4G/5G cellular networks metrology and management

Imane Oussakel

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES

Délivré par :
l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Présentée et soutenue le *17/07/2020* **par :**

IMANE OUSSAKEL

4G/5G Cellular Networks Metrology and Management.

JURY

ISABELLE GUÉRIN-LASSOUS	Professeure des Universités	Rapporteure
VÉRONIQUE VÉQUE	Professeure des Universités	Rapporteure
DAVID COUDERT	Directeur de Recherche	Examinateur
LAURENT HOUSSIN	Maître de Conférences	Membre du jury
PASCAL BERTHOU	Maître de Conférences	Co-directreur de thèse
PHILIPPE OWEZARSKI	Directeur de Recherche	Directeur de thèse
BRUNO COUDOIN	Ingénieur	Invité

École doctorale et spécialité :
ED : MITT – Mathématiques, Informatique et Télécommunications de Toulouse, Spécialité : Informatique et Télécommunications

Unité de Recherche :
LAAS-CNRS – Laboratoire d'Analyse et d'Architecture des Systèmes (UPR 8001)

Directeur(s) de Thèse :
Philippe OWEZARSKI et *Pascal BERTHOU*

Rapporteurs :
Isabelle GUÉRIN-LASSOUS et *Véronique VÉQUE*

Acknowledgments

Everything happens for a reason! And the reason behind my PhD was to explore new Horizons. This wasn't possible without my close association with many people, whether the ones I have met in the lab or outside in Toulouse or even abroad during conferences, seminars and training days. Even though words are never enough to express our deep feelings, they can at least expose our gratitude for making such a journey a special and unforgettable one.

First and foremost, i would like to thank the ones who gave me the chance to outpace my limits, the freedom to explore new research paths. To the ones who went far and beyond to support me, pushing me to keep a fighting spirit and who were generously sharing their wisdom, knowledge and assistance, to my supervisors and mentors: Philippe & Pascal. I feel proud and fortunate to be part of their institution, which has a mission to impart quality based education with ethics and values as its bedrock. I owe them lots of gratitude for showing me this way of research. I really value your existence in my life and I don't think you are getting rid of "les IMANERIES" by the end of this thesis.

A special thanks to Prof. Isabelle Guerin-Lassous and Prof. Veronique Veque who has accepted to review my thesis. I would like to thank Dr. David Coudert as well for examining my work and accepting to be the president of my jury.

My special appreciations to Dr. Laurent Houssin for accepting to be on my jury and for introducing me to the optimization field of research. Chapter 4 wouldn't have been accomplished without his collaboration.

Even though every Phd student is left on their own, researching and debugging their problems, the ups and downs are common experiences, especially the chasm. Being surrounded by the ones who can relate is so helpful. Many joys have been shared and many obstacles have been overcome in the corridors, cafeteria and canteen. I would like to thank my coworkers for these special memories. First and foremost, my heartfelt appreciation to the ones who turn the office atmosphere productive and sometimes just unsupportable. To "au secours" who has supported my changing mood and who finally after 3 years and just days before the quarantine has learned how to open the office door smoothly without startling me. To the Wednesday peacock for all the "blblblbl A-ha-ha" moments and closed window to avoid any disaster. To the invisible one (Etienne) who knocks our door once a month to say a mature "Hi". To the ones who motivated me during the first months: Gilles & Rémy. Thank you all, you have made my three years special in this office. I would also like to thank my teammates: Tanissia, for her outstanding patience when I'm telling her my foolish moments during her objective function optimization and for all the weekends spent working at LAAS reminding me "Our life has become the Phd life". Nicola, for all the moments teasing each other. Drs. Tik & Tac for approving my model "Phd behind curtains". Josue for unintentionally pushing me to drink tea without sugar. Soufiane for all the 2 minutes questions that turn to be 2 hours discussion. Marine for always offering coffee breaks. Nico, for his positive energy. Ezekiel, for reminding me to keep my smile during the downs. Inès, for

trying to initiate me to the football world. Tom, for always being obliging. Santi, for letting me discover the Maté. Laurent, for his inspirational and enthusiastic mindset. Benoit & Fadel, for the late lunches and loud laughters we have shared together. Yovi for the delicious greek pastries. Nga & Han for their peaceful talks. Raoua for all the networking TP classes that we have monitored together. Nour for missing the flight, I hope it is the first and last time for both of us. Overall, I would like to thank all SARA members who were there for me when needed: Either by motivating, supporting, providing advice or just making up my day with a smile or a joke.

I would like to thank the MINC team members for welcoming me in the characterization room. For always asking about the state of my 4G private network, and motivating me to continue debugging the Openairinterface until it worked finally. I have really appreciated your support and the ambiance we have shared.

A special regards to Marie-Agnès, Justine, Belle Urica and Amandine who helped me in all the administrative procedures and arranged all my travels. I would also like to thank Jean-Michel, Julien and Isabelle from Sysadmin, for always dealing with the material configurations change during the 4G testbed deployment.

I warmly thank INSA-Toulouse for welcoming me within the GEI department as teaching assistant and letting me develop a good teaching pedagogy. I would also like to thank Christophe Chassot and Slim Abdelatif for generously mentoring me during this teaching period.

My special regards to my teachers through the years. Thanks to their teaching at different stages of education has made it possible for me to discover my passion for wireless networks. Thanks to their kindness and guidance I feel I was able to reach a stage where I could write this thesis.

Similarly to Frodo Baggins and his fellowship, my journey wasn't without friends and companions and they played a big role in reaching my destination: Mariam (bestie) for always being by my side, Noura & Younes who always answered my requests when I asked, Aymen & Arij for all the adventures that we have and will share, Khaoula for the midnight philosophical discussions, Assya, Firdaous, Meryem, Mohammed & Mourad for always asking me "how's your research going ?" even if they knew they should not ask such a question to a Phd student, but were always there supporting me. Latifa for her emotional support. Fati & Lamia & Youssra for all our discussions looking for a reason to get motivated again.

I owe my deepest gratitude towards my family members for their constant inspiration and encouragement. Despite the distance and their limited knowledge about my research field, they were always supporting me and pushing me to persevere until accomplishing this thesis. Their infallible love and support have always been my strength.

I shouldn't forget about some of the most important people in my life: My siblings. To my big brother, the Tom to my Jerry, the Sylvester to my Tweety, my guardian angel and protector, to the one who made my childhood full of great moments: Mouad. To the beauty and the kindness, to the pearl and the diamond, to the physically identical but individually different, to my precious sisters: The

twins Khadija & Rachida! Thanks to you, I've lived awesome times of joy and laughter and unforgettable memories. My love for you knows no limits.

Baba, Mama! There are no words in any language that exists or ever existed that can express how grateful I am for you. You are the light at the end of the tunnel, the moon in the darkness of night, the sun in my day, the stars who guide my way: You are my everything! You were there when I first cried, you pulled me up when I first fell, you stayed up during the feverish nights and kept me warm during the cold, you sacrificed everything without hesitation to keep us comfortable and happy. Everything I am and will be, everything I have reached and all I have achieved is a proof that you were perfect parents and magnificent role models. Thank you, thank you and thank you. **I love you to the infinity and beyond and this work is also for you ...**

*– **It always seems impossible until it's done**–*
Nelson Mandela

Contents

Introduction

In the scope of smart cities, where everything is connected at anytime and anywhere, various sectors are evolving, e.g. automotive, homes and healthcare. It is envisioned to turn each machine connected and generate useful data. As the concerned sectors are different, various performances are required for the range of devices. Notably, some connected devices will need high throughput, while ultra low latency might be the critical performance metric for other machines.

Wireless communication is in fact considered as the cornerstone for this vision realization. Accordingly, existing technologies are enhanced and new ones are developed to support this diverse performance demands. Particularly, the short-range technologies such as Bluetooth [Ferro 2005] and Zigbee [G. 2015] widely used since 1996 and 2004 respectively are not adapted for energy sensitive devices communications. For that, the Low-Power Wide Area Networks (LPWAN) are designed to support these devices requirements, such as the low data-rate, low energy consumption, low cost and long range. This devices' connectivity is labeled Internet of Things (IoT). Its leading technologies include LoRaWAN, Sigfox and NB-IoT [Mekki 2019]. They target a long battery life up to 10-15 years. LoRaWAN and Sigfox use the unlicensed ISM (Industrial, Scientific and Medical) frequency band, i.e. 915 MHz in the Americas and 868 MHz within the European Union. As to minimize the battery energy consumption, the connected devices send their data at a rate up to 100 bps with Sigfox and 50 Kbps with LoRaWAN. Contrary to Sigfox that limits the number of device generated messages per day, with LoRaWAN it is controlled by a duty cycle. Further, the NB-IoT (Narrow-band IoT) is developed and standardized by 3GPP (3rd Generation Partnership Project) since 2016. It makes use of the 4G licensed frequency band. It has higher power consumption and lower coverage compared to LoRaWAN and Sigfox. Nevertheless, the NB-IoT is simple to deploy, especially within the existing 4G networks with only a software upgrade in the operator's base stations. The IoT devices data-rate can reach 200 Kbps on a narrow bandwidth of 200 kHz.

Although the IoT devices generate small packets at a low frequency, the widespread adoption of multiple devices by each consumer and the IoT applications/services progression produce a huge amount of data. It is expected that IoT devices generate up to 847 Zeta Bytes of data by 2021[1]. Thus, LoRaWAN, Sigfox and NB-IoT face the challenge to support this traffic explosion mainly concentrated at the upload. With NB-IoT, it is up to the cellular network operators 4G and upcoming 5G to cope with this upload traffic growth.

Out of the IoT box, numerous machines are forecasted to be connected wirelessly such as vehicles and medical robots. The Intelligent Transport System - ITS - emphasizes many services/applications relying on car communication system. The

[1]Cisco Global Cloud Index: Forecast and Methodology, 2016–2021 White Paper. Last access 01/2020 `Paper`

main V2X (Vehicle to everything) applications are classified into four ITS classes: active road safety, cooperative traffic efficiency, cooperative local services and global internet services. Many car companies have been engaged in a research effort to tackle the upcoming ITS issues. Particularly, Continental Digital Service France (CDSF) and LAAS-CNRS have launched a collaboration in the framework of the eHorizon project (2017-2021) to cope with ITS systems[2]. In such systems, the vehicle is connected to pedestrians (V2P), other vehicles (V2V) and the ITS servers (V2I: vehicle to infrastructure) via wireless networks to increase driving safety and efficiency. Two technologies are proposed in literature to sustain the car communication: 4G-LTE (Long Term Evolution) and the DSRC (dedicated short-range communication). The DSRC [Li 2012] is an 802.11p-based wireless communication. An on board unit is inserted on the connected car to communicate with RSUs (Roadside units) installed along the road. It is therefore an ad-hoc based communications. Whereas, the 4G is a cellular network standardized by 3GPP and offered by mobile operators. The DSRC is highly effective for cars safety applications, such as collision avoidance. While the 4G-LTE has demonstrated its high effectiveness for non safety applications, notably the traffic information transmission [Xu 2017]. It could be therefore seen as two complementary technologies to enable the cooperative-ITS. Nevertheless, the cellular networks are evolving rapidly over the years. For instance, the D2D (device-to-device) LTE interface has been adapted for V2V communications [Nshimiyimana 2016]. In addition, the 5G system is strongly growing as to support the ultra-reliable transmissions. Thus, cellular networks might serve most of the ITS applications in the future.

Most of the envisioned sophisticated car applications send data (i.e. data uploading, traffic monitoring and vehicular video conferencing) to remote internet servers through the wireless interface. The generated data by car sensors can reach 3 Gbits/s and in some scenarios 40 Gbits/s. Research efforts are ongoing to reduce/compress the car transmitted data to servers. Yet, that procedures might not be possible in other areas such as healthcare which expect high performance from 5G to achieve their objectives. In fact, various applications and services are designed with different requirements. They range from the patient applications for remote monitoring to remote procedures such as the complex surgical procedures in real-time. These procedures are critical. They are conducted using high definition image streaming. It therefore demands a very low-latency, high reliability and high throughput for both uplink and downlink communication.

With all the third parties interest toward cellular networks, one should not forget the conventional usage of such networks, i.e. subscriber data download via smartphone/tablets. Regarding this case, changes have already been remarked during the last years. Ericsson has conducted cellular traffic analysis during the FIFA 2014 football games in Brazil. It showed that social media and texting were highly used during matches. Interestingly, the ratio of uplink traffic in the total data traffic was about 50%. It is important to notice that normal ratio in the

[2]Continental Automotive. Last access 03/2020 `www.continental-automotive.com`

same location is between 12% and 17%. In fact, during the final world cup match, only 25 % used the internet to download or look for content related to the match, while 62% of subscribers have posted/sent pictures via the network. In addition, the same study has indicated that 33% of users have posted videos and only 16% watched them. Such statistics report network traffic usage in crowded scenarios. Nevertheless, the number of smartphone users keep increasing over the years. By only the end of 2024, Ericsson forecasts around 1.5 billion 5G subscriptions for enhanced mobile broadband and 4.1 billion for cellular IoT connections [3]. This is promoted by the sophisticated applications on smartphones and tablets as well as their evolving capabilities, e.g. high camera definition (4K) and virtual reality (VR). Cisco has categorized this trend of applications and services such as the VR and Ultra HD video streaming within the ones highly uploading data as much as downloading it. The rapid growth of applications usage and smartphone penetration result in the subscribers traffic growth at steady rate for download as well as upload.

With all the smart cities fields, e.g. ITS, IoT, healthcare and mobile broadband, data resident in the client devices is now moving to data centers through mainly the 4G and 5G wireless interface. It is forecasted that the total data center capacity storage will reach 2.6 zeta bytes by 2021 compared to only 663 eta bytes in 2016 [Heinrich 2017]. Thus, contrarily to the Human-to-human applications where downlink transmission dominates, the upcoming applications/services mostly upload data. This emphasizes the importance to focus on the uplink transmissions monitoring as well as the resources usage to support this increasing various demand over the cellular networks.

The wireless channel in cellular networks make the transmissions monitoring challenging. It is characterized by the coexistence of multiple unpredictable radio phenomena. Such channel is complex to characterize and therefore it is hard to understand its impact on data transmissions. The most critical KPI (Key Performance Indicator) in LTE systems is the users average goodput, as LTE is to a large extent designed to maximize the system capacity for broadband data. It is nevertheless a vital objective for 5G systems as well. Further, with the explosion of data upload, the network average goodput is also becoming a crucial element in these systems. Conventionally, the average user goodput (download) is maximized through an aggressive use of re-transmissions and sophisticated schedulers techniques implemented at the base station (BS), in addition to other procedures at the BS physical and MAC (Medium Access Control) layers. The objective behind such techniques is to overcome the channel impairments. They incorporate the channel parameters or KPIs estimations [Richter 2005]. The KPIs might include throughput, and reliability (e.g. packet loss). Whereas, the radio channel parameters involve the channel impulse response, signal angle of arrival and so on. Particularly, the throughput estimation has allowed not only high rate when integrated within downlink schedulers, but also it is beneficial for increasing the video streaming application quality [Yin 2015] as well as network congestion control and

[3] Ericsson incorporated: Ericsson Mobility Report, November 2018. Last access 02/2020 `Report`

avoidance [Lu 2015]. It is then clear that such estimation would be also required for uplink transmission monitoring, as the upload speeds will continue to increase over time. With dissimilarities between the connected devices and BSs capabilities, the embedded techniques for uplink transmissions are different from the downlink. It therefore restricts the extension of the used approaches for downlink throughput estimation to the uplink estimation. For that, the users instantaneous uplink throughput estimation is considered as one of this thesis contributions.

Contrary to the downlink huge amount of traffic over the current deployed networks, the future traffic generated by the various applications/services of each area (e.g. automotive, healthcare, IoT..) in both uplink and downlink has diverging performance requirements. Within the same sector, applications/services needs are various. Notably, the high throughput is pivotal for vehicle infotainment, whereas, ultra low latency and high reliability are critical metrics for safety applications. These diverse requirements have to be sustained by the serving network, mainly the 5G. The architecture of 4G and previous cellular network generations are not adapted for such objective, as the network is considered a monolithic unit. Therefore, slicing the network to the various needs is seen as a key enabler to support such diverse QoS needs. With that, the operator opens up its infrastructure and resources to third parties (e.g. healthcare, automotive, YouTube). Each third party is considered as a slice or tenant. The operator should have the ability to manage each slice resources independently of the other slices. The resources include also the radio access network (RAN) resources, i.e. spectrum. Such perspective is constrained as these resources type are scarce and the performance depends on both the amount of allocated resources and the channel conditions. Regarding the radio phenomena, 5G comes with advanced techniques to mitigate/take advantage of the inter-cell interference based on tight cooperation between cells resources. Therefore, the Mobile Network Operator should have the ability to enable the appropriate techniques on each slice allocated resources. On the other hand, it should manage efficiently the radio resources to serve at best the slices with their demand distribution variation over time. The RAN slicing is in fact at exploration phase especially from a multi-BS perspective. Its enforcement is driven from multiple prospects. It is therefore considered in this thesis work, mainly from resource perspective in the context of multi-BS.

Accordingly, this thesis manuscript is structured as follows:

The first chapter introduces the 4G cellular networks. It includes the main wireless mechanisms supported by the network to mitigate the negative effect of the wireless channel conditions as well as to monitor the users QoS. As we focus on the users instantaneous throughput estimation to enhance the uplink monitoring, the existing work in this research field are reviewed. It is uncovered that most contributions have been interested mainly in downlink throughput estimation as only downlink was interesting. The relevant dissimilarities between the uplink and downlink transmissions limiting the extension of downlink approaches are then elaborated.

The second chapter deals with the developed estimation model for the instanta-

neous uplink throughput estimation. The supervised machine learning techniques (MLTs) are used for this aim. Therefore, the selected MLTs are reviewed with an exploration of their main application challenges. Also, the followed methodology for the estimation error generalization is explained. The global developed model for scalable estimation over time is then depicted.

As the work concerns the uplink bandwidth, we are therefore positioned at the network side instead of device side. Therefore, the accessibility to the network measurements is vital. Such metrics are under the confidentiality agreements of each operator. The third chapter then exhibits the deployed 4G real time testbed to collect the relevant metrics in order to feed the developed estimation model. Then, datasets building process is depicted with an analysis of the collected metrics importance for the estimation model. Further, an evaluation of the users instantaneous uplink throughput estimation is conducted with a discussion and open issues.

The fourth chapter expands to the 5G enforcement through the enabling of the RAN slicing. It introduces the 5G architecture and its main deployment challenges at the RAN level. It then deals with 5G RAN slicing enforcement from a resource perspective in a multi-BS multi-slice context. The slicing requirements at the RAN are summarized and their enforcement is formulated with two allocation strategies. Mathematical models are proposed to enforce the RAN slicing. Nonetheless, their real time deployment is limited. Thus, heuristics are proposed and evaluated based on the optimal or upper bound values given by the optimal models.

Finally, the last chapter summarizes the contributions of this thesis, pointing out their limitations and opens up to possible future work.

Chapter 1

4G Throughput monitoring

Contents

1.1 Introduction

The cellular networks are expected to embody the C-V2X (Cellular-Vehicle to everything), mainly because of their ability to address the different V2X categories, i.e. V2P, V2I, V2V, in end-to-end manner with the same technology. Moreover, with C-V2X, the vehicle can also access the network (V2N) to benefit from the commercial services (e.g. video streaming) as well as the safety related features. Both 4G and 5G are envisioned to support the V2X services/applications. With this, the connected car generates huge amount of data from its local sensors and live video images. It then needs high throughput and low latency connectivity to enable the exchange of either raw or processed data. For example, C-V2X could be used for the advanced driver assistance systems (ADAS), where cars can cooperate, coordinate and share information collected by sensors. Such system requires high throughput for both the upload and download. Thus, the throughput is considered as a crucial QoS metric for C-V2X. This can be added to the other services/applications highly uploading data as much as downloading it, such as video conferencing and Virtual Reality (VR) streaming. From that, the uplink and downlink throughput

monitoring is vital. Downlink has been already investigated in literature as the previous cellular network generations were conceived mainly for download. However, the upload is evolving more and more with these advanced applications/services. Therefore, we focus on the uplink throughput monitoring in cellular networks.

The chaotic radio phenomena characterizing the real environment leads to a degradation for the different QoS metrics, especially the throughput. Such phenomena include noise, multipath fading, and interference, etc. It is increasingly hard to understand how the throughput changes rapidly depending on the environmental situations. Thus, radio phenomena should be considered when monitoring the uplink throughput in such environments.

As researchers recently endorse the use of 4G standard for the short and middle term services/applications, the QoS monitoring in such networks is then covered in this chapter. The lower layers are the ones responsible for error correctness and treating any signal deformation due to the wireless channel conditions. They play a pivotal role in the QoS enforcement. Consequently, the major 4G lower layer mechanisms affecting the throughput QoS are explained. Subsequently, the importance of estimating the throughput is pointed out. And, to improve the uplink throughput monitoring in these networks, we propose to investigate its predictability by incorporating the rich lower layers information. Therefore, the remarkable proposed approaches for data-rate estimation are reviewed. Then, the main differences between uplink and downlink, limiting the pertinence of the downlink developed techniques in the uplink case, are discussed.

1.2 4G mechanisms and QoS monitoring

Several Mobile Network Operators (MNOs) are available in each country. In order to avoid interference between those MNOs, the frequency spectrum is subdivided over the MNOs, i.e. each MNO is regularized to operate over a specific frequency bandwidth. Hence, each MNO is facing the challenge to offer a good QoS to its users with a high spectrum efficiency regardless of the radio phenomena (multipath fading, shadowing, etc). For that, many mechanisms are carried in the system, such as the AMC (Adaptative Modulation and coding), CRC (Cyclic Redundancy Check) and the HARQ (Hybrid Automatic Repeat reQuest) [Villa 2012], as it is explained later in this section.

1.2.1 4G architecture and wireless mechanisms

Fig. 1.1 shows a basic architecture of the 4G network when a User Equipment (UE) is connected via the LTE access network to the Evolved Packet Core network (EPC). The EPC maintains the sessions, registration procedures and routing of UE IP-packets. The base station for 4G radio is named evolved NodeB (eNB); it ensures mainly radio resources management and scheduling in both uplink and downlink. Uplink (UL) refers to transmissions from the UE toward the eNB and Downlink

(DL) for the other direction (eNB toward UE). The wireless interface linking the UE with the eNB is called LTE air interface.

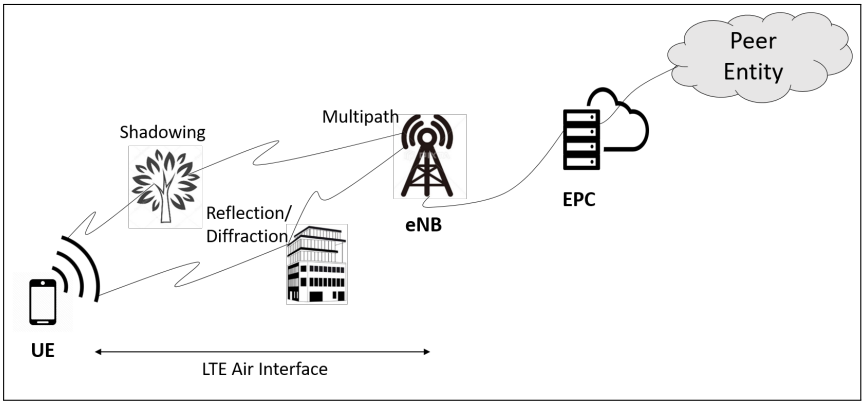


Figure 1.1 – 4G basic architecture.

Contrarily to the wired channel, the radio channel is characterized by its randomness. In fact, multiple radio phenomena can affect negatively the transmission over the LTE air interface, such as shadowing, multipath fading, noise, etc. It is clear that such radio issues will degrade the QoS metrics, e.g. data-rate. For that, the system incorporates many mechanisms in the lower layers of both UE and eNB to handle the radio channel variation. In the following, the main lower layers mechanisms characterizing the 4G system are briefed.

1.2.1.1 Adaptive Modulation and coding (AMC)

When a UE is connected to a 4G network and has data to be transmitted over the LTE air interface, the data is coded and modulated according to the radio channel condition. In fact, based on the measurements exchange between the eNB and the UE, reflecting the channel quality and the network capacity, an MCS (Modulation and Coding Scheme) is selected. The later refers to the modulation order Qm and the coding rate. The MCS is a crucial element for the UE data-rate determination.

Channel coding is introduced in the system to make data more robust against bad channel conditions. For that, the most used techniques are the ones where redundant bits are added, such as turbo coding [Heegard 1999]. Further, the coding rate indicates how much of data stream is actually being used to transmit useful data (non-redundant). For instance, a rate of 5/6 (83.3%) means that for every 5 bits of useful data, a total of 6 bits is generated with 1 redundant bit.

On the other hand, Qm is related to the modulation type and has a direct impact on the data-rate, e.g, QPSK (Quadrature Phase Shift Keying), 16 QAM (Quadrature Amplitude Modulation) or 64 QAM. The encoded user bits are separated into symbols with a specific length equal to the modulation order. For instance, when using a higher-order of QAM such as 64 QAM ($64=2^6$), i.e. Qm=6, each symbol will contain 6 bits.

1.2.1.2 Cyclic Redundancy Check (CRC)

In 4G system, the data coming from upper layers (or even the Medium Access Control (MAC) layer) toward the PHY layer is referred by Transport Block (TB). The transport block size (TBS) is determined by the selected MCS and the number of radio resources.

At the PHY layer, each TB is divided by a Cyclic Generator Polynomial to generate a 24 parity bits. The bits are appended at the end of the TB to form the CRC. This mechanism allows the receiver to detect errors introduced by the wireless channel. Further, each TB with CRC is segmented to multiple code blocks [Koopman 2004]. Each code block is then passed through the channel coding and other modules. The resulting code blocks are concatenated. Then, the modulation and other processing modules are performed [1].

1.2.1.3 Hybrid Automatic Repeat reQuest (HARQ) mechanism

When the channel conditions are bad, a high level of errors might be observable. Thus, the CRC and the coding scheme might be insufficient to reconstruct the transmitted data. Therefore, HARQ (Hybrid Automatic Repeat reQuest) is introduced in the communication system. It combines the retransmission mechanism ARQ (Automatic Repeat reQuest), as well as the error correction mechanism. The ARQ consists of requesting a retransmission when a TB is not well received. This is supported with an ACK/NACK exchange. For each transmission, the receiver sends either an ACK or NACK. ACK is sent for a successfully decoded TB and NACK when the receiver is unable to succeed the TB decoding.

In systems where only ARQ is performed, the receiver discards each TB not well decoded and requests its retransmission. Contrary, Hybrid ARQ (HARQ) adds memory in the system. That is, when a receiver is unable to successfully decode an erroneous TB, the latter is stored in a buffer and its retransmission is requested. Then, once the retransmission is completed, the receiver combines the recorded TB with its re-transmission for a successful decoding. It is possible that one retransmission is insufficient for a successful decoding, for that another retransmission might be carried depending on the HARQ configuration.

When a sender transmits a packet and waits for an ACK/NACK from the receiver, an HARQ process is then triggered.

We differentiate between synchronous and asynchronous HARQ processes. In asynchronous HARQ, the retransmission occurs any time after receiving the NACK. In this case, during the retransmission, the sender indicates the HARQ process details being used. In contrast, for the synchronous HARQ, the retransmission is scheduled at a specific time after receiving the NACK.

Overall, each of the aforementioned mechanisms is considered as a cornerstone for the wireless communications. They are functional as to mitigate the effects of

[1] 3GPP TR 36.212: Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding

the uncontrollable radio phenomena. Their impact on the QoS is prominent. They are part of the low level QoS monitoring. Further, an overview on the high level QoS monitoring in 4G networks is over-viewed in the following.

1.2.2 QoS monitoring

The QoS in cellular networks is handled in an end-to-end (E2E) manner. As illustrated on fig. 1.2, the e2e connectivity is established using an e2e bearer service, connecting the UE to a specific entity, over the 4G components (i.e. the EPC and the Radio Access Network (RAN) containing the eNBs).

The bearer profile includes the following QoS parameters: QoS class identifier (QCI), Allocation and Retention Priority (ARP), Guaranteed Bit Rate (GBR), and Maximum Bit Rate (MBR). To ensure a given QoS, the e2e bearer service is decomposed into small bearers. Each bearer links two major entities, for instance, the radio bearer is the one handling the communication between the UE and eNB over the LTE air interface. Such decomposition of the e2e bearer enables the QoS monitoring over each small bearer. Hence, the fulfillment of the e2e required QoS leverages on each part (bearer) achieved QoS. And, It has been recognized that a key determinant of QoS is the allocated resources. Keeping aside the EPC and cloud resources, that could be over-provisioned, the radio resources allocation at the eNB side still remains a challenge. Radio resources are scarce, and the number of UEs is increasing with a high throughput demand. Also, the amount of resources required to achieve a given data-rate depends on the wireless channel quality, e.g. when the channel conditions are poor, much more radio resources are prerequisite to achieve a data-rate. It implicates the necessity of taking the channel conditions into consideration when monitoring the throughput in cellular networks.

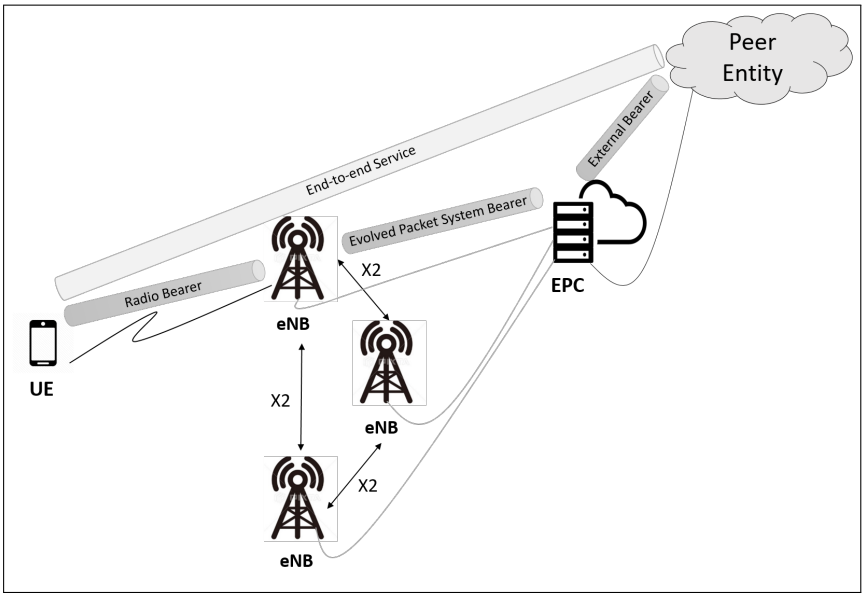


Figure 1.2 – End-to-End (E2E) QoS scheme in 4G network.

1.2.3 LTE/LTE-A performance evaluation

With the massive data that is going to be transmitted by the UE (machine, vehicle...) toward networks to reach the application servers, several researches show that LTE/LTE-A suffers from congestion [Luoto 2016, Trichias 2011, Phan 2011, Vinel 2012], especially in uplink. L. Trichias [Trichias 2011] argues that uplink is the 4G system bottleneck for intelligent transport based communications. And, on the same track, authors in [Vinel 2012] prove the limitation reached by the uplink channels for simple cars transmission scenarios. As a result, it degrades the QoS metrics. Therefore, a focus on the uplink transmissions is required as the usage case of the cellular network is changing for future applications.

Accordingly, we propose in this work to turn our focus on enhancing uplink transmissions through monitoring its QoS. Such objective might be realized either with a reactive or proactive systems. The former indicates the system reaction after receiving an event. For example, the network congestion control mechanisms are triggered only after detecting a QoS degradation due to the congestion. On the other hand, pro-active systems anticipate any positive or negative variation in the system and activate the appropriate procedures to avoid any QoS deterioration. It is clear that the QoS is maintained better in proactive systems compared to the reactive ones. Therefore, the uplink QoS monitoring in a proactive context is considered.

The key enabler of system pro-activeness is the anticipation of the QoS metrics evolution. And, throughput is considered as a crucial QoS metric. For that, the uplink throughput predictability is investigated. In the following, a state of art of the major throughput estimation approaches in literature is given.

1.3 Data-rate estimation: state of art

Over years, throughput estimation and prediction has been widely studied in wired networks and WLAN. D. Koutsonikolas et al. [Koutsonikolas 2011] reveal the ineffectiveness of using these wired techniques in cellular networks as they are characterized by the large short-scale fluctuation of bandwidth. Nevertheless, [Winstein 2013, Liu 2008] prove the predictability of throughput in cellular networks. Consequently, only the proposed techniques in that environment (i.e. cellular network) are discussed. They can be categorized into two categories, as shown below. The approaches estimating the data-rate based on higher layer metrics such as TCP metrics, and the ones considering the lower layer metrics (i.e. physical and MAC (Medium Access Control) layers) are over-viewed.

1.3.1 Estimation based on higher layer metrics

The higher layer metrics, starting from the network layer up to the application layer are used to estimate the throughput. For instance, Winstein et al. [Winstein 2013] propose the use of packet inter-arrival time to infer link bandwidth

and further determine the number of packets that can be transmitted. Q. Xu et al. [Xu 2013] estimate the throughput based on the historical throughput and the instantaneous sending rate using regression trees. The developed protocol monitors the network traffic passively. Hence, it limits the possibility of predicting the achievable throughput of the following time window at an arbitrary sending rate. It assumes also that the sending rate of the upcoming time window is similar to the previous one. Such presumption might be limited in network with high throughput variability.

The proposed techniques in these studies focus only on the higher layer metrics, neglecting the rich lower-layer information.

1.3.2 Estimation based on lower layer metrics

Considering the fact that data-rate performance is affected by radio phenomena, and lower layer metrics reflect the channel conditions, several studies have investigated the throughput estimation based on lower layer metrics for different use cases. For instance, to improve the TCP cross-layer congestion control mechanism in 3G networks, F.Lu et al. [Lu 2015] propose a prediction of downlink capacity based on CQI (Channel quality indicator) and DRX (Discontinuous Transmission). However, it uses a basic CQI-rate matching and the presence or not of DRX during a given time interval to predict the link capacity for the upcoming time interval. On the other hand, Margolies et al. [Margolies 2014] developed another version of the proportional fair scheduler, adapted for mobile users. The developed scheduler takes the predicted feasible data-rate as an input argument. Such prediction is based on the reproducibility of signal quality over the same path and user trajectory tracking. Simulations results are promising as the throughput has increased by 15%–55% compared to the traditional schedulers, while improving fairness.

On the other hand, with the emergence of the machine learning techniques and its good performance in different application domains, recent studies estimate the throughput with an underlying machine learning algorithm. It includes Support vector machine (SVM) and Random forest (RF). For instance, in order to minimize the energy consumption by cellular UEs, A.Chakraborty et al. [Chakraborty 2013] propose a protocol based on a specific LoadSense technique to increase the UE coordination efficiency for transmissions. The LoadSence approach is based on support vector machine classifier. It uses features such as the link quality, power ratio and its variation to estimate the availability of low or high throughput for the UE. The throughput was measured by downloading data from a server located in the same geographic area as the UE. In this study, it is considered that any throughput variation is mainly due to the wireless channel contention and not the operator wired part. In fact, this might be a misleading hypothesis as the authors had no control of the operators side.

Then, with the objective to help content provider choose the most appropriate representation (e.g. picture resolution, video resolution and rate) before the connection establishment, A. Samba et al. [Samba 2017] consider the throughput

estimation before establishing the connection. For that, they conduct a measurement campaign involving 60 users connected to a production network in France. They have proven that using either radio measurements on the UE side or the RAN measurements (e.g. average cell throughput, average number of connected users, BLER (Block Error Rate) of the cell) lead to good estimations. The application of random forest on UE radio measurements, RAN metrics and their combination result respectively in 50%, 59% and 65% of the relative prediction error within ±20% of errors. The relative prediction error of an estimation is computed by dividing the estimated value minus the actual value, by the actual value. Further, C Yue et al. [Yue 2017] developed a machine learning based framework to estimate the UE average throughput, named bandwidth. A server sends an UDP traffic to the UEs in different scenarios, i.e. stationary and mobility scenarios. Then an extensive measurements campaign is conducted in two commercial LTE networks in the US. As a result, five lower-layer measurements are identified as the most correlated with the downlink bandwidth. The measures include RSRP (Reference signal Receive Power), RSRQ (Reference Signal Receive Quality), CQI (Channel Quality Indicator), BLER, and the number of handover attempts. Based on the collected metrics and historical bandwidth, accurate predictions were obtained. For instance, for the walking scenario, 69% of the relative prediction errors are within ±10%. Overall, when using the machine learning techniques, mainly the average throughput is considered, as the work of [Samba 2017] estimated the instantaneous throughput before the connection establishment.

Although, both client and network throughput are important in cellular networks, link bandwidth estimation related work considered only downlink transmissions. The uplink and downlink have many dissimilarities from multiple perspectives. In the following, the major differences that might affect the uplink performance and its throughput estimation are covered.

1.4 Uplink vs Downlink

In DL transmissions, the user handles only its dedicated traffic coming from one eNB. In contrast, the eNB requires instead to support the traffic coming from all its connected users (UL case). Therefore, in terms of throughput estimation, in DL transmissions each user estimates its incoming throughput from one eNB at a time. In contrast, during UL transmissions, the eNB will need to estimate the incoming throughput of multiples users simultaneously. Thus, it makes the task more challenging for uplink communications.

Moreover, 4G handles the uplink and downlink differently. As one of the main concerns of this study is the estimation of the uplink throughput by incorporating the information from lower layers, an overview of the major differences between UL and DL in these layers is exhibited. It includes a comparison of the UL and DL access techniques, radio resources allocation and the radio measurements. This way, the possibility of extending the developed DL techniques for throughput estimation

based on lower layer metrics is also investigated.

1.4.1 Access technique

In cellular networks, the efficient utilization of the scarce frequency bandwidth by multiple users is primordial. For that, different access techniques have been developed such as TDMA (time division multiple access), where users use all the frequency bandwidth over different time slots, FDMA (Frequency division multiple access) that divides the frequency bandwidth into multiple frequency sub-channels and allocates each sub-channel to a separate user. On the other hand, when CDMA (Code division multiple access) is used, the users share the same frequency bandwidth by using different codes with spread spectrum technique. But, with the increase of connected users requiring high throughput, the 4G standard comes with another techniques to ensure the users access to the network, i.e. OFDMA in downlink and SC-FDMA in uplink. Orthogonal Frequency Division Multiple Access (OFDMA) is a multiple carrier system, where each symbol (consecutive modulated data bits) is transmitted over one sub-carrier. This transmission of multiple symbols in a parallel manner leads to a high PAPR (Peak to Average Power Ratio). PAPR causes a high-energy consumption for the transmitter. For uplink transmissions, the efficiency of power amplifier becomes crucial as the UE has a limited battery power. To avoid this OFDMA drawback in uplink, the 4G deploys the Single-Carrier Frequency Division Multiple Access (SC-FDMA), where symbols are transmitted in series and each symbol is carried by a wider bandwidth. Hence, contrarily to OFDMA that reduces/vanishes the inter-symbol interference (ISI), SC-FDMA is prone to ISI. ISI refers to the correlation between several symbols transmitted over different time instants, as illustrated on fig 1.3. Each symbol contains an amount of modulated bits as explained in paragraph 1.2.1.1. The UE transmits three symbols (symbol 1, symbol 2, symbol 3) in series to the eNB. Due to the objects in the radio channel between the UE and eNB, the multipath phenomenon is observed. Each path introduces a delay. Thus, it is possible that the eNB receives the three symbols at same time instant, i.e. the three symbols interfere. This issue is called Inter-Symbol Interference (ISI). The ISI increases the error rate in the system. It results in data-rate reduction when no compensation is present. For that the eNB deploys as a first step a complex frequency equalizer to mitigate such distortion. Therefore, UL data-symbols are not only affected differently by the channel variation compared to DL, but signal treatment is also different in the eNB.

1.4.2 Radio resources allocation

The efficient radio resources utilization is a challenging task for 4G QoS monitoring. Therefore, The 4G resources decomposition is illustrated in the following with an insight on the characteristics of the uplink radio resources allocation.

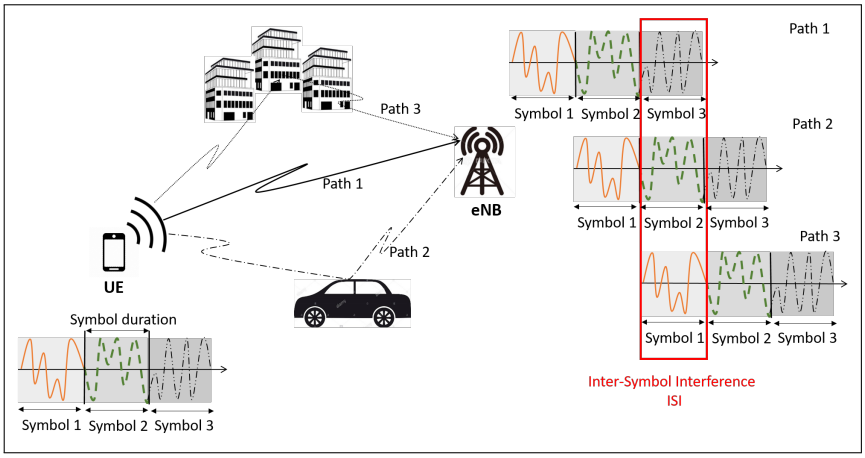


Figure 1.3 – Illustration of Uplink Inter-Symbol Interference (ISI).

1.4.2.1 Radio resources

Fig. 1.4 illustrates the frame structure for the 4G FDD (Frequency Division Duplex) mode, where different frequency bands are used for downlink and uplink. The frame length is fixed to 10 ms, containing 10 sub-frames of 1 ms. A resource block is defined over frequency and time domains by 0.5 ms over 12 subcarriers spaced by 15 kHz (i.e. 180 kHz). Then, data symbols are transmitted over one or several resource blocks. Each resource block holds seven SC-FDMA symbols. In the physical layer, the 4G radio resource is referred as Physical Resource Block (PRB) of size 180 kHz x 1 ms. In time domain, the PRB contains fourteen SC-FDMA symbols expanded over 12 subcarriers in frequency domain and the smallest block holding data is referred as Resource Element (RE).

