuracy. The clustering algorithm evaluates all UEs fulfilling the signal quality condition. The Nearest neighboring (NN) search algorithm [39] or similar can be used to reduce the number of evaluated clusters when they are close to each other. The proposed positioning algorithm can be extended to reflect the number of UEs satisfying the signal quality condition in the cluster or evaluate the number of UEs negatively influenced by the FlyBS. Furthermore, the proposed algorithm for positioning of the FlyBS can be further extended via transmission power control of the FlyBS. Finally, to evaluate its functionality in the real world, it should be implemented to the existing experimental setup and tested.

11 References

[1] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. Giordano, A. Garcia-Rodriguez and J. Yuan, "Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges", *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3417-3442, 2019.

[2] M. Zuckerberg, "The technology behind Aquila", 2016. [Online]. Available: https://www.facebook.com/notes/mark-zuckerberg/the-technology-behind-aquila/10153916136506634/. [Accessed: 2020-04-13].

[3] L. Nagpal and K. Samdani, "Project Loon: Innovating the connectivity worldwide", *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 1778-1784, 2017.

[4] M. Shabanighazikelayeh and E. Koyuncu, "Outage-Optimized Deployment of UAVs", *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1-6, 2019.

[5] M. Abdel-Malek, A. Ibrahim and M. Mokhtar, "Optimum UAV positioning for better coverage-connectivity tradeoff", *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1-5, 2017.

[6] A. Fotouhi, M. Ding and M. Hassan, "Dynamic Base Station Repositioning to Improve Performance of Drone Small Cells", *2016 IEEE Globecom Workshops (GC Wkshps)*, pp. 1-6, 2016.

[7] V. Sharma, M. Bennis and R. Kumar, "UAV-Assisted Heterogeneous Networks for Capacity Enhancement", *IEEE Communications Letters*, vol. 20, no. 6, pp. 1207-1210, 2016.

[8] C. Yu and Z. Wang, "UAV path planning using GSO-DE algorithm", *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*, pp. 1-4, 2013.

[9] Q. Song, F. Zheng and S. Jin, "Multiple UAVs Enabled Data Offloading for Cellular Hotspots", *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6, 2019.

[10] "ETSI TS 125 215 V15.0.0 (2018-07): Universal Mobile Telecommunications System (UMTS); Physical layer; Measurements (FDD)". Sophia Antipolis Cedex, France, 2018.

[11] "ETSI TS 125 123 V10.2.0 (2011-07): Universal Mobile Telecommunications System (UMTS); Requirements for support ofradio resource management (TDD)". Sophia Antipolis Cedex, France, 2011.

[12] "ETSI TS 138 133 V15.6.0 (2019-07): 5G; NR; Requirements for support of radio resource management". Sophia Antipolis Cedex, France, 2019.

[13] "ETSI TS 136 214 V14.3.0 (2017-10): LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements". Sophia Antipolis Cedex, France, 2017.

[14] "3GPP TS 36.133 V16.5.0 (2020-03): 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management (Release 16)". Sophia Antipolis Valbonne, France, 2020.

[15] "ETSI TS 136 331 V15.9.0 (2020-04): LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management". 3GPP TS 36.331 version 15.9.0 Release 15, Sophia Antipolis Cedex, France, 2010.

[16] J. Plachy, Z. Becvar, P. Mach, R. Marik and M. Vondra, "Joint Positioning of Flying Base Stations and Association of Users: Evolutionary-Based Approach", *IEEE Access*, vol. 7, pp. 11454-11463, 2019.

[17] A. Al-Hourani, S. Kandeepan and S. Lardner, "Optimal LAP Altitude for Maximum Coverage", *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569-572, 2014.

[18] M. Hata, "Empirical formula for propagation loss in land mobile radio services", *IEEE Transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317-325, 1980.

[19] S. Zhang, Y. Zeng and R. Zhang, "Cellular-Enabled UAV Communication: Trajectory Optimization under Connectivity Constraint", *2018 IEEE International Conference on Communications (ICC)*, pp. 1-6, 2018.

[20] S. Roth, A. Kariminezhad and A. Sezgin, "Base-Stations Up in the Air: Multi-UAV Trajectory Control for Min-Rate Maximization in Uplink C-RAN", *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1-6, 2019.

[21] F. Fritsch and R. Carlson, "Monotone Piecewise Cubic Interpolation", *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. 238-246, 1980.

[22] N. Mesbahi and H. Dahmouni, "Delay and jitter analysis in LTE networks", *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 122-126, 2016.