Also, to handle the communication between the UE and eNB, the use of these PRBs is standardized using channels formats in the lower layers. For instance, for uplink transmissions at Physical layer level, the UE activates the PRACH, Physical Random Access Channel to demand access to the network; it communicates the control signaling information using the Physical Uplink Control Channel (PUCCH). PUCCH is mainly located in the bounds of the used bandwidth in frequency domain (see fig 1.4). The UE transmits its data on the Physical Uplink Shared Channel (PUSCH), which uses the remaining PRBs.

Table 1.1 – Available PRB in 4G for each frequency bandwidth

$Bandwidth(Mhz)$	1.4	3	5	10	15	20
Number of PRB	6	15	25	50	75	100

Based on the used frequency bandwidth, as shown on Table 1.1, a defined number of useful RBs is available. However, when using the aforementioned frame type, the frequency bandwidth is divided per 180 kHz (i.e. 12 subcarriers spaced by 15

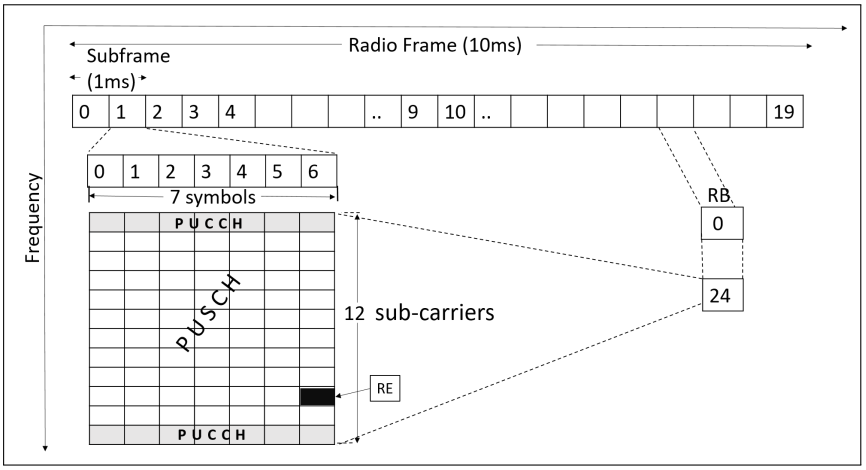


Figure 1.4 – Uplink radio resources in FDD mode.

kHz to construct the PRB). For example, for a 5 MHz frequency bandwidth, 25 PRB could be assigned to the users attached in the cell.

1.4.2.2 Radio resources allocation

As stated earlier, the radio resources utilisation is a crucial task in the QoS monitoring, due to their scarcity nature and the UEs increasing demand. Over years, researchers were focused on enhancing DL transmissions, as only DL was challenging, by improving or proposing new algorithms for an efficient flow scheduling. The utility function of the proposed schedulers aims mainly to reduce the downside of the chaotic wireless channel while considering the QoS metrics [Margolies 2014], [Bang 2008]. The scheduling task is performed by the eNB.

Further, once a UE/flow is scheduled, it is allocated an amount of RBs. Note that however in downlink, the allocation of RBs is made mainly per flow or bearer. But, in uplink, the radio resources allocation is instead made per UE. And, the RBs assignment to bearers is performed by the UE as an uplink control function.

Accordingly, to improve the services/applications uplink performance, an estimation of the UE throughput is primary, instead of service flow estimation. Several DL schedulers [Margolies 2014, Kushner 2004, Bang 2008] and congestion control mechanisms [Lu 2015] increase their efficiency when taking the estimated data-rate as input argument. For instance, F. Lu et al. [Lu 2015] come up with a cross-layer congestion control design taking into consideration the cellular link capacity estimation. And, the throughput over TCP has been boosted. Further, we forecast that similar results might be achieved when integrating the estimated uplink user data-rate into the uplink and congestion control algorithms in the future.

Meanwhile, the achieved throughput is conditioned by the scheduler output. Although schedulers are not specified by 3GPP, it is reported to consider one TTI (Transmission Time Interval) of 1 ms as the smallest scheduling time unit [Trivedi 2014]. Hence, for each TTI a new resources allocation scheme is proposed

by the eNB. In uplink LTE, to retain a low PAPR in sc-fdma, a contiguous resources allocation is designated. This constraint affects the network performance. In LTE-A, the non-contiguous allocation is allowed in a single Component Carrier (CC). It increases the uplink spectral efficiency while enabling the frequency-selective scheduling. But, the non-contiguous allocations directly increases the PAPR and other challenges as stated in [Abu-Ali 2014].

Moreover, the UL HARQ is synchronous in LTE while DL is asynchronous. Thus, contrary to the DL retransmission that occurs anytime, the UL retransmission always occurs eight subframes after the prior transmission attempt for the same HARQ process. This way, the transmissions of the HARQ process details are not sent to the eNB (receiver). This has the advantage of overhead avoidance. But as a result, in time domain, the uplink schedulers always prioritize users with pending re-transmissions independently of other users' priority and channel conditions.

Overall, the uplink is quite limited compared to the flexible design of downlink schedulers. Such challenges restrict the scheduler to reach the required data-rates by users experiencing different radio conditions. Thus, the data-rate variability at small time granularity for uplink might be different from the downlink.

1.4.3 Radio measurements

From section 1.3 some researchers are able to estimate accurately the DL throughput when incorporating the radio measurements at the UE side. Yet, we address the uplink transmissions and based on the 3GPP standardization of the physical layer measurements[2] on both UE and eNB, the most used metrics at the UE are unavailable at the eNB. The common used metrics for throughput estimation in both [Yue 2017, Samba 2017] include RSRP, RSRQ. The RSRP and RSRQ qualify the downlink channel quality. They are determined based on the power of the exchanged reference signals (RS). The later is predetermined by antenna configuration and carried on predefined REs in each slot in DL. Hence, the RS is known by both UE and eNB. It is qualified as a mandatory feature, as it allows the UEs to demodulate coherently their data and also estimate the channel.

On the other hand, the uplink transmission is characterized by two types of RS, the Demodulation Reference Signal (DM-RS) and SRS (Sounding Reference Signal). The DM-RS is time multiplexed with uplink data and transmitted on the fourth or third SC-FDMA symbol. It enables the coherent data demodulation at the eNB. SRS is transmitted over the last SC-FDMA symbol of 1 ms subframe and could be shared between users with different transmission bandwidth. Contrarily to DM-RS, the SRS allows the channel dependent uplink scheduling. But, the SRS is defined as an optional feature. Hence, the measurements based SRS could not be taken as input arguments in generalized systems. Moreover, the SRS results in about 7% uplink capacity reduction. Thus, contrarily to the available downlink radio measurements used for downlink throughput estimation, the uplink radio

[2]3GPP TR 36.214: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer Measurements

measurements based on SRS are not always available at eNB. Moreover, the uplink radio measurements are different from the downlink ones.

Therefore, the DL throughput estimation approaches based radio measurements proposed in [Yue 2017, Samba 2017], might not be extended to the uplink data-rate estimation as the core of the proposed techniques are based on unavailable radio measurements at eNB side.

1.5 Conclusion

This chapter sheds light on our motivations behind the instantaneous uplink throughput estimation in cellular networks. The 4G QoS monitoring with some major wireless mechanisms are briefed. They point the importance of taking advantage of the lower layer metrics related to the radio condition variations in the throughput monitoring. Given the QoS performance assurance in proactive systems, the throughput estimation is targeted and a state of art of the major approaches for throughput estimation techniques is over-viewed. Even though the forecasted need to monitor the uplink transmissions, research has not yet focused on the instantaneous throughput estimation for that link. Thus, the main dissimilarities between uplink and downlink restricting the portability of DL results to uplink are discussed.

This work proposes the UEs instantaneous 4G data-rate estimation as part of the uplink QoS monitoring. For that, the following chapter motivates the use of machine learning approaches as an underlying technique for such objective. Therefore, the machine learning techniques chosen for the estimation are explored. Then, the developed model for the instantaneous data-rate estimation at different time granularities is detailed.

Chapter 2

Estimation methodology and model

Contents

2.1 Introduction

The main objective of this work is the enforcement of an efficient monitoring system for cellular networks, from 4G up to the 5G. As stated in the previous chapter, one of the major upcoming challenges in cellular networks is the uplink transmissions monitoring. And for that, we propose to enforce the proactive systems through estimating the UE instantaneous uplink throughput.

The wireless link between the UE and eNB is in fact the major critical part in the transmission process, due to its randomness [Andersen 1995]. And, any error/deterioration caused by the channel conditions is treated in the lower layers, i.e. the eNB lower layers for uplink transmissions. Hence, we consider the importance of taking into account the impact of wireless channel in the throughput estimation. For that, the estimation involves the eNB lower layer metrics.

On the other hand, the data analytic techniques such as machine learning, are considered as strong approaches for estimation task. In fact, the machine learning algorithms have the ability to learn patterns from data and give accurate estimations. The estimation performance might be improved by the choice and amount of the introduced data. Therefore, for the UE instantaneous uplink throughput estimation based on the eNB lower layer metrics, a model with an underlying machine learning technique (MLT) is developed as detailed in this chapter.

The MLTs are categorized into unsupervised and supervised techniques. The former groups data in several clusters with similar characteristics. The later divides data into input and output sub-groups. Then, it builds a mapping function between the input and output features. With this, the supervised techniques aim to estimate the output feature from the input ones. In our case study, the objective is to estimate the instantaneous throughput for each user from its lower layers metrics. It is clear then that the supervised technique category is the suitable one for our objective, as the lower layer metrics could be considered for input features in the system and the instantaneous throughput as the output feature to estimate. Therefore, we introduce in this chapter the supervised learning techniques and their implementation challenges to reach accurate estimations. As many techniques are proposed in the literature, we depict then the chosen ones for our objective achievement. Three supervised learning techniques are selected, linear regression (LR), support vector regressor (SVR) and random forest (RF). These techniques are favored as to encounter the linear and non-linear relationships between the input and output features, i.e. the throughput and lower layer features. Once the estimations are performed with any machine learning technique, the generalization of the estimation error is vital. For that, the followed methodology for estimation error generalization is exhibited. Finally, the built model for the throughput estimation is highlighted. It is a scalable estimation model, as the throughput can be measured over different granularities depending on the application case.

2.2 Supervised machine learning approach

The main reason behind using machine learning algorithms is to estimate/predict each UE instantaneous uplink throughput in 4G networks. In uplink transmissions, UEs send their data to the network. Via several mechanisms, as discussed in the previous chapter, the eNB (receiver) handles the errors/deterioration caused by the radio channel to reconstruct the UE data. Therefore, several metrics related to those mechanisms are available at the eNB. From that, we propose an estimation of the uplink throughput incorporating those metrics, i.e. the eNB lower layer metrics. Thus, we suppose the access to the throughput measurements as well as the eNB lower layers metrics to apply the machine learning techniques. This way, the problem could be treated with a supervised learning approach, as explained next.

2.2.1 Regression Supervised learning

Supervised learning is considered as a type of machine learning technique where the learning process requires input and output variables. Let's consider one input variable, called also feature, $X = \{X_1, X_2, ..., X_p\}$ and an output variable $Y = \{Y_1, Y_2, ..., Y_p\}$, the machine learns a function (f) that maps each sample X_i of X to the output Y_i, $i \in \{1, ..., p\}$, i.e. $Y_i = f(X_i)$. In the case of this work, the X could be considered as one of the eNB lower layer metrics, and Y the throughput measurements.

Two phases are primordial in the MLT process, training and test phases. During the former phase i.e. training, the machine is fed with input as well as output variables. And, the objective behind is to reach an accurate approximation of the mapping function (f), to estimate accurately $Y^{train} = \{Y_1^{train}, Y_2^{train}, ..., Y_p^{train}\}$ from the input variable $X^{train} = \{X_1^{train}, X_{2^{train}}, ..., X_p^{train}\}$. It results then in a trained MLT on X^{train} and Y^{train}. Further, the trained MLT is tested on new data in the second phase, i.e. test phase. For that, new samples of the same input variable, $X^{test} = \{X_1^{test}, X_{2^{test}}, ..., X_n^{test}\}$, are given to the trained MLT in order to estimate $Y^{test} = \{Y_1^{test}, Y_2^{test}, ..., Y_n^{test}\}$. Then, the estimated $\hat{Y}^{test}$ are compared to the real Y^{test} values to evaluate the MLT accuracy.

The output variable Y is either discrete or continuous. When it is discrete or categorical, then the problem is addressed as a classification task, i.e. each new estimation is classified within the predefined classes. Otherwise, if the output variable Y is numerical or continuous, the problem is then handled with regression, as the variables have real values. In this work, we aim to estimate the instantaneous throughput, therefore we suppose continuous throughput measurements (Y). Hence, the objective is addressed as a regression learning task.

2.2.2 Machine learning performance

In order to define the MLT performance, the estimated $\hat{Y}$ values are compared to the real Y values. In the case of throughput estimation, a comparison of the estimated throughput is induced. Different statistical metrics are available to conduct such comparison. In this study, the RMSE (Root Mean Squared Error) is chosen. RMSE is attractive from a statistical and scientific perspective. It represents the average error prediction in the model, expressed in the units of the variable of interest. It is computed as follows: $RMSE = \sqrt{(1/n * \sum_i^n (y_i - \hat{y_i})^2)}$ where $\{y_1 ... y_n\}$ are the actual values and $\{\hat{y_1}, ... \hat{y_n}\}$ the estimated ones. By squaring the error, a high weight is given to the large errors. RMSE score is negatively oriented, hence lower values are better.

Further, the throughput estimations are evaluated based on RMSE, i.e. $Y \longleftrightarrow$Throughput.

2.2.3 Machine learning challenges

The MLT is trained on a given dataset, then tested on new samples. The ultimate objective is to reach good estimation performance on the new data as the obtained performance when training the model, i.e. low RMSE values in both training and test phases. Such objective is known by generalization. A good model is the one able to generalize the performance for new data in the same domain. This generalization comes with several challenges, namely the over-fitting and under-fitting problem, as well as the bias-variance trade-off, as explained in the following.

2.2.3.1 Bias-variance trade-off

The major challenge in the application of MLT is the bias-variance trade-off. The variance occurs when the model gives a different estimation value for the same output variable when trained with different datasets. In other words, Let's $(\alpha 1, \beta 1)$ and $(\alpha 2, \beta 2)$ be two datasets, with $\alpha 1 = \{\alpha 1_1, ..., \alpha 1_t\}$, $\beta 1 = \{\beta 1_1, ..., \beta 1_t\}$, $\alpha 2 = \{\alpha 2_1, ..., \alpha 2_t\}$, and $\beta 2 = \{\beta 2_1, ..., \beta 2_t\}$. Let's $y1$ and $y2$ be the estimation of the output variable y based on x using the model M when trained with $(\alpha 1, \beta 1)$ and $(\alpha 2, \beta 2)$ respectively. When $y1 \neq y2$, the model is considered with high variance. In fact, the model is sensitive to the training dataset. It is then unable to give good estimations on new data.

On the other hand, the bias is considered as the error caused by erroneous assumption of the model during the learning process. A perfect bias-variance trade-off allows a generalization of the error over new data. In other words, to have a good error generalization a low bias and low variance should be accomplished.

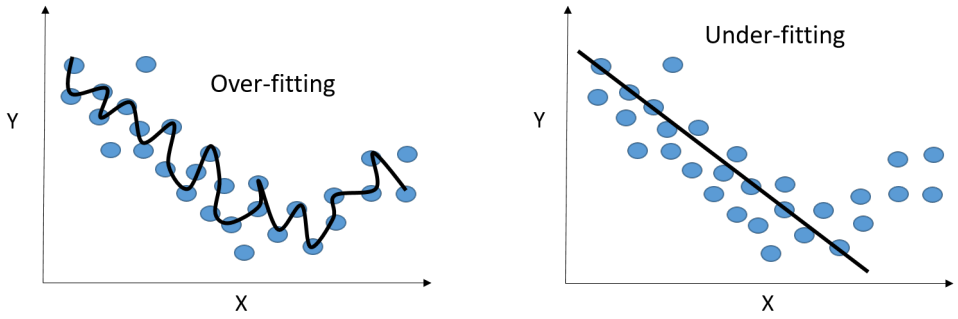
2.2.3.2 Over-fitting and under-fitting

In machine learning, the model ability to approximate the mapping function between input and output variables is known by fitting data. Over-fitting refers to the model fitting well the training data. It gives then poor estimations on unseen data. Fig 2.1 (a) illustrates this problem. It happens when the model learns so much details about the training data, including its noise. This issue occurs mainly with non-parametric and non-linear MLTs. Non-parametric technique refers to the MLT lacking any user parameters, i.e. parameters fixed by the user to control the model behavior.

In the opposite of over-fitting, there is under-fitting as shown on fig 2.1 (b). In this case, the model is unable to either fit the training data or generalize to the new data. Poor performances are observed already in the training phase. Thus, it implies that the MLT is unsuitable for the problem resolution.

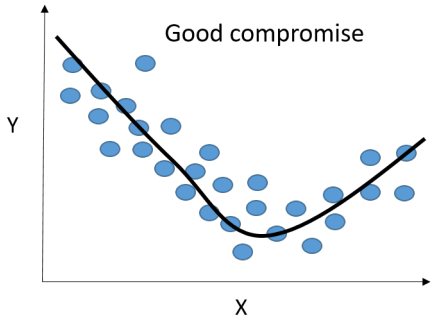
Therefore, the optimal model is the one with a good compromise between under-fitting and over-fitting, as depicted on fig 2.1 (c). The model learns the data pattern instead of learning all the details or less information.

Over-fitting and under-fitting are relied to the variance-bias trade-off as shown on fig 3.11. The under-fitting comes with a low variance and a high bias. Whereas,



(a) Model over-fitting the data

(b) Model under-fitting the data



(c) The optimal compromise for fitting the data

Figure 2.1 – Model fitting data: over-fitting vs under-fitting and optimal model

the over-fitted model is characterized by a high variance and low bias. In order to generalize the performance, the model should have a low variance and low bias. It corresponds to the encircled zone on fig 3.11, named optimal model.

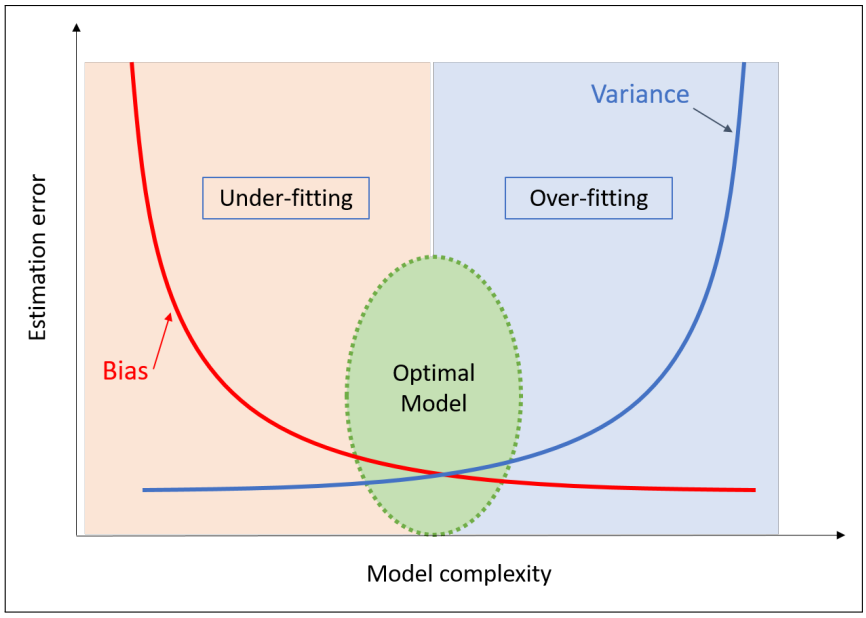


Figure 2.2 – Machine learning technique challenges.

2.2.4 Machine learning techniques (MLT)

The regression supervised machine-learning algorithms considered to estimate the UL data-rate based on the eNB lower layers metrics, as the most encountered in the literature, include the Linear Regressor (LR) [Weisberg 1980], Support Vector Machine [Smola 2004] and Random Forest (RF) [Liaw 2002]. The main reason of using LR is to investigate the linearity between the lower layers metrics and the high QoS metric, i.e. data-rate. In contrast, SVM is selected with non linear kernel as to exploit the non linear relationship between the eNB and the data-rate metrics. RF has the ability to create the linear and non linear boundaries during the trees building, which leads to accurate estimations. RF is then applied. Moreover, RF and SVR algorithms are known to lead to good estimations in different application domains with accurate time series [Wu 2007, Sapankevych 2009], and it is reported also to be insensitive to high dimensional feature spaces [Zekić-Sušac 2014]. In the following, the background of the three machine learning techniques is explained in the context of throughput estimation based on the eNB lower layers metrics.

2.2.4.1 Linear Regressor (LR)

Linear Regressor is considered as the simplest technique in literature. It looks for the statistical relationship between the variables. In other words, given $X =$

$\{X_1, ..., X_n\}$ and $Y = \{Y_1, ..., Y_n\}$ the input and output variables for the MLT respectively, LR looks to fit the variables X_i and Y_i into a linear relationship between them.

For the objective of this work, i.e. uplink throughput estimation based on eNB lower layer metrics, let's presume a UE is sending data to a given server through a 4G network. The eNB, as the first 4G reception component, performs the primary measurements on the received signals such as the signal received power, RX_power. Hence, RX_power might be considered as one of the input variables for LR algorithm, i.e. $RX\_power = \{RX\_power_1, ..., RX\_power_n\}$, with RX_power_i corresponds to the measured received power during the time interval i, $i \in \{1, .., n\}$. Let's denote the throughput measurements with $Thr = \{Thr_1, ..., Thr_n\}$, where Thr_i is the amount of data received during the time interval i, $i \in \{1, .., n\}$. Therefore, in the case of throughput estimation based only on the received power, during the LR training phase, LR looks for the linear relationship between the received power and throughput measurements, i.e. $Thr = f(RX\_power)$, f() being a linear function. If the model gives accurate estimations during this training phase, it means that the MLT was able to find a linear relationship between the RX_power and throughput. To generalize the performance, the trained model is tested on new values of RX_power to estimate its corresponding throughput values.

2.2.4.2 Support Vector Regressor (SVR)

SVR is based on geometrical margin to separate data. Given a dataset with input and output variables $X = \{X_1, ..., X_p\}$ and $Y = \{Y_1, ..., Y_p\}$, where $X \in \mathbf{X}$ and $Y \in \mathbf{R}$. SVR approach looks for a function $f(X)$ with maximal deviation inferior to ε for the estimation samples Y_i.

In the case of linear regression, f has the form indicated in equation 2.1, w denotes the weight vector and b is named the bias. In such case, the regression is performed in the same input space $\mathbf{X}$.

$$f(X) = < w, X > + b \quad with \quad w \in \mathbf{X} \quad and \quad b \in \mathbf{R} \tag{2.1}$$

$$< ., . >: \textit{scalar product}$$

On the other hand, for non-linear case, a naive method is used, where data is mapped from the input space $\mathbf{X}$ to a higher dimensional feature space $\mathbf{F}$ using a non-linear function ϕ, i.e. $\phi : \mathbf{X} \rightarrow \mathbf{F}$. Therefore, the function f is redefined as in equation 2.2.

$$f(X) = < w, \phi(X) > + b \quad with \quad w \in \mathbf{X} \quad and \quad b \in \mathbf{R} \tag{2.2}$$

$$< ., . >: \textit{scalar product}$$

In order to estimate w and b, the minimization of equation 2.4 is performed. Where C is an hyper-parameter fixed by the user, it concerns the trade-off between the

model uniformity ($||w||^2$) and the learning error.

$$L_e(Y_i, f(X_i, w) = \begin{cases} \|Y_i - f(X_i, w)\| - \varepsilon, & \text{if} \quad |Y_i - f(X_i, w)\| \geqslant \varepsilon \\ 0 \qquad\qquad\qquad\quad \text{otherwise} \end{cases} \tag{2.3}$$

$$R(f, C) = C \sum_{i=1}^{n} L_e(Y_i, f(X_i, w)) + 1/2||w||^2 \tag{2.4}$$

$< ., . >$: *scalar product*

In fact, the function L_e allows only the penalty of errors superior to ε. Further, it is proven that computing the non-linear features explicitly does not scale well with the number of input features. For that, kernel approaches are introduced as to avoid the step of explicitly mapping data to high dimensional space. This is possible by a Lagrange dual representation of the decision boundaries. Hence, the function f is redefined as in equation 2.5, with α_i are Lagrange coefficients.

$$f(X) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) < \phi(X_i), \phi(X) > + b \tag{2.5}$$

This way, it is unnecessary to find the function ϕ, but instead the value of $< \phi(X_i), \phi(X) >$. The scalar product function, $K(x, x') =< \phi(x), \phi(x') >$, is called a kernel. SVR is considered as a versatile method as it offers the possibility of using different kernel functions for decision-making. The kernel choice of the related SVR parameters impacts the output performance [Sánchez A 2003].

In the case of our objective of throughput estimation using the lower layer metrics/measurements, the input space could be multi-dimensional with the use of multiple metrics as input variables. The use of a kernel function is then required. The RBF (Radial basis Function) is applied and is defined as follows:

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2)$$

where K denotes the RBF kernel function and γ an hyper-parameter fixed by the user, and that impacts the model generalization capacity. The RBF objective function uses the Euclidean distance along its calculations. Thus, the estimator could be dominated by the variable having a high variance order. To avoid such problem, the RBF SVR objective function assumes that all the input features are centred around zero with same order of variance.

Further, given a dataset of eNB lower layer metrics and throughput measurements, the SVR is applied with an RBF kernel. The configuration of the related hyper-parameters (i.e. parameters fixed by the user) is then required. There is mainly three hyper-parameters, each one impacts differently the estimation performance:

- Epsilon ε: it defines the margin of tolerance, where no penalty is given to errors.

- Gamma γ: it is the kernel coefficient. Higher gamma values reflect the attempt of the model to fit exactly the training dataset. Thus, a high γ value might lead to a high accuracy but with a biased model.

- C: it is the Cost parameter. Large values of C indicates poor accuracy but low biased model.

The followed methodology to find the good compromise between the different hyper-parameters is detailed in section 2.3.1.

2.2.4.3 Random Forest (RF)

The random forest (RF) technique is considered as an ensembling learning model. The ensembling approach combines the estimations from multiple learning algorithms with the objective of increasing the model performance.

RF is based on building multiple trees, called Decision tree (DT). The DT models the problem in a tree structure, as illustrated on fig 2.3. from a root node the algorithm takes several decisions to reach the leaf nodes. Each leaf node is an estimation value. And, each internal node split refers to the taken decision.

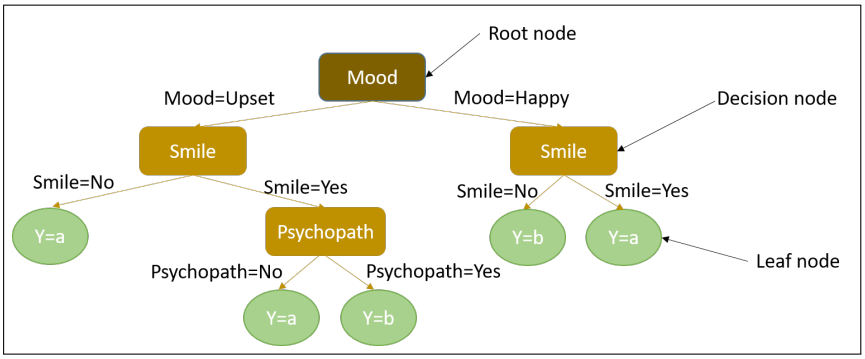


Figure 2.3 – Example of decision tree (DT).

The individual decision tree is sensitive to data and prone to over-fitting [Bramer 2013]. For that, RF combines at training phase many individual decision trees by adding randomness in order to select the training subset and the feature in each node, as it is outlined in the following algorithm steps:

1. Draw multiple trees with bootstrap sampling. The number of trees (n_estimators) is an hyper-parameter to fix by the user.

2. For each node split during the tree building process, use the best split among a subset of predictors randomly chosen, instead of choosing the best among all the predictors. At this step, the random chosen number of features (max_feature) controls the randomization strength in the feature selection process.

3. Estimate the new data by averaging the estimations of the built trees in case of regression.

Further, the forest created in the context of this work incorporates the lower layer metrics. Each tree root contains a random subset of metrics, and each internal node is a decision concerning those metrics values. The leaf nodes hold the throughput estimation. In order to estimate the throughput value of Thr_j during time interval j, the built trees estimate Thr_j, then the model averages their estimations to obtain the $\hat{Thr}_j$, i.e. throughput estimated value.

To be noted that the randomness introduced by RF gives the model a wide diversity and turning it robust against over-fitting, resulting in prediction improvement. Also, RF is known for its simplicity and for providing good results most of the time. It runs efficiently on large datasets and has the ability to handle thousands of input features without any feature deletion. In order to find the optimal forest that gives good performance, RF puts forward multiple hyper-parameters characterising the trees building.

- n_estimators: the number of trees built in the forest. Generally, higher number of trees gives better performance. But, it comes with more computational cost.
- max_features: the maximum number of features to select during the node splitting phase. Small value reflects the strong randomization in the feature selection process.
- max_depth: the maximum depth of the tree. It refers to the tree length from the root to the leaf node. Larger trees induce more information about the data structure, whereas small trees are mostly less precise.
- min_samples_split: it is the minimum required number of samples for the node split. With higher value, the trees are more constrained, as they should consider more samples at each node split.
- min_samples_leaf: the minimum required number of samples for a leaf node. It refers to the leaf size. With a small leaf size, the model is prone to capture noise in the data.

The estimation performance highly depends on these hyper-parameters values. The used approach to fix them is explained in the next section.

In summary, three algorithms are chosen for the uplink throughput estimation based on the eNB lower layer metrics. Each model incorporates the eNB metrics to estimate the throughput, i.e. throughput is carried as an output variable in the learning process. Then, the model performance is based on the comparison between the estimated and the real measured throughput metric. Good performance

would be reached only by a well chosen values for the hyper-parameters. Thus, the hyper-parameters configuration of each model is critical. Accordingly, the approach followed to fix the hyper-parameters is explained in the following. Further, once the estimations are performed based on a given set of measurements, the estimation error should be generalized on new data (i.e. unseen measurements). For that, the used technique to generalize the throughput estimation errors is explored next.

2.3 Estimation error

The aim of a MLT is the accurate estimations of a given variable. But, as explained in part 2.2.3, the use of MLT comes with different challenges as well as multiple hyper-parameters to setup to reach the optimal model. The optimal configuration of these hyper-parameters helps the model to fit data well. It then results in good performance, i.e. lower estimation error. For that, this section deals with the hyper-parameters tuning procedure and the generalization of the estimation error.

2.3.1 Hyper-parameters tuning

Each machine learning technique has various hyper-parameters to tune before the learning process, and fixing them is critical for the model' performance. For decades, a grid search [Bergstra 2011] has been the standard technique for hyper-parameters optimization. It consists of an exhaustive search over subset values of parameters for a given estimator. With that approach the optimal combination of hyper-parameters is definitely selected but it is criticized for its high computation complexity. In practice, the expert applied a manual refinement procedure. They started with a coarse grid of hyper-parameters space and then reduce iteratively the searching space of hyper-parameters. It is also a time consuming strategy. Over the years, several approaches have been proposed, including population-based search [Guo 2008], gradient search [Sato 2009] and Genetic based strategies [Di Francescomarino 2018]. Recently, it has been proven that random search [Bergstra 2012] performs better than grid search [Mantovani 2015]. It performs a random selection in the hyper-parameters search space. In fact, instead of looking for the combination exhaustively, it tries randomly different parameters from the hyper-parameters grid. The user of random search can setup the number of trials to find the optimized hyper-parameters. From that, we use the random search strategy to figure out the optimal hyper-parameters combination that produce the best estimation performance, i.e. lower estimation error.

2.3.2 Estimation error generalization: Cross-validation technique

The setup of hyper-parameters for a given MLTs forms a model. The estimation of a given model accuracy is vital to estimate its future performance (i.e. error generalization) and also to choose between different models. As explained in

2.2.3.2, we would like to have an estimation with low bias and low variance. The most used approaches to estimate a model accuracy on unseen data (i.e. data not used during the model training phase) includes cross-validation (CV) and bootstrap [Friedman 2001]. The bootstrap is a re-sampling method, i.e. sampling with replacement. But it is less reliable in most case studies compared to the K-fold CV [Kohavi 1995] as explained later. CV is categorized into exhaustive and non-exhaustive categories. The exhaustive strategies include the leave-one-out approach. Given a dataset with n samples, the leave-one-out method trains the model on (n-1) samples. It then tests its performance on the left sample. It is repeated n times. This method is almost unbiased, but it is shown to have a high variance. Thus, it produces unreliable estimations. Also, it is more computational consuming for high dimensional datasets. The non-exhaustive cross-validation techniques such as the K-fold CV overcome this downside. With the K-fold, the initial dataset is split into K folds, and K iterations are performed as illustrated on fig 2.4. On each round, 1 fold is taken as a test set and K-1 combined folds as training set. For instance, for the first iteration, fold 1 constitute the test set and the remaining folds (i.e. from fold 2 to fold k) are combined to form the training set. Hence, for each iteration, a data subset is considered as a test set (i.e. new data). Thus, the model performance is evaluated in each iteration. At the end, the model is tested over K-folds. this way all the samples in the initial dataset are tested. Then, the model performance is computed as the mean over the K test sets performance. That is, let's denote $RMSE_p$, the estimation error with the test set of fold p. And, let $RMSE_M$ be the estimation error of the model M after CV. It is computed as follows: $RMSE_M = 1/k \sum_{p=1}^{k} RMSE_p$

The value of K is crucial, as it might result in over/under estimation of the model's performance, with high variance or high bias. The 10-fold CV, i.e. K=10, is shown to produce a good compromise between variance and bias of the model [Friedman 2001]. Therefore, the K-fold CV is used in this work to generalize the models performance.

Further, in order to find the optimal hyper-parameters combination with a generalization of its estimation error, we propose the use of a combination between CV and random search. This technique is called RandomizedSearchCV [Geisser 1975]. It works as summarized in algorithm 2.1.

Algorithm 2.1 Random Search CV technique

Input: MLT, dataset.
Output: The error estimation $\overline{RMSE_M}$, best hyper-parameter configuration.

1: **for each** trial: select an hyper-parameters combination **do**
2: Do K-fold CV
3: Compute the model estimation error $RMSE_M$
4: **end for**
5: The hyper-parameter combination with minimum $RMSE_M$, noted $\overline{RMSE_M}$, is the best configuration.
6: The error estimation for model M is $\overline{RMSE}_M$

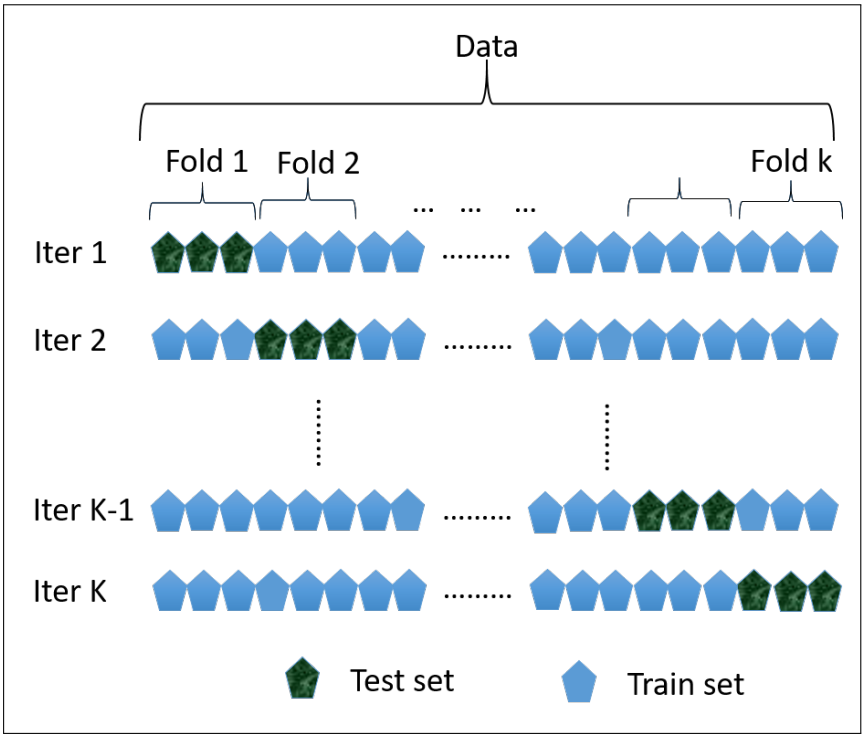


Figure 2.4 – Illustration of K-fold cross-validation technique.

With this approach, the selection of the optimal model is fulfilled (i.e. optimal hyper-parameters combination) and the estimation performance can be also concluded thanks to the embedded CV. Nevertheless, authors in [Varma 2006] have shown that such performance estimation are biased as the model selection and the error estimation are performed simultaneously. Accordingly, to generalize the estimation error, we use the nested k-fold CV as explained in the next paragraph.

2.3.3 Unbiased estimation error: nested k-fold

By applying the RandomizedSearchCV technique, the model is seeking the optimal estimator parameters on each iteration during the K-fold CV execution. Varma and Simon [Varma 2006] report that the estimated prediction error from the cross-validation used to tune hyper-parameters is biased, and recommend the use of nested cross-validation instead, where an inner CV is used to select the optimized model (executed with RandomizedSearchCV) and an outer CV to estimate the prediction error.

Algorithm 2.2 depicts the steps of the nested CV techniques. Let denote K_1-fold and K_2-fold the inner and the outer CV respectively. Given an input dataset, a split is performed to construct training and test sets. In fact, the dataset is split into K_2-folds, one fold is used for testing and the others K_2-1 folds constitute the training set. For each hyper-parameter combination from the random search, K_1-fold is applied on the training set. It divides the training set (K_2-1 folds) into K_1

equal folds; K_1-1 folds are used for training and the remaining fold for evaluation. It computes the prediction error and iterates until all the folds are used for both training and validation, then the estimation error is averaged over all the K_1 cases of CV. The hyper-parameter combination achieving a minimized prediction error is selected as the best optimized model, noted M_{k_1}. In order to generalize the selected model M_{k_1}, it is tested on unseen data, i.e. the test fold. This is done for the k_2 folds. Then, the true estimation error is the average of the estimated prediction error over the K_2 tested sets. It corresponds to the generalized estimation error.

Algorithm 2.2 Nested K-fold technique

Input: MLT, dataset, k_1, K_2.
Output: The generalized estimation error.

split dataset into K_2 folds.
for each iteration in K_2-fold CV **do**
  for each hyper-parameters combination **do**
    Do K_1-fold CV on the $K_2 - 1$ folds
    Compute the model estimation error $RMSE_{M_{k_1}}$
  end for
  The optimal hyper-parameter combination is the one with minimum $RMSE_{M_{k_1}}$, noted M_{k_1}.
  Test the model M_{k_1} on the test set.
  Compute the estimation error on test set, $RMSE_{k_2}$
end for
Average the error over K_2 tests.

2.4 Estimation parameters and model

The basic implementation of the aforementioned MLT in section 2.2 assumes an estimation of the output variable Y_i from the input variable X_i, $i \in \{1, .., p\}$. But, the objective is to estimate throughput over an upcoming time interval. And, the later depends on the application/service function needs. Hence, in order to have a flexible estimation model over different time granularities, we propose to add a system parameter, named forecast window. Also, to exploit the pattern created by the variation of the input variable values X_i, $i \in \{1, .., p\}$, lag window parameter is introduced as explained later.

2.4.1 Forecast window

From the related work 4.3.3, the main chosen estimation time interval is fixed to 1 s. Thus, it delimits the applications type that might employ the estimated throughput.

With that, we propose to have a scalable estimation model. Thus, the notion of forecast window is introduced. Let's denote β the forecast window and δt as the smallest estimation time interval, the relation between δt and β is expressed by $w = k * \delta t$, $k \in \mathbf{N}$. Thus, β refers to the largest estimation time interval. Hence, the estimation system can be modeled as follows: For each input variable X_i, we estimate the set $\{Y_i, Y_{i+1}, ..., Y_{i+k}\}$, where $k = \delta t/\beta$. The model is then flexible to estimate the throughput, up to a certain time interval size, for diverse services.

2.4.2 Lag window

The lag window parameter, denoted l, $l = j*\delta t$, $j \in \mathbf{N}$, refers to the largest time interval in the past. It is introduced as to test the impact of the past values of a metric on an estimation. In other words, given an input and output variable X and Y respectively, instead of an estimation of the sample Y_i based only on X_i, the model considers also the past values of X_i in the time interval l, i.e. $\{X_{i-l}, ..., X_{i-1}, X_i\}$. It is valuable in the case of this study, as the radio phenomena might have a low and large scale variation over time. Hence, for low scale variations, it is interesting to investigate the effect of the metrics past values on the instantaneous throughput estimations.

In summary, two parameters are integrated in the estimation model, lag size l and forecast window β. The later is to have flexibility in term of time interval data-rate forecast and the former takes advantage of any significant variable variation. Accordingly, for a given forecast window $\beta = k * \delta t$ and lag size $l = j * \delta t$, $k, j \in \mathbf{N}$, and a dataset $X = \{X_{m-j}, ..., X_{i-1}, X_{n+k}\}$, $Y = \{Y_{m-j}, ..., Y_{i-1}, Y_{n+k}\}$, the estimation model looks for the mapping function (f) between X and Y as follows:

$$\overleftarrow{\text{lag size}} \qquad \overrightarrow{\text{forecast window}}$$

$$\begin{pmatrix} X_{m-j} & \cdots & X_{m-1} & X_m \\ X_{m+1-j} & \cdots & X_m & X_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n-j} & \cdots & X_{n-1} & X_n \end{pmatrix} \xrightarrow{\text{(f)}} \begin{pmatrix} Y_m & Y_{m+1} & \cdots & Y_{m+k} \\ Y_{m+1} & Y_{m+2} & \cdots & Y_{m+1+k} \\ \vdots & \vdots & \ddots & \vdots \\ Y_n & Y_{n+1} & \cdots & Y_{n+k} \end{pmatrix}$$

2.4.3 Estimation model

For each estimator from the aforementioned techniques in section 2.2, the final implemented model to estimate the throughput per time granularity δt is summarized in algorithm 2.3.

For a given dataset of eNB lower layer and throughput metrics, we test the three MLT, i.e. RF, LR and SVR. Let's β and l be the forecast window and lag size respectively. For each MLT, the forecast and lag window (β, l) are fixed. Then, nested CV is applied as to generalize the estimation error and find the optimal hyper-parameters for the MLT. This way, the estimation error of each model is extracted, noted $RMSE_{l,w}^{MLT}$.

2.5 Conclusion

In this chapter, the estimation methodology for the users uplink throughput is detailed. It incorporates the supervised machine learning techniques (MLT). For that, the implementation requirements of such approaches is briefed. Then, the selected MLTs are depicted. They include the random forest (RF), linear regressor

Algorithm 2.3 Estimation model

Input: MLT, dataset, k_1, K_2.
Output: The generalized estimation error $RMSE_{l,w}^{MLT}$.

for each MLT in {RF,SVR,LR} **do**
  for each forecast window β and lag window l **do**
    split dataset into K_2 folds.
    for each iteration in K_2-fold CV **do**
      for each hyper-parameters combination **do**
        Do K_1-fold CV on the $K_2 - 1$ folds
        Compute the model estimation error $RMSE_{M_{k_1}}$
      end for
      The optimal hyper-parameter combination is the one with minimum $RMSE_{M_{k_1}}$, noted M_{k_1}.
      Test the model M_{k_1} on the test set.
      Compute the estimation error on test set, $RMSE_{k_2}$
    end for
    Average the error over K_2 tests, noted $RMSE_{l,w}^{MLT}$.
  end for
end for

(LR) and the support vector regressor (SVR). The three MLTs are chosen as to investigate the linear and non linear relationships between the eNB lower layer metrics and throughput. The RF and SVR have several hyper-parameters to tune by the user. The setup of this parameters is crucial for the estimation performance. Accordingly, the strategy used in this work to find the optimal hyper-parameters combination is explained, it concerns the random search. Moreover, with MLTs implementation, one should be able to generalize the observed estimation error with each model. Therefore, an approach for the error estimation generalization is required. We have concluded that a combination between nested-CV and random search is the best approach for such task, as it leads to an estimation with low bias and low variance. Further, we propose to have a scalable estimation model, where the forecast window can be fixed based on the application type. Also, another system argument is added as to investigate the impact of historical lower layer metrics values on the instantaneous throughput estimation, i.e. lag window. Finally, the estimation model is summarized with all the involving parameters.

In this chapter, we have supposed the existence of datasets to estimate the instantaneous uplink throughput from lower layer metrics. Thus, there is a need for datasets with both eNB lower layer metrics and throughput to feed the estimation models. Accordingly, in the next chapter we explore the possible environments to collect these metrics, e.g. simulators and software defined radio (SDR) based platforms. The selection of such environment is quite challenging as a realistic real time 4G transmission should be accessible to gather the required metrics. Therefore, a real time 4G testbed is deployed. The testbed deployment and datasets building process are then detailed. Furthermore, the performance evaluation of the estimations based on the various built datasets, using the developed model in this chapter with the three MLTs, is finally discussed.

Chapter 3

Instantaneous uplink throughput estimation

Contents

3.1 Introduction

Several mechanisms are integrated in the eNB lower layers to reconstitute the data sent by the UEs. They use numerous metrics and measurements for their underlying procedures to reach their predefined objectives. For instance, radio measurements such as the signal to noise ratio (SNR) might be taken as a decision metric in the scheduling task. Also, the HARQ (Hybrid Automatic Repeat reQuest), decoding and demodulation functions incorporate various metrics specific to their

algorithms. Most of those metrics value change with the radio channel condition variation and also the network capacity. It then impacts the variation of the received throughput. It is therefore interesting to investigate the importance of those metrics in the throughput estimation objective. For that the eNB lower layer metrics are taken as input variables for the throughput estimation model developed in the previous chapter.

In the framework of this thesis - the eHorizon project- there is no accessibility to eNB lower layer measurements via a Mobile Network Operator (MNO). Hence, another 4G environment is required to collect these metrics and build the dataset for the estimation model.

Therefore, in this chapter, the potential 4G test environments to collect the eNB metrics are exhibited. It includes simulators and software defined radio (SDR) platforms. The 4G simulators are revealed to be limited for our objective realization. A testbed based SDR is then chosen. Therefore, the testbed deployment is detailed with the collection process of the required metrics and measurements. From the collected metrics, datasets are built to train the estimation model developed in chapter 2 and perform the UEs instantaneous uplink throughput estimation.

Later, an evaluation of the obtained results for the throughput estimations is conducted.

3.2 4G test environments

The aim is to monitor the UL QoS efficiently through investigating the UL performance and its throughput estimation. It then concerns the observed QoS at the network side instead of UE side. For that, a need to access the network measurements, especially the eNB metrics is inevitable. In fact, the eNB is the first 4G component that receives the UEs transmitted data. However, the open access datasets such as [Raca 2018] and [Li 2016] collect the measures/metrics mainly from the UE-side in real or simulated environments, i.e. metrics concerning the downlink transmissions and not the uplink. Furthermore, the cellular networks are private and proper to the MNO (Mobile Network Operator). Hence, the eNB metrics and measurements are confidential. Taking such fact into consideration, researchers have developed simulators and SDR based platforms to test the cellular networks and integrate evolutionary services. In this part, the existing 4G simulators and SDR (Software Defined Radio) based platforms are over-viewed.

3.2.1 Simulations based solution

Even though the high demand for cellular network test environments, a few open-source contributions have been proposed over the last decade. We differentiate mainly between the physical layer and high system level simulators. For instance, Mehlfuhrer et al. [Mehlfuhrer 2009] have developed a matlab based simulator for DL physical layer, proposing different capabilities such as one UE- one eNB, multi-users connected in one single cell and multi-user in a multi-cell environments.

On the other hand, mainly four strong contributions could be considered for the high system level simulations, modeling almost the entire protocol stack of the 4G components. J.Ikunu et al. [Ikuno 2010] developed a matlab based simulator taking only downlink transmissions into account. LTE-sim proposed by Piro et al. [Piro 2011] implements the main resources scheduling techniques for LTE network in multi-user/multi-cell environment. The later doesn't offer any support for the simulation workflow automation, e.g. definition and measures collection with a lack of one of the main eNB mechanisms, HARQ. Because of the physical layer complexity requiring a high computational effort, the aforementioned simulators implement an analytical model of the PHY layer, with no time notion, instead of a complete one.

Other scientists have developed the 4G protocol stack within the ns-3 framework[1] and simuLTE [Virdis 2016] based on omnet++. Although simuLTE integrates many PHY layer mechanisms, it still remains incomplete as the physical channels are not modeled down to the OFDMA symbols level. In other words, the later developed simulators are packet oriented based on discrete-event simulations. They either model the layer protocols or abstract them partially/altogether.

3.2.2 SDR based solution

All of the above mentioned simulators are either implementing the LTE from a system level, abstracting some of the protocol layers, or only the physical layer. Thus, it turns the 4G network implementation incomplete. Therefore, in order to compensate mainly the lack of the physical layer implementation in simulators, recently the Software Defined Radio (SDR) is taking place. The SDR based platforms offer a high level of realism and flexibility due to the introduced softwarization. SDR [Mitola 2000] refers to the radio transceiver/receiver system implementing, in software, the traditional hardware components, i.e. amplifiers, filters, etc. Different boards are now developed to support the SDR capabilities such as USRP (Universal Software Radio Peripheral) by National Instrument[2] and limeSDR by MyriadRf[3].

Several SDR based platform implementing LTE have emerged recently. The platforms include LTE 100 which provides the use of eNB and EPC full network functionalities over a standard linux-based PC interfaced with USRP SDR platform. Unfortunately, it is not open source and commercialized by Amarisoft [4]. Other closed-source LTE based SDR implementation are available such as PicoSDR10 by Nutaq [5]. There is also Open5GCore [6] implementing the 4G and 5G core networks adapted for integration with 5G new radio or eNB, but it is available with paid license.

[1] N.Baldo, "The ns-3 LTE module by the LENA project". Last access 02/2020: `tutorial`.
[2] National Instrument SDR products `http://www.ni.com`
[3] MyriadRF Sdr components `https://www.myriadrf.org`
[4] Amarisoft `https://www.amarisoft.com/technology`
[5] PicoSDR Series `https://www.nutaq.com/`
[6] Open5GCore `https://www.open5gcore.org/`

The LTE open-source SDR based platforms are emerging. For instance, LibLTE [Rondeau 2014] offers a pre-alpha development of SDR UEs and eNBs. OpenLTE[7] implements the 3GPP LTE specifications involved only in the downlink transmissions and reception. And, gr-LTE [Demel 2015] develops the receiver part solely. Thus, the aforementioned platforms implementations are incomplete.

Two complete open-source SDR based systems are available: srsLTE [Gomez-Miguelez 2016] and OpenAirInterface (OAI) [Nikaein 2014]. At time of experimentation, srsLTE implemented the full software protocol stack of LTE release 8, whereas OAI deployed fully LTE release 8 with a subset of release 10. Both platforms are highly realistic compared to the aforementioned simulators in the previous part. Through experimentation, authors in [Gringoli 2018] have proven that OAI is more accurate than srsLTE. Also, srsLTE consumes four times more CPU (central processing unit) than OAI, especially for uplink transmissions. With that, the OAI is chosen for our testbed deployment and will be detailed in the next paragraph.

3.2.2.1 OpenAirInterface platform

The OpenAirInterface (OAI) platform is the only fully open source SDR based platform that spans all the protocol stack of all the 4G components (UE, eNB, core network), including features from LTE-Advanced and LTE-Advanced-Pro for both E-UTRAN and EPC. Its deployment is possible over different frequency bandwidths, i.e. 5, 10, and 20 Mhz. All the UL/DL channels and their pre-processing mechanisms are realized. And, a fully integrated protocol stack from the physical layer to the network layer for both UE and eNB is developed, respecting the frame timing constraint. The software is compliant with 3GPP standards and developed in C/C++ under real-time Linux. OAI proposes emulation and real-time experimentations based SDR. It supports the USRP Hardware Driver software (UHD) of USRP B210 and X300 (USRP-Rio) family of products. Thus, the use of OAI platform comes with a high level of flexibility and realism. It then allows a repeatable and scalable system evaluation.

The deployment of an experimentation based OAI could have different configurations integrating the commercial components, i.e. sold components for public/ research usage. For instance, a 4G testbed using OAI platform can be built with connecting a commercial UE (e.g. a smartphone) to an OAI eNB and OAI EPC, or other combinations between OAI and commercial components as illustrated below, where $\leftrightarrow$ refers to a wireless connection and + is the wired link:

1. Commercial UE $\leftrightarrow$ OAI eNB + OAI EPC
2. Commercial UE $\leftrightarrow$ OAI eNB + Commercial EPC
3. Commercial UE $\leftrightarrow$ Commercial eNB + OAI EPC

[7] OpenLTE: `https://sourceforge.net/p/openlte/wiki/Home/`

4. OAI UE ↔ Commercial eNB + Commercial EPC
5. OAI UE ↔ Commercial eNB + OAI EPC
6. OAI UE ↔ OAI eNB + Commercial EPC
7. OAI UE ↔ OAI eNB + OAI EPC

This choice diversity for the 4G network deployment with OAI allows the researcher/developer to select the appropriate setup following its experimentation objective and material availability.

In our case study, the UE is required only to generate the 4G uplink traffic. Hence, a commercial UE is sufficient for the experimentation. Yet, an access to the metrics and measurements performed in the eNB layers is crucial. The OAI eNB is then chosen for implementation. Due to the unavailability of a commercial EPC in the laboratory, the OAI EPC is the best alternative for a complete 4G network deployment. Accordingly, the first setup, i.e. a commercial UE connected to an OAI eNB and OAI EPC, is used in this work.

3.3 Testbed deployment

The real world environment is characterized by its chaotic and uncontrolled radio phenomena. The laters degrade the QoS metrics in unpredictable manner. For such reason, we propose the use of an anechoic room where the RF propagation is controlled thanks to the microwave absorbers materiel on the walls, scattering any wireless signal that comes across. In fact, the absorbers take the form of cones of

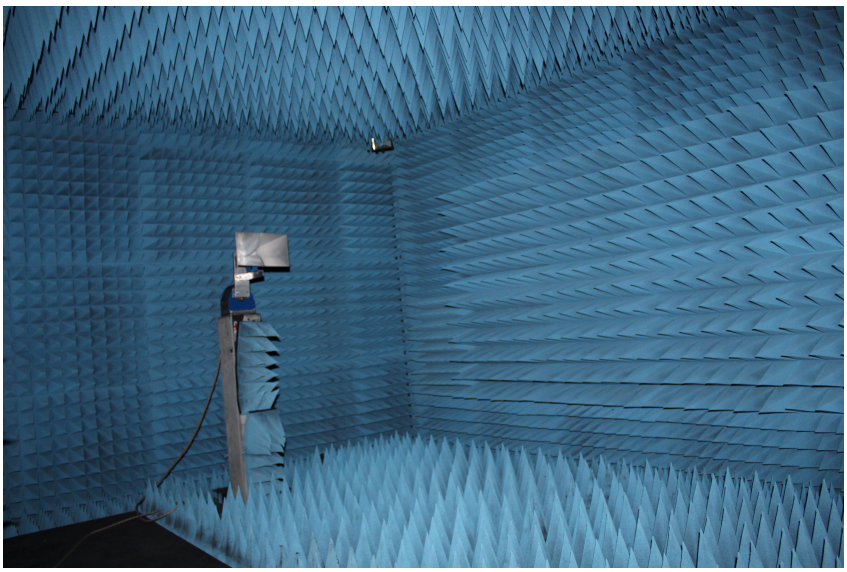


Figure 3.1 – Anechoic room in LAAS-CNRS.

different sizes able to capture the reflected waves, as shown on fig. 3.1. The cones'

size refers to the wave frequency able to be absorbed. Further, thanks to those insulators inside the room, the environment is free from inside and outside radio perturbations. This allows a clear analysis of encountered behaviors in the QoS metrics variation, as the room is isolated from any uncontrolled radio phenomena. Also, it is worth noting that one of the advantages of running experiments in a controlled anechoic room lead to unambiguous and reproducible work.

On the other hand, the 4G frequency bands are licensed to each MNO by a specific organism in each country. For instance, in France, it is up to the ARCEP to regulate the usage of the 4G frequency bands between the MNOs [8]. Therefore, it is restricted to use these frequency bands by any other organism to avoid interference. Hence, the use of the real-time 4G testbed inside an anechoic room respects the frequency regulation in the country.

With the use of the anechoic room, the uncontrolled effects during a wireless transmission are limited. As the throughput variation is related to the radio phenomena fluctuation, we have introduced in the anechoic room the major frequent radio phenomena in real world impacting the throughput, such as noise, multipath fading and radio congestion. Thus, it allows an investigation of each radio phenomena impact on the throughput. We started with a basic testbed deployment, referred as testbed 1 (see 3.3.1), where only a specific noise profile is added in the radio environment. Then, a more complex one is deployed in 3.3.2, labeled testbed 2. It deploys more radio issues and enlarge the eNB metrics benchmark. Once the testbeds are functional, the real time 4G communications are initiated to collect the relevant metrics for the estimation.

3.3.1 Testbed 1: LAAS-CNRS anechoic room

Fig.3.2 schematizes the testbed deployment. The setup is inside the anechoic room of LAAS-CNRS of dimension 4,10 m * 2,50 m. Openairinterface (OAI) software based platform is implemented for both eNB and EPC components. The OAI softmodem is connected with a hardware platform for SDR: USRP B210. The later is connected to a host computer to perform processing, and then connected to a PC running the core network, and accessing the internet and a server.

To emulate a connected machine, a commercial UE (Samsung Galaxy J3 2017) is used and controlled remotely. The network transmission mode is Single Input Single Output (SISO). The eNB antennas (Tx and Rx antennas) and the UE are placed inside the anechoic room, which is free from any multipath phenomena and radio perturbation or degradation. The testbed works on frequency band 7 with 5 MHz bandwidth using FDD mode, which corresponds to the traditional and stable version of OAI platform with USRPB210 at the time of the experimentations. Using this testbed the undermentioned scenario is realized.

[8] ARCEP: Autorité de Régulation des Communications Électroniques et des Postes. `https://www.arcep.fr`

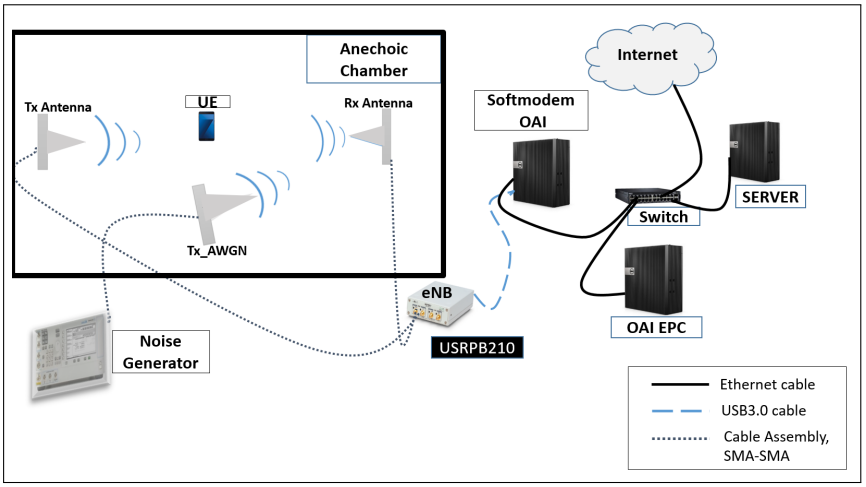


Figure 3.2 – Testbed deployment in LAAS-CNRS anechoic room.

3.3.1.1 Scenario 1: Noisy based transmissions

In real environments, multiple radio phenomena are scrambling the communications. Example phenomena include multipath fading leading to InterSymbol Interference (ISI) noise, pathloss and random processes such as AWGN (Additive White Gaussian Noise). These phenomena tend to attenuate aggressively the transmitted signal which causes a significant amount of signal strength reduction. Thus, in time varying scenarios, the received signal amplitude undergoes rapid fluctuation that is often modeled as a random variable with a particular distribution.

In this testbed, we consider the Gaussian distribution, AWGN, which is characterized by its amplitude that affects the signal strength. Moreover, noise (AWGN) is introduced as it causes transmission errors and may disrupt the communication with ISI production for high power noise [Bolat 2003]. Contrarily to work in literature where AWGN is often taken with constant attenuation, we introduce randomness in the attenuation in order to have attenuation fluctuations of the signal over time. For that, we developed an assisted labVIEW program on the noise generator (Anritsu MG3700A). Given an interval of maximum and minimum noise levels, each 10 s the noise level takes randomly a value in the specified interval. The programmed step for noise level change (10 s) is chosen as to have sufficient samples for each noise level. Therefore, low noise level values keep the channel flat, while high noise level disrupt totally the communication, with the probability of introducing ISI. Accordingly, the noise interval bounds are chosen after several tests. Also, the abrupt changes in noise levels during the transmission, as illustrated on fig 3.3, tend to reflect the real environments, where the user's mobility across different shadowers leads to aggressive/alleviated signal attenuation. The noise is injected inside the anechoic room using a signal generator with a directional antenna toward the eNB receiver. The frequency bandwidth is fixed to 5 Mhz to scramble the full UL bandwidth.

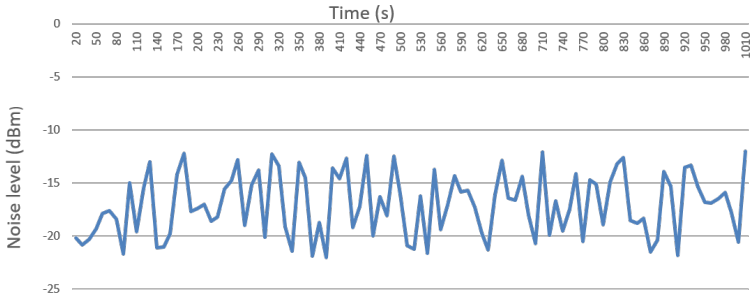


Figure 3.3 – Noise level variation during the test scenario 1

3.3.2 Testbed 2: INRIA anechoic room

3.3.2.1 R2lab platform

In order to extend the first testbed and complexify the tests, R2lab [Parmentelat 2018] is used as an underlying testbed. It is located at Inria, Sophia Antipolis, France. It offers a wireless network with multiple computers interfaced Wi-Fi and SDR nodes inside an anechoic room of size about 90 m^2, as shown on fig 3.4. 37 available nodes in total, each one being Icarus off-the-shelf computers with CPU Intel®$Core^{TM}$ i7-2600, 8M Cache at 3.40 GHz, 8GRAM and 240GB SSD. The nodes are dispatched on a grid, which allows the implementation of different scenarios with and without line of sight.

Figure 3.4 – R2LAB open wireless testbed [Parmentelat 2018].

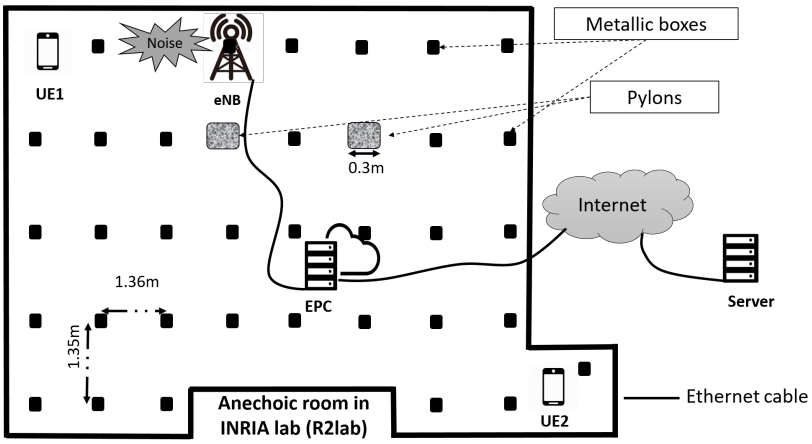


Figure 3.5 – Testbed 2 deployment.

3.3.2.2 Testbed 2 description

The SDR based nodes enable a full access to realistic physical metrics/ measurements. With remote access to the indoor nodes, the deployment of LTE-A network is accomplished. Fig. 3.5 schematizes the deployed testbed. Openairinterface (OAI) software based platform is implemented for eNB and EPC. The OAI eNB is connected with an USRP B210. A 2.6 Ghz antenna is attached to the URSP via a duplexer. The later allows one antenna to work in both transmission and reception at the same time. It is introduced as to remove interference between the two nearby antenna connectors. However, FDD mode is used in frequency band 7. The USRP B210 is then connected to the computer for processing, and to the PC running the EPC. To emulate 4G connected nodes, two commercial UEs are used and placed inside the anechoic room (Nexus 5 and Moto E4G smartphones). The hardware metallic enclosure boxes scattered on a grid inside the room are considered as fixed multipath sources, which we take advantage of in investigating the impact of multipath phenomenon on throughput variation.
The upcoming scenarios tend to investigate the impact of each phenomenon on throughput variation. For each test scenario, a radio phenomenon is added in the system to complexify the tests.

3.3.2.3 Scenario 2: Multipath fading based transmissions

We investigate the impact of multipath fading and pathloss on throughput variation in this scenario. In other words, the two UEs used in the experimentation are at different distances from the eNB, i.e. each UE has a different pathloss. Over the room, multiple metallic boxes surround the eNB (fig. 3.5). They introduce multipath fading leading to ISI. The two pylons, covered with absorbers are considered as shadowers for the transmissions, especially for UE2. Therefore, only pathloss, shadowing and multipath fading are present in the system as scramblers for the transmission, i.e the anechoic room is isolated from any other radio phenomena.

In order to avoid any degradation/losses due to the insufficiency of radio resources the total transmitted data by the two users is inferior to the maximum network capacity.

3.3.2.4 Scenario 3: Noisy and multipath fading based transmissions

In this scenario, another radio phenomenon is added in the anechoic room to investigate its presence on the throughput estimation, i.e, noise. For that we generate in a controlled manner a specific noise profile to scramble the communication. As in scenario 1, the AWGN is the noise introduced in the testbed. But, this time the gain is also changing randomly with the amplitude as to increase the effects of the noise profile. Further, noise level is defined as noise with a given gain and amplitude. We used mainly GNUradio on USRP B210. Given a list of gain levels and an interval of maximum and minimum amplitude levels, each 10 seconds a random value of amplitude and gain are chosen. In fact, the amplitude value affects the statistical characteristics of the noise source, i.e, the standard deviation of the Gaussian noise. The gain affects the transmitted signal power. The programmed step for noise level change (10 s) is fixed in order to have sufficient samples per each noise level. Further, the ISI could be introduced by high noise level and also the fixed mutipath generated by the metallic boxes inside the room. To scramble all the UL bandwidth, the bandwidth is fixed to 5Mhz.

3.3.2.5 Scenario 4: Congested radio, noisy and multipath fading based transmissions

During the previous scenarios, the total transmitted data by the two UEs is much lower than the maximum available capacity. In this part, another radio phenomenon leading to throughput degradation is introduced in the testbed, i.e. radio congestion. The later occurs when the total capacity required by the connected UEs outpaces the maximum eNB capacity. Mobile application based speedtest is tested in scenario 2 for UE 1. It reaches the maximum throughput of 8 Mbps. Therefore, in order to realize radio congestion, the total transmitted bit-rate by both UEs outpaces the network maximum capacity. Noise is introduced as described in scenario 2 to complexify the test.

3.3.3 Real-time 4G traffic transmission

3.3.3.1 Iperf: traffic generator

Iperf[9] is a tool for IP traffic generation over TCP, SCTP and UDP. Various parameters could be tuned as to monitor the traffic transmission, such as timing, buffers and protocols. It consists of a clients and a server. The clients transmits the generated traffic and the server is the receiver. There is mainly 3 versions of iperf,

[9] A.Tirumala et al. iPerf: TCP/UDP bandwidth measurement tool. Last access 01/2020. `https://iperf.fr/`

i.e. iperf, iperf2 and iperf3. Iperf3 is the last developed and sophisticated version of this tool, developed by ESnet / Lawrence Berkeley National Laboratory. Contrarily to the older version, Iperf3 allows capacity monitoring at small time slots under 1 s. Hence, in the following tests, we have chosen Iperf3 as a traffic generator as we are interested in estimating the throughput at small time granularities.

3.3.3.2 Traffic transmission

For all the above scenarios, Iperf3 generates traffic at the UE side and Iperf3-server monitors throughput reception in the server. In order to have a fixed transmission amount of data during the whole test duration, UDP (User Datagram Protocol) is used as a transport protocol. In fact, TCP (Transport Control Protocol) changes the transmission window based on the perceived packet losses in the window. Given this, any observed bandwidth degradation is essentially due to radio environment variation. Using a speed test throughput application in the UE, the maximum UL data-rate achieved in the testbed is around 8 Mbps. From that, the traffic consists of UDP flows where the UE data-rate is fixed differently regarding each scenario as explained further. The size of the packets is set to 1350 bytes to avoid any segmentation during transmission.

- Scenario 1: The UE data-rate is fixed to 8 Mbps, which allows the use of all the available PRBs for transmission.
- Scenario 2: UE 1 transmits during the whole test duration a fixed amount of data of 2 Mbps and UE 2 transmits 3 Mbps. This way, the total transmitted data is inferior to the network maximum capacity. Hence, it avoids any degradation due to radio congestion.
- Scenario 3: As in scenario 2, i.e. multipath fading based transmissions, UE 1 and UE 2 transmit their data at a rate of 2 Mbps and 3 Mbps respectively to prevent any congestion in the network.
- Scenario 4: The objective is to reach the unavailability of resources in the network during scheduling, i.e. radio congestion. For that, the required data-rate of both UEs is increased as to outpace the maximum available capacity offered by the network, i.e. UE 1 transmits its data at a fixed amount of 4 Mbps and UE 2 at 5 Mbps.

3.4 Datasets collection

The constitution of datasets is required for the application of the machine learning techniques to estimate/predict the instantaneous uplink throughput. In the aforementioned deployed scenarios in both testbeds, the radio environment is disturbed in a controlled manner. And, in the 4G networks, it is up to the eNB to correct/treat the errors and negative effects of the radio phenomena on the received

signals. Also, the eNB schedules the connected users. Those tasks are mainly performed in the lower layers, i.e. MAC and PHY layers. From that, the first collection point is the eNB, principally from the lower layers. Then, as the objective is the throughput estimation, the second collection point of metrics is the server receiving the data from the served UEs. The received throughput is recorded per specific slots of time during the whole tests duration.

In the following, the eNB collected metrics in each testbed is presented with an insight on the approach followed in order to build the datasets.

3.4.1 Testbed 1 datasets

The objective of the first testbed is to investigate the possibility of estimating/predicting throughput from radio measurements when the UE transmits its data at the maximum network capacity with only noise disturbing its transmissions. The test duration is 1530 s. For that, the radio measures are the main collected metrics in this testbed. In fact, the eNB performs different radio measurements in order to decode the received data and adapt to channel variation. With SDR at eNB side, we are able to collect all the performed eNB measurements, especially from the lower layers. After a benchmark of physical layer measurements, the following metrics are extracted:

- RIP (dBm): Received Interference Power measured within the bandwidth of each PRB. The eNB measures the noise power over the PRBs each 1 ms.

- UL_RSSI: Uplink Received Signal Strength Indicator. It measures the total wide band received power over the full bandwidth - 5 MHz - including noise and interference. UL_RSSI states the quality in the cell. Too low RSSI reflects the inability of the cell to communicate with any UE, while too high value indicates a high level of interference in the cell.

- SNR: Signal to Noise Ratio compares the level of the desired received signal to the level of noise. Taking P_{signal} and P_{noise} as the average power of the received signal and noise respectively, SNR is defined (in decibels) as follows: $SNR_{dB} = 10.log_{10}(P_{signal}/P_{noise})$. It is measured for each received PUSCH holding UE's data and PUCCH containing the UE's control signaling information. Let's denote PUSCH_SNR and PUCCH_SNR the corresponding metrics.

- Rx_power (dBm): The received power. It measures the received power in the eNB based on the demodulation reference signal (DMRS), which is used to get a coherent detection and demodulation of UL channels.

The ultimate objective of this work is to realize instantaneous throughput estimation over flexible time granularities. The throughput measurements are performed in a discrete time manner. As the minimum time report interval in IPERF3

Table 3.1 – Constitutions of datasets based on scenario 1, testbed 1.

Dataset	Sample notation
Dataset_RIP	{RIP_min, RIP_max, RIP_mean}
Dataset_SNR	{PUSCH_SNR_min, PUSCH_SNR_max, PUSCH_SNR_mean, PUCCH_SNR_min, PUCCH_SNR_max, PUCCH_SNR_mean}
Dataset_RSSI	{UL_RSSI_min, UL_RSSI_max, UL_RSSI_mean}
Dataset_Rx_power	{Rx_power_min, Rx_power_max, Rx_power_mean}
Data_ALL	{Dataset_RIP, Dataset_SNR, Dataset_RSSI, Dataset_Rx_power}

is 100 ms, we fix δt=100ms. The estimations are then made every δt. This granularity is in fact smaller than the related work predicting throughput at minimum of 1 s, as stated in chapter 1 (see section 1.3). On the other point, The eNB measurements are performed per subframe scale (1ms). Therefore, the collected metrics from the two network components are on different scales, which is not accepted by the machine learning process. To overcome this scale conflict, we propose the use of the main representative statistical metrics for each eNB collected measure. Thus, we compute the maximum, minimum, and the mean of each measure per δt to construct the datasets. This way, each radio measurement has its representative measure per δt as illustrated on table 3.1. The later depicts the constitution of datasets used as input for learning algorithms. The first column represents the datasets labels, and the second column explicits the constitution of datasets in the form of samples. We built 5 datasets based on the aforementioned metrics. Four of them are based on a single feature and the last one combines all the features. For instance, dataset_RSSI contains the single UL_RSSI feature. Each sample in dataset_RSSI is composed of the maximum, minimum and mean measured UL_RSSI over δt, i.e. {UL_RSSI_min, UL_RSSI_max, UL_RSSI_mean}, dataset_All is a combination of the four features.

3.4.2 Testbed 2 datasets

The second testbed is considered as an extended version of the first one, not only by adding another UE but also by exploring the impact of other radio phenomena on the throughput degradation. Furthermore, we propose a deep investigation of the impact of radio environments on mostly all the lower layer metrics as well as throughput. Hence, using the advantage of SDR, we collect almost all the performed lower layers (PHY and MAC layers) measurements and metrics from the eNB. As the connected UEs communicates their control signaling information using PUCCH, and transmit their data on PUSCH channels, we perform a deep benchmark of the main metrics/measurements linked with the two channels. The OAI eNB metrics/measurements are performed as depicted by the 3GPP standard. The

collected metrics are mainly extracted during the lower layers data processing as described further.

On one hand, radio measurements are collected as they are crucial for higher layer mechanisms and reflect the channel quality. The main measures are:

- SNR: It is measured for each received PUSCH holding UE's data.
- received UL_CQI: Uplink Channel Quality Indicator. It is computed at the eNB based on the observed SNR.
- PUCCH received power and noise power: the two measures are estimated principally for each PUCCH handling a scheduling request.
- PUCCH threshold: It is the threshold to detect the pucch format1.

On the other hand, several metrics are collected during the real time processing of transmitted data by both UEs. As explained in chapter 1, the eNB lower layers fulfill different mechanisms to deal with the errors introduced by the wireless channel as well as to support the required QoS, e.g. MCS, HARQ, CRC. Therefore, the metrics incorporated in these procedures and their functions are extracted. Particularly, the TBS metric that refers to the block size of data in the physical layer. Moreover, the metrics related to the decoding/demodulation processes are also retrieved, e.g. decoding time, decoding iteration. The length of each received SDU/PDU (Service Data Unit/Protocol Data Unit) on the MAC layer is also considered with the buffer size when data is handled.

Overall, a total of 43 metrics is collected during the whole test duration, i.e. 400 seconds (see annex A).

Similar to the first testbed, the received throughput measurements is performed in a discrete time manner, each $\delta t = 100$ ms. Hence, the estimations granularity is δt. As the eNB metrics are collected per subframe scale (1 ms), and the scaling task is required to build datasets for the estimation model, i.e. all the metrics must be on same granularity, we propose this time to use all the statistical metrics of each eNB measures. In other words, we compute the maximum, minimum, mean, median and the standard deviation of each eNB metric per δt to build the datasets. Therefore, each metric γ is represented in the dataset as follows: $\{\gamma_{min}, \gamma_{max}, \gamma_{mean}, \gamma_{median}, \gamma_{std}\}$. For each UE u, $u \in \{1, 2\}$ and each scenario s, $s \in \{2, 3, 4\}$, a dataset is built, noted $dataset\_s\_u$. Each $dataset\_s\_u$ contains all the lower layers metrics collected for the given UE u during the scenario s, including historical received throughput.

3.5 Results evaluation

3.5.1 Throughput analysis

The scenarios deployed in the real time 4G testbed are chosen as to reproduce the major frequent radio phenomena in real world impacting the throughput. However,

the throughput undergoes rapid fluctuation during each scenario. For instance, for the first scenario, with the random noise profile, the throughput variation is illustrated on fig 3.6, where throughput reaches is received at minimum rate for higher noise values. Hence, an analysis of the impact of radio issues on the received throughput is relevant for our study.

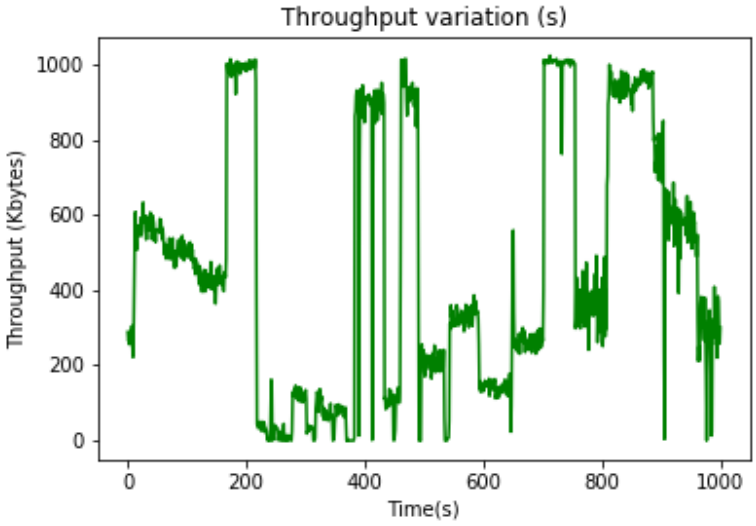


Figure 3.6 – Scenario 1 throughput variation.

Firstly, an outlook on the received throughput statistics is given in table 3.2. It represents the minimum, maximum and mean received throughput for all the UEs during all scenarios. It is to be noted that henceforward, throughput refers to the received amount of data (Kbytes) per $\delta t = 100ms$. Then, to push the analysis, fig. 3.7 plots the standard deviation of the received throughput for each UE during each scenario. Each bloc is scenario specific.

The first scenario scrambles the uplink transmissions with a specific radio noise profile using a directional antenna. The UE data-rate was fixed to 8 Mbps during the whole test duration. Hence, for ideal performances, the server shall receive 100 Kbytes/δt. But, during this scenario, the received throughput varies between 0 Kbytes/δt and 128 Kbytes/δt (see table 3.2). And, based on fig. 3.7, it is clear that the throughput distribution undergoes a rapid fluctuation over the wide range, i.e. between 0 Kbytes/δt and 128 Kbytes/δt. In fact, in this scenario a high level of packet loss is observed and the retransmission mechanism was active. This could be explained by the higher and abrupt change of the noise values impacting the transmissions. Further, it discerns the radio change consequences on the throughput variation. And, such fluctuation is critical for services with a granted QoS.

During scenario 2, the two UEs transmit at a fixed low level data-rate, i.e. 2 Mbps and 3 Mbps for UE1 and UE2 respectively, in an environment where only a multipath fading and fixed shadowing exist. The mean received throughput from UE1 and UE2 is 24.41 Kbytes/δt and 36.61 Kbytes/δt respectively, which is near the

Table 3.2 – The received throughput statistics (Kbytes/δt.) for all UEs over all scenarios

Throughput (Kbytes/δt)	UEs	Minimum	Mean	Maximum
Scenario 1	UE0	0	51.1	128
Scenario 2	UE1	0	24.41	69.7
	UE2	0	36.61	76.7
Scenario 3	UE1	0	24.39	82.3
	UE2	15.3	36.6	64.1
Scenario 4	UE1	0	48.8	86.5
	UE2	16.7	58.34	105

theoretical amount of data that should be received during the whole test duration, i.e. 25 Kbytes/δt and 37.5 Kbytes/δt from UE1 and UE2 respectively. Nevertheless, a considerable throughput variation is observed. From the figure 3.7, UE1 throughput distribution is straight compared to the UE2 received throughput. In fact, the amount of received throughput from UE2 is spread out over much wider range. Such issue could be mainly explained by the radio phenomena experienced by UE2. The UE2 transmissions face a high level of multipath fading and shadowing compared to UE1 transmissions. UE1 has a near-line-of-sight with the eNB. This observation points out the sensitivity of the 4G based communications to the multipath fading and shadowing issues.

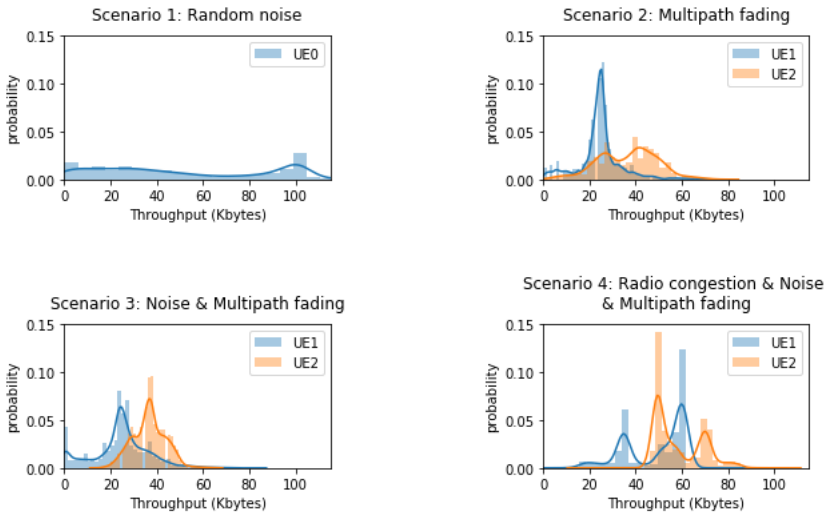


Figure 3.7 – Throughput standard deviation over all scenarios in both testbeds.

For scenario 3, noise profile is injected in the testbed. The noise source is placed between the UE1 and the eNB. Both users keep transmitting their data at the same rate as scenario 2. On the third sub-figure of fig. 3.7, it is clear that

noise has added variance to the UE1 distribution, with the curve becoming right skewed. Such impact is expected as the noise variation introduces errors and ISI, especially for UE1 transmissions. The maximum received UE1 throughput has increased to 82 Kbytes/δt instead of 69.7Kbytes/δt in scenario 2. This is in fact conform to the observations from the first scenario. In contrast, surprisingly the UE2 throughput variance is decreased in this scenario. The minimum received throughput is increased too, from 0 Kbytes/δt to 15.3 Kbytes/δt (table 3.2). Recall that UE2 transmissions are highly affected by multipath fading compared to UE1 transmissions. From that, the throughput variation in this scenario for UE2 exhibits the positive impact of the noise profile on the unwanted multipath signals.