[23] T. Diallo, A. Pizzinat, P. Chanclou, F. Saliou, F. Deletre and C. Aupetit-Berthelemot, "Jitter impact on mobile fronthaul links", *Optical Fiber Communication Conference*, pp. W2A.41-, 2014.

[24] R. Kress, *Numerical Analysis*. New York, United States: Springer, 1998.

[25] K. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. John Wiley and Sons, 1988.

[26] I. Schoenberg, "Contributions to the problem of approximation of equidistant data by analytic functions. Part A. On the problem of smoothing or graduation. A first class of analytic approximation formulae", *Quarterly of Applied Mathematics*, vol. 4, no. 1, pp. 45-99, 1946.

[27] C. Bishop, *Pattern recognition and machine learning*. New York, United States: Springer, 2006.

[28] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition", *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32-40, 1975.

[29] Q. Zhang, W. Saad, M. Bennis, X. Lu, M. Debbah and W. Zuo, "Predictive Deployment of UAV Base Stations in Wireless Networks: Machine Learning Meets Contract Theory", *arXiv preprint arXiv:1811.01149*, pp. 1-30, 2018.

[30] . Yizong Cheng, "Mean shift, mode seeking, and clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790-799.

[31] K. Krishna and M. Narasimha Murty, "Genetic K-means algorithm", *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 3, pp. 433-439.

[32] M. Stroot, B. Becks, D. Algarra and M. Fontanilla, "UAV-aided Wireless Networks", unpublished. Praha, 2018.

[33] N. Nikaein, M. Marina, S. Manickam, A. Dawson, R. Knopp and C. Bonnet, "OpenAirInterface: A Flexible Platform for 5G Research", *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33-38, 2014.

[34] F. Kaltenberger, G. Souza and R. Knopp, "The OpenAirInterface 5G New Radio Implementation: Current Status and Roadmap", *EURECOM*, pp. 1-5, 2019.

[35] E. Dahlman, S. Parkvall and J. Sköld, *4G LTE/LTE-advanced for mobile broadband*. Burlington, Mass.: Academic Press, an imprint of Elsevier, 2011.

[36] X. Foukas, N. Nikaein, M. Kassem, K. Kontovasilis and M. Marina, "FlexRAN: A flexible and programmable platform for software-defined radio access networks", *International on Conference on emerging Networking Experiments and Technologies* 2016.

[37] S. Abeta, T. Kawahara, A. Umesh and R. Matsukawa, "O-RAN Alliance Standardization Trends", *O-RAN Alliance Standardization Trends NTT DOCOMO Technical Journal*, vol. 21, no. 1, pp. 1-8, 2019.

[38] "O-RAN Alliance", 2020. [Online]. Available: https://www.o-ran.org/. [Accessed: 2020-05-19].

[39] D. Knuth, *The art of computer programming*. NJ, United States: Addison-Wesley, 2011.

12 List of figures

13 List of tables

14 List of abbreviations

3G	Third Generation Network
3GPP	3rd Generation Partnership Project
4G	Fourth Generation Network
5G	Fifth Generation Network
BS	Base station
CN	Core Network
E-M	Expectation - Maximization
ETSI	European Telecommunications Standards Institute
FIFO	First in first out
FlyBS	Flying Base Station
FSL	Free Space Losses
GMM	Gaussian Mixture Model
GSM	Global System for Mobile Communications
IEEE	Institute of Electrical and Electronics Engineers
ITU	International Telecommunication Union
KDE	Kernel Density Estimation
LTE	Long Term Evolution
MAC	Medium Access Layer
MME	Mobility Management Entity
MS	Mean Shift
MSC	Mean Shift Clustering
NAS	Non-Access Stratum
Near-RT RIC	Near Real-Time RAN Intelligent Controller
NFV	Network Function Virtualization
NN	Nearest Neighbor
NR	New Radio
OAI	OpenAirInterface
OFDM	Orthogonal Frequency Division Multiplexing
OSA	OpenAirInterface Software Alliance
PDCP	Packet Data Convergence Control
PHY	Physical Layer
Pchip	Piecewise Cubic Hermite Interpolating Polynomial
RAN	Radio Access Network
RF	Radio-frequency
RLC	Radio Link Control
RRC	Radio Resource Control
RSCP	Received Signal Code Power
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SBS	Static BS
SINR	Signal to Noise and Interference Ratio
SNR	Signal to Noise Ratio
SS-RSRP	Synchronization Signal RSRP
UAV	Unmanned Aerial Vehicle
UE	User Equipment
USRP	Universal Software Radio Peripheral