When the transmitted data outpaces the network capacity in scenario 4, two modes are highly distinguishable with low variance for both users. Also, the maximum received throughput is higher compared to previous scenarios (table 3.2. Such result is expected, as during the transmission, a high level of packet loss is observed and retransmission was highly active. It is worth mentioning that multiple metrics values were missed during this scenario. It reflects the real complex wireless systems, where the presence of multiple radio phenomena lead to severe throughput degradation.

Overall, this illustration sheds light on the high link between radio phenomena and high level QoS metric, i.e. throughput. The eNB lower layer metrics treating these radio phenomena are then vital for the throughput estimation. With that, an explanation of the input features space management (i.e. eNB lower layers metrics) is given in the next section.

3.5.2 Features space

In order to exploit the linear relationship between the collected metrics and the throughput, LR (Linear Regressor) is applied as an underlying ML technique. On the other hand, RF has the ability to create the linear and non linear boundaries during the trees building, which leads to accurate predictions. RF (Random Forest) is then applied. SVR (Support Vector Regressor) uses the non-linear kernel function to extract the non-linear relationship between the input and output metrics. As many features are available, the feature space reduction selection is crucial, especially for LR. To this end, the statistical technique Principal Component Analysis (PCA) is used [Jolliffe 1986]. PCA reduces the high features space dimension with variance maximization of each component. For instance, it uses orthogonal transformation to generate linear combination of orthogonal features vectors. In contrast, an approach for features importance determination is implemented in RF models. RF combines multiple decision trees to get a more accurate estimation. For each node split, it looks for the most important feature among a random subset of features. In fact, a given feature is considered as important if its perturbation leads to larger error. Hence, RF relies on features importance for building the forest. Thus, it turns PCA unnecessary in the case of RF. Furthermore, as we are using the random search technique to tune the MLT hyper-parameters for the SVM. The SVM

kernel function is then configured as to maintain only the important feature space and have the most adapted hyper-parameters for a given dataset. Also, RF and SVM are well known for their insensitivity for higher-dimensional features space. With this, the space features for RF and SVR is not reduced. The estimation model developed in chapter 2 is applied with different MLTs on the various datasets. In the following, the RMSE score equals the mean RMSE over the outer loop CV of the nested CV.

3.5.3 Estimation accuracy

In order to compare the estimation performance of the three ML techniques, fig 3.8 exhibits the observed RMSE with all the datasets for window forecast of $1\delta t$. The RMSE with the five datasets built in scenario 1 is higher compared to the ones of the other scenarios (i.e. scenario 2, 3 and 4). The scenario 1 datasets are built with only radio metrics. That is, a throughput estimation based on radio measurements is conducted. The estimation based LR with these datasets leads to an error between 37 and 38 Kbytes. Recall that the maximum received bandwidth is 128 Kbytes. The error is then about 28%. RF has quite similar estimation error as LR. With SVR, the RMSE has increased smoothly, i.e. it ranges between 39 Kbytes and 41 Kbytes. Interestingly, even the combination of all the radio metrics in one dataset, i.e. dataset_all, doesn't produce good estimations.

In order to figure out whether the cause is the length of training samples per noise level, another test is performed, where the time step is 60 s instead of 10 s. Hence, 600 samples are available for each noise level, and a total of 26 noise levels are tested. The same methodology is applied, and similar results of RMSE are obtained (fig.3.8). It clearly exhibits that the cause behind the higher observed values of RMSE is not the low number of training samples per noise level.

In order to generalize the estimation errors and be sure that the 10-fold nested CV doesn't introduce any bias during the estimation stage, we study CV bias with varying folds. Different K values for the K-fold method are tested. A small value of K forms folds containing multiple noise level transitions, while higher values of K forms folds with a low number of samples per noise level. For K=15, the number of folds is exactly the number of noise levels. In this part K_2=K_1 for the nested CV. Fig. 3.9 shows the distribution of the observed RMSE per CV size for a forecast window of $1\delta t$ with dataset_Rx_power and the underlying RF MLT. For each K-fold, a vertical boxplot is drawn. It consists of a box from the lower quartile of the observed RMSE to the upper quartile, with a crossbar in the mean of RMSE. The upper and lower fences outside the box represent respectively 95% and 5% of the estimated RMSE. For all the tested K-folds, the mean RMSE is around 36 $Kbytes$. Then, the generalized estimation error for dataset_Rx_power in an environment with random noise is approximately 36 $Kbytes$.

On the other hand, the datasets generated from testbed 2, where cross-layer metrics benchmark is fulfilled under multiple radio phenomena coexistence, produce an estimation with an error less than 13 Kbytes with the three MLTs. Fig 3.8 shows

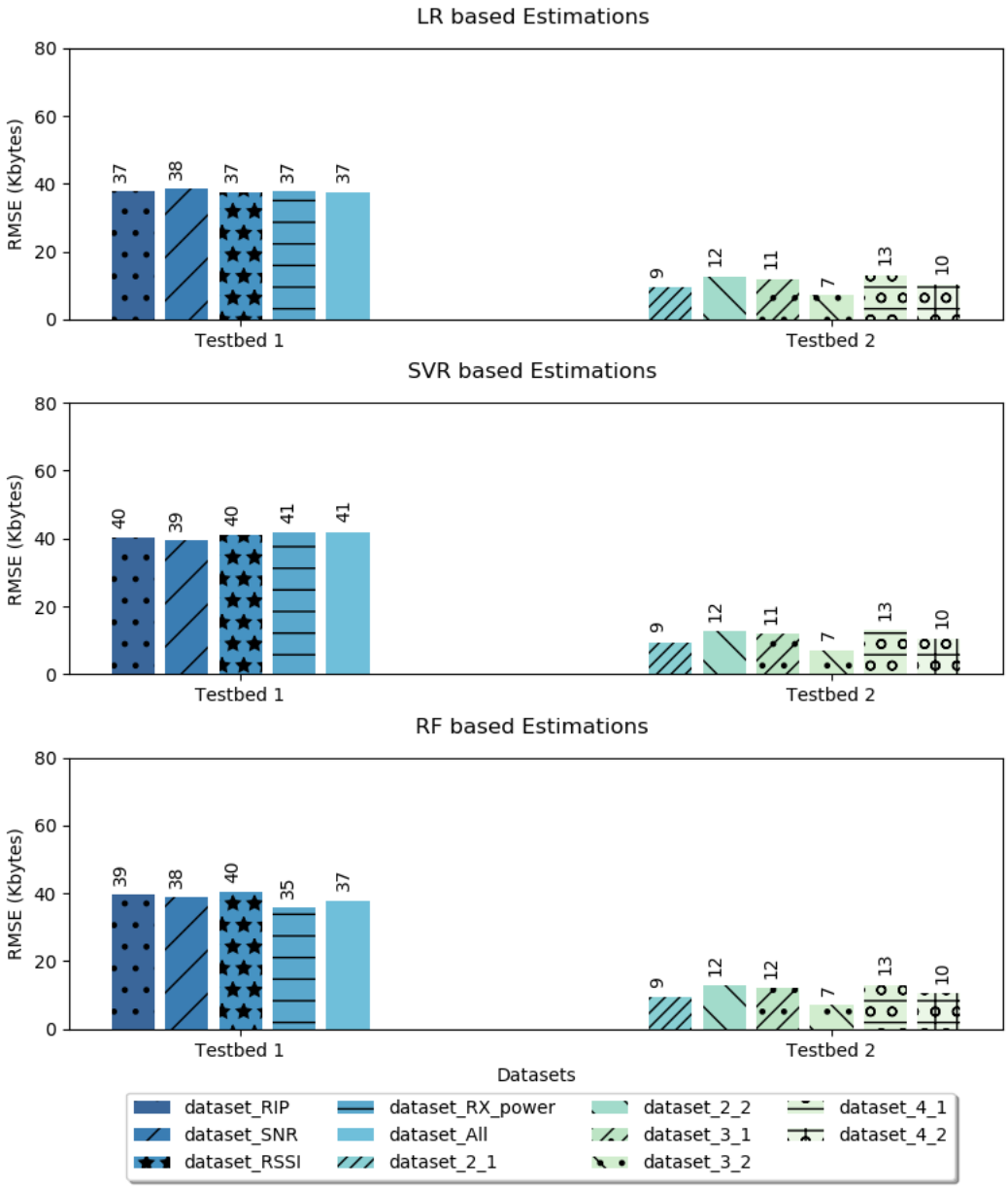


Figure 3.8 – Estimation errors with the different MLTs as a function of the different datasets of both testbed 1 and testbed 2.

the statistics for a forecast window of $1\delta t$ and with only the PHY and MAC layer metrics. In other words, the historical throughput is not introduced as an input argument in the estimation model.

The three MLTs lead to similar RMSE for all datasets at a difference of some bytes only. The observed RMSE is changing based on the used dataset for each scenario. Particularly, for scenario 2, the UE 1 RMSE is about 9 Kbytes while the

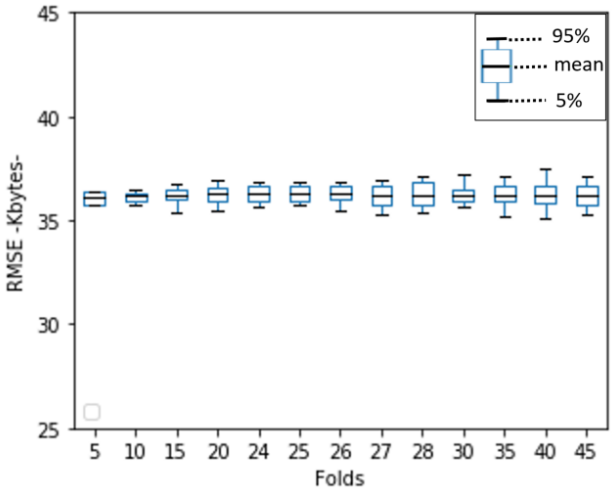


Figure 3.9 – Cross-validation bias.

one for UE2 is 12 Kbytes. Recall that both UE 1 and UE 2 send at same fixed data-rate in both scenarios 2 and 3, i.e. 2 Mbps (UE1) and 3 Mbps (UE2). The RMSE has increased for the UE1 from 9 Kbytes to 12 Kbytes when random noise in introduced in the environment (scenario 3). Whereas, the estimation performance for UE2 has increased in scenario 3, i.e. the mean RMSE is reduced by 5 Kbytes. Then, when the radio environment is complexified through the setup of radio congestion (i.e. scenario 4), the UE2 RMSE is around 10 Kbytes, and 13 Kbytes for UE1. Further, the three MLTs produce accurate estimations, ranging between 9% and 15% for complex radio environment.

Overall, the instantaneous throughput estimation based on only radio measurement is not accurate as the estimation error can reach 32%. It is actually less accurate compared with DL estimations based radio measurements, already mentioned in part 3.1 (chapter1). On the other hand, the cross-layer metrics increase the models accuracy in different radio environments. It is therefore possible to estimate the instantaneous uplink throughput at small time granularities based only on the cross-layer metrics, i.e. datasets built from testbed 2.

3.5.4 Higher scale estimation: forecast window

As the forecast window size of the instantaneous throughput depends on the application/service needs, we investigate the scaling of the initial forecast window $1\delta t$ in this part. Fig 3.10 shows the estimation error as a function of forecast windows with the three MLTs for all the datasets, i.e. testbed 1 datasets are plotted on the left figures while the right figures encompass the testbed 2 datasets.

Regarding the testbed 1 datasets, all the tested datasets keep the same RMSE evolution over the different forecast windows, i.e. curves are quite overlapping. The RMSE decreases with larger forecast windows with both RF and LR. It drops down from 39 Kbytes to 17 Kbytes when forecasting up to $7\delta t$. Then it remains at

the same scale for larger windows. Contrarily, the SVR RMSE varies in a random manner with the forecast windows. Notably, the RMSE is reduced to 18 Kbytes with $4\delta t$, then it mounts up to 31 Kbytes. Further, higher forecast windows lead to a reduced RMSE when applying both RF and LR on radio metrics.

For the estimation based on cross-layer metrics, the RMSE with dataset_2_1 (scenario 2, UE1) decreases from 9 Kbytes to 7 Kbytes when forecasting for larger windows with all MLTs. Furthermore, the estimation based on dataset_3_1 improves with higher windows size. Contrarily, the RMSE for the estimations based dataset_2_2 increases proportionally to the window forecast. It reaches 15 Kbytes for estimations at a granularity of 1 s. The other datasets i.e. dataset_3_2, dataset_4_1 and dataset_4_2 produce a quite stable estimation error for almost all window forecasts.

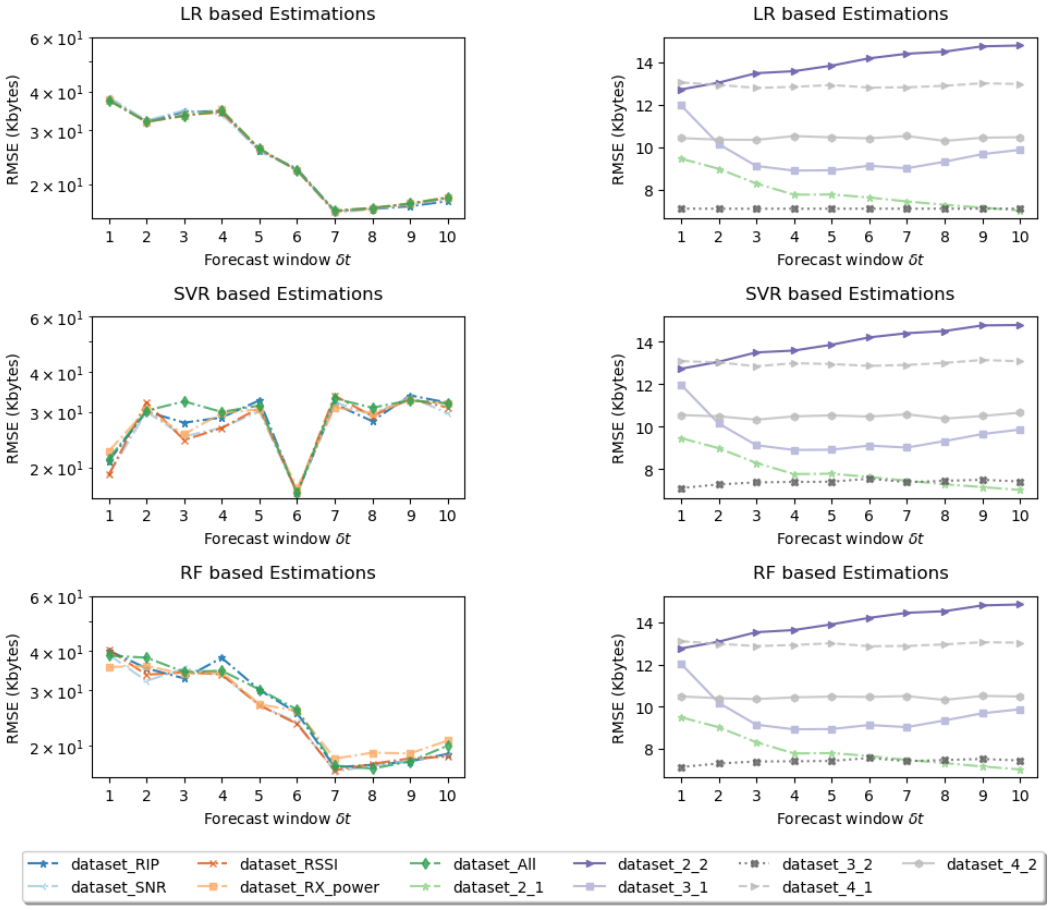


Figure 3.10 – RMSE as a function of the forecast window for all datasets of both testbeds. Left sub-figures represent the testbed 1 datasets, and right sub-figures plots the testbed 2 datasets.

To investigate deeply the high observed RMSE for estimations based on radio measurements, another experimentation is configured, where the noise level changes linearly instead of random variation. In fact, in scenario 1, the noise level

was changing randomly during the test duration. Hence, the selected X_train contains a significant amount of random noise level variations. In this part we study the impact of the presence of such transitions in the training set, we set up another experimentation process. Keeping the same testbed, we changed the assisted LabVIEW program to control remotely the noise generator. This time, given an averaged granularity of 0.1 dBm as a step for noise level variation, the noise level is increasing every 10 s by the programmed step during the whole test duration (800s). This later experiment is aimed at providing a dataset less complex than the previous one in order to investigate the impairments introduced by the random change of noise levels on the system. The same process of data collection in the other scenarios is followed. We evaluate each dataset from table 3.1 over $i\delta t$, with $i \in \{1, .., 10\}$ and $w = 0$. During throughput estimations, the forecast window changes from $1\delta t$ to $10\delta t$. The maximum, minimum and mean observed RMSE with K_2-fold CV is represented in fig. 3.11. For each dataset, dashes represent the maximum and minimum of the observed RMSE per δt, and the line links the mean of the observed metric per δt. Based on dataset_SNR and dataset_RIP, the RMSE doesn't exceed 9.7 $Kbytes$ and 13.4 $Kbytes$ respectively for all the forecasted windows. It is to be noted that the minimum, maximum and mean received throughput are 1.32 $Kbytes$, 133 $Kbytes$ and 75.04 $Kbytes$ respectively. Comparatively, based only on the received power (dataset_Rx_power), poor predictions are obtained; an RMSE of 23.5 $Kbytes$ is observed for a window forecast of $1\delta t$. Dataset_all that combines all the features, gives pretty similar results to the ones obtained with dataset_SNR. This is explained by the presence of SNR measurements in dataset_all. Therefore, this remarkable decrease of RMSE with the datasets containing linear generated noise levels is essentially due to the absence of randomness in noise level variation. In other words, the lower observed values of RMSE in the linear testbed are essentially due to the consecutive noise level variation in the training set for the model. Hence, It is due to the randomness in noise level variation that high RMSE is obtained, as expected.

In summary, better estimations are produced for larger forecast windows especially with radio metrics when using RF and LR as an underlying MLTs. Furthermore, the higher RMSE observed for small time granularities is mainly due to the abrupt noise level variation, as better performance are induced when a linear noise is introduced in the environment. On the other hand, improvement have been remarked mainly for estimation based on UE 1 datasets from both scenarios 2 and 3. It is to be noted that the first UE 1 is in line-of sight with the noise generator. Thus, its related metrics are highly variable at low and large scale to mitigate the negative effect of noise. The MLTs were therefore able to learn a pattern from this variations that impacted the throughput at large scale. The positive point is that higher window forecasts in complex environment such as scenario 4 doesn't necessarily impact the estimations.

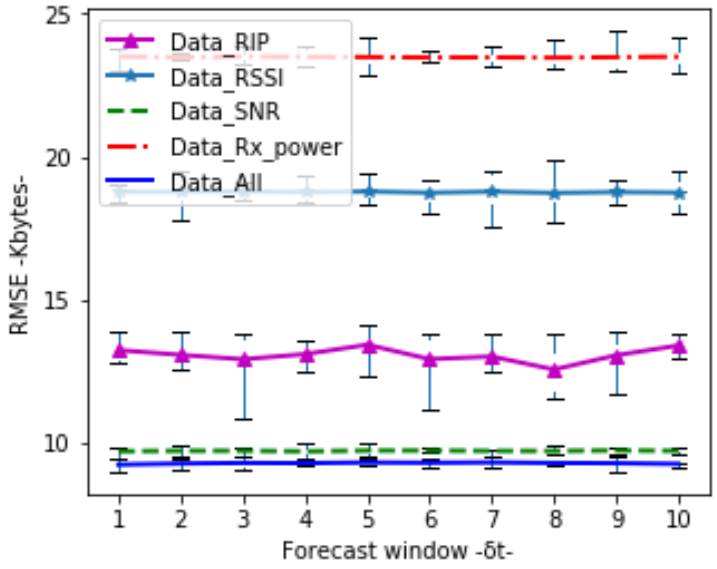


Figure 3.11 – Impact of noise level variation.

3.5.5 Lag window impact

The metrics varies over the test time duration, in this part we analyse the impact of their small scale variation on throughput estimation. For that, fig 3.12 plots the observed RMSE with all datasets and MLTs as a function of the lag window size for a forecast window of $1\delta t$. The main tested lags are $1\delta t$, $3\delta t$, $5\delta t$, $7\delta t$, $9\delta t$ and $10\delta t$.

It is remarked that introducing the past radio measurements in the model with an underlying LR MLT improves slightly the performance for datasets with only radio metrics (i.e. testbed1 datasets), except the estimations based dataset_SNR that remain at same score for all the lag sizes. With SVR, the model performs better when only the past measurements of $1\delta t$ or $9\delta t$ are introduced. The RMSE is reduced by approximately 50%, i.e. 20 KBytes. For other windows, the RMSE is around 30 Kbytes. Contrarily, with RF, the estimation performance based on *dataset_All* and *dataset_RIP* degrades with larger lag windows.

For testbed 2, the historical throughput is also introduced as input argument for larger lag size. Interestingly, the estimations are insensitive to the lag window size with both LR and SVR. Yet, with RF the estimations are improved mainly with lag size of $1\delta t$ and $3\delta t$. Then, larger windows don't influence the estimation performance.

Fig 3.13 plots in 3D the RMSE variation over various lag size and window forecast for estimations based on both dataset_3_1 and dataset_SNR. It is remarked that even for higher window forecast, the large lag windows don't necessarily enhance the estimations for both estimation cases. Quite similar observations are concluded when using other datasets (see Annex A).

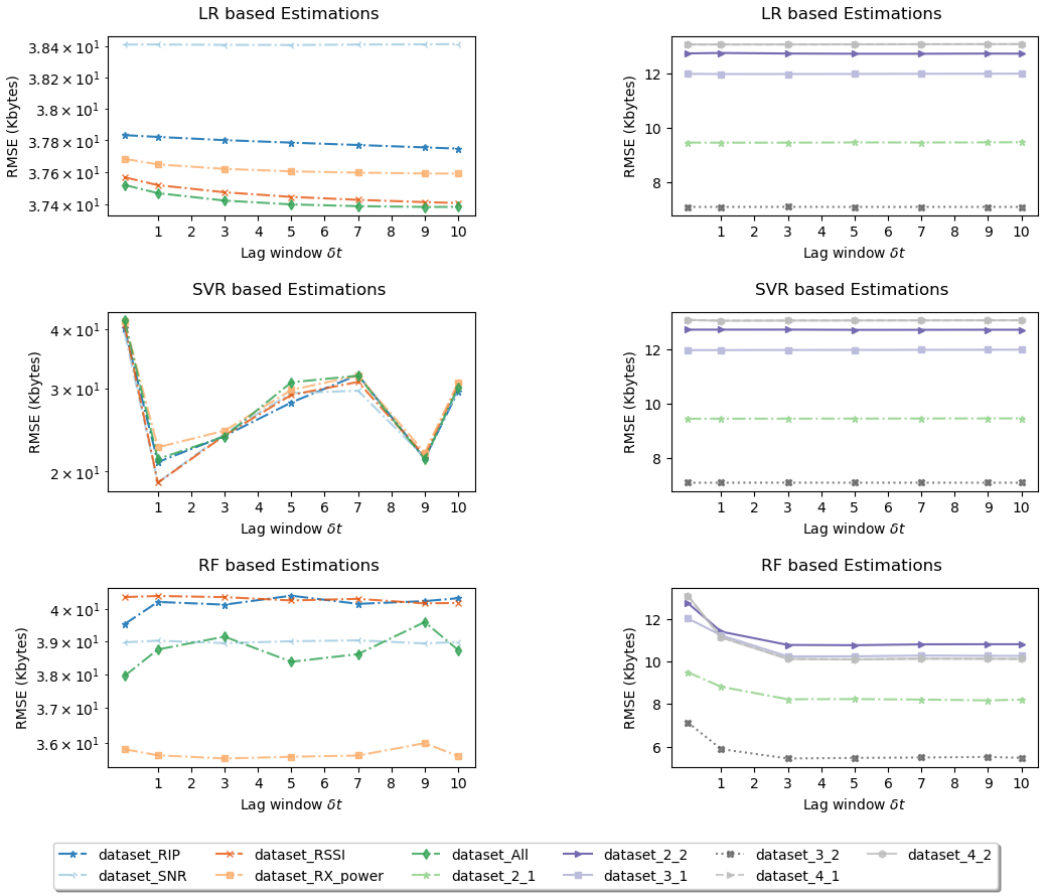
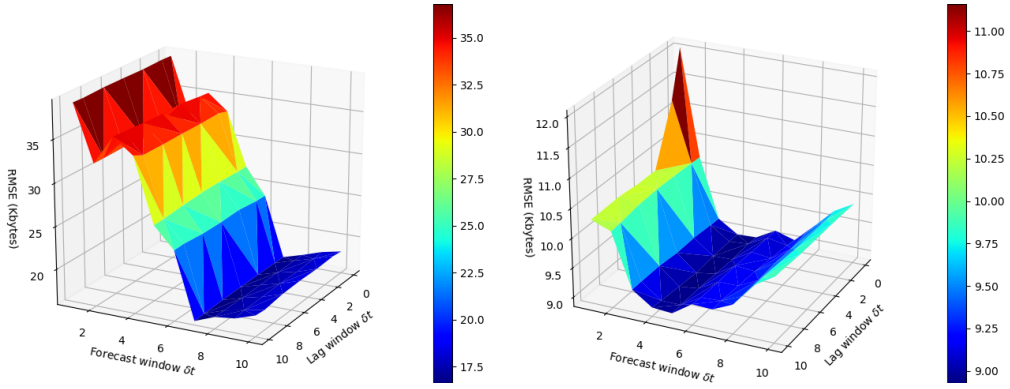


Figure 3.12 – RMSE as a function of the lag window for all datasets of both testbeds with a forecast window of $1\delta t$. Left sub-figures represent the testbed 1 datasets, and right sub-figures plots the testbed 2 datasets.

Overall, the lag window impact depends on the used MLT and dataset. In fact, the estimation performance is improved when using RF or LR as underlying MLT mainly with testbed 2 datasets and some datasets generated from testbed 1 respectively. It increases on the other hand the model complexity when introduced, i.e. time consumption.

3.5.6 Training Time (TT)

Fig 3.14 plots the training time for each dataset with the different MLTs as a function of the forecast window. Over all the tested MLTS, LR is the only MLT with very low TT. It ranges between $8 * 10^{-4}$ s and 10^{-3} s for scenario 1 datasets. For the other scenarios, the TT is between 10^{-3} s and 10^{-2} s. It is in fact expected as LR is non-parametric and so it doesn't implement any hyper-parameter tuning technique (i.e. RandomizedSearchCV).



(a) Estimations Based dataset_SNR. (b) Estimations Based dataset_3_1.

Figure 3.13 – 3D plot for RMSE as a function of lag and forecast windows for estimations based on both dataset_SNR and dataset_3_1 with RF as the underlying MLT.

The RF and SVR TT are in the order of seconds. Nevertheless, RF outperforms SVR with respect to TT. Particularly, for scenario 1 datasets, the RF TT is less than 20 s, whereas, the SVR TT is between 10^2 s and 10^3 s.

Both dataset_RSSI and dataset_Rx_power have quite similar TT over the different forecast windows when using LR. Dataset_RIP training lasts too long compared to the other datasets. And, small TT variation is observed over the forecast windows with dataset_All, dataset_SNR. With SVR and RF, it is dataset_all that lasts longer in the training phase. It is normal, as this dataset contains all the radio measurements, i.e. 15 features. The TT increases for the window forecast of $2\delta t$ with the other datasets and it then varies in a small interval range.

Interestingly, although the testbed 1 datasets are larger than the ones of testbed 2, i.e. total samples about 15300 (testbed 1) and 4000 samples (testbed 2), the LR and RF TT for estimations based testbed 2 dataset are higher than the ones of the first testbed. The TT is always high when estimating throughput based on dataset_3_2 with all MLTs. The generated datasets in scenario 4 are trained in less time with both RF and LR, i.e. 20 s (RF) and 10^{-3} (LR). Overall, the estimation model TT is insensitive to the forecast window when using SVR and RF, quite similar TT is observed over the various forecast windows.

3.5.7 Estimation Time (ET)

Once the model is trained, it is tested on one fold from K2 folds of the outer loop CV, referred as test fold. The time consumed for test fold estimation of each dataset is represented on fig 3.15. As the TT, the LR has marked lower ET for all datasets. It is centred around 10^{-4} s for testbed 1 datasets and between $10^{-4} - 10^{-3}$ for the testbed 2 datasets. For all the datasets, a small ET variation is observed over the various window forecats. Particularly, the ET for the throughput estimation using RF is around 10 s for all datasets. This result highlights the insensitivity of the

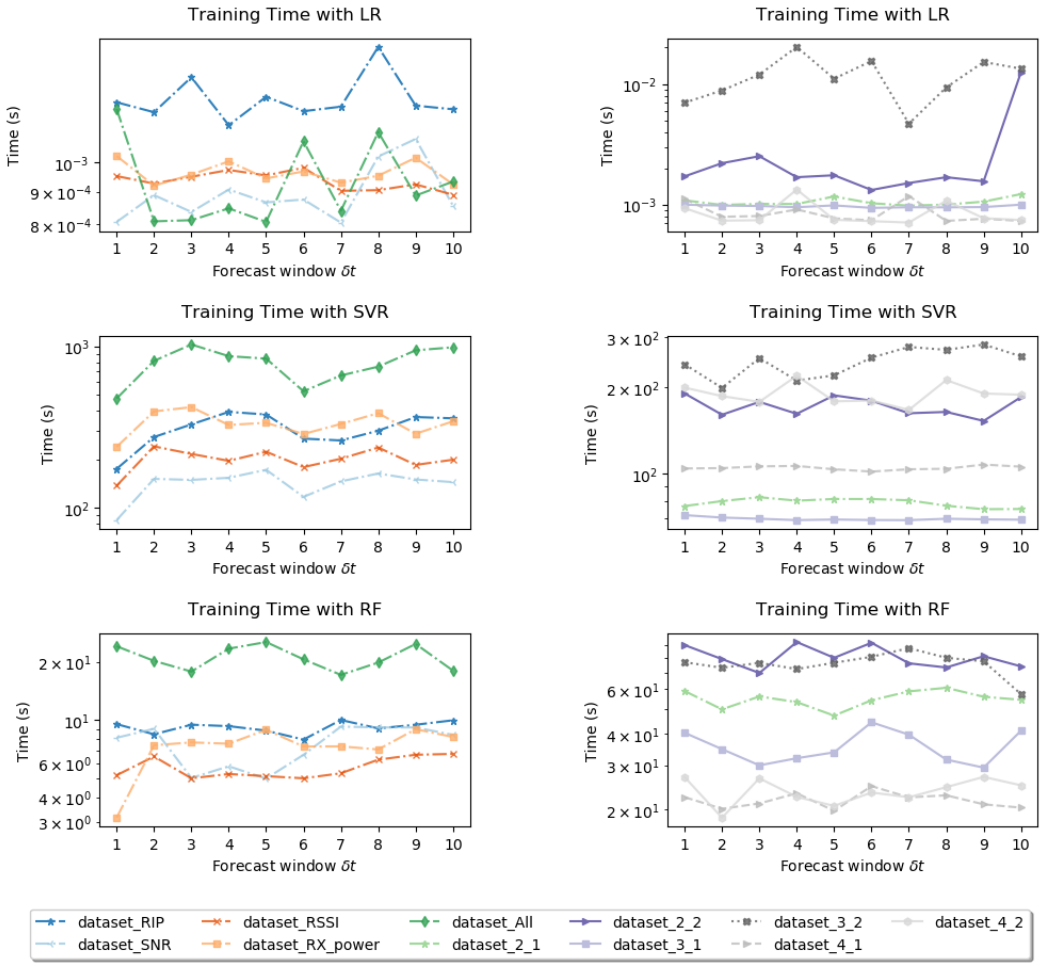


Figure 3.14 – Training time (s) with the different MLTs as a function of the forecast window for all datasets of both testbeds. Left sub-figures represent the testbed 1 datasets, and right sub-figures plots the testbed 2 datasets

model to forecast window with respect to ET.

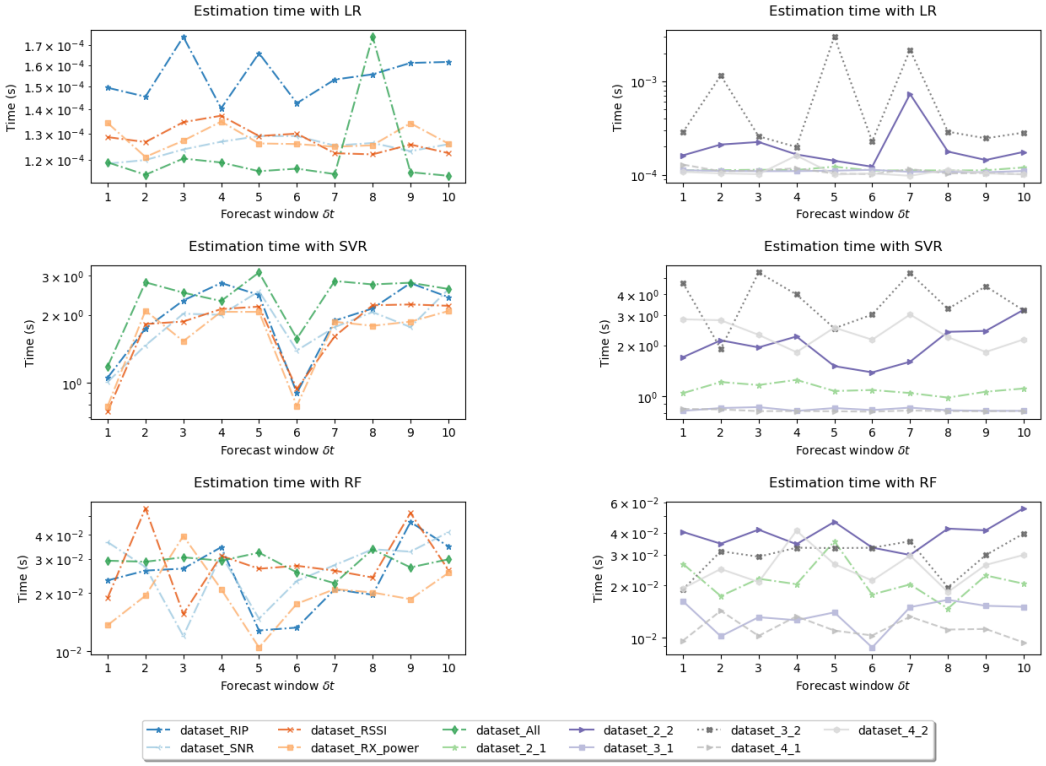


Figure 3.15 – Estimation time (s) with the different MLTs as a function of the forecast window for all datasets of both testbeds. Left sub-figures represent the testbed 1 datasets, and right sub-figures plots the testbed 2 datasets

3.6 Conclusion

In this chapter, we have investigated the possibility of users instantaneous uplink throughput estimation in cellular network based on the lower layer metrics. For that, real time 4G testbeds are deployed in an anechoic room where the radio phenomena are controlled. This allows a clear analysis of encountered behaviors in throughput variation. Several radio phenomena are introduced. The radio phenomena are chosen as the most encountered in real environment. Notably, noise, multipath fading and radio congestion. The testbed complexity is increased gradually on each radio scenario. Then, exhaustive benchmark of lower layers measurements/metrics is performed to build datasets reflecting the network response to radio environment variation. Some datasets contain only the radio measurements, while others include almost all metrics incorporated in the PHY and MAC layers mechanisms to mitigate the radio conditions and UEs scheduling.

With the deployed testbed, the smallest granularity that could have been used is 100 ms. It is limited by the used traffic generator. Therefore, the developed estimation model in chapter 2 is used with a time granularity of 100 ms. That is, the smallest forecast window for the instantaneous throughput estimation is fixed

to 100 ms. The three MLTs, LR, SVR and RF are used as an underlying MLT for the estimation model. The analysis of the received throughput exhibit the strong link between throughput and radio environment variation.

The DL approaches results for the average throughput estimation have shown that the combination of RSRP, RSRQ and other radio metrics are sufficient to have accurate DL estimations for a forecast window of 1 s. Due to the non-presence of these metrics in uplink measurements, we have fulfilled a benchmark of the relevant radio measurements at the eNB to estimate the instantaneous UEs uplink throughput. After applying different MLTs, it is conducted that radio metrics only are not sufficient for small time granularities estimation, i.e. at the order of 100 ms. Nevertheless, they lead to better performance for higher forecasts more than 700 ms, or in simple environments with only a linear noise variation. With that, contrarily to the downlink average throughput estimations for 1 s, we are able to achieve accurate estimations for UEs instantaneous uplink throughput for forecast windows of 700 ms, 800 ms, 900 ms and 1 s when using one of the uplink radio metrics (i.e. SNR, RIP, RSSI or Rx_power). These results are achieved with both LR and RF machine learning techniques. For estimation at very small time granularities, i.e. less than 700 ms, the radio metrics solely are insufficient. Accordingly, exhaustive eNB lower layer metrics collection is fulfilled. With the 43 metrics collected from the eNB lower layers, accurate estimations are produced for very small time granularities (i.e. at order of 100 ms) by the three MLTs in different radio environments.

An investigation of the impact of historical measurements have been also conducted. A slight improvement is realized when adding a lag window of 100 ms, especially with RF. Otherwise, it doesn't really have any impact on the estimation other than increasing the model complexity. Interestingly, the estimation model with LR as underlying MLT is able to offer quite similar estimations at some bytes of difference as the RF and SVR in much lower time compared to the two techniques. The SVR is the slowest model for all the tested cases. This is interesting as only LR can be used for quick convergence times. In this work, we have proved the possibility to reach accurate estimations for the users instantaneous uplink throughput at small time granularities. However, the extension of these results to the real complex environment remains an open issue.

Chapter 4

5G RAN Slicing Enforcement

Contents

4.1 Introduction

The tremendous growth of services and/or applications demand is increasing over the years. The diversity of such services results in several QoS (Quality of Service) requirements to be fulfilled by the serving network. 3GPP and other organizations aim to support this variety of services requirements through 5G system.

For that, an agile architecture is required. This is expected to be achieved through the network slicing (NS), where the MNO shares its infrastructure with different stakeholders/third parties, called tenants or slices. Even though, the slicing is not a novel idea, its deployment in the 5G context remains challenging especially that it is hard to achieve it at the radio part. The RAN is characterized by scarce radio resources and chaotic radio channel. Moreover, it is expected that the 5G embodies diverse and multiple cells forming an heterogeneous network. It arises then a high inter-cell interference level. Thus, it limits the 5G objectives in term of QoS guarantee. Therefore, the MNO is compelled to manage efficiently the rare resources while mitigating at best the negative impact of the radio phenomena. Regarding the interference, the 5G intends to deploy advanced transmission techniques, such as the Cooperative Multi-Point (CoMP) schemes. The latter involves a tight cooperation between BSs (base stations), mainly at the resource level to take advantage and/or mitigate the interference effect.

Consequently, in addition to the global NS requirements such as the slices isolation, the RAN slicing shall ease the way for the 5G radio techniques deployment. In this work, we focus on the RAN slicing enforcement from resource perspective in a multi-BS 5G context.

Therefore, this chapter introduces the network slicing with its deployment limitations in the 5G context. Then, the focus is turned on the most challenging part, i.e. RAN slicing at the radio resources level. The 5G enables more flexibility at this level. The 5G resources structures are then exhibited. Further, the RAN slicing requirements are highlighted and the ongoing research study is over-viewed. Few works are fulfilled for the resource slicing at a multi-BS prospect with respect to the inter-cell interference. From that, the RAN slicing problem is formulated as a multi-objective optimization problem (MOOP) for radio resources allocation. Each objective seeks the assurance of particular RAN slicing requirement. Particularly, one objective targets an allocation easing the 5G advanced techniques integration. Whereas, the other objective assures the RAN scalability. A non pareto approach is followed to approximate the optimal model for this MOOP. Based on simulation, it is revealed that such models are limited for real time RAN slicing deployments whereas it is required for the future of the 5G orchestration. Therefore, heuristics are proposed to assure at best the RAN slicing requirements. Later, simulations are performed to evaluate the different algorithms with respect to some predefined performance metrics.

4.2 5G system and deployment challenges

The 5G system is argued to be revolutionary compared to the previous generations. Other than its objective of reaching a higher throughput and utlra low latency, 5G aims to serve diverse UEs with diverging performance requirements. This is expected to be achieved through a service based architecture. The latter is based on the network slicing paradigm. Therefore in this section, this approach is outlined in the 5G context with an overview of its main principles. Then, the proposed architecture for its embodiment is depicted with a discussion about its potential challenges.

4.2.1 5G Network slicing (NS)

A range of vertical industries are innovating with an underlying hypothesis of a reliable wireless connectivity, including automotive and healthcare. Their services/applications cope with different use cases, each one being characterized by a specific QoS demand. Thus, it implies a diverging set of performance and service requirements.

On the other hand, the existing architecture in cellular networks "one-size-fits-all" is unable to address such services diversity. The later is designed with a limited pre-defined QoS classes based on a monolithic network. 5G is expected to differentiate between the diverse QoS needs and open up for advanced innovative opportunities. An approach to reach this aim is through a service-based architecture, where the network is tailored exactly to achieve the service performance demands. In other words, an architecture that slices the network on per-service basis is envisioned, called also "end-to-end network slicing". The latter is defined by 3GPP as "A logical network that provides specific network capabilities and network characteristic". With this vision, each MNO physical infrastructure is shared between several tenants as slices. Each slice has a business service with certain quality of service (QoS) requirements. The tenant could refer to the vertical segment (e.g. automotive industry), application provider (e.g. YouTube), as well as the virtual MNO (VMNO).

During the UE registration, the UE uses a slice identifier to request a connection to a specific slice. Currently, 3GPP has standardized three slice/service types (SST): Enhanced Mobile Broadband (eMBB), ultra-reliable low latency (URLLC) and massive internet of things (mIoT). This differentiation between slices is principally due to the diversity of slices demand, e.g. eMBB slices' services require higher throughput than IoT services. Nevertheless, the slice generation is not limited by the three standardized SSTs, i.e. a slice might be created based on a given SLA (Service-Level Agreement) with specific QoS needs. Yet, its creation and management are constrained with the predefined NS basis, as explained in the following section.

4.2.2 Network slicing principles

In order to achieve the service based architecture, the NS comes with different principles. These principles aim to support efficiently the various users QoS. The key requirements are summarized in this paragraph.

- Isolation: it must be sustained over the different system levels, i.e. resources, network and service levels. For instance, the outage performance of one slice, i.e. congestion, attack or QoS degradation, should not impact negatively the other available slices in the network. Moreover, the slice performance must be guaranteed even in the existence case of other slices with conflicting performance requirements.

- Elasticity: it refers to the ability to decrease/increase the slice allocated resources as well as modifying the network functions. In fact, the radio, network conditions and served users are changing over time. The MNO needs to assure the slice SLA under these variations. Particularly, if the slice users increase in a given geographical area due to users mobility, the initial allocated resources must be scaled up to fulfill the slice service requirements. Nevertheless, it should be performed without affecting the other served slices, i.e. isolation.

- Customization: it refers to the efficient utilization of the allocated resources by each slice. Although it concerns all the network resources, it is more critical for the rare radio resources.

- Automation: this requirement characterizes the dynamic creation of slices over time. The slices configuration is performed on-demand and on-the fly without any manual intervention.

- Programmability: it gives the tenants the ability to control and manage their slices allocated resources. Thus, it enables the slice owner to customize its resources. This can be achieved through an open API exposing the network capabilities.

Although the NS is not a novel technology (e.g. cloud computing), but its e2e deployment for 5G system needs to be adapted. The next section outlines the generic architecture for its enforcement, pointing out the relevant challenges.

4.2.3 Network slicing concept and challenges

The 5G is advocated to respond to the third parties needs in terms of scalability, availability and performance demand. Network slicing is figured to be the key enabler for such objective. In order to realize this approach in the 5G context with the aforementioned principles, the system framework is subdivided into three layers, infrastructure, network and service layers.

- Service layer: Each service instance points to the service provided by the network tenants. This layer deals with its description. Then, it copes with the mapping between each service description and the e2e infrastructural/ function elements to realize the service demands. Each slice is modeled as a set of network services connected with each others. The slice management and orchestration (MANO) is of essence in this layer. Particularly, the MANO[1] framework deploys a resource orchestrator for the different virtualized resources and a network service orchestrator for the life-cycle management of network services. With that, the slices automation is enabled. In other words, the slice is created on the fly, and the MANO assures its life-cycle with respect to the network resources.

- Network layer: Once the service is described and the network functions are selected, the network layer configures and places the related functions on the infrastructure. It should offer the e2e service requirements when chained together. The enabling technologies for the deployment/management of network functions have attracted the research interest for this layer setup. Virtualization and softwarization based solutions are the most nominated technologies, mainly NFV (Network Function Virtualization) and SDN (Software Defined Network) based solutions. The former allows flexibility of NFs via virtualization, and the latter separates the control from the user data functions with a centralized controller. Several architectures based on NFV and SDN have been proposed to address the Core (CN) [Qazi 2017] and (RAN) [Foukas 2017] slicing. The SDN implementation at the RAN part is referred by SD-RAN (Software Defined RAN). Both NFV and SDN/SD-RAN bring flexibility, scalability and service-oriented adaptation. These characteristics are crucial for the network slicing implementation. However, their deployment is challenging, especially in the RAN part due to the shared and rare physical resources. The network functions granularity is also exposed to be critical for this layer deployment. Existing work proposes either a coarse or fine grained functions. The coarse network function handles a large task of network operations. Contrarily, the fine grained approach subdivides each network entity into functions, which in turn are decomposed into sub-functional entities until achieving a fine grained functions. The first approach comes with less flexibility and adaptability to the network change. Whereas, the second concept offers a more flexible and adaptive way for the network functions management at the expense of interfaces explosion and network functions chaining. This trade-off resolution depends also on the infrastructure layer flexibility level as discussed next.

- Infrastructure/resource layer: it refers to the physical infrastructure with its control and management mechanisms. It spans both the CN (core network) and RAN. Mainly two research axis for this layer are evolving: the infras-

[1]ETSI GS NFV-INF 001, "Network Functions Virtualisation (NFV). Infrastructure Overview.

tructure architecture and its virtualization. With respect to the 5G high flexibility, it is assumed to have access to the Infrastructure as a Service (IaaS). Even though this is not a novel concept (e.g. cloud computing), its enforcement in the 5G context involves adjustment. The central cloud architecture [Zhou 2016] and mix between central and edge computing infrastructures [Rost 2016] are proposed for the CN deployment. Also, the core network virtualization can be extended easily from the cloud computing existing work. Nonetheless, the 5G RAN architecture and virtualization is still at its infancy phase. In fact, the 5G RAN is characterized by the coexistence of different RAT (Radio Access Technologies), including 3GPP and non-3GPP technologies (e.g. LTE, WiFi). Few research proposals for the RAN architecture [Zhou 2016, Akyildiz 2015] nominate a generic software-defined base stations, with a central baseband processing units and remote radio heads. Regarding the RAN virtualization, the VM and container based solution proposed for cloud computing are restricted, as another dimension is added, i.e. radio resources (spectrum and radio hardware). The scarcity of these resources type ensues the infeasibility of resources over-provisioning. Existing work dealing with this virtualization level can be categorized into two models: dedicated and dynamic shared model. With the dedicated resource model, each tenant leases a specified part of wireless resources. Thus, this static fragmentation of radio resources offers a high level of isolation, but it lacks in terms of slice flexibility, scalability and multiplexing gains. Therefore, it falls in the inefficient utilization of the rare radio resources. Namely, the slice demand varies in time. The allocation of static resources amount violates the NS principles, i.e. customization and programmability. Alternatively, the dynamic shared model emphasises the radio resources sharing between all the slices through a common underlying physical and lower MAC layers. Even though, the resource utilization is more efficient than the dedicated model, it doesn't respect the required isolation at the wireless resource level. Thus, sophisticated allocation strategies enabling the resource isolation with high customization level are required for the RAN slicing deployment.

Overall, the RAN slicing is a pivotal element for the e2e network slicing deployment. However, its enforcement is challenging. This is due to the scarce radio resources and the uncontrolled radio channel. In fact, the perceived UE QoS depends on both the amount of allocated resources and the channel condition. The allocation of same resources amount to users experiencing different radio phenomena results in different QoS for each UE. Hence, an assignment of static resources part to a slice (i.e. dedicated model) doesn't guarantee the users slice satisfaction. Thus, there is a need for a dynamic resource assignment. The dynamic shared model limits the network flexibility, as only the coarse network functions can be used. Therefore, a dynamic dedicated model is required. In other words, a model with dynamic resources slice assignment, i.e. slice resources assignment varies with the slice demand, as well as the slice resources isolation. Each resource is occupied

by maximum one slice at a time. With such allocation strategy, a high level of flexibility can be achieved enabling the fine grained network functions. Moreover, the NS principles are respected.

In this work, we are positioned at the infrastructure layer. The objective is to enforce the RAN slicing from a wireless resource perspective as it is still at early stage. For that, a dynamic dedicated resource approach is intended. In the following, the RAN resources slicing is explored in more details to enfold our contribution.

4.3 RAN slicing

The network slicing is conceived for an end-to-end deployment. It then copes with the RAN as well as the CN. In the previous part, it is concluded that the wireless resources are the main obstacle toward the e2e slicing achievement with respect to all NS requirements. In fact, the wireless resources allocation has to meet the service requirements for each slice regardless the channel or network conditions, while efficiently using the scarce available resources (i.e. customization and programmability NS requirements). Moreover, the isolation between slices should be maintained. The proposed models for radio resources management, i.e. dedicated and dynamic shared resources models, address only a part of the NS principles. The dynamic shared model offers resources customization and programmability without isolation, whereas the dedicated model puts forward the isolation at the expense of the other NS principles. Thus, it manifests the need for a dynamic dedicated resources management model to enforce the e2e slicing objective. As it concerns the wireless radio resources, let overview the 5G radio resources (spectrum) forms in this section. The 5G system is also expected to enable some advanced techniques to achieve its objectives through bad channel conditions mitigation, e.g. interference. The developed techniques, as discussed in the following, deal with the wireless resources. Accordingly, another dimension is added for the RAN slicing enforcement, i.e. it should allow the deployment of these advanced techniques for each slice. Therefore, the main RAN slicing requirements to accomplish the global vision of the e2e 5G network slicing are summarized. Later, the RAN slicing ongoing research is discussed.

4.3.1 5G radio resources

In 5G system, the physical layer is more flexible with respect to the previous generations. Recall that radio resources in 4G are uniformly distributed over a time-frequency grid, i.e. the later is decomposed in physical resource blocks (PRB) of 1 ms over 12 sub-carriers spaced by 15 kHz as illustrated in fig 4.1 (a). The PRB is of size 180 kHz* 1 ms.

In order to fulfill the variety of services requirements, increase the network reliability and adapt to frequency range, 5G introduces different radio frames numerologies for sub-6 GHz and above-6 GHz bands. Especially, table 4.1 exhibits the

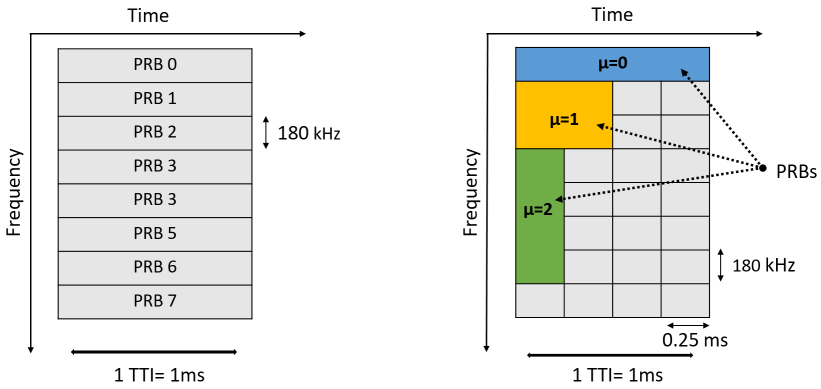


Figure 4.1 – Comparison between 4G and 5G radio resources: (a) illustrates the 4G resources, (b) schematizes 5G resource structures.

different numerologies for sub-6 GHz bands. Each given numerology μ [2] defines the time-frequency resource size in one Transmission Time Interval (TTI), TTI=1ms. That is, a numerology μ refers to the sub-carrier spacing (SCS) in frequency domain and the slot duration in time domain. For instance, as depicted in fig 4.2, for $\mu = 1$ the radio resource size is fixed to 0.5 ms over 12 sub-carriers spaced by 30 kHz. And, for $\mu = 2$, 60 kHz is chosen for the SCS and 0.25 ms for the slot time duration. In general, the SCS scales by $2^{\mu} * 15kHz$ and the slot duration decreases with higher numerology (μ). With that, the 5G radio resources have different shapes, as shown on fig 4.1 (b). Such flexibility is essentially introduced as to achieve the diverse services requirements. For example, it is preferable to transmit URLCC services, that are latency sensitive, in shorter time interval with larger sub-carrier spacing, e.g. $\mu = 3$.

Table 4.1 – 5G Radio frames numerologies for sub-6 GHz bands

μ	SCS (kHz)	Slot time duration (ms)
0	15	1
1	30	0.5
2	60	0.25

In order to support the coexistence of the multiple numerologies on the same carrier, the resources are structured in the so-called tiles [Elayoubi 2019, Pedersen 2016]. The tile is the smallest subset of frequency and time resources allocated to a particular slice/service with same numerology μ. Hence, for sub-6 GHz three tiles structures are tailored as shown in fig 4.2. For instance, the tile

[2] 3GPP, TR 38.802, TR 38.804: Study on new radio access technology Physical layer aspects, Study on new radio access technology Radio interface protocol aspects. `https://www.3gpp.org/`

structure for $\mu = 0$ is 1 ms over 12 sub-carriers spaced by 15 kHz. Further, multiplexing over time and frequency is required for the transmission of the different numerologies, e.g. over time 3GPP imposes symbol alignment between tiles to insure orthogonality. Furthermore, same as LTE, each 5G frame is 10 ms and each subframe is 1 ms, i.e. 1 frame contains 10 subframes.

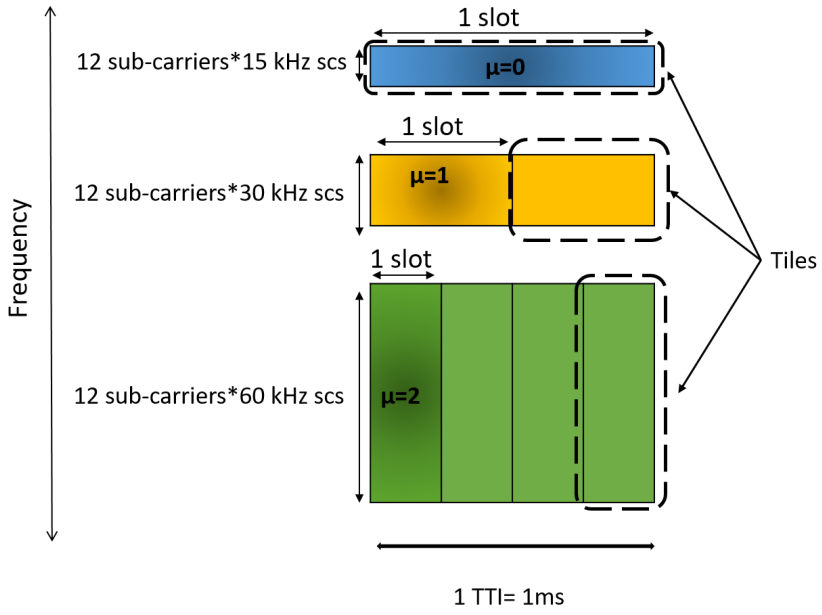


Figure 4.2 – Sub-6 GHz numerologies

4.3.1.1 5G advanced radio techniques

In cellular networks, a cell is defined as a set of collocated antennas serving a geographical area sector. The adjacency of these cells determines the MNO coverage. With the transmission power variance between cells, the inter-cell interference (ICI) is highly observed between the adjacent cells. The ICI lowers the network average data-rate. This leads to a degradation of the network performance.

Accordingly, advanced techniques to mitigate this phenomenon are designed. They include Inter-base station power control (IBSPC), Coordinated multi-point (CoMP) with different schemes (e.g. Coordinated scheduling/coordinated beamforming (CS/CB), Joint Transmission (JT)) [Hossain 2014, Lee 2012, Sawahashi 2010]. These techniques are based on a tight cooperation between the cells to mitigate/exploit the inter-cell interference for the MNO benefits. Particularly, the JT-CoMP is based on simultaneous transmission of data to a UE from multiple cooperating transmission points (TPs). It has the advantage of exploiting the ICI by converting the interfering signal to a required one. On the other hand, the IBSPC techniques target the ICI mitigation through coordination based control power approaches. Notably, the latter technique imposes some limitations on the

usage of the different resources. It might forbid the transmission of data in a cell on specific resource or limit the transmission power on certain resources.

Overall, a tight cooperation between cells is required to deploy such advanced techniques. This cooperation enforcement is indeed challenging. A time synchronization between the TPs is intentional for such prospect, as well as the UE allocated radio resources.

In the context of 5G NS, the deployment of such techniques is more though than ever as the MNO resources are sliced between the tenants. The tenant should implement the relevant cooperation technique on its resources independently from the other tenants (i.e. slices). Therefore, the advanced techniques are envisioned for deployment at slice level instead of the global MNO level.

Consequently, another requirement is added for the RAN slicing. It concerns the development of a slicing policy with respect to the 5G radio techniques. In the following the RAN slicing requirements are outlined.

4.3.2 RAN slicing: requirements

Considering the NS requirements, the scarcity of the radio resources and the envisioned RAN technologies to reach the 5G objectives, the RAN slicing requirements regarding the radio resources can be formulated and summarized as follows:

- Orthogonality (resource isolation): It must be guaranteed between slices. Each radio resource, in terms of time and frequency, must be allocated to only one slice to avoid interference, thus, ensuring the slice isolation at the radio resource level.

- Satisfaction: it is based on the slice performance requirements, the traffic demand and the channel/network conditions, the amount of the slice required resources should be decided. This is known by resource slicing policy. Therefore, each slice has to be allocated the amount of assigned resources based on the slicing policy. Consider the example of a slicing policy expressed by the amount of tiles with a specific numerology, if a given slice is assigned 20 tiles with $\mu = 2$, it should receive approximately the 20 tiles with same numerology during the allocation process, without excess. This way the slice demand is satisfied. Furthermore, each slice uses fully its resources, i.e. it assures the slices resource customization.

- Scalability: the MNO should be able to scale up/down the slice allocated resources with respect to the network conditions and slice demand variation. One example includes users of a given slice experiencing a bad channel condition, the slicing policy might decide to increase the amount of the slice allocated resources to overcome the negative impact of the radio phenomena. Moreover, as the slices are created dynamically and on-demand, the radio resource model should allow the MNO to serve new slices requests. This can be achieved through the reuse of the unallocated resources during the allocation window. This requirement implicitly assures the NS programmability.

- Cooperation enabling: the 5G advanced radio techniques involve a tight cooperation between the base stations (i.e. gNBs in case of 5G) to achieve their objective. As the RAN slicing imposes the isolation between the slices resources, the activation of the appropriate technology is based on the slice performance requirements and SLA (i.e. programmability aspect). The slices radio resources allocation should therefore ease the deployment of these advanced technologies for each tenant, e.g. beamforming, IBSPC, CoMP.

4.3.3 RAN slicing: state of art

In the context of RAN slicing, resource management and orchestration have received significant interest from the research community. Many frameworks [Devlic 2017, Ferrus 2018, Ksentini 2017, Rost 2017, Ordonez-Lucena 2017] have been proposed to deal with the high level wireless resource orchestration and management, where virtualization/abstraction of wireless resources is performed. While the proposed approaches are effective in resource control and orchestration, they might lack effectiveness for fine grained control scenarios, where performing and enabling advanced 5G transmission techniques are required. Also, a major challenge with these frameworks is the efficient resource allocation while preserving the radio resources isolation and NS requirements.

For that, the resources slicing has attracted academia and industry researchers. The proposed contributions are established either from one BS or multi-BS basis. In the following, we review the most relevant proposals.

4.3.3.1 One BS perspective

The static segmentation of the available resources between the served slices is argued to be limited. Even though, this allocation offers a high isolation level, the other RAN/NS requirements are not satisfied. For that, researchers have been focused on enhancing the existing 4G schedulers or proposing new scheduler schemes for the RAN slicing. Consequently, the resources are shared between slices. The authors in [Yan 2019] propose the integration of deep learning and reinforcement learning to schedule and allocate the slices resources to each slice with respect to only the isolation at a performance level. A MAC scheduler is proposed by S.Mandelli et al.[Mandelli 2019] to achieve each slice targeted throughput. This is implemented with a counter tracker for the aggregated rate and resource allocations. B. Han et al. [B. Han 2018] propose the use of Genetic algorithm to optimize resource management between heterogeneous slices with maximized long-term network utility. The authors in [Papa 2019] tackle the RAN slicing with a Lyapunov approach. They target resource usage minimization with QoS accounting, principally delay and throughput. The isolation is supported from a QoS level, ensuring that each slice throughput doesn't exceed a fixed maximum throughput. The users of different slices share the radio resources. Although the proposed methods gain in terms of multiplexing, they lack of programmability and resources isolation aspects,

that allow each tenant to manage its resources independently.

Moreover, resources sharing through virtualization has been also proposed in several works. The authors in [Kokku 2012] propose the Network Virtualization Substrate (NVS) based on WiMaX systems. It consists of a two step-process based allocation. Firstly, the tenant maximizing the MNO utility function (i.e. MNO revenue) is chosen to be served. Then, each tenant customizes its resources. This way, the slice selection is decoupled from the flow selection. It allows then a high level of customization for the slice owner, i.e. satisfaction requirement. With such level of virtualization, the isolation is enabled only from a packet-level. Thus, the programmability required by NS is limited at the packet-level. Chang et al. [Chang 2018] propose a 5G partitioning algorithm that maximizes the percentage of satisfied slices while allocating the minimum resources to each slice. For that, the problem is formulated with knapsack approach by mean of virtualization. The slices demand is expressed by the number of PRB and allocation type, e.g. contiguous or non-contiguous resources. The slice resources are then pooled, i.e. virtualization process. As a second step, the slices numerology type is chosen as to solve the knapsack optimization problem, leaving the largest unallocated rectangle in the resource grid. The resource-level isolation and customization is reached by this approach. Nevertheless, the authors assume the flexibility of the slice numerology to achieve their objective. Such flexibility is not always possible. Notably, a numerology $\mu = 2$ is tailored to serve at best a slice with URLLC traffic, and the change of this numerology to maximize the served slices and increase customization come at the expense of the slice performance.

Further, all of the above-mentioned contributions consider a network with only one BS, which limits the deployment of their approaches in a multi-BS network, where each tenant requires a different amount of resources on each BS, based on the channel condition and the number of connected users. Moreover, the inter-cell interference is not addressed, i.e. cooperation enabling requirement.

4.3.3.2 Multi-BS perspective

Few work has been done in the context of resources allocation in multi-BS system. Authors in [Mahindra 2013] developed NetShare, a framework for network resources management. NetShare is designed with a centralized gateway deciding the amount of resources to be allocated to each tenant over each BS. The gateways assure the MNO resources customization and isolation at network level. Then each BS implements the resources scheduler as proposed by the NVS [Kokku 2012]. Therefore, the system ensures isolation at best from packet-level. Thus, it is limited from programmability point of view. Also, interference between BSs is not taken into consideration when allocating resources. AppRAN, an application-oriented framework for sharing RAN resource is proposed in [He 2015]. The slice is considered as an application. The proposed framework defines a set of abstract applications with different QoS. Then, it maps each incoming slice demand (concrete application) to the predefined applications. The resources allocation for each application is

performed centrally with a controller. Once the controller decides of the resources amount for each slice, the slice resources allocation is executed on each BS. The proposed approach eases the high level programmability thanks to the controller, but it is limited at the resources level. All of the aforementioned works didn't take the RAN slicing cooperation enabling requirement in their approaches.

On the other hand, the contribution in [Sallent 2017] sheds light on four approaches for the radio resources management from multi-cell multi tenant perspectives. Each approach addresses the radio slicing at a level of granularity. Their study includes spectrum, ICIC (Inter Cell Interference Coordination), packet scheduling and admission control levels. The spectrum level is related to the resources allocation for each slice on each cell, the dedicated resources model is then addressed. At the ICIC level, the resources are shared between slices but their assignment takes into account the expected spatial distribution of traffic for each slice over the cells. That is, the resources assignment is performed at a multi-cell level. With the packet scheduling slicing, the resources allocation is emphasized on each cell separately. Besides, the slicing at the admission control level decides only if the slice is accepted to be served or not. All the slices flows are combined through the different resources. The authors have concluded, that the best RAN slicing approach in terms of isolation and customization is the one performed at the spectrum and ICIC levels. In contrast, the AC and packet scheduling slicing levels offers high flexibility at the expense of the isolation. Although, the fine grained resource management is covered (i.e. ICIC and spectrum levels) to mitigate inter-slice interference, they didn't propose any algorithm to enforce their approach.

D'oro et al. [D'Oro 2019] proposed an algorithm to enforce the RAN slicing policies with interference mitigation. This is enabled through guarantying that the same (or similar in time/frequency) resource blocks (RB) are assigned to the same slices when BSs are close enough to interfere among themselves. Although, their approach is efficient from interference mitigation perspective in 4G networks, it might be ineffective in resource utilization in the 5G system. In fact, contrarily to 4G system with only one fixed numerology, the variety of numerologies in 5G system involves a fine grained allocation.

4.3.4 Problem formulation

The 5G RAN slicing approach involves an enforcement to realize the envisioned 5G objectives. It rises many requirements, as indicated in section 4.3.2, mainly scalability, orthogonality, slices satisfaction and easing the inter-base stations cooperation. In the following, resources refers to the radio resources.

To ensure the scalability and satisfaction requirements, the RAN should be more flexible about the radio resources allocation. This could be achieved through an efficient resources allocation with low and high level customization. The low level customization is related to the efficient use of each slice resources, i.e. intra-slice customization. For that, each slice should be assigned the amount of resources satisfying its demand without an extravagant allocation, whereas the high level

customization is related to the inter-slice customization level, i.e. MNO efficient resources utilization. A high efficiency is attained when the MNO manages its available resources between the slices requests without any resource waste.

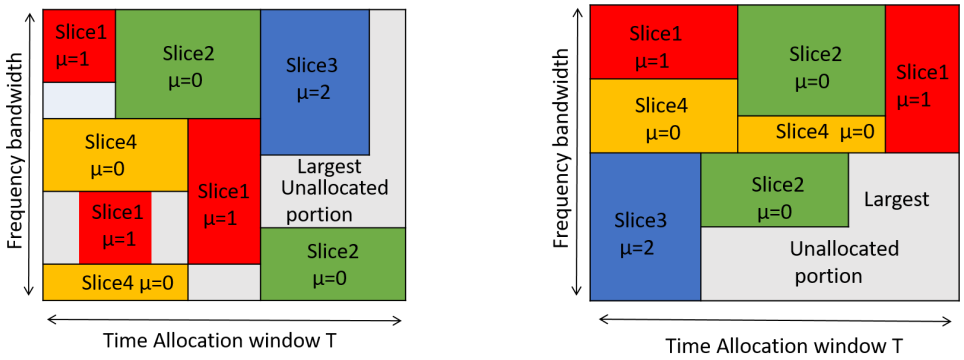
Let suppose that the MNO is able to realize the low level customization and let investigate the inter-slice allocation with the different 5G resources numerologies, as illustrated in fig. 4.3. Each slice demands a different amount of resources with specific numerology during the allocation window T, i.e. a number of tiles. Two time-frequency resource grids are schematized, (a) and (b). With the allocation on (a), a quite small sparse unallocated resources are induced over the resource grid. These scattered resources don't fit any tile structure. They are therefore considered as wasted as the MNO is unable to reuse them while scaling up another slice demand or to serve a new slice. This leads to an inefficient high-level customization. The only resources portion that could be taken for the MNO benefit is the largest continuous one. With this illustration, 2 tiles with $\mu = 0$ and 1 tile with $\mu = 2$ or 2 tiles with $\mu = 1$ and 1 tile with $\mu = 2$ can fit in this largest portion.

From that, it is clear that the optimal allocation is the one minimizing these small leaved resources, as illustrated on fig 4.3 (b). The adjacency of these small sparse unallocated resources forms a large space that could fit a tile structure. The optimal allocation would be therefore the one letting the largest continuous unallocated space of resources. It allows the MNO to further reuse the unallocated resources in an efficient manner, as different tiles structures could fit in this portion. Notably, from 4.3 (b) different tiles allocations are possible: 3 tiles with $\mu = 1$ and (2 tiles with $\mu = 2$ or 2 tiles with $\mu = 0$) or 4 tiles with $\mu = 1$ and 1 tile with $\mu = 2$ and so on.

Overall, even though both allocations (a) and (b) satisfy the four slices requests during the allocation window T, it is clear that old fashioned resource allocation strategy in (a) is sub-optimal compared to (b). The later increases scalability and resource utilization efficiency.

Accordingly, to achieve the high-level customization, the RAN slicing enforcement algorithms should allocate resources in a way leaving the largest unallocated portion of resources, instead of sparse small unallocated resources.

On the other hand, in 5G context, each geographical area is covered by different cells types. Thus, high level of interference is expected. It includes the inter-slice as well as the intra-slice interference. To illustrate the inter-slice interference, let consider two adjacent BSs (BS1 and BS2), close enough to interfere. The top scheme of figure 4.4 schematizes their resources allocation for 3 slices. Each slice has a different time-frequency resources portion on each BS. As the BSs are adjacent and the slices deploy different techniques to manage the transmission over their resources, the inter-slice interference is induced. Particularly, an inter-slice interference is observed between slice 2 and slice 3, as slice 3 is allocated the left lower resources portion on BS1, while the same portion on BS2 is allocated to slice 2. Such interference type is hard to manage, as each tenant monitors its resources independently from the other tenants. Moreover, 5G strategies for interference mitigation rely on a tight cooperation and coordination among the adjacent BSs in the network, i.e.



(a) Sub-optimal allocation (small continuous unallocated area)

(b) Optimal allocation (large continuous unallocated area)

Figure 4.3 – Optimal and sub-optimal resource allocation

cooperation enabling requirement.

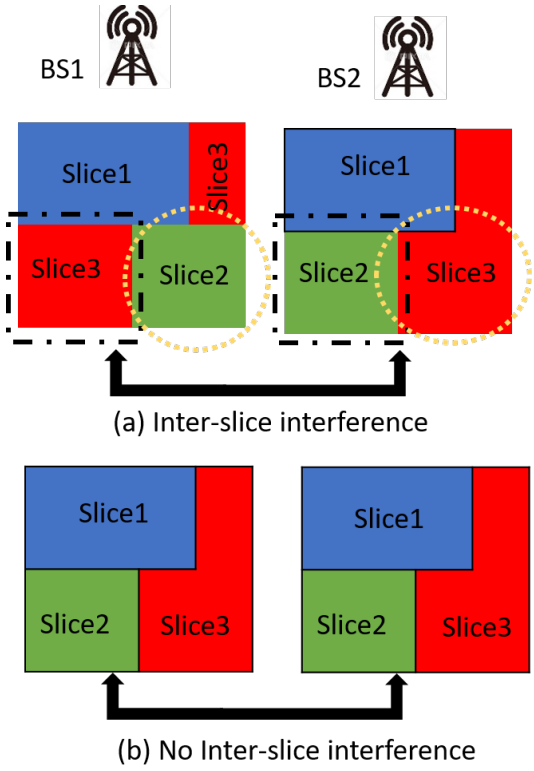


Figure 4.4 – Resource allocation for inter-slice interference mitigation

Therefore, the RAN slicing enforcement algorithms should guarantee the allocation of the same radio resources over time and frequency to the same slices among the adjacent BSs. Such allocation eases not only the deployment of 5G advanced

techniques such as MIMO and beamforming, but also the transmission schemes for inter-slice interference mitigation. Thus, it improves the overall network performance. Through experimentation, D'Oro et al [D'Oro 2019] proved the ability of such allocation to double the network throughput compared to a random allocation.

Fig. 4.4 (b) depicts the idea, each slice is allocated the same time-frequency portion of resources on both BSs. Hence, inter-slice interference is absent. Also, the slice owners have more flexibility to mitigate intra-slice interference and enable the advanced 5G techniques. Clearly the allocation strategy in (b) is optimal for the RAN slicing enforcement in 5G compared to the random allocation approach in (a), where each BS allocates its resources independently from the adjacent BSs. That is, the RAN slicing enforcement requires a coordinated resources allocation over adjacent BSs.

Further, we argue that the combination of both allocation strategies (fig. 4.4 (b) and fig. 4.3 (b)) allows the realization of the challenging RAN slicing requirements. For that, we address the optimization of such strategy in the upcoming parts.

4.4 RAN slicing enforcement

Considering the importance of the RAN slicing to achieve the global NS requirements and further the 5G objectives, we are focused on its enforcement. As stated earlier, the RAN slicing comes in its turn with another sophisticated requirements to accomplish the NS ones: slices orthogonality, satisfaction, scalability and cooperation enabling. With the concluded objectives in the previous part to reach those requirements, we proceed to the system design of this work. It highlights the 5G RAN vision where the RAN is controlled in a centralized manner. This is crucial, as a cooperation between BSs is required for a global resources allocation. Moreover, the presence of the flexible resources structures involves a fine grained resource management. Therefore, the resource grid decomposition is exhibited. Further, we investigate the possibility of deploying the already discussed optimal allocation strategies. System models are then depicted.

4.4.1 System design

Let consider a set of BSs covering a geographical zone. The 5G base station (BS) is named gNB. The gNBs cluster is controlled by a centralized SD-RAN (Software Defined RAN) controller, as illustrated in figure 4.5, noted R. This is essentially due to the high cooperation level required between the gNBs. The SD-RAN controls the RAN traffic, e.g. it receives the slices demand on each gNB (5G Base station) and all the RAN signaling information. We assume that SD-RAN copes with the scheduling and radio resources allocation over the specific zone. With the advanced implementation of intelligence in the radio part, the estimation of the slice demand traffic is possible [Sciancalepore 2017]. On the other hand, several researches have been interested in the slicing profile generation, i.e. the slice demand and resources assignment [Foukas 2017, Ferrus 2018, Gebremariam 2018, Caballero 2017, Jia 2018].

Therefore, the slicing profile is considered as an input argument for our system.

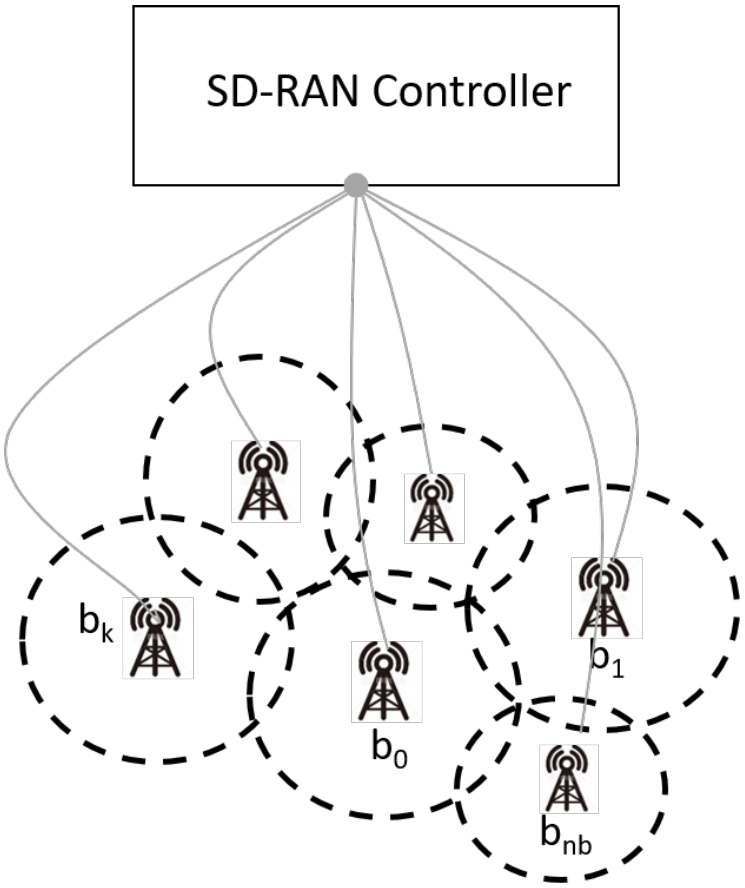


Figure 4.5 – System design

Furthermore, we propose to take advantage of the RAN intelligence in the 5G and the upcoming cellular networks to build a proactive allocation system for slices resources. For instance, with the pre-knowledge of the slices demand over the gNBs set, the SD-RAN proposes a resource allocation for the upcoming 10 ms, which corresponds to the frame duration in current cellular networks. This has the advantage to minimize the signaling exchange between the SD-RAN controller and the gNBs over the cluster.

On the other side, with the different tiles structures proposed by the 5G, an efficient resource management involves a fine grained access to the resources. Therefore, we put forward a new scheme for the resources grid decomposition as explained in the following.

4.4.1.1 Radio resources grid decomposition

With the diverse services flows and the increasing demand of cellular traffic, the 3GPP emphasizes the importance of treating the 5G radio resources differently

from the earlier standards. For that, it introduces different numerologies on each frequency band. Each numerology is efficient for a specific service flow, particularly, the $\mu = 2$ is much required for services with low latency. In the same perspective, we push this flexibility a step forward, and propose to handle the radio resources at small time and frequency granularities.

To that end, each gNB is entitled by its resource grid. With the variety of numerologies, we consider a resource grid decomposed into the smallest granularity in time and frequency. For instance, for the sub-6 Ghz bands, where μ can take values in $\{0, 1, 2\}$, the smallest resource block (sRB) is of size 180 kHz*0.25 ms. Fig 4.6 illustrates a decomposition for a small resource grid of 1 ms over 1.4Mhz. Three tiles structures are considered. Namely, the tile structure for a given slice with $\mu = 1$ is a square of 2*2 sRBs.

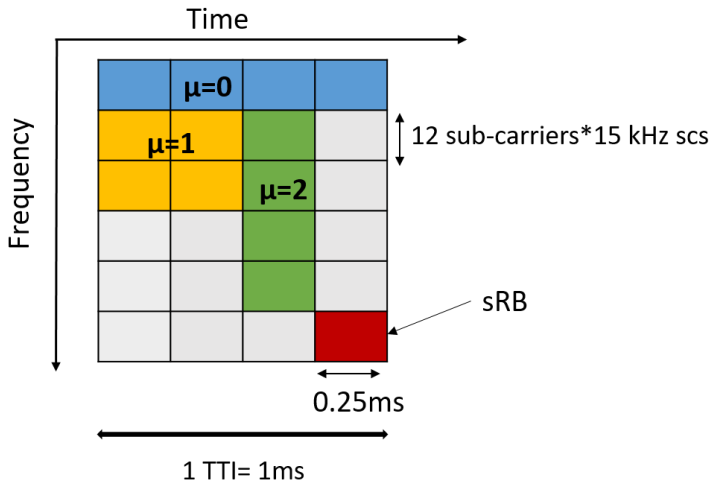


Figure 4.6 – Resource grid decomposition

The proposed resource grid decomposition allows a fine grained manipulation of the available resources, as they are shaped based on the slices numerologies requirements. Moreover, this decomposition results in an efficient control and management of the scarce radio resources. Also, it eases the way for the tight cooperation required by the 5G advanced techniques.

4.4.1.2 Objective formulation

Given the SD-RAN controller of a given zone, each gNB is characterized by its decomposed radio resource grid. For a specific allocation window, each slice is assigned an amount of tiles on each gNB over the RAN. From that, the SD-RAN proposes an allocation for the slices tiles taking into account both of the following objectives:

The first objective aims to maximize the placement of the tiles in the resource grid with respect to the BSs set, i.e. the maximization of the number of allocated

tiles in the same or similar position (time/frequency), for each slice, over the gNBs set. This is because of the tight cooperation and coordination involved over the RAN for the 5G advanced techniques deployment. Thus, it is imperative to ensure an allocation of resources to the same slice over adjacent BSs as explained in fig.4.4.

On the other hand, while allocating the slices tiles, not only their placement in similar position over the gNBs set is crucial but also an efficient radio resources utilization in each gNB. The latter could be achieved by an allocation that minimizes the sparse wasted unallocated resources. Or from another vision, maximizes the largest continuous unallocated resources space of each resource grid. Such allocation allows the MNO to accept new slices requests with different numerologies. Also, as the slices demand is variable over time, the MNO can reuse the unallocated resources portion to respond to the increased demand of the served slices. Hence, it ensures the scalability, especially when the largest portion is left over the time axis. Thus, the objective is implicitly multi-objective.

This allocation strategy, combining space and position optimization, ensures an enforcement of the RAN slicing. It then can be treated as a multi-objective optimization problem (MOOP) [Hutchison 2008], i.e. both objectives are conflicting. In such case, no objective can be satisfied totally. Solutions must find a compromise between both objectives. To solve the MOOP, three approaches are proposed in literature: aggregated, non-pareto and pareto approaches. The concept of the aggregated methodology is the transformation of a multi-objective problem to a mono-objective one. This is fulfilled through an attribution of suitable weight to each objective. The non-pareto technique seeks the optimal solution for each objective separately. Contrarily, the pareto based approach looks for all pareto optimal allocations. An allocation is taken for pareto optimality when it is impossible to make one objective better off without turning the other objective worse off. The accuracy of the pareto approach come with higher computational complexity. It restricts then the real time implementation vision. For that, we propose in this work the use of non-pareto technique. Thus, each objective is formulated individually. Then, heuristics are proposed as to solve the MOOP in aggregated manner.

4.4.2 System Model

Let denote $B = \{b_1, ..., b_{n_b}\}$ the cluster/set of n_b gNBs covering a geographical area. Notice that the gNB might offer a macro as well as small cell coverage. They are controlled by a centralized SD-RAN (Software Defined RAN) controller R, as illustrated in fig. 4.5. The multiple gNBs are adjacent to cover efficiently the geographical area. Such adjacency is highly vulnerable to interference.

Let us consider that R receives n_s slices requests to be served simultaneously during the allocation window T, $S = \{s_0, s_1, ..., s_{n_s}\}$. Based on the slices requirements on each BS (gNB), R generates the slicing profile $\Gamma = (\gamma^{\mu}_{s_i,k})_{s_i \in S, k \in B}$, with $\gamma^{\mu}_{s_i,k}$ is the amount of tiles to be allocated to slice s_i in BS b_k with numerology μ during T. Each slice s_i is supposed to have the same numerology over R, but requests a different amount of resources on each gNB. The slicing profile Γ is con-

sidered as an input argument in our system model. Therefore, once it is generated, it is primary to test its feasibility before the allocation process. In other words, a verification step of the possibility to allocate all the assigned slices resources in the appropriate gNB resource grid. In the following, we propose an exact method with an underlying constraint programming (CP) approach to test the slicing profile feasibility and tiles placement objective. This is because the CP eases the resolution of discrete problems through high level constraint propagation and controlled search behaviors [Hebrard 2017, Wallace 1996]. A constraint problem is stated as a set of variables, where each variable has a finite domain of values, and a set of relations on subsets of these variables.

4.4.2.1 Slicing Profile Feasibility Model (SPFM)

Let $g_k = (r_{k,x,y})_{0 \leq x \leq N_r, 1 \leq y \leq T}$ be the matrix representing the resource grid of gNB b_k, $k \in \{0, .., n_b\}$, with T and N_r represent the number of temporal slots (0.25 ms) and frequency channels of 180 kHz respectively, i.e. $r_{k,x,y}$ symbolizes the sRB in gNB b_k in position (x,y).[3] The resource grid size is therefore $A = N_r.T$.

A slicing profile Γ is considered as feasible, if all the tiles assigned to a group of slices on a given gNB b_k can be allocated over g_k without any overlapping, for all $k \in \{0, .., n_b\}$.

Let $\zeta_{s_i} = \{\tau_j \quad for \quad j \in \{0, .., \gamma^{\mu}_{s_i,k}\} \forall b_k \in B\}$ be the set of tiles requested by slice s_i over B.Each tile has a form of a rectangle based on the slice numerology (see fig 4.6). From that, we represent each tile τ_j of slice S_i in gNB b_k by two interval variables $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$. They refer to the tile allocation over frequency and time axis respectively. The length of the intervals is fixed as to reproduce the rectangle form of the tile. Particularly, if the tile τ_j corresponds to a slice resource with $\mu = 2$, the length of $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$ are fixed to 4 sRBs and 1 sRB respectively. Therefore, a non overlapping between two tiles τ_j and τ_h on a given g_k refers to their non overlapping over X and Y axis, i.e. $X_{b_k,s_i,j} \cap X_{b_k,s_i,h} = 0$ and $Y_{b_k,s_i,j} \cap Y_{b_k,s_i,h} = 0$.

Let α^{j}_{x,b_k}, $\alpha^{t^j_i}_{y,b_k}$ be the variables referring to the starting point of the two intervals $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$ respectively. And β^{j}_{x,b_k}, β^{j}_{y,b_k} point out their ends. The SPFM can be therefore formulated using CP approach as follows:

$$\alpha^{j}_{x,b_k} \leq N_r \qquad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \tag{4.1}$$

$$\alpha^{j}_{y,b_k} \leq T \qquad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \tag{4.2}$$

$$\begin{aligned} &(\alpha^{j}_{x,b_k} \geq \beta^{r}_{x,b_k} \vee \alpha^{r}_{x,b_k} \geq \beta^{j}_{x,b_k}) \wedge \\ &(\alpha^{j}_{y,b_k} \geq \beta^{r}_{y,b_k} \vee \alpha^{r}_{y,b_k} \geq \beta^{j}_{y,b_k}) \quad \forall r, j \in \zeta_{s_i} \quad \forall b_k \in B \end{aligned} \tag{4.3}$$

The constraints 4.1 and 4.2 limit the allocation bounds of each tile over both

[3] In this work, the sub-6 GHz band is treated, but it can be extended easily to the above-6 GHz band, where the sRBs will be of size (1440 kHz*62.5μs).

X and Y axis respectively. Then, the second constraint 4.3 ensures the allocation of the required slices tiles on each gNB without any overlapping between two tiles. This way the model is feasible when all the slices demand in a given gNB are allocated to the appropriate resource grid.

For feasibility model (SPFM) implementation, the IBM CPOptimizer constraint programming solver IBM ILOG (CPO)[4] is used. It provides a high level scheduling constraints. The allocation of tiles taking into consideration the constraint 4.3 can be directed by the *searchPhase* function. It guides the search for positions over X-axis and Y-axis for each tile with respect to non-overlapping constraint. Let VX_k denotes all the interval variables over X-axis representing the tiles assigned for the allocation on b_k, and VY_k the ones over Y-axis. The use of *searchPhase* is therefore writen as:

$$SetSearchPhases(searchPhase(VX_k), searchPhase(VY_k)) \quad \forall b_k \in B \tag{4.4}$$

Once the SPFM is verified and the slicing profile is feasible, a RAN slicing enforcement policy Ψ is required to fulfill the requirements depicted in section 4.3.2. It should lead to an optimal radio resources allocation over the b_{n_b} gNBs.

As stated earlier, the problem is treated as a MOOP. One objective carries the maximization of slice' tiles placement in the same frequency-time position in the resource grid of the adjacent BSs. Then, the other objective deals with the maximisation of the largest unallocated continuous portion of radio resources on each gNB.

In the following, the radio resources placement objective is modeled with an exact optimization method, as well as the approach followed to carry the largest continuous unallocated space during the resources allocation.

4.4.2.2 Enforcement of Slice Resources Placement (ESRP)

The policy Ψ has the objective to maximize the tiles placement of a given slice in the same position over the set of gNBs, B. For that we introduce the notion of tied tile.

Définition 1 (Tied tile) *A given tile τ_j is tied to a slice s_i over B if and only if the tile τ_j is placed in the same position over all the gNBs in the cluster B, i.e. τ_j has the same frequency and time position on each g_k, $\forall b_k \in B$*

Each tile τ_j of slice S_i in gNB B_k is represented by two interval variables $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$ as explained in 4.4.2.1. With α^j_{x,b_k}, $\alpha^{t^j_i}_{y,b_k}$ are the variables indicating the starting point of the two intervals $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$ respectively, and β^j_{x,b_k}, β^j_{y,b_k}

[4]CPLEX Optimization studio 12.9 `www.cplex.com`

their ends. With that, a tile is tied if and only if $\alpha^j_{p,b_k} = \alpha^j_{p,b_{k'}}$ and $\beta^j_{p,b_k} = \beta^j_{p,b_k}$ $\forall b_k \in B$, $p \in \{x, y\}$.

In other words, a tile of a given slice is tied if all its sRBs are allocated in the same position over the set of involved gNBs, i.e. gNBs where tile τ_j is present, as the slice demand varies over the gNBs. Consequently, we introduce the concept of tied sRB:

Définition 2 (Tied sRB) *A given sRB $r_{k,x,y}$ is tied to a slice s_i over B if and only if the sRB is allocated to the same slice over each gNB in B, i.e. $(x_k, y_k) = (x_{k'}, y_{k'})$ $\forall b \in B$.*

Even though the allocation is performed per tile, it is clear that the maximization of the amount of tied tiles for all the slices turns out to the maximization of the total amount of tied sRBs. Accordingly, we model mathematically the system as to maximize the total amount of tied sRBs. We have conceived two ways to model this objective. Both model versions are depicted in the following. It is in fact interesting to compare their scores as to find the best optimal model. Notably, the convergence time that is considered as a pivotal performance metric for each model implementation.

- **ESRP model version 1 (ESRP-v1)**

  For a given tile τ_j of s_i, let denote θj the amount of its tied sRBs over B. As each tile τ_j is symbolized by two interval variables on each g_k, $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$, θj corresponds to the overlap length between both intervals over all the involved gNBs. It can be formulated as follows:

  $$\theta_j = \Pi_{p \in \{x,y\}}(\Psi^j_p - \Upsilon^j_p)$$

  Where $\Psi^j_p = \min_{\forall b_k \in B_j} \beta^j_{p,b_k}$ and $\Upsilon^j_p = \max_{\forall b \in B_j} \alpha^j_{p,b_k}$, $p \in \{x, y\}$, identify the start and end position of the overlap between the rectangles over the involved gNBs over both frequency and time axis. It is worth noting that the overlap score between two intervals I_1 and I_2 is given by the CPO function *OverlapLength*, i.e. $OverlapLength(I_1, I_2)$. The same function has been used for assignment problem in [Kiatmanaroj 2016]. By way of illustration, let us consider the allocation over two gNBs of tile τ_j with $\mu = 2$ as shown in fig 4.7. Let suppose that both tiles have the same y-axis position. The amount of tied sRBs is exactly the surface of the overlap between the τ_j in gNB b_0 and τ_j in gNB b_1. The starting point of this surface over each axis can be computed by $\max(\alpha^j_{p,b_0}, \alpha^j_{p,b_1})$ and the end position by $\min(\beta^j_{p,b_0}, \beta^j_{p,b_1})$. On the X-axis, they are equal to α^j_{x,b_0} and β^j_{x,b_1} respectively. Thus, it represents two sRBs, i.e. $\theta_j = 2$. That is, two sRBs are tied between the two gNBs.

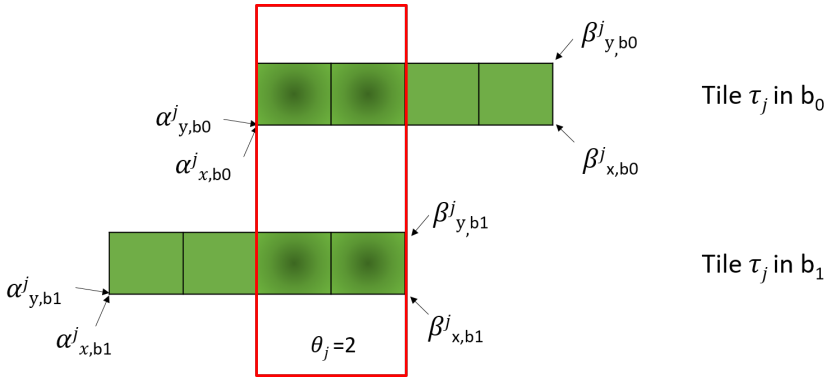


Figure 4.7 – Illustration of tied sRBs of a tile between two gNBs

Therefore, the total tied sRBs for a given slice s_i over B is given by:

$$\Theta_{s_i} = \sum_{j \in \zeta_{s_i}} \theta_j$$

Further, the total tied sRBs over B can be expressed as the summation of the total tied sRB of each slice over B:

$$\chi = \sum_{s_i \in S} \Theta_{s_i}$$

The objective is then formulated as to choose the slicing enforcement policy that maximizes χ. It is developed with a constraint programming (CP) approach as the SPFM (section 4.4.2.1) resolution. In the CP implementation, the constraints are explicitly stated to shape the aimed solution, i.e. in this case the maximization of χ.

$$\max_{\psi \in \Psi}(\chi) \qquad \text{(ESRP-v1)}$$

subject to

$$\alpha^j_{x,b_k} \leq N_r \qquad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \qquad (4.5a)$$

$$\alpha^j_{y,b_k} \leq T \qquad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \qquad (4.5b)$$

$$\sum_{j \in \zeta} \theta_j \leq \gamma^{\mu}_{s_i,k} \qquad \forall k \in B \quad \forall s_i \in S \qquad (4.5c)$$

$$(\alpha^j_{x,b_k} \geq \beta^r_{x,b_k} \vee \alpha^r_{x,b_k} \geq \beta^j_{x,b_k}) \wedge (\alpha^j_{y,b_k} \geq \beta^r_{y,b_k} \vee \alpha^r_{y,b_k} \geq \beta^j_{y,b_k}) \qquad \forall r, j \in \zeta_{s_i} \quad \forall b_k \in B \qquad (4.5d)$$

$$\Psi^j_p \geq \Upsilon^j_p \qquad \forall p \in \{x, y\} \quad \forall \tau_j \in \zeta_{s_i} \quad \forall s_i \in S \qquad (4.5e)$$

The constraints (4.5a) and (4.5b) ensure that all the allocated tiles are inside the resource grid, i.e. the allocation doesn't outpace the gNB grid limits on both frequency (4.5a) and time (4.5b) axis. The second constraint (4.5c) guarantees that each slice receives at maximum its required amount of tiles over each gNB. Then, the constraint (4.5d) addresses the non overlapping between tiles on the same gNB, i.e. each sRB is allocated at maximum to one slice. Hence, the slices orthogonality is achieved (i.e. resource isolation). Then, the last constraint (4.5e) assures the non negativity of each tied sRB surface.

- **ESRP model version 2 (ESRP-v2)**

In this ESRP version, in order to get the tied sRBs of a given tile τ_j over a gNBs set, we propose to add its overlap over both axis X and Y taking into account the numerology type. In other words, let denote ξ_j the total tied sRBs of a tile τ_j over a set of gNBs. It is expressed by:

$$\xi_j = [\sum_{p \in \{x,y\}} (\Psi_p^j - \Upsilon_p^j)] - \delta$$

With $\Psi_p^j = \min_{\forall b_k \in B_j} \beta_{p,b_k}^j$ and $\Upsilon_p^j = \max_{\forall b \in B_j} \alpha_{p,b_k}^j$, $p \in \{x, y\}$. δ is a binary variable, $\delta \in \{0, 1\}$, it equals 1 when the tile numerology is $\mu = 0$ or $\mu = 2$, and 0 otherwise. In fact, ξ_j refers to the tied sRBs surface computed without multiplication.

Further, for a given slice, the total tied sRBs (TTR) over B can be expressed by:

$$\Xi_{s_i} = \sum_{j \in \zeta_{s_i}} \xi_j$$

With ζ_{s_i} is the set of tiles requested by s_i over B. Thus, the total amount of tied sRBs over all B is formulated by:

$$\Phi = \sum_{s_i \in S} \Xi_{s_i}$$

The slicing enforcement policy that maximizes Φ is formulated as follows with CP resolution approach:

$$\max_{\psi \in \Psi}(\Phi) \qquad \text{(ESRP-v2)}$$

subject to

$$\alpha^{j}_{x,b_k} \leq N_r \qquad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \tag{4.6a}$$

$$\alpha^{j}_{y,b_k} \leq T \qquad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \tag{4.6b}$$

$$\sum_{j \in \zeta} \xi_j \leq \gamma^{\mu}_{s_i,k} \qquad \forall k \in B \quad \forall s_i \in S \tag{4.6c}$$

$$\begin{aligned}&(\alpha^{j}_{x,b_k} \geq \beta^{r}_{x,b_k} \vee \alpha^{r}_{x,b_k} \geq \beta^{j}_{x,b_k}) \wedge \\ &(\alpha^{j}_{y,b_k} \geq \beta^{r}_{y,b_k} \vee \alpha^{r}_{y,b_k} \geq \beta^{j}_{y,b_k}) \qquad \forall r, j \in \zeta_{s_i} \quad \forall b_k \in B\end{aligned} \tag{4.6d}$$

$$\Psi^{j}_{p} \geq \Upsilon^{j}_{p} \qquad \forall p \in \{x, y\} \quad \forall \tau_j \in \zeta_{s_i} \quad \forall s_i \in S \tag{4.6e}$$

$$\Psi^{j}_{x} - \Upsilon^{j}_{x} \geq 0 \Leftrightarrow \Psi^{j}_{y} - \Upsilon^{j}_{y} \geq 0 \qquad \forall \tau_j \in \zeta_{s_i}, \forall s_i \in S \tag{4.6f}$$

The constraints (4.6a) and (4.6b) ensure the allocation inside the gNB grid limits. The constraint 4.6c assures that each slice receives at maximum its required amount of tiles over each gNB. Furthermore, the orthogonality between slices is ensured by the constraint (4.6d), i.e. each sRB is allocated to only one slice. Then, the non negativity of each overlap either over X or Y axis is verified by the constraint (4.6e). Further, the last constraint assures that the model looks for the overlapping over X axis as well as over Y axis. This way, the tied sRB surface is realized.

Several symbols will be reused further in the manuscript. Accordingly, table 4.2 summarizes them.

Symbol	Signification
T	Allocation window
n_b	Number of gNBs (BSs) controlled by the SD-RAN
B	Set of n_b gNBs
n_s	Number of served slices by B
S	Set of n_s slices
Γ	Slicing profile
$\gamma^{\mu}_{s_i,k}$	Number of tiles to be allocated to slice s_i in BS b_k with numerology μ during T
ζ_{s_i}	Set of tiles requested by slice s_i over B
g_k	Resource grid of gNB b_k

Table 4.2 – Summary of the most used system symbols.

4.4.2.3 Largest Continuous Unallocated Space (LCUS)

The second objective targets the maximization of the continuous unallocated space on each gNB resource grid. Such allocation allows an efficient utilization of the rare radio resources. Also, it enables the scalability requirement in the RAN slicing.

In order to let the largest continuous portion of unallocated radio resources in each gNB resource grid g_k, $k \in B$. The problem is tackled as a two-dimensional rectangle bin packing (2DBP) optimization problem. In such problem, given a sequence of rectangular objects with specific height and width, the objective is to place the maximum of these objects inside a minimum bins of fixed size. With constraint of no-overlapping between the rectangles. The NP-Hardness of this problem is proven by a reduction from the 2-partition problem [Garey 2009, Karp 1972]. Although, the nonexistence of an asymptotic polynomial time approximation scheme (APTAS), it is APX-hard [Bansal 2004].

Let project the 2D bin packing to the context of the resource allocation with LCUS objective. In our case, the rectangular objects to pack in the bins are the slices tiles with their specific numerologies μ. Each tile τ_j has a form of a rectangle based on the slice numerology. Particularly, the tiles of a given slice with numerology $\mu = 2$ have a rectangular form of width and height equal to 4 and 1 respectively. Each gNB decomposed resource grid g_k is represented by a bin. It is supposed that the bins have the same size over all the gNBs set B, i.e. $\forall k \in B \quad size(g_k) = A$. Only one bin is available for the packing for each gNB. Its size is exactly the size of the resource grid in terms of time and frequency resources, i.e. allocation time over the carrier bandwidth. This can be considered as Knapsack use case problem of 2DBP.

The Knapsack problem is argued to be NP-hard. The realization of the LCUS objective then turns to be also NP-hard. Over decades, the research is focused on the development of efficient heuristic algorithms that approximate the optimal solution for 2DBP. Hence, several algorithms are proposed in the literature. They include heuristics with objects orientation and rotation possibility. In this work, the objects are non oriented. Also, the rectangles rotation is not allowed, as the rotation of a tile with $\mu = 0$ results in another different tile type with $\mu = 2$.

The suggested heuristics include the Skyline algorithm proposed in [Jylänki 2010]. It starts by placing the first rectangle object in the bottom left (BL). Then, each new rectangle object is left-aligned on top of the skyline level that results in the top side of the object lying at the bottom-most position of the bin. The topmost edges of already packed objects is tracked as illustrated by the red line in fig 4.8. The example shows the packing of 6 tiles with $\mu = 1$ and 5 tiles with $\mu = 0$ using the skyline algorithm. The algorithm then maintains the list of these horizons or "skyline" edges. The later grows linearly in the number of the packed rectangle objects. And for each rectangle packing top of a hole, it is possible and easy to compute the free rectangle that would be lost after packing. Thus, it is stored and evaluated for an aforementioned use. Such approach is referred as a waste map (WM) improvement for the skyline (BL) heuristic.

The authors tested a benchmark of 2DRP heuristics and variants of the skyline algorithms. They proved that skyline-BL-WM outperforms all the best tested online packers, in terms of packing efficiency as well as the run-time performance, when packing to one bin at a time. As the algorithm packs the objects in a way to minimize the wasted space between the packed objects, it results in letting the

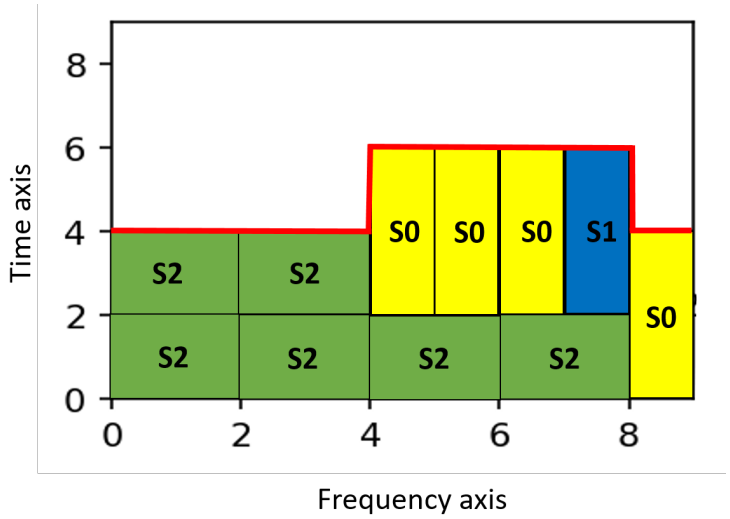


Figure 4.8 – Illustration of Skyline algorithm packing 3 slices tiles

largest unallocated space. Therefore, the skyline-BL-WM heuristic is chosen to approximate the LCUS solution. Mainly for two reasons:

- The algorithm is highly performing in both time convergence and packing efficiency on one bin. This corresponds to the use case of this work, i.e. each g_k is represented by one bin and the allocation is allowed only in this bin.
- The algorithm approach seeks to pack the objects (tiles) as to have the lowest skyline (contour). This is advantageous, as we are seeking to let the maximum of unallocated space over time axis. Hence the bottom could be chosen as the frequency axis. The skyline is then aligned over time as shown on fig 4.8.

The Skyline heuristic approximates the LCUS solution. Thus, there is a need to evaluate its performance. For that, an optimal score is necessary. In this work, the naive method that encounters the LCUS topmost upper bound is used, as depicted in the following. It is therefore considered as the optimal LCUS solution.

LCUS topmost upper bound LTUB:
On a given resource grid g_k, $k \in B$, the topmost LCUS upper bound can be achieved when all the tiles of all the slices are allocated without overlapping and with no space left in between, i.e. non existence of wasted space between allocated tiles. The size of each resource grid can be computed, as stated before, by $A = N_r * T$, with N_r is the number of frequency channels and T the allocation window. A refers also to the total number of available sRBs on each g_k. Given the slices demand $\gamma^{\mu}_{s_i,k}$ in terms of tiles, the total required tiles on each b_k can be computed by: $\rho_k = \sum_{s_i \in S} \gamma^{\mu}_{s_i,k}$. Therefore, the total allocated sRBs on each g_k is equal $4 * \rho_k$,

as each tile contains 4 sRBs. From that, the highest upper bound of LCUS, noted $LCUS_k$, can be quantified by

$$LTUB_k = A - 4 * \rho_k$$

The topmost upper bound over all B is then:

$$LTUB = \sum_{k \in B} LTUB_k$$

4.5 Models evaluation

In the previous section, the two objectives are modeled separately. Two mathematical models are proposed for the ESRP objective. Their evaluation is necessary, as well as the comparison of their performance. The second objective is treated as a 2DBP optimization problem. With the NP-hardness of such resolution, the skyline heuristic is used for this resolution. In order to evaluate its performance in term of LCUS, the LTUB is taken as the optimal solution. In this section, the evaluation of both objectives solutions is conducted. The metrics of this evaluation are: the total tied sRBS (TTR) for ESRP versions, the LCUS for skyline and the convergence time for all algorithms (i.e. ESRP-v1, ESRP-v2, Skyline)

4.5.1 Performance metrics computation

In the following, the used performance metrics in this evaluation are highlighted with the followed methodology.

4.5.1.1 Total tied sRBs (TTR)

The objective behind the implementation of both ESRP models is to maximize the total tied sRBs during an allocation of a slices set over a gNBs set. Therefore, the basic evaluation metric is the achieved total tied sRBs, noted TTR. The two ESRP models are developed with CPO. Both models are configured with a time limit fixed to 600 s. The later limits the time for models to reach the optimal TTR score. In fact, if the model doesn't reach the optimal TTR score during the 600 s, the compilation is stopped and the upper bound score is saved as optimal score. Otherwise, the objective score is retained.

4.5.1.2 Convergence Time (CT)

The time convergence is critical metric for the long-term implementation of the slicing enforcement policy. Especially that the policy should be extended to real time deployment scenarios in the future. The aim is to conduct a comparison of the convergence time between all the tested algorithms. For that the time python module is used. The CT is started at the time of building the model. It ends once

the optimal model reaches the optimal solution or time limit. In the case of the heuristic, the time is stopped at the end of the tiles packing.

4.5.1.3 Largest Continuous Unallocated Space (LCUS)

In order to count the largest continuous unallocated space (LCUS) after each allocation, we propose the use of the Connected Component Labeling (CCL) with the Depth First Search (DFS) method [He 2017]. The later is based on graph traversal approaches in graph theory and can be used as well for binary images (matrix). A connected component in a matrix is the subset of matrix elements with same value, where each element is reachable by the other elements. Thus, in our work case, we derive a binary matrix from the resource grids after the allocation completion. Each allocated sRB to a given slice corresponds to an element matrix with value equals 1 and the unallocated sRBs (elements) worth 0. That is, each binary matrix M_k refers to one g_k after allocation, $k \in B$. The objective is then adapted to find the maximum subset of zeros among each matrix, i.e. continuous unallocated sRBs. Let denote it $LCUS_k$, $k \in B$. It can be achieved by the CCL approach. Particularly, given an input binary matrix as illustrated in figure 4.9, the CCL based DFS resolution lead to the labeled matrix (right matrix). Each group of connected elements is labeled by the same number. For instance, the input matrix in 4.9 contains three subsets of zeros, i.e. 9 zeros, 2 zeros and 4 zeros at the top left, top right and bottom right of the matrix respectively. After the CCL, each group is labeled differently with an integer. Particularly, the group of zeros at the bottom right is labeled by 6. This means that each group forms a connected component of zeros. Therefore, the derivation of the biggest connected zeros group is straightforward and accurate. In this example, it is the group of 9 zeros at the top left of the matrix, labeled by 2. This group represents exactly the largest continuous unallocated space in a resource grid.

Further, we compute the total LCUS over the set B as:

$$LCUS = \sum_{k \in B} LCUS_k$$

4.5.2 ESRP evaluation: Slice Resources Placement Problem

The resource grid size is fixed, i.e. $N_r = 27$, $T = 40$. For each test, the number of slices requests and the number of adjacent BSs are fixed. The slicing profile Γ is randomly generated for each simulation run as to have at maximum 80% of the grid usage. 100 independent simulation runs are realized for each given test. For each simulation run the model feasibility (SPFM) is tested. Then in case of its feasibility, the same instances are used for both optimal models and skyline. It is worth noting that solving time for the ESRP models is fixed for maximum 600 s. In other words, if the ESRP (ESRP-v1 or ESRP-v2) model doesn't find the optimal solution during 600 s, then the upper bound solution is saved. This is because of the long time consumed by both models to converge.

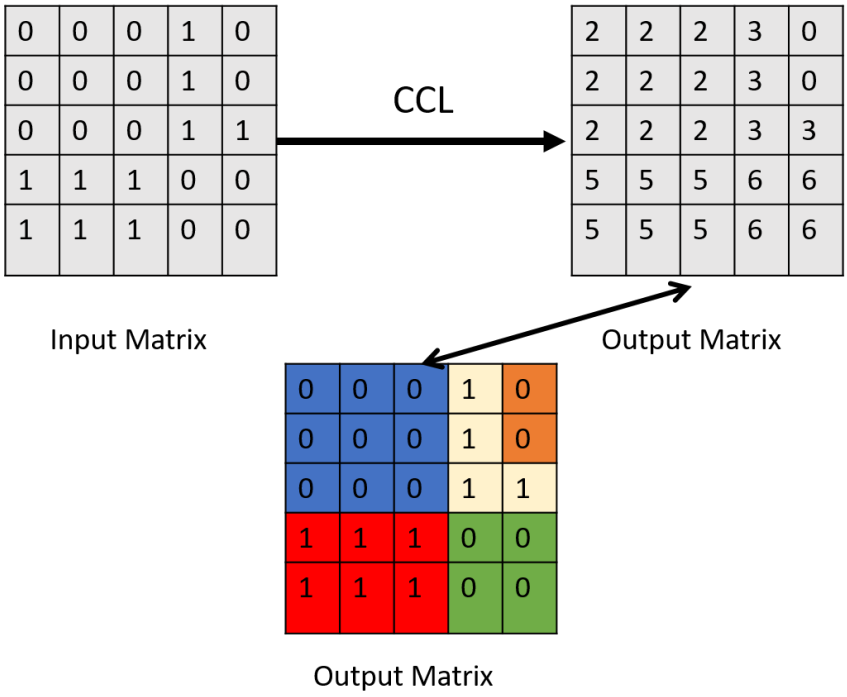


Figure 4.9 – Application of CCL-DFS on a resource grid (binary matrix)

- Total Tied sRBs (TTR): The total tied sRBs corresponds to the objective score achieved by the model as explained earlier. Both ESRP models optimal score is highlighted in this paragraph through an analyse of the B and S size impacts.

  a) Impact of B size: Fig 4.10 illustrates the optimal/upper bound TTR score with both models as a function of B size when serving 3 and 9 slices, left and right figures respectively. The two models curves are superposed for both slices set size. Hence, the same TTR score is achieved by both models over the different B sizes. The score is increasing with B size growth. It ranges from 278 sRBs to 720 sRBs for both use cases (i.e. 3 and 9 slices). Thus, larger B sets produce higher TTR scores.

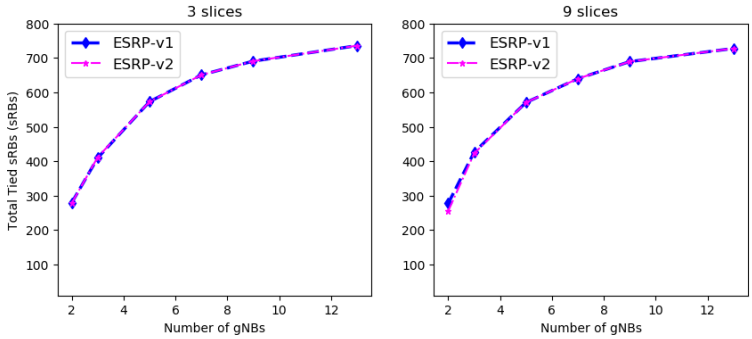


Figure 4.10 – Total tied sRBs with both ESRP-v1 and ESRP-v2 as a function of B set size.

b) Impact of S size: The S size is an important parameter for the system evaluation, as the NS aims the creation of various slices. Fig 4.11 plots the TTR score by both ESRP-v1 and ESRP-v2 as a function of the S size for both a system with 3 and 9 gNBs. Both models achieve quite similar score over S for both use cases. Their corresponding curves are superposing (see the fig 4.11). A small variation of TTR score is observed with S variation. This reflects the TTR insensitivity of both models to the slices set growth.

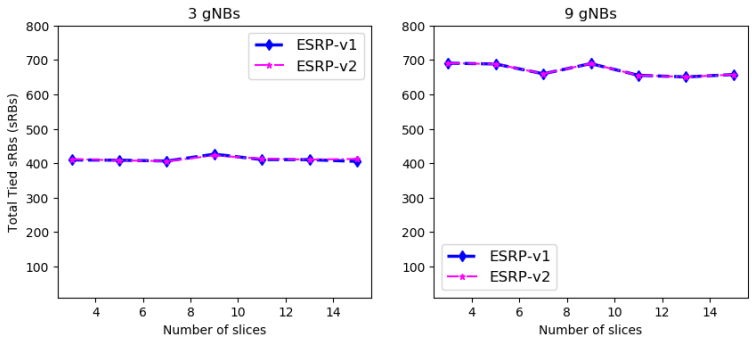


Figure 4.11 – Total tied sRBs of both ESRP-v1 and ESRP-v2 as a function of B set size.

In the previous analysis, the TTR score evolution over both B and S sizes is depicted. The TTR optimal score is either achieved by the ESRP models or not during the time limit. For that, table 4.3 indicates the statistics of both ESRP-v1 and ESRP-v2. It highlights the number of times each model reaches the optimal solution during the prefixed time limit, referred by successful optimal TTR. The highest TTR points out the number of simulations a model outperforms the other one based on the achieved TTR score.

Both ESRP versions have marked pretty similar number of optimal solutions. ESRP-v1 reaches the optimal solution 653 times, and 617 times achieved by ESRP-v2. It is to be noted that the total simulation runs is 4198. Thus, the successful optimal TTR is achieved only 15,55% and 14,7% with ESRP-v1

Table 4.3 – Comparison between ESRP-v1 and ESRP-v2

Number of	ESRP-v1	ESRP-v2
Successful optimal TTR	653	617
Highest TTR	163	53

and ESRP-v2 respectively. Yet, the difference between the two models is not significant. Therefore, both models are unable to achieve the optimal TTR score in 600 s. Details about the instances for which the ESRP models have achieved the optimality within the time limit are depicted in Annex B. Most of the optimal scores are reached for the small systems composed of 2 gNBs. Also, in terms of the number of highest TTR score, the ESRP-v1 performs better than ESRP-v2 in 163 simulations. While ESRP-v2 reaches higher TTR compared to ESRP-v1 only 53 times. It can be therefore concluded that ESRP-v1 outperforms smoothly ESRP-v2 with respect to TTR.

- Convergence Time (CT): Considering the convergence time as a crucial metric for real time slicing implementation, we evaluate both models CT in this paragraph.

  a) Impact of B size: to show the impact of B size on the convergence time of each model, fig 4.12 plots the convergence time in seconds as function of the B size when the system serves 3 slices (left figure) and 9 slices (right figure). In both case studies, the ESRP (ESRP-v1 and ESRP-v2) models have a quite similar convergence time, i.e. their curves are superposed. With three slices and 2 gNBs, the models converge approximately at 100 s. Then, the CT increases gradually with the B set size growth. The maximum CT is marked to 600 s. It corresponds to the prefixed time limit. When the S set size increases to 9 slices (right figure), The minimum CT for both models is about 480 s. Then, it reaches quickly the 600 s for larger B size. This points out the sensitivity of the models to B and S sizes as analysed next.

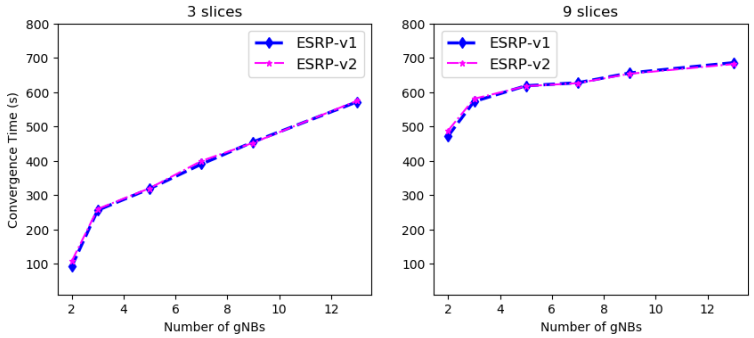


Figure 4.12 – Convergence Time (s) of both ESRP-v1 and ESRP-v2 as a function of B set size.

b) Impact of S size: fig 4.13 plots the CT of both ESRP versions as a function of S set size with a system composed of 3 gNBs and 9 gNBs, left and right figure respectively. The minimum CT is about 260 s. It is reached when serving 3 slices in a set of 3 gNBs. Then, it increases with higher number of slices. A small difference in CT can be observed between ESRP-v1 and ESRP-v2 in the left figure. In this case, the CT gap between ESRP-v1 and ESRP-v2 is in the order of ten seconds. Moreover, based on statistics results, the ESRP-v1 converges rapidly compared to ESRP-v2 in 2122 simulations. Whereas, the ESRP-v2 outperforms ESRP-v1 in 2076 simulations with respect to CT.

With 9 gNBs, the minimum CT is in the order of 450 s. It is achieved with small S size. Then, a similar CT evolution as in the 3 gNBs system is observed. One can remark that maximum CT is higher than 600 s in this illustration. This is because the threshold fixed time of 600 s starts after building the model. Whereas, the CT computation in this work includes also the model building time. From that, we can conclude that larger B size results in higher building time for both models. In this case, it reaches 50 s.

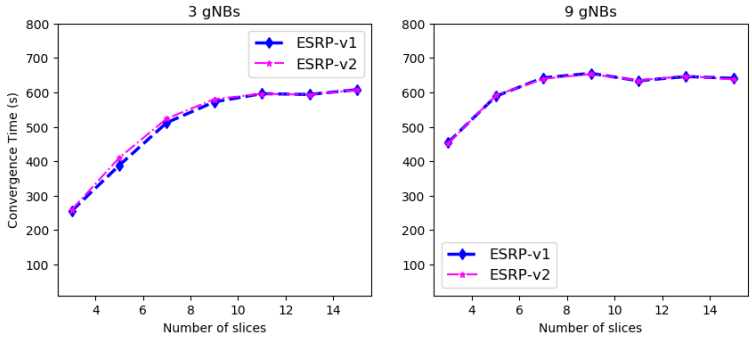


Figure 4.13 – Convergence Time (s) of both ESRP-v1 and ESRP-v2 as a function of S set size.

4.5.3 LCUS evaluation: Unallocated Space Problem

The second objective concerns the LCUS. For that, we have proposed the use of the Skyline heuristic. In this part, the achieved LCUS with skyline is compared with the upper bound LCUS, LTUB with respect to the S and B size.

- Largest Unallocated Space (LCUS):

a) Impact of B size: fig 4.14 shows the variation of LCUS by skyline and LTUB as a function of B size in a system serving 3 slices (a) and 9 slices (b). Over the different B sizes, the skyline heuristic reaches the topmost upper bound LCUS (LTUB). The two curves representing the LTUB and skyline score are similar. This reflects the capability of skyline to allocate efficiently the slices tiles without any space waste in between.

On both use cases 3 slices and 9 slices, the LCUS is increasing with B size. This is expected, as the LCUS is the sum of $LCUS_k$, $k \in B$, over B.

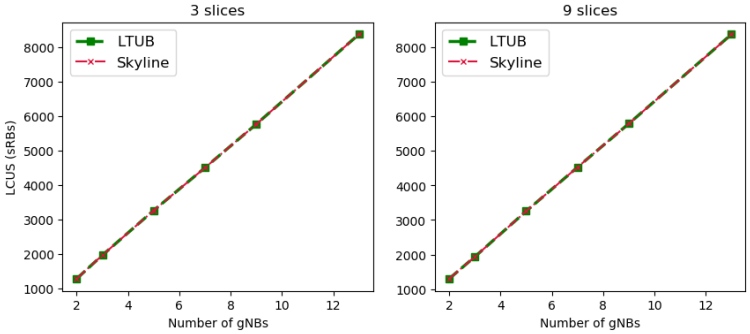


Figure 4.14 – LCUS (sRBs) for both Skyline and LTUB as a function of B set size.

b) Impact of S size: regarding the S variation impact, fig 4.15 indicates the LCUS score by skyline and the LTUB as a function of S set size in the case of 3 gNBs and 9 gNBs. For both case studies, the LCUS varies smoothly with the S size variation. Also, with different S sets, the skyline always attain the LTUB. From that, it is clear that LCUS skyline is insensitive to the number of served slices or either the B set size.

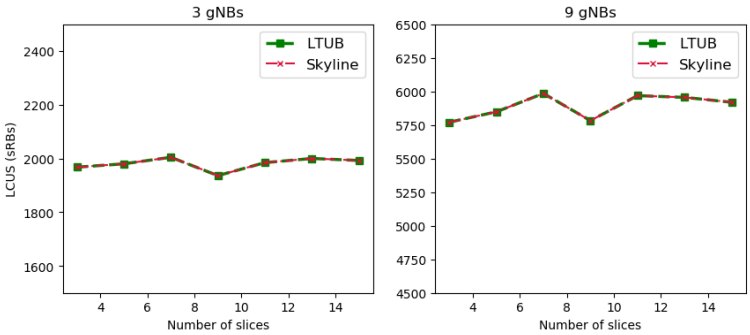


Figure 4.15 – LCUS (sRBs) for both Skyline and LTUB as a function of S set size.

- Convergence Time (CT): the CT for skyline as a function of B size and S size is shown on fig 4.16. The skyline converges in the order of hundreds of milliseconds. The CT increases for higher B sets (fig 4.16 (a)). Nonetheless, it doesn't outpace 0.45 s. A small CT difference is remarked when serving 3 and 9 slices. It is zoomed out on fig 4.16 (b). For lower gNBs set, the CT doesn't exceed 0.1 ms for different slices set sizes. Then, for larger B size, the CT is between 0.16 s and 0.45 s for the various S set size. Therefore, the skyline is interesting with respect to CT, as it can be implemented for SD-RAN real time allocations.

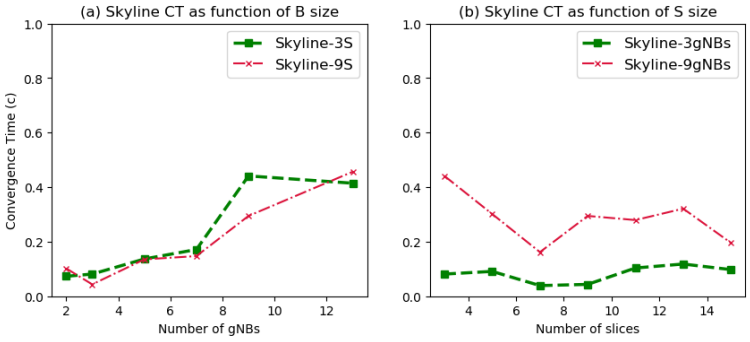


Figure 4.16 – CT for Skyline as a function of B set size (fig. (a)) and S set size (fig. (b)).

4.5.4 Discussion

Overall, both ESRP models converge slowly in the order of hundreds of seconds. This is with the fixed time limit before the simulations run, i.e. 600 s. Hence, the ESRP versions might converge at higher time scales. For the trial simulations, the scale is at a granularity of hours for larger sets. Also, the percentage of reaching the objective (i.e. optimal TTR) within this time limit is feeble. Thus, higher time limits is required to achieve this objective. This in fact restricts their implementation for real time allocation strategies. There is therefore a need for heuristics to realize the cooperation enabling requirement rapidly. Nevertheless, the ESRP models could be advantageous for the SD-RAN large time scale decisions, especially for the case of slices with small traffic variation.

On the other hand, the NP-hardness of the LCUS objective drove us to use the heuristic solution, especially the skyline heuristic. The later has demonstrated good performance in terms of largest continuous unallocated space as well as convergence time. The LCUS attained by skyline is exactly the same as the LTUB (LCUS topmost upper bound) for different B set and S set sizes. In addition, it converges in less than 0.45 s for various B and S sets. This turns the skyline suitable for real time deployments of RAN slicing without enabling cooperation requirement.

Further, the RAN slicing comes with the four requirements together, i.e. orthogonality, satisfaction, scalability and enabling cooperation. The ESRP models target the three RAN slicing requirements (i.e. satisfaction, orthogonality and cooperation enabling) at the expense of scalability. Whereas, the skyline assures the RAN slicing without enabling cooperation requirement. With the aim of real time RAN slicing enforcement, a solution based heuristic seems to be the best choice, as the ESRP models converge slowly in the order of hundreds of seconds. Accordingly, we propose three heuristics to support the RAN slicing requirements. With the multi-objective criterion, the space objective is prioritized, as explained in the next section. This is because of the scalability importance from the MNO perspective.

4.6 Proposed heuristics

The aim of this work targets the real time RAN slicing enforcement. For that, an allocation strategy combining the maximisation of the total tied sRBs over a given set of gNBs as well as the largest continuous unallocated space is argued to reach such aim. The non-pareto approach is used to resolve such MOOP. The LCUS objective is uncovered to be NP-hard. And, the developed ESRP models converge slowly. Thus, it limits the real time slicing enforcement. Therefore, we propose to tackle this MOOP with heuristic based approach.

Given the muti-objective criterion, a compromise between both objectives is unavoidable. The slice owner might have the possibility to allocate its TTR to the users at the cells boards highly affected by ICI. Thus, it can enable the cooperation techniques on only these TTR. Certainly, a slice with higher TTR is much more beneficial, as the slice owner would have more flexibility on its resources. On the other hand, the LCUS enables the scalability. Thus, the MNO can serve more slices. Moreover, an improvement of the current served slices QoS can be achieved by scaling up their resources. In addition, it allows a high level of spectral efficiency. Accordingly, The LCUS is prioritized in the heuristic development.

The aforementioned used skyline heuristic reaches the optimal LCUS in a small time scale, i.e. maximum 0.45 s. This is attractive from the real time implementation perspective. For such reason, the proposed heuristics use the skyline as an underlying allocation technique. Three heuristics are developed. With the skyline approach, the LCUS is guaranteed while the TTR is targeted at best to find the optimal slicing enforcement policy. In this section, the developed heuristics are depicted.

4.6.1 Heuristic 1: Highest Slice First HSF

Each slice is assigned a different amount of resources on each BS. Thus, the slices requiring higher amount of resources over B are expected to generate high number of tied sRBs. Considering such fact, the total required resources over B is computed for each slice based on the slicing profile $\Gamma = (\gamma^{\mu}_{s_i,k})_{s_i \in S, k \in B}$ ($\gamma^{\mu}_{s_i,k}$ is the amount of tiles to be allocated to slice s_i in BS b_k with numerology μ during T) as follows:

$$\lambda_{s_i} = \sum_{k \in B} \gamma^{\mu}_{s_i,k}$$

The algorithm first try to set the bigger slices at the same position in the interfering BS. Therefore, the slices are sorted in a decreasing order based on λ_{s_i}. Hence, the slice with higher amount of tiles is first served and the lower last. We denote such method as the `Highest Slice First, HSF`. It is represented in Algorithm 4.1 and performs as follows:

1.) compute the total required resources of all the slices over B, $\Lambda = (\lambda_{s_i})_{s_i \in S}$.

2.) sort the slices in decreasing order based on λ_{s_i} and generate the set S^o with ordered slices.

3.) insert the first object of the first slice in S^o in the bottom left of the bins.

4.) allocate the current slice object in the bottom-most position leaving the largest unallocated space over time in each bin and minimizing the wasted space between objects of same bin.

5.) keep inserting the objects of the current slice on all the bins with respect to 4 until the required objects of the current slice are allocated on all the bins.

6.) repeat 4 and 5 as to allocate the slices in sequential order as in S^o, until all the objects of each slice on each BS are allocated.

Algorithm 4.1 Heuristic1- HSF

Input: B, S, Γ
Output: HSF sRBs allocation $G^{HSF} = (g_k^{HSF})_{k \in B}$
set $g_k^{HSF} = (\alpha_{k,f,t}^{s_i,\mu})_{f,t} = 0 \; \forall k \in B \quad \forall s_i \in S$
Compute $\Lambda = (\lambda_{s_i})_{s_i \in S}$
$S^o \leftarrow$ sort S in decreasing order based on Λ
for each BS $b_k \in B$ **do**
  for slice $s_i \in S^o$ **do**
    while $\gamma_{s_i,k} \neq 0$ **do**
      allocate s_i tile subsequent sRBs with LCUS account.
      update g_k^{HSF}
      remove the allocated object from $\gamma_{s_i,k}$
    end while
  end for
end for
end

The first instructions of the total required resources computation and slices sorting run in $O(n_b * n_s + n_s log(n_s))$. Let denote ρ the total required tiles of all slices over all the gNBs, i.e. $\rho = \sum_{s_i \in S} \lambda_{s_i}$. The heuristic core code run in $O(n_b * \rho^2)$. This is because the packing time on each gNB is ρ^2. Consequently, the HSF converges with a time complexity of $O(n_b * \rho^2)$.

4.6.1.1 HSF example

Let us consider an SD-RAN controlling three gNBs, $\{b_0, b_1, b_2\}$ and serving three slices $\{s_0, s_1, s_2\}$. The slices tiles demand is depicted in table 4.4. For each slice with a given numerology μ, its required resources on each BS is depicted, e.g. s_0 demands four tiles with numerology $\mu = 0$ on gNB b_2. Let us follow the HSF algorithm 4.1. The total required resources by s_0, s_1 and s_2 are 8, 4 and 9 tiles respectively. Consequently, the sorted slices set in decreasing is $\{s_2, s_0, s_1\}$.

Therefore, the first served slice is s_2. Its first tile is allocated at the bottom left on each gNB as shown in fig 4.17. Then, the allocation for s_2 tiles is completed with respect to its demand (table 4.4), e.g. three tiles with $\mu = 1$ is allocated for s_2 on b_1 (Yellow tile). For each tile allocation, table 4.4 is updated by removing the allocated tiles. Once s_2 allocation is completed, s_0 tiles are inserted consecutively keeping the lowest skyline on each gNB and updating the resource demand. Later, s_1 resources are inserted on each gNB. With that all the slices are served over the three gNBs (see fig 4.17).

	$s_0(\mu = 0)$	$s_1(\mu = 2)$	$s_2(\mu = 1)$
b_0	2	2	1
b_1	2	1	4
b_2	4	1	4

Table 4.4 – Slices required tiles on each gNB: $(\gamma_{s_i,k})$.

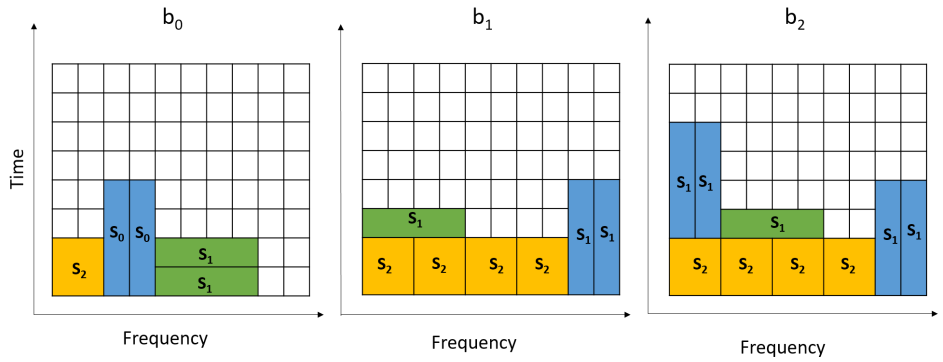


Figure 4.17 – HSF Heuristic example

4.6.2 Heuristic 2: Iterative Minimum Allocation IMA

The previous heuristic, HSF, allocates in sequential order all the tiles of a given slice over all the involved gNBs, starting with the slice requesting the highest total number of tiles over B.

Given that the slices request different amount of tiles over a set of gNBs, it is clear that the maximum tied sRBs between a subset of gNBs equals the minimum required resources over the same subset. From that, with the aim to maximize the total tied sRBs over B, we propose an iterative allocation of the non null minimum required tiles of each slice over the involved BSs. We refer to such approach as an Iterative Minimum Allocation (IMA) approach.

Let denote m_{s_i} the non null minimum required objects for slice s_i over B. It is computed as follows:

$$m_{s_i} = \min_{\substack{k \in B \\ \gamma_k^{s_i} \neq = 0}} \gamma_k^{s_i}$$

.

The IMA procedure works as follows:

1.) compute the total required resources of all the slices over B, $\Lambda = (\lambda_{s_i})_{s_i \in S}$.
2.) sort the slices in decreasing order based on λ_{s_i} and generate the set S^o with ordered slices.
3.) Compute the minimum required objects for all slices, $s_i \in S^o$ over all B, $M = (m_{s_i})_{s_i \in S^o}$.
4.) allocate m_{s_i} sequentially with leaving the LCUS over each bin.
5.) update $\gamma_{s_i,k}$ by subtracting m_{s_i}.
6.) repeat 3, 4 and 5 for all $s_i \in S^o$. If $\gamma_{s_i,k} = 0$. Remove s_i from b_k.
7.) if $\Gamma = 0$, stop. Otherwise, repeat vi until all the slices are assigned the required tiles over B.

Algorithm 4.2 Heuristic2- IMA

Input: B, S, Γ
Output: IMA sRBs allocation $G^{IMA} = (g_k^{IMA})_{k \in B}$
set $g_k^{IMA} = (\alpha_{k,f,t}^{s_i,\mu})_{f,t} = 0 \ \forall k \in B \quad \forall s_i \in S$
Compute $\Lambda = (\lambda_{s_i})_{s_i \in S}$
$S^o \leftarrow$ sort S in decreasing order based on Λ
while $\Gamma \neq 0$ **do**
  Compute $M = (m_{s_i})_{s_i \in S^o}$
  for each BS $b_k \in B$ **do**
    for each slice $s_i \in S^o$ **do**
      add m_{s_i} to the allocation with LCUS
      Update g_k^{IMA} by allocating m_{s_i} tiles subsequent sRBs
      remove the allocated objects from $\gamma_{s_i,k}$
    end for
    Update Γ by removing the allocated m_{s_i} objects from $\gamma_{s_i,k}$
    if $\gamma_{s_i,k} = 0$ **then**
      remove s_i request in b_k
    end if
  end for
end while
end

4.6.2.1 IMA example

Let us consider the same system as the one of HSF example in part 4.6.1.1, i.e. three gNBs serving three slices. The slices resources demand over the system

is depicted in table 4.4. With IMA algorithm, the total required resources of each slice over the system is computed. It equals 8 tiles for s_0, 4 tiles for s_1 and 9 tiles for s_2. Let the sorted slices set in decreasing order be $\{s_2, s_0, s_1\}$. Thus, the first served slice is s_2. The minimum required resources of each slice over B is computed. From table 4.4, the non null minimum required tiles of s_0, s_1 and s_2 equal 2, 1 and 1 tile respectively. With that, IMA allocates one tile for s_2, then two tiles for s_0 and later one tile for s_1 over the gNBs system as illustrated on fig 4.18. Then, the table 4.4 is updated by removing these allocated resources. For instance, the remaining required resources of s_1 on gNB b_0 is one tile as one tile has already been allocated on this gNb.

The slices that have been served totally on a given gNB in the first step are removed, i.e. s_0 request is removed in b_1 and b_0. Thus, in the next step, the slices remaining resources are allocated with respect to the order in $\{s_2, s_0, s_1\}$. In other words, three remaining tiles of s_2 are allocated over b_1 and b_2, then two tiles of s_0 are allocated on b_2. Later, one tile is allocated for s_1 on b_0 (see fig 4.18). With that, all the slices requests are served, preserving the lowest skyline on each gNB resource grid.

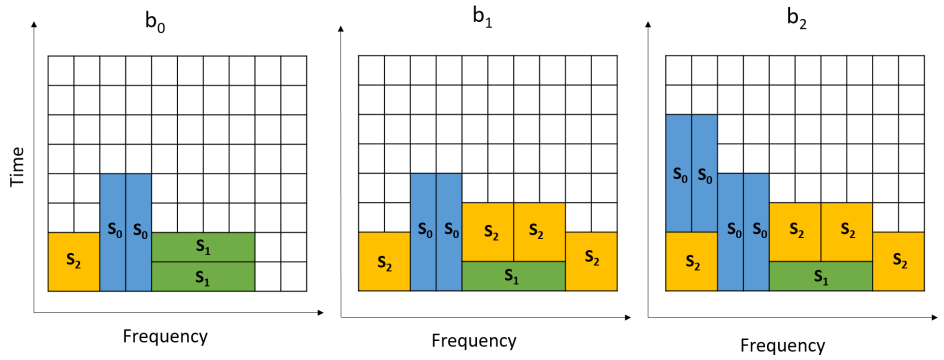


Figure 4.18 – IMA Heuristic example.

4.6.3 Heuristic 3: Highest Minimum First HMF

With IMA, the slices are sorted in decreasing mode based on the total required resources over the involving BSs. HMF instead sorts the slices in decreasing order based on the minimum required resources over the gNBs set. Therefore, the minimum is computed at each iteration and the algorithm proceeds as follows:

1.) Compute the minimum required tiles for all slices, $s_i \in S^o$ over all B, $M = (m_{s_i})_{s_i \in S}$.

2.) sort the slices in decreasing order based on $m_{s_i s_i \in S}$ and generate the set S^o with ordered slices.

3.) allocate m_{s_i} sequentially with leaving the LCUS over each bin.

4.) update $\gamma_{s_i,k}$ by subtracting m_{s_i}.

5.) repeat the steps from 1 until 4 for all $s_i \in S^o$. If $\gamma_{s_i,k} = 0$. Remove s_i from b_k.

6.) if $\Gamma = 0$, stop. Otherwise, repeat 5 until all the slices are assigned the required tiles over B.

Algorithm 4.3 Heuristic3- HMF

Input: B, S, Γ
Output: HMF sRBs allocation $G^{IMA} = (g_k^{IMA})_{k \in B}$
set $g_k^{IMA} = (\alpha_{k,f,t}^{s_i,\mu})_{f,t} = 0 \; \forall k \in B \quad \forall s_i \in S$
while $\Gamma \neq 0$ **do**
  Compute $M = (m_{s_i})_{s_i \in S^o}$
  $S^o \leftarrow$ sort S in decreasing order based on M
  for `each BS` $b_k \in B$ **do**
    for `each slice` $s_i \in S^o$ **do**
      add m_{s_i} to the allocation with LCUS
      Update g_k^{IMA} by allocating m_{s_i} tiles subsequent sRBs
      remove the allocated objects from $\gamma_{s_i,k}$
    end for
    Update Γ by removing the allocated m_{s_i} objects from $\gamma_{s_i,k}$
    if $\gamma_{s_i,k} = 0$ **then**
      remove s_i request in b_k
    end if
  end for
end while
end

HMF and IMA are implemented with time complexity of $O(n_b * \rho^2)$.

4.6.3.1 HMF example

Considering similar system composition as the IMA and HSF examples in parts 4.6.1.1 and 4.6.2.1 respectively, we illustrate in this paragraph the HMF algorithm. Firstly, the minimum required tiles computation for each slice over the three gNBs is computed. It gives: 2, 1 and 1 tile for s_0, s_1 and s_2 respectively. Based on that, the slices serving order is defined. The slices are sorted in decreasing order based on the computed minimum, i.e. $S^o = \{s_0, s_1, s_2\}$. Therefore, the minimum tiles are allocated consecutively for each slice in S^o as shown in fig 4.19. For instance, as slice s_0 minimum demand is two tiles and it is the first served slice, both tiles are allocated at the bottom left of each gNB (see fig 4.18). Once the three slices minimum required resources are served, the table 4.4 is updated. The minimum is again computed and the slices are re-sorted. In this step, the minimum tiles for s_0, s_1 and s_2 are 2, 1 and 3 tiles respectively. Therefore, the slices ordered set is

$S^o = \{s_2, s_1, s_0\}$. Three tiles of s_2 are allocated on both b_1 and b_2. Then, two tiles are allocated for s_0 on b_2. Later, s_1 tile is inserted in b_0 (see fig 4.19). The update of table 4.4 gives null values, i.e. all slices requests are served. During the whole process, a lower skyline is ensured.

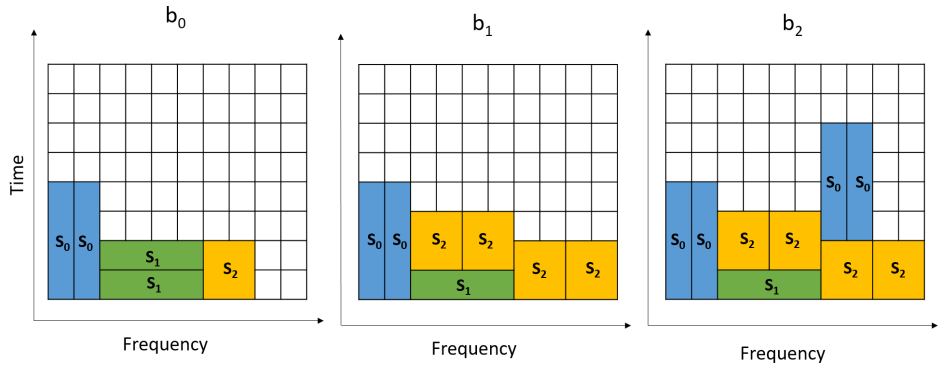


Figure 4.19 – HMF Heuristic example.

4.6.4 TTR computation

Once the allocation is performed, an evaluation of the total amount of tied sRBs, the largest continuous unallocated space in a grid and convergence time is prominent. The LCUS and CT are computed as explained in section 4.5.1. With ESRP models, the TTR was exactly the objective score. With the heuristics, the computation of TTR is required once the allocation finishes. For that, we propose an algorithm to count the total tied sRBs over the gNBs.

An sRB is considered as tied if it is allocated to same slice over the involved gNBs. In fact, each slice requests a different amount of resources on each gNB, the maximum tied sRBs between a given subset of gNBs is then equal to the minimum required resources over the same subset. An example includes, a slice s_1 that requires 2 tiles (8 sRBs) on gNB b_1 and 1 tile (4 sRBs) on gNB b_2. If the allocation is optimal, we will have at maximum 1 tied tile for s_1 over b_1 and b_2, i.e. 4 tied sRBs.

From that, given the resource grid with complete allocation, $G_c = g_k^c$, we propose to count the total tied sRBs as summarized in algorithm 4.4 and explained in the following for each slices$s_i \in S$.

1.) compute the total required sRBs for each slice, $Max(s_i) = \sum_{k \in B} \gamma^{\mu}_{s_i,k}$.
2.) select the gNBs where the slice requests the resources, noted B_{s_i}
3.) compute the minimum required resources over B_{s_i}, i.e. $Min(s_i) = \min_{k \in B_{s_i}} \gamma^{\mu}_{s_i,k}$.
4.) compute the tied sRBs from $Min(s_i)$ without redundancy.

5.) update $Max(s_i)$ and repeat from step 2.

6.) repeat from 2 until the total required sRBs is reached or there is no minimum sRBs between any gNBs to be tied.

Algorithm 4.4 Total tied sRBs (TTR)

Input: G_c, S, B, Γ.
Output: Total tied sRBs over B.
for `each slice` $s_i \in S$ **do**
  Compute $Max(s_i) = \sum_{k \in B} \gamma^{\mu}_{s_i,k}$
  Compute the $Min(s_i) = \min_{k \in B} \gamma^{\mu}_{s_i,k}$
  while $Max(s_i) \geq 0$ and $Min(s_i) \neq 0$ **do**
    $B_{s_i} \leftarrow \{b_k, k \in B \text{ where } \gamma^{\mu}_{s_i,k} \neq 0\}$
    $Min(s_i) = \min_{k \in B_{s_i}} \gamma^{\mu}_{s_i,k}$
    count θ_j from $Min(s_i)$ and check the non redundancy.
    update $Max(s_i) \leftarrow Max(s_i) - 4 * Min(s_i)$
  end while
  $\Theta_{s_i} = \sum_{j \in \zeta_{s_i}} \theta_j$
end for
$\chi = \sum_{s_i \in S} \Theta_{s_i}$

This algorithm is validated based on the ESRP models. For all the instances of ESRP simulation runs, the achieved optimal TTR by ESRP is compared to the one computed by algorithm 4.4.

4.7 Heuristics Evaluation

In this section, the performance evaluation of the three heuristics is performed. Mainly, a comparison based on convergence time (CT), total achieved number of tied sRBs (TTR) and largest continuous unallocated space (LCUS) is conducted. For that, the heuristics achieved TTR is compared with the optimal TTR given by ESRP models. In section 4.5, it is shown that quite similar results are given by both ESRP versions and that ESRP-v1 outperforms smoothly ESRP-v2. Only ESRP-v1 scores are then considered in this part. Regarding the LCUS, although the skyline has reached the optimal LCUS for the different B and S sizes, the comparison between the three heuristics LCUS is performed with respect to the LTUB.

Furthermore, for all evaluations, the impact of the gNBs number and the served slices is also investigated. The frequency bandwidth is fixed to 5 MHz, hence $N_r = 27$. The allocation window is over 10 ms, i.e. $T = 40$. For each test, the number of slices requests and the number of adjacent BSs are fixed. For each simulation run, the slicing profile Γ is generated randomly as to have at maximum 80% of the grid usage. 100 independent simulation runs are performed for each test. The model feasibility is tested for each simulation run. In case of its feasibility, the

same instances are then used for both optimal models as well as heuristics. The time limit for the ESRP models is fixed to 600 s. In this part, only some cases are analyzed, the reader might refer to annex B for other results.

4.7.1 TTR Analysis

One of the objectives is the maximization of the total number of tied sRBS (TTR) over a set of BSs. Hence, an evaluation of the achieved TTR by the three heuristics is intended. It is compared to optimal TTR score given by ESRP-v1 model.

We start by analysing the heuristics performance for low number of interfering gNBs, 2 and 3 gNBs. For that, fig 4.20 plots the violin of the achieved TTR by each algorithm. Each violin shape reflects the TTR distribution of a given algorithm. On each distribution plot, the white dot in the middle represents the TTR median value. While the black bar inside each violin plot refers to the interquartile range.

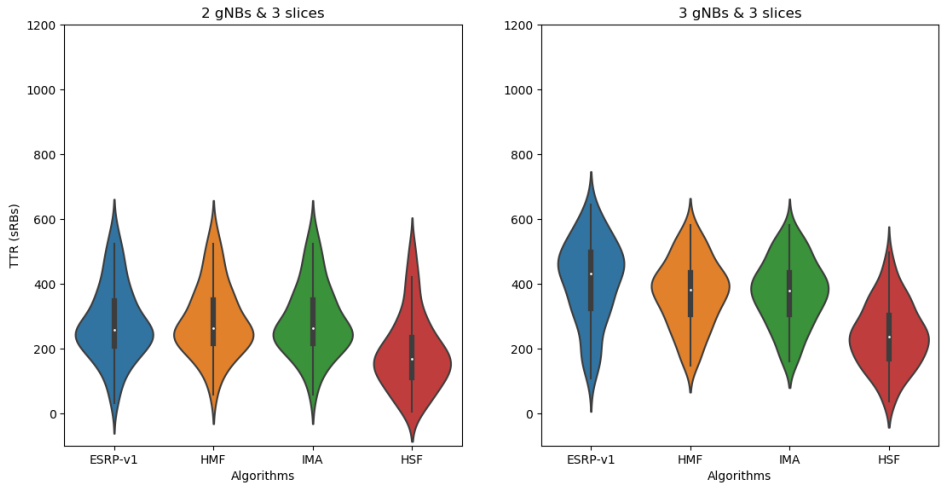


Figure 4.20 – Achieved TTR (s) distribution over 2 gNBs and 3 gNBs with 3 served slices.

The left figure in fig 4.20, exhibits the TTR distribution when only 2 gNBs are present in the gNBs set and 3 slices are served during T. It appears that the ESRP-v1, HMF and IMA achieve similar results, as they have similar TTR distribution and median. The TTR values are concentrated around the median for the three algorithms, i.e. 240 sRBs. Hence, IMA and HMF attain the optimal TTR. The HSF has marked a reduced TTR median value compared to the three algorithms. Its TTR distribution is centred around 170 sRBs. But it is still considered as good performance, as it reaches 70% of the optimal score.

On the right figure, The gNBs set is increased to three. A smooth difference is remarked between IMA, HMF and ESRP TTR scores distribution. IMA and HMF still outperform HSF. The IMA and HMF achieve 89% of the optimal TTR score, whereas HSF marks only 56% of optimality. In addition, the median TTR

has increased for all algorithms with the gNBs set size augmentation. Particularly, 240 is the median TTR with 2 gNBs and 430 is the one reached in the case of 3 gNBs. It is therefore interesting to analyse the impact of gNBs set size on the TTR variation.

4.7.1.1 Impact of B size

Fig 4.21 illustrates the TTR optimality gap achieved by the 3 algorithms as a function of B size. The optimality gap (OG) is obtained from the difference between the optimal score given by ESRP-v1 and the achieved score by a given algorithm divided by the optimal score. It refers to the gap between the reached score and the optimal one. IMA and HMF have quite similar results over the various B set sizes, i.e. their curves are superposing. IMA and HMF reach the optimality for lower B set sizes when serving 3 or 5 slices, i.e. OG=0%. Then, their scores decrease proportionally to B size augmentation. Particularly, the optimality gap with 3 gNBs is 0.9% and 48% with 9 gNBs when serving 3 slices. Both algorithms outperform HSF for B sizes lower than 9 gNBs. The HSF optimality gap ranges from 33% to 49% with 2 gNBs and 13 gNBs respectively, when 3 slices are served (see left figure 4.21). For higher slices number (right figure), a smooth increase in terms of OG is observed. It is therefore interesting to analyse the impact of S size on the heuristics OG with respect to TTR.

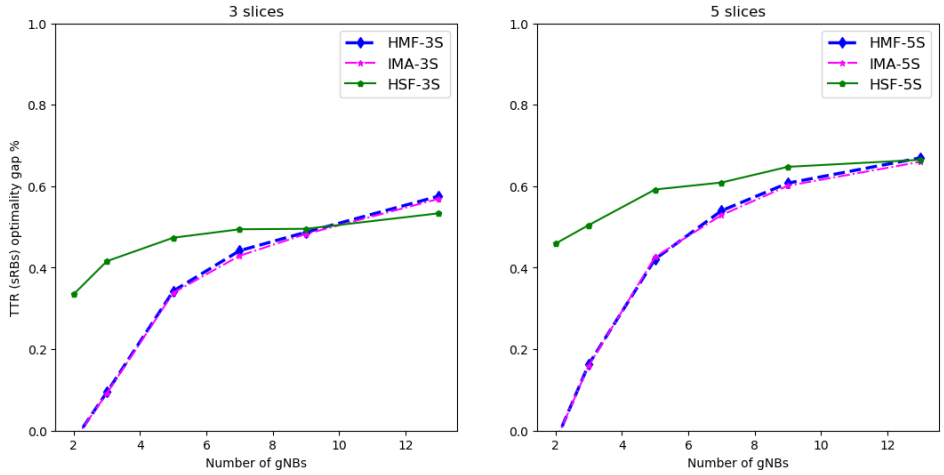


Figure 4.21 – Optimality gap of the achieved TTR (sRBs) by the three algorithms as a function of B size for different S sets

4.7.1.2 Impact of S size

The NS proposes the possibility to create slices on the fly. Moreover, the allocation task is forecasted to be challenging with high number of slices. As each slice requires a different amount of resources on each gNBs. Thus, it is primordial to investigate the impact of their variation on the achieved TTR. Fig 4.22 shows the

impact of S size on the reached TTR in the system. The OG is represented for two system sizes: 3 and 5 gNBs.

The OG increases when moving from a system serving 3 slices to the one with 7 slices. Then, the OG is quite stable around 22% with 3 gNBs and 47% with 5 gNBs with both IMA and HMF. It can be concluded that HMF and IMA become insensitive to larger S set size starting from 7 slices. It is mainly the B set size that has an impact on the TTR score. The HSF has higher OG compared to HMF and IMA. Its OG score ranges between 47% and 74% with a system of 5 gNBs. Thus, IMA and HMF always outperform HSF in the case of 3 gNBs as well as 5 gNBs.

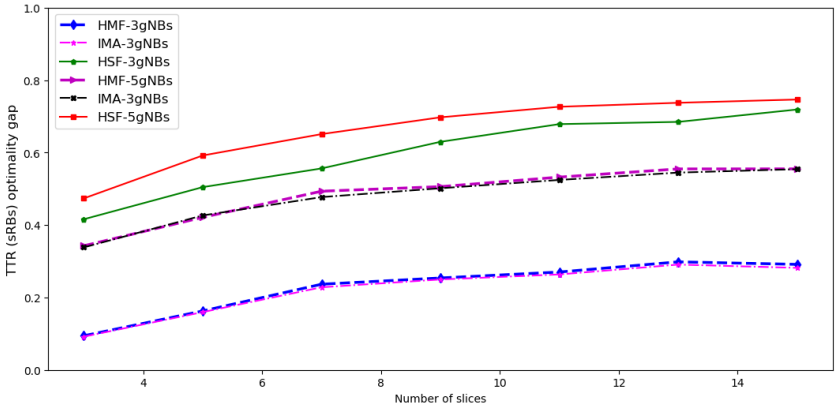


Figure 4.22 – Optimality gap of the achieved TTR (sRBs) by the three algorithms as a function of served slices for different B size.

4.7.2 CT Analysis

The convergence time is a vital performance metric, as the objective is the real time slicing enforcement. In this part, an analysis of the CT is conducted. Especially, the impact of the B and S sizes growth is investigated.

4.7.2.1 Impact of B size

Fig 4.23 shows the convergence time of the three heuristics in seconds as function of the B size for two use cases: system with 3 and 5 slices. The three heuristics converge quickly at a time scale of hundreds of milliseconds. IMA and HMF have similar CT in both case studies. Over all the tested B sizes, the convergence time increases proportionally with B size growth. It expands gradually in the order of milliseconds. For small B set size, the CT is less than 10 ms. Nevertheless, for the higher B size superior to 9, the CT is quite stable when it serves 3 slices. It is around 0.53 s.

The HSF converges rapidly compared to IMA and HMF over all the gNB sets. This could be explained by the repeated minimum computation and sorting instructions in both algorithms. For higher slice set size, the same evolution of CT

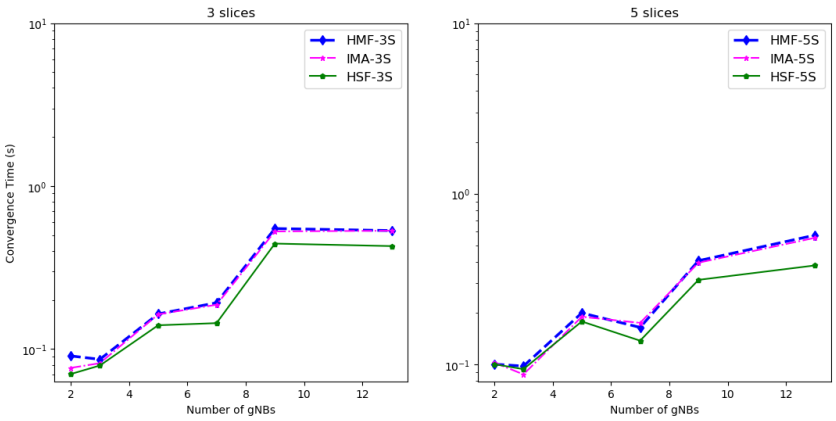


Figure 4.23 – Convergence time (s) performance evaluation over varying B set size.

is observed. Interestingly, the HSF CT is lower when serving 5 slices compared to the system serving 3 slices. Therefore, the impact of slice set size on the CT is investigated next.

4.7.2.2 Impact of S size

In this section, the impact of S size on CT is analysed. For that, fig 4.24 plots the three heuristics CT in function of S size with a system composed of 3 gNBs and 9 gNBs, left and right figures respectively. From both use cases, system with 3 and 5 gNBs, the HMF and IMA converge in similar time granularities. HSF marked lower CT than both IMA and HMF. The CT variation is independent of the S size. It varies in small interval size. Notably, for a system with 3 gNBs, the IMA CT ranges between 0.04 s and 0.11 s.

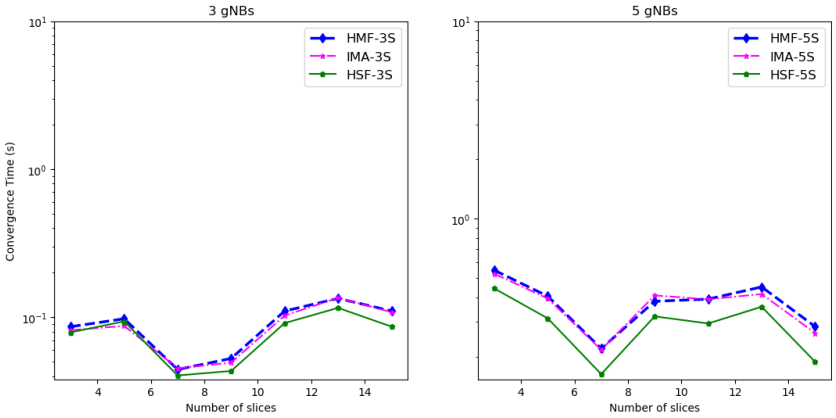


Figure 4.24 – Convergence time (s) for all algorithms over varying S size with different B sets.

4.7.3 LCUS Analysis

The second objective is to maximize the continuous unallocated resources over a set B. For that after allocation, the LCUS is computed over B as described in 4.5.1.3. The heuristics performance (i.e. HSF, IMA and HMF) are compared with LTUB.

Let analyse the models performance regarding the LCUS for a basic set of gNBs, 2 and 3 gNBs. The number of served slices during T is fixed to three. Fig 4.25 shows the LCUS score distribution by each algorithm and the LTUB, for a system with 2 gNBs and 3 gNBs, left and right figures respectively. All the algorithms achieve similar LCUS score for both B sets, i.e. similar LCUS score distribution. The median for a system with 2 gNBs and 3 slices is 1310 sRBs.

On the other hand, with the B set size growth, the LCUS score increases for all models. This is because, the LCUS is summed over the gNBs of the set B. Therefore, more gNBs in the set, higher is the LCUS.

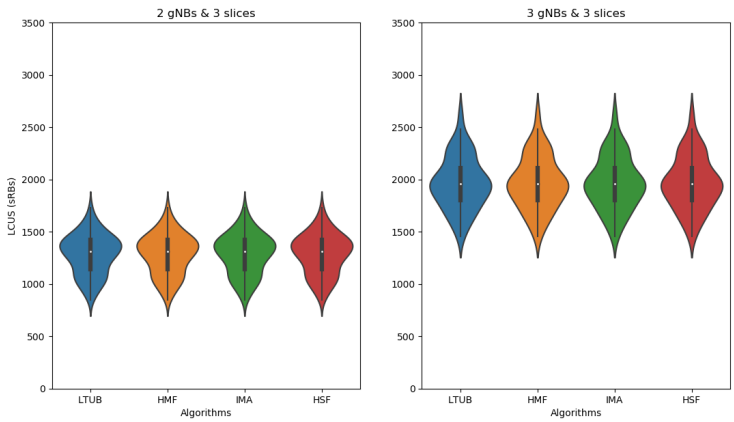


Figure 4.25 – LCUS performance evaluation over varying Bs set size.

In the following, a study of the LCUS variation as a function of B and S sizes is addressed.

4.7.3.1 Impact of S size

The number of served slices is pivotal for the RAN slicing enforcement. Therefore, the MNO have to allocate efficiently the available resources to the accepted slices. Accordingly, the LCUS allows the MNO to scale up/down the resources for the current served slices or even accept new slices requests.

Fig 4.26 highlights the impact of S size on the LCUS score in the system. Two gNBs set sizes are considered, 3 and 5 gNBs. The three heuristics achieve the optimal LCUS score given by LTUB. This is expected, as the skyline is used for the allocation. With S variation in both case studies, the LCUS also varies in a small interval. This variation is independent of the S set size evolution. For instance,

with 3 gNBs, the LCUS ranges between 1937 sRBs when serving 9 slices and 2005 sRBs with 7 slices.

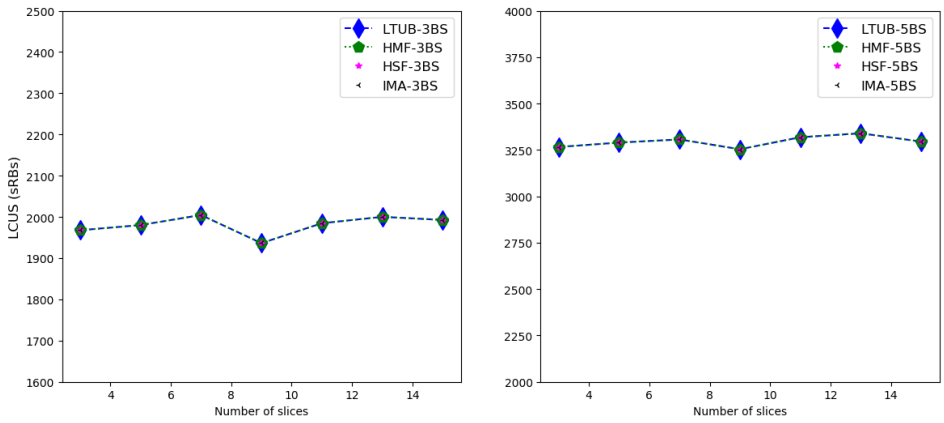


Figure 4.26 – Achieved LCUS (sRBs) as a function of served slices for different B size.

4.7.3.2 Impact of B size

The growth of B size is inevitable in the 5G context. Therefore, it is important to analyse the approaches performance in terms of LCUS with respect to the B size growth. Fig 4.27 plots the total unallocated space over each B set size. All heuristics achieve the optimal LCUS score over the different B set sizes, i.e. their curves are superposing. It is observed that larger B sets allows a higher gain in terms of LCUS by all approaches. The LCUS score is then proportional to the B set size. Therefore, it could be considered that HMF, IMA and HSF LCUS score are insensitive to the S size variation.

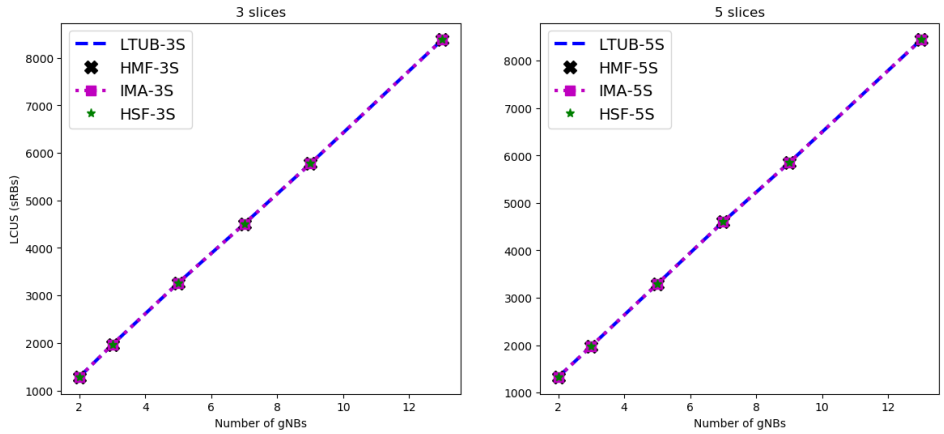


Figure 4.27 – Achieved LCUS (sRBs) as a function of B size for different Slices sets.

This is advantageous as the SD-RAN will have the possibility to scale-up/ down the radio resources for each gNB based on the slices traffic distribution over R.

4.7.4 Discussion

With the multi-objective criterion of the allocation strategy, i.e. maximization of TTR and LCUS, each objective is modeled separately. ESRP models the TTR maximization while skyline approximates the LCUS. In the previous parts, an evaluation of all algorithms is realized. Regarding the convergence time (CT), both ESRP versions converge at time scale of hundreds of seconds. Even though the ESRP-v1 marks a lowest CT compared to ESRP-v2, but the CT for both models reaches the time limit (600s) for almost all the tests. On the other hand, skyline has achieved a lower CT to achieve the optimal LCUS. The CT is at granularity of hundreds of seconds.

With the objective to enforce the real time RAN slicing, the ESRP models have shown their limits in terms of convergence time. Thus, it implies their non adaptability for real time allocation. Nevertheless, they might help the SD-RAN controller for large-scale decisions. Contrarily, the skyline demonstrates its capability to converge quickly with an optimal score over different BSs and Slices set sizes. From that, we proposed to use a solution based heuristics to resolve the MOOP. The LCUS is prioritized while the TTR is achieved at best. Three heuristics are developed, i.e. IMA, HMF and HSF. A comparison between all the algorithms to enforce the RAN slicing is fulfilled. The key metrics are the CT, TTR and the LCUS.

Contrarily to the ESRP models, the three heuristics converge at time scale of milliseconds. For lower B and S sizes, the CT is in the order of 10 ms. The CT increases smoothly with larger B set size. But, it doesn't outpace 0.7 s, 0.65 s, and 0.47s when tested with HMF, IMA and HSF respectively in a large B set of 13 gNBs serving 15 slices simultaneously. From that, these heuristics demonstrate their capability for real time deployment within the 5G SD-RAN.

Further, the HMF, HSF and IMA heuristics are compared to the optimal/upper bound solution given by ESRP-v1 for the TTR score. For the LCUS, the heuristics are compared with the upper bound LCUS (LTUB). For a small set of gNBs and slices, e.g. 2 or 3 gNBs with 3 slices, the IMA and HMF achieve quite similar results in terms of TTR as the optimal score given by ESRP-v1. The HSF has lowest scores in this case, but only a very small gap from optimality, i.e. 30 %. The optimal LCUS is reached by all of HMF, HSF and IMA for similar case study. With the CT scored for such case (small B size up to 5 gNBs) in the order of 10 ms. The IMA and HMF are highly enforcing the RAN slicing for real time system deployment by reaching the optimality for TTR and LCUS in very small time scale. This could be the case of macro cells deployment, as well as a small group of other cells type covering a specific geographical zone.

The 5G NS vision includes the dynamic creation of slices over time. Thus, higher S size might be carried by the SD-RAN over time. Regarding the TTR, the

IMA and HMF have shown insensitivity to the S size larger than 7 slices. In other words, higher S size doesn't have a big impact on both the tested algorithms. This is advantageous for the 5G RAN enforcement.

Another system parameter is the number of gNBs. With the optimal models, i.e. ESRP, it is observed that TTR optimal/upper bound increases with higher B sizes. The HSF follows the same change. The IMA and HMF performance degrade with larger B sets. Nonetheless, the worst case study with 13 gNBs serving 15 slices, at least 32% of the optimal TTR is reached by both heuristics. This score might be higher as the ESRP doesn't converge within the time limit for this instances, and then the comparison is conducted with the upper bound TTR. Nevertheless, this score still advantageous as the heuristics prioritize the LCUS at the expense of TTR. In fact, the highest B size produce the highest LCUS when applying both heuristics. It corresponds exactly to the optimal LCUS marked by LTUB. Thus, they lead to an allocation without resources waste. In fact, the LCUS increases with the B size growth. This prioritization is intended because of the crucial task of efficient resource allocation required by the MNO.

In summary, although the ESRP models give the optimal allocation with higher total tied sRBs, their high convergence time and non assurance of resources efficient usage make their real time deployment questionable. The IMA, HMF and HSF heuristics achieve a good results in terms of TTR, CT and largest space for lower B sets. This proves the possibility of their real time deployment for such cases. The growth of B size allows a larger continuous unallocated space at the expense of TTR with all the developed heuristics. Even though this priority prospect, the TTR is assured at best by HSF, IMA and HMF. The HMF and IMA are outperforming the HSF. Thus, the slice owner could use the tied resources to enable the advanced transmission schemes for the critical transmissions. Moreover, all heuristics highly enforce the RAN slicing with respect to the resource orthogonality, satisfaction, scalability and enabling cooperation requirements. In fact, the orthogonality, satisfaction and scalability are guaranteed, while the enabling requirement is assured at best.

4.8 Conclusion

The RAN slicing comes with challenging requirements such as resources isolation, slices satisfaction, programmability and the cooperation enabling. In this work, we aimed to enforce it from resource perspective in the 5G context. For that, we have formulated the problem as a multi-objective optimization to allocate efficiently the slices resources with respect to the diverse RAN slicing requirements. The first objective addresses the programmability of the RAN slicing through the maximization of the largest continuous unallocated space on each gNB resource grid. Then, the second objective handles the cooperation enabling requirements by means of resource allocation in similar position over frequency and time for a given slice over the set of gNBs. The second objective involves a tight management

of resources. Therefore, a resource grid decomposition is proposed as to have a fine grained resources monitoring. Both slices orthogonality and satisfaction are guaranteed by means of constraint.

With the multi-objective criterion, the optimal solution for each objective is targeted. Two mathematical models are developed for the first objective, whereas the second objective is tackled as a 2D bin packing optimization problem. An heuristic is then used to approximate rapidly the optimal score, as the problem is known to be NP-hard. As well as the upper bound LCUS solution is computed. The NP-hardness of the optimal models converge slowly, which limits their deployment for real time use cases. Nevertheless, they could be advantageous for the SD-RAN large scale decisions.

Therefore, three heuristics are implemented with the aim to enforce the allocation strategy for the RAN slicing. The programmability is prioritized with these heuristics at the expense of the enabling cooperation requirement. All the algorithms are evaluated in terms of convergence time, total tied resources and largest continuous unallocated space.

Contrarily to the optimal models, the developed heuristics, i.e. IMA, HMF and HSF achieve good results in different case studies. Especially, for lower set of gNBs, the IMA and HMF reach the optimal scores for both tied resources and LCUS with a very low convergence time in the order of 10 ms. In such case the four RAN slicing requirements are guaranteed. Moreover, all the tested algorithms show insensitivity to the number of served slices during the allocation window. Such results encourage the real time deployment test for the three approaches.

Conclusion and future work

4.5 Conclusion

Several sectors have high expectations from cellular networks 4G/5G to support their sophisticated services/applications. They come with various performance requirements and huge traffic amounts both in upload and download. This is challenging for this type of wireless networks. This thesis addresses this double challenge and proposes two related contributions:

The first contribution is the intelligence integration in cellular networks through the estimation of the users instantaneous uplink throughput at small time granularities. For that, a scalable estimation model with underlying machine learning techniques is proposed. Then, a real time 4G testbed has been deployed with various radio phenomena to reproduce the wireless channel effects on throughput. And, an exhaustive eNB lower layers metrics benchmark is then fulfilled to build representative datasets. The estimation model is tested with three machine learning techniques, i.e. LR, RF and SVR, based on the built datasets. The forecast window ranges between 100 ms to 1 s. Accurate estimations are reached with various datasets incorporating cross-layers eNB metrics, i.e. errors less than 15%. It is also concluded that radio measurements are not sufficient for small time scale throughput estimations. Good estimations are achieved for higher time granularities from 700 ms.

The second contribution is the 5G RAN slicing enforcement at resource level from multi-cell perspective. This is because the 5G is expected to support the various applications/services performance requirements through slicing the network. The core network slicing is achieved by incorporating the existing cloud approaches. Contrarily, the RAN involves various requirements hindering its deployment, especially at the resource level. The main requirements can be summarized into: slices orthogonality, satisfaction, scalability and cooperation enabling. An exact optimization model with constraint programming is developed to enforce the slicing with respect to the orthogonality, satisfaction and cooperation enabling needs. And, a 2D bin packing heuristic is adapted for the enforcement of scalability, orthogonality and satisfaction at the expense of cooperation enabling need. The developed model converges slowly for large instances. Nevertheless, it is usefull for large time scale allocations. Further, three heuristics are proposed to enforce the four RAN slicing requirements with a prioritization of scalability over the cooperation enabling. Results show good performance with two heuristics. They are then highly enforcing the real time RAN slicing deployment.

4.6 Future work

The extension of this thesis work is discussed in this part.

4.6.1 Toward smart systems

The emergence of cellular network softwarization eases the intelligence integration in their systems as the network is becoming more and more flexible. Our first contribution can be highly merged in this trend to design effective systems. Particularly, the instantaneous throughput estimation can be taken as input argument in advanced schedulers, congestion control and avoidance mechanisms. Also, with the SDN vision, the proactiveness is strongly advantageous. The controller is the decision maker for traffic management over the network.

4.6.2 5G platform for slicing evaluation

The evaluation of the developed slicing heuristics and the upcoming ones on real time communication is prominent. Unfortunately, there is no open access platform to test eventual slicing techniques at RAN resource level. OpenAirInterface (OAI) on the other hand is chalking the road for 5G. With that, an extension of the deployed testbed in this work to support also the 5G is expected. For that, a development of an open source platform upon 5G OAI is planed. The platform is envisioned to setup multiple users with OAI as to emulate the versatile 5G users. Further, the platform would be open to the research community for a remote access in order to test/validate their advanced slicing ideas.

4.6.3 Slicing enforcement: A tailored allocation strategy

The scarcity of radio resources and the complex radio environment are restricting the 5G RAN slicing deployment. The isolation, satisfaction, scalability and cooperation enabling are the main RAN slicing requirements. The achievement of all these principles is challenging. In this work, the proposed heuristics prioritize the scalability at the expense of the cooperation enabling with respect to the other requirements. In this section, we discuss a potential extension of this allocation strategy to enforce the cooperation enabling as well.

With such aim, one of the potential allocation strategies is the enhancement of the proposed approach for the scalability. Instead of letting the largest continuous unallocated space for further reuse, the unallocated space could be sparse in an efficient way. For this, the small unallocated resources portion should fit at minimum a tile structure. In other words, the allocation should aim the maximization of the tied tiles, with a verification that each chunk of unallocated resources can fit another slice need. Of course, even in this case the trade-off between scalability and cooperation enabling might arise. For that, an aggregation heuristic is the ultimate solution, where the user can define a weight for each requirement based on the network needs. The shape of sparse unallocated resources will highly depend on the slices needs evolution over time. For that, a forecast of the slices traffic demand will be beneficial as the SD-RAN would have the possibility to prioritize a given tile shape among others. Particularly, if a traffic demand of slice with a numerology $\mu = 1$ is expected to increase, it is more advantageous to let higher

percentage of sparse resources with square form instead of rectangle. Eventually, this strategy entails a tracking of those available resources and verification during each tile allocation. Such task might increase the convergence time. Therefore, time limitation should also be considered when designing the allocation to enforce the real time deployments.

4.6.4 Large scale RAN slicing: potential strategies

For the slicing system design, we have been focused on a given controlled zone by an SD-RAN. The extension of this work over a large scale with multiple SD-RANs is intended. For that, mainly two potential designs are designated, cooperative and coordinated slicing as discussed in the following. In both cases, the system is envisioned with numerous SD-RANs controlling adjacent zones.

4.6.4.1 Coordinated slicing

This approach is based on a coordination between SD-RANs before the allocation process, where each SD-RAN shares the slices traffic distribution over its zone, and one SD-RAN is elected to fulfill the first allocation of its traffic. The selection of this SD-RAN, labelled SD-RAN reference, might be based on the amount of required resources of all slices or the number of prioritized slices, e.g. slices with critical mission requirements. The basic idea is that once the SD-RAN reference is selected and its allocation is fulfilled, its neighboring SD-RANs can repeat the same allocation structure with respect to their traffic distribution. The developed heuristics can be adapted for this case. It is possible to replicate the same allocation (i.e. slices tiles placement over the resource grid), but it might lead to non accomplishment of the satisfaction requirement, especially, when a slice demand is higher on a given zone than the SD-RAN reference zone. Also, the NS customization might occur for a slice with lower demand compared to the SD-RAN reference.

With this strategy, it is mainly the SD-RAN reference zone that gain from the coordination, as the RAN slicing would be highly enforced in this zone. For the other SD-RANs areas, the RAN slicing might be less effective. Nevertheless, it is a simple and rapid strategy to deploy at large scale.

4.6.4.2 Cooperative slicing

Instead of choosing one SD-RAN reference, the SD-RANs can communicate with each other as to find an optimal allocation over the large zone. In fact, the slice resource allocation is critical mainly at the cells at the boarder of an SD-RAN zone. With this, an allocation strategy have to consider these cells in the RAN enforcement process. Another requirement would then be added. It concerns the tied tiles between adjacent gNBs under the control of different SD-RANs. This cooperation could be enabled either with a centric unit managing of all the involved SD-RANs or an interface development for communication between SD-RANs. The

former prerequisites the transfer of all the local SD-RANs information to the central unit. Apart from the huge signaling exchange, it might be time consuming. The latter involves a decentralized decision over the adjacent SD-RANs. In other words, each adjacent SD-RAN zones forms a group. Thus, the signaling would be exchanged only between a given SD-RAN and its neighboring SD-RANs, i.e. cooperation among only the neighbors. This is advantageous as less information transfer can be achieved.

Overall, with this deployment strategy, the slice will have more flexibility to manage its resources over all the large zone efficiently, and can deploy its advanced 5G techniques with an abstraction of the SD-RAN existence.

List of Acronyms

2DBP	2 Dimensional Bin Packing
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
AWGN	Additive White Gaussian Noise
BLER	Block Error Rate
BS	Base Station
CDMA	Code Division Multiple Access
CP	Cyclic Prefix
CDSF	Continental Digital Service France
CoMP	Cooperative Multi-Point
CN	Core Network
CPO	CPLEX CPOptimizer solver
CS/CB	Coordinated scheduling/coordinated beamforming
CT	Convergence Time
D2D	Device-to-Device
DL	Downlink
DSRC	Dedicated Short-Range Communication
E2E	End to End
eNB	evolved NodeB
EPC	Evolved Packet Core network
ESRP	Enforcement of Slice Resources Placement
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
GBR	Guaranteed Bit Rate
HARQ	Hybrid Automatic Repeat reQuest
HDR	Hardware Driver software
HMF	Highest Minimum First
HSF	Highest Slice first
IBSPC	Inter-base station power control
ICI	Inter-Cell Interference

ICIC	Inter Cell Interference Coordination
IMA	Iterative Minimum Allocation
ISM	Industrial, Scientific and Medical
ISI	Inter-Symbol Interference
IoT	Internet of Things
ITS	Intelligent Transport System
JT	Joint Transmission
KPI	Key Performance Indicator
LCUS	Largest Unallocated Space
LR	Linear Regressor
LTE	Long Term Evolution
LTE-A	Long Term Evolution-Advanced
LTUB	LCUS Topmost Upper Bound
MAC	Medium Access Control
MBR	Maximum Bit Rate
MIMO	Multiple Input Multiple Output
MLT	Machine Learning Techniques
MNO	Mobile Network Operator
MOOP	multi-objective optimization problem
NB-IoT	Narrow-band Internet of Things
NS	Network Slicing
NVS	Network Virtualization Substrate
OAI	OpenAirInterface
OFDM	Orthogonal Frequency Division multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PAPR	Peak to Average Power Ratio
PHY	Physical layer
PRB	Physical Resource Block
QCI	QoS class identifier
QoS	Quality of Service
RAN	Radio Access Network
RB	Resource Block
RE	Resource Element
RF	Random Forest

RIP	Received Interference Power
RSSI	Received Signal Strength Indicator
SC-FDMA	Single Carrier Frequency Division Access
SCTP	Stream Control Transmission Protocol
SD-RAN	Software Defined Radio
SDR	Software Defined Radio
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
sRB	smallest Resource Block
SPFM	Slicing Profile Feasibility Model
SVR	Support Vector Regressor
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TTI	Transmission Time Interval
TTR	Total Tied sRBs
UDP	User Datagram Protocol
UE	User Equipement
UHD	USRP Hardware Driver software
UL	Uplink
URLCC	Ultra Reliable and low Latency Communications
USRP	Universal Software Radio Peripheral
VR	Virtual Reality
V2N	Vehicle to Network
V2X	Vehicle to everything
V2P	Vehicle to Pedestrians
V2V	Vehicle to Vehicle

Appendix A

Throughput Estimation Evaluation

This annex depicts the eNB lower layer metrics used for building testbed 2 datasets, for the three scenarios, i.e. multipath fading, noise and radio congestion. In addition, the estimation performance of the instantaneous throughput with the three machine learning techniques is plotted.

A.1 Cross Layer metrics

Table A.1 highlights the collected metrics from the eNB lower layers used to build the testbed 2 datasets. For each radio scenario in testbed 2, and for each UE, the following metrics set is collected to constitute the dataset reflecting the UE in the same scenario. For instance, dataset_2_1 contains the depicted metrics for UE1 in scenario 2.

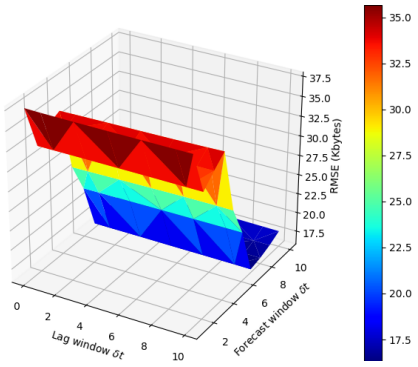
Table A.1 – eNB lower layer metrics used for testbed 2 datasets

Abbreviation	**Parameter**
SNR	Signal to Noise Ratio
CQI	Uplink Channel Quality Indicator
PUCCH_power	PUCCH received power
PUCCH_noise	PUCCH noise power
PUCCH_thr	PUCCH reception threshold power
PUSCH_Rx_pw	PUSCH received power
MPR	Maximum Power Reduction
MPR_correction	Added Correction for MPR
MCS_old	Modulation adn coding scheme from last scheduling
MCS_current	current Modulation and coding scheme
Qm	Modulation order
TBS	Transport Block Size
Timing_advance	Used to control signal delays
Timing_advance_update	Used to control signal delays
Nb_rb	Number of allocated resource blocks after scheduling
Total_allocated_rb	Total allocated resource blocks
HARQ_pid, round, RV, o_r1, O_Ack, A, G, O_RI	From the ulsch decoding/demodulation phases (related to HARQ process)
NDI	New Data Indicator
Nsymb_pusch	Number of symbols at the PUSCH
First_rb	First resource block containing data (ulsch)
First_cce	Control Channel Element
L, LCID, cyclic_shift, Phase_max, num_ce	Extracted during the UE ulsch scheduling process
CRC_status	cyclic redundancy check
Buff_size_IQ	Buffer size containing the IQ data
PDU_length	Length of Protocol Data Unit (MAC) (SDU concatenated)
PDU_length_total	Total Mac PDU bytes
Decoding_time	Time spent to decode the UE ulsch
Buff_size_data	Buffer size containing data (MAC)
SDU_length	Length of Service Data Unit (MAC)
SDU_length_total	Total MAC SDU bytes
Stats_max	Total ulsch bitrate

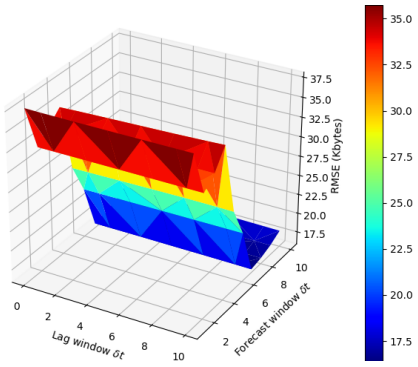
A.2 Error estimations as a function of forecast and lag windows

A.2.1 Estimations based on radio metrics (testbed 1 datasets):

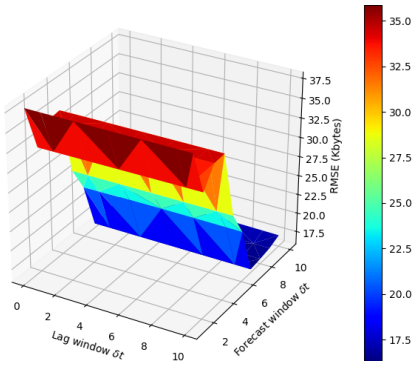
A.2.1.1 LR based Estimations:



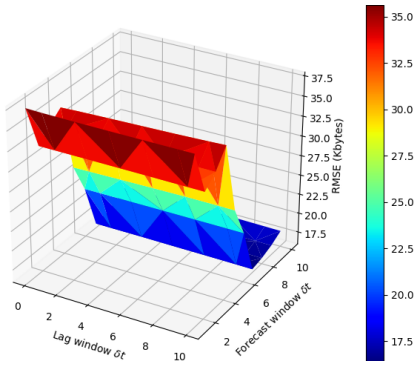
(a) Estimations Based *dataset_RSSI*.



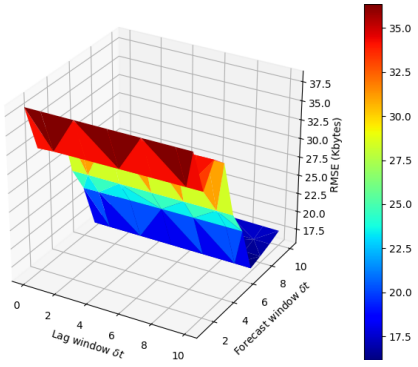
(b) Estimations Based *dataset_RX_power*.



(c) Estimations Based *dataset_RIP*.



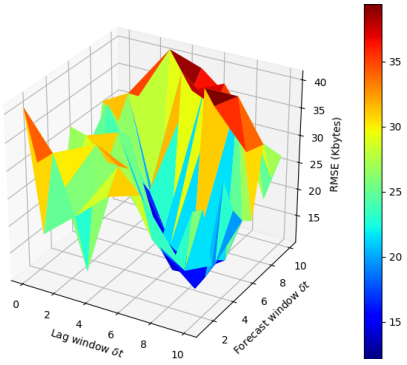
(d) Estimations Based *dataset_All*.



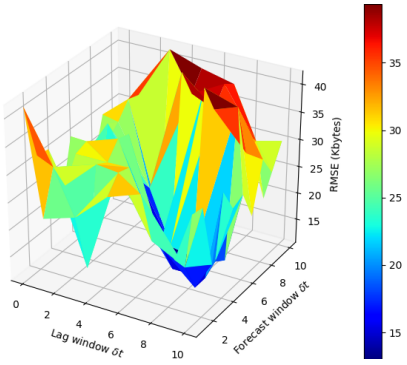
(e) Estimations Based *dataset_SNR*.

Figure A.1 – 3D plot for RMSE as a function of lag and forecast windows for estimations based on *dataset_RX$_p$ower*, *dataset_RSSI*, *dataset_RIP*, *dataset_All* and *dataset_SNR* with LR as the underlying MLT.

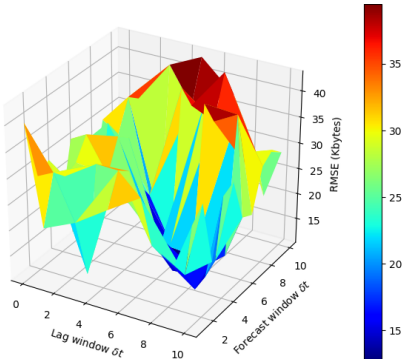
A.2.1.2 SVR based Estimations:



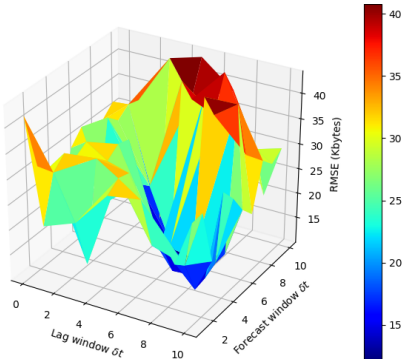
(a) Estimations Based *dataset_RSSI*.



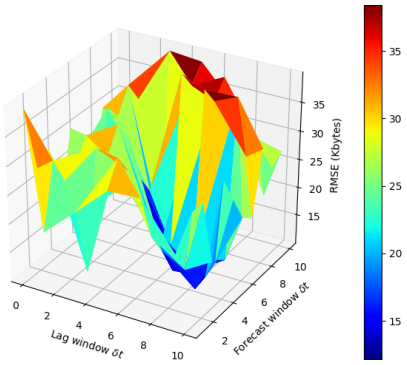
(b) Estimations Based *dataset_RX_power*.



(c) Estimations Based *dataset_RIP*.



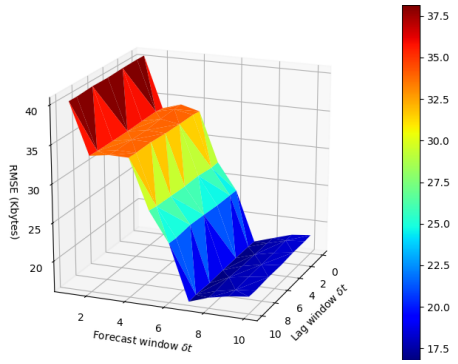
(d) Estimations Based *dataset_All*.



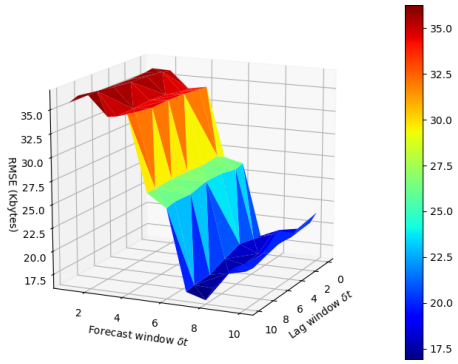
(e) Estimations Based *dataset_SNR*.

Figure A.2 – 3D plot for RMSE as a function of lag and forecast windows for estimations based on *dataset_RX_power*, *dataset_RSSI*, *dataset_RIP*, *dataset_All* and *dataset_SNR* with SVR as the underlying MLT.

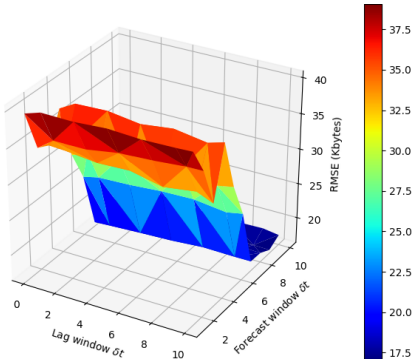
A.2.1.3 RF based Estimations:



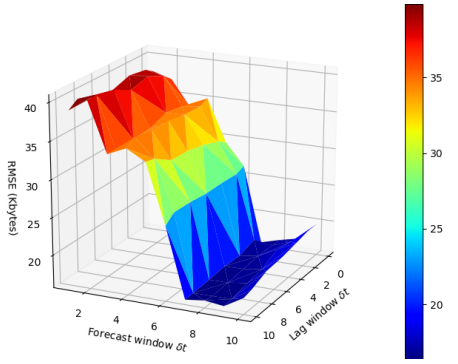
(a) Estimations Based *dataset_RSSI*.



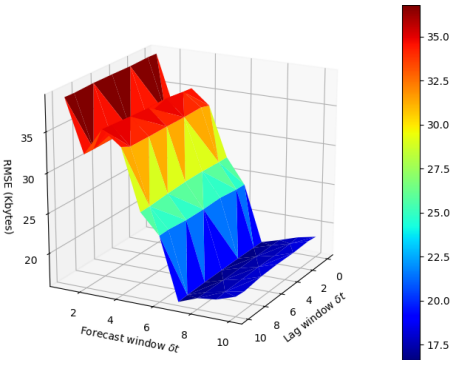
(b) Estimations Based *dataset_RX$_p$ower*.



(c) Estimations Based *dataset_RIP*.



(d) Estimations Based *dataset_All*.

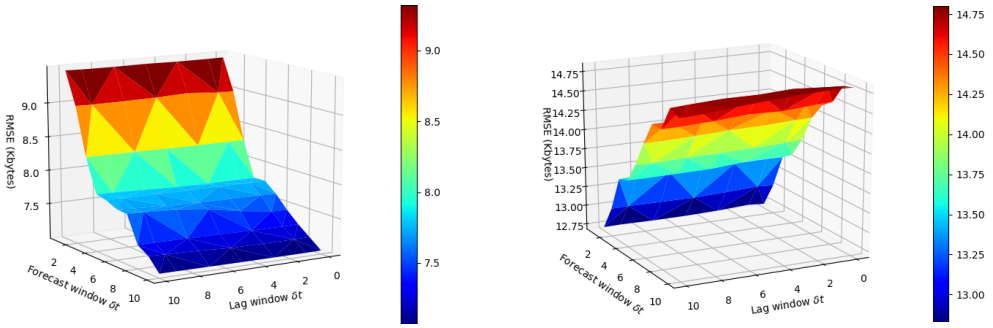


(e) Estimations Based *dataset_SNR*.

Figure A.3 – 3D plot for RMSE as a function of lag and forecast windows for estimations based on *dataset_RX$_p$ower*, *dataset_RSSI*, *dataset_RIP*, *dataset_All* and *dataset_SNR* with RF as the underlying MLT.

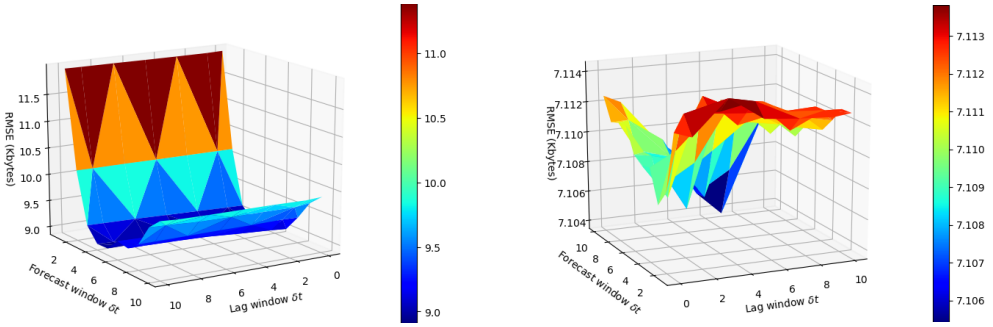
A.2.2 Estimations based on cross-layers metrics (testbed 2 datasets):

A.2.2.1 LR based Estimations:



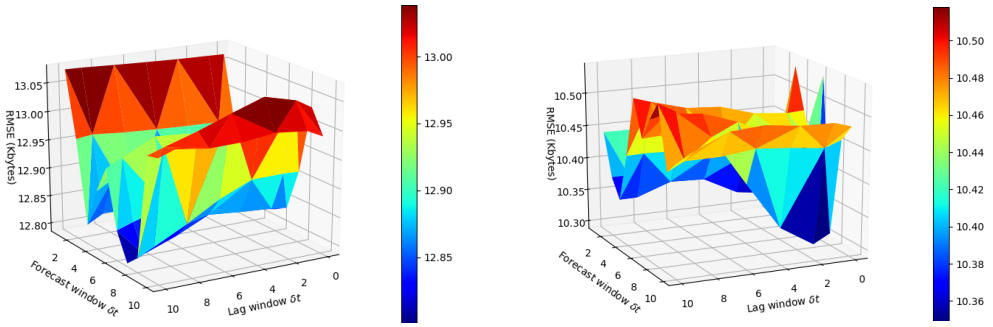
(a) Estimations Based *dataset*_2_1.

(b) Estimations Based *dataset*_2_2.



(c) Estimations Based *dataset*_3_1.

(d) Estimations Based *dataset*_3_2.

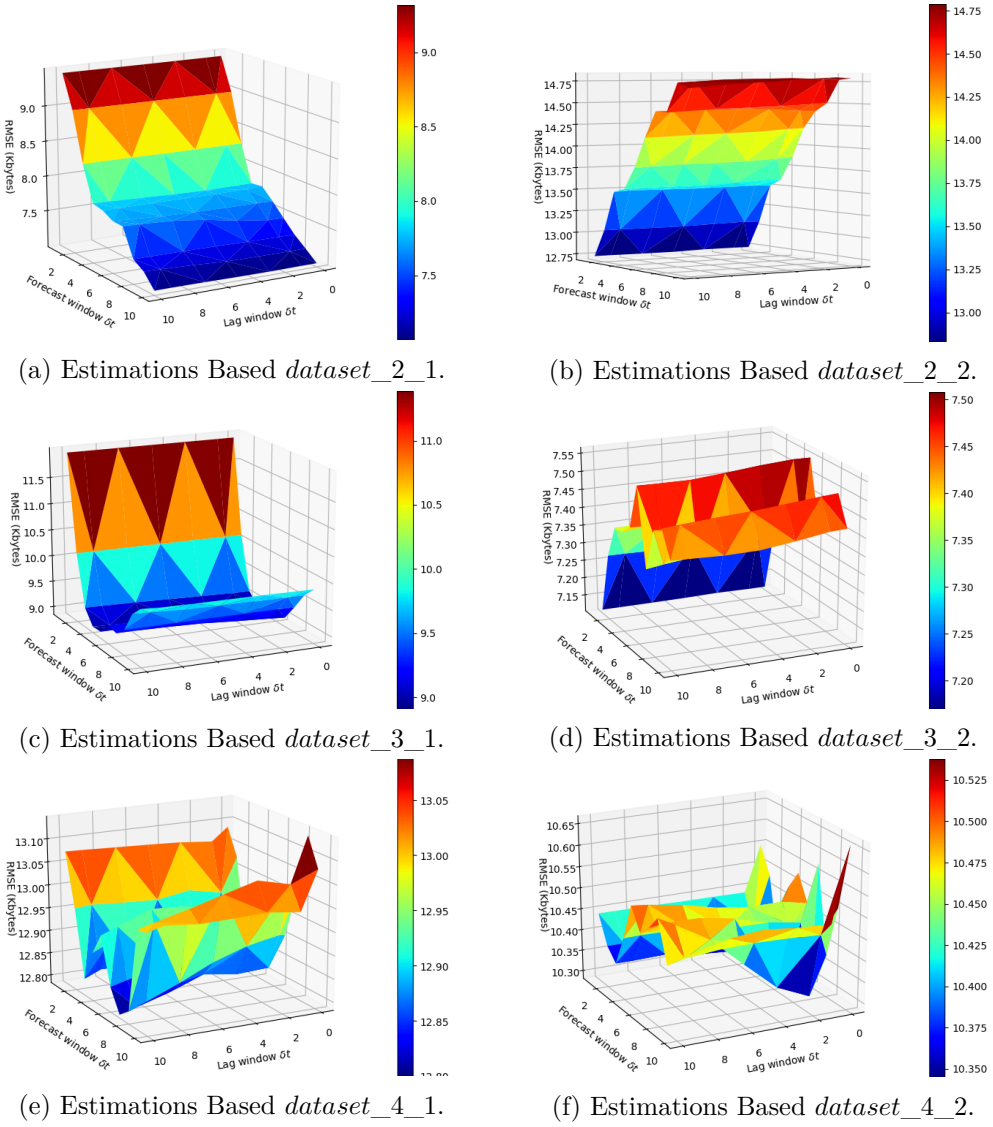


(e) Estimations Based *dataset*_4_1.

(f) Estimations Based *dataset*_4_2.

Figure A.4 – 3D plot for RMSE as a function of lag and forecast windows for estimations based on *dataset*_2_1, *dataset*_2_2, *dataset*_3_1, *dataset*_3_2, *dataset*_4_1 and *dataset*_4_2 with LR as the underlying MLT.

A.2.2.2 SVR based Estimations:



(a) Estimations Based *dataset*_2_1.

(b) Estimations Based *dataset*_2_2.

(c) Estimations Based *dataset*_3_1.

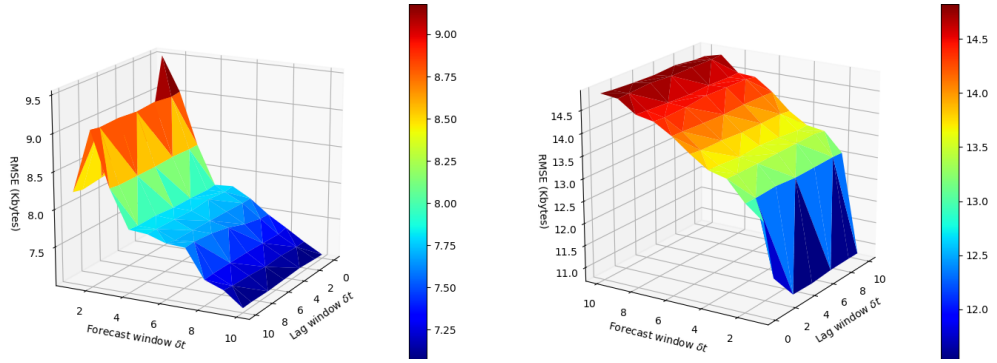
(d) Estimations Based *dataset*_3_2.

(e) Estimations Based *dataset*_4_1.

(f) Estimations Based *dataset*_4_2.

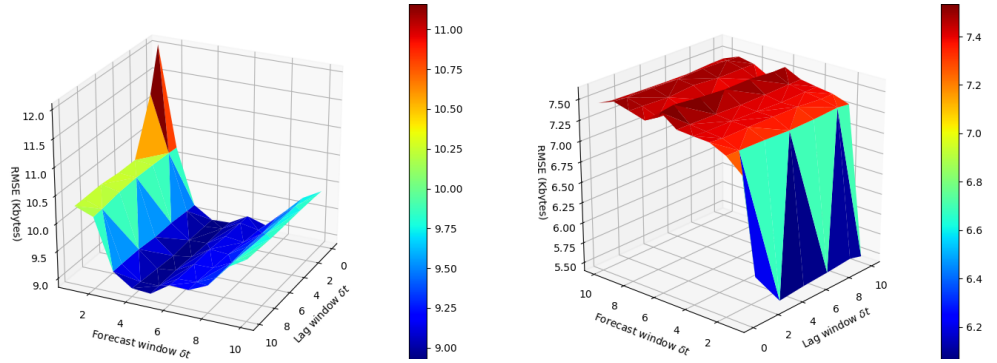
Figure A.5 – 3D plot for RMSE as a function of lag and forecast windows for estimations based on *dataset*_2_1, *dataset*_2_2, *dataset*_3_1, *dataset*_3_2, *dataset*_4_1 and *dataset*_4_2 with SVR as the underlying MLT.

A.2.2.3 RF based Estimations:



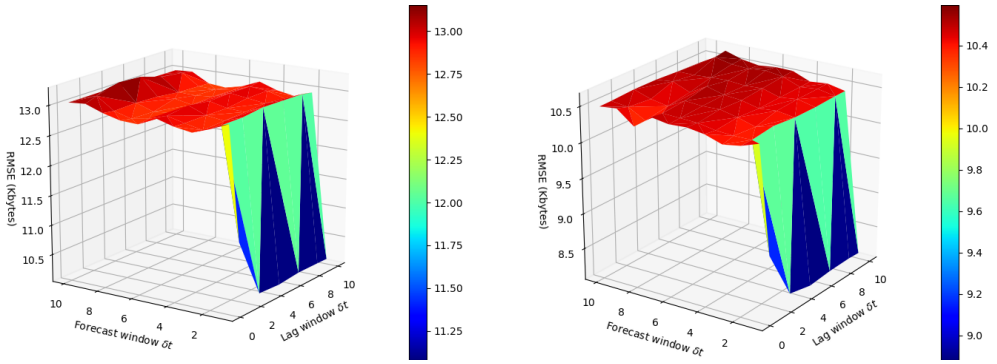
(a) Estimations Based *dataset*_2_1.

(b) Estimations Based *dataset*_2_2.



(c) Estimations Based *dataset*_3_1.

(d) Estimations Based *dataset*_3_2.



(e) Estimations Based *dataset*_4_1.

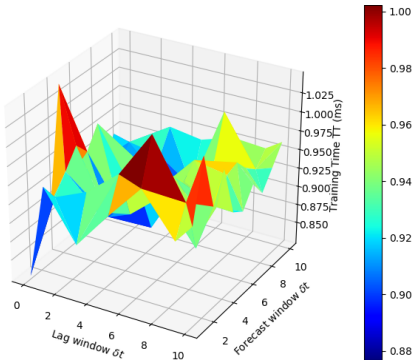
(f) Estimations Based *dataset*_4_2.

Figure A.6 – 3D plot for RMSE as a function of lag and forecast windows for estimations based on both *dataset*_2_1, *dataset*_2_2, *dataset*_3_1, *dataset*_3_2, *dataset*_4_1 and *dataset*_4_2 with RF as the underlying MLT.

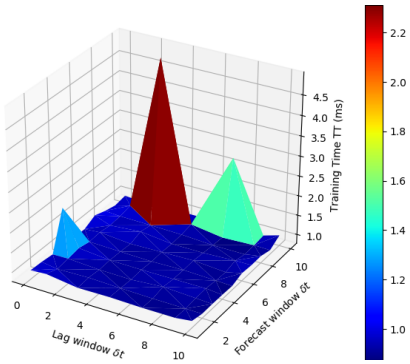
A.3 Estimation models Training Time (TT)

A.3.1 Training Time using radio metrics (testbed 1 datasets)

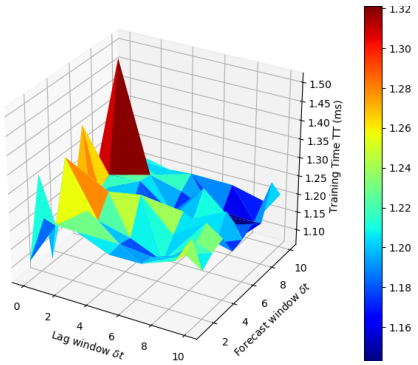
A.3.1.1 LR based Estimations:



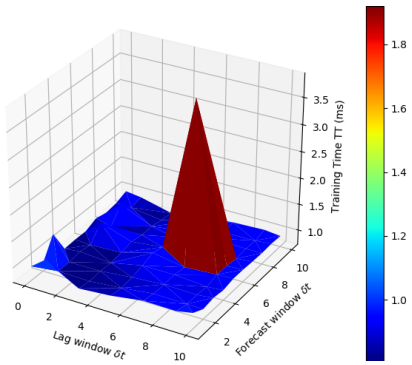
(a) TT for Estimations Based *dataset_RSSI*.



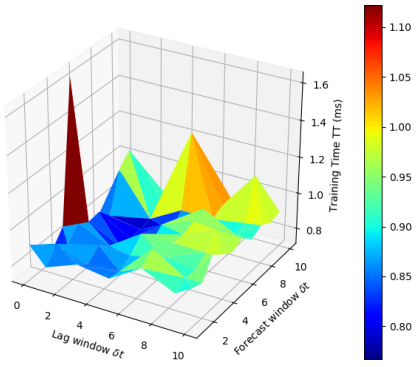
(b) TT for Estimations Based *dataset_RX_power*.



(c) TT for Estimations Based *dataset_RIP*.



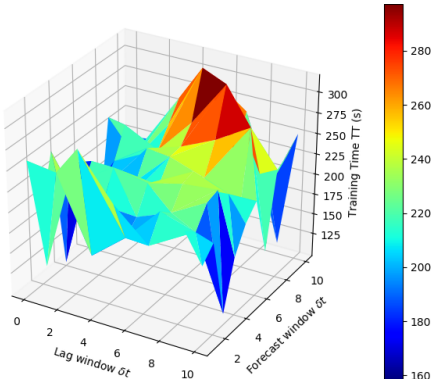
(d) TT for Estimations Based *dataset_All*.



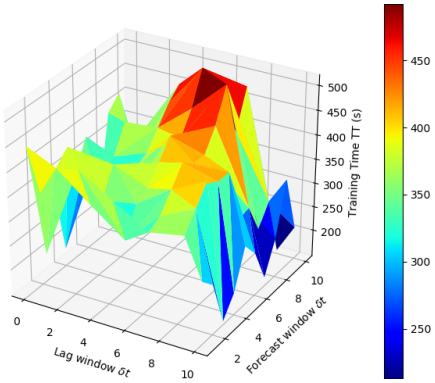
(e) TT for Estimations Based *dataset_SNR*.

Figure A.7 – 3D plot for Training Time (ms) as a function of lag and forecast windows for estimations based on *dataset_RX_power*, *dataset_RSSI*, *dataset_RIP*, *dataset_All* and *dataset_SNR* with LR as the underlying MLT.

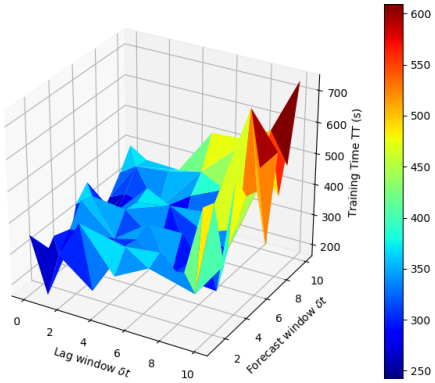
A.3.1.2 SVR based Estimations:



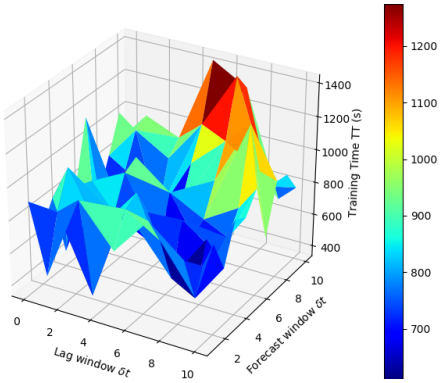
(a) TT for Estimations Based *dataset_RSSI*.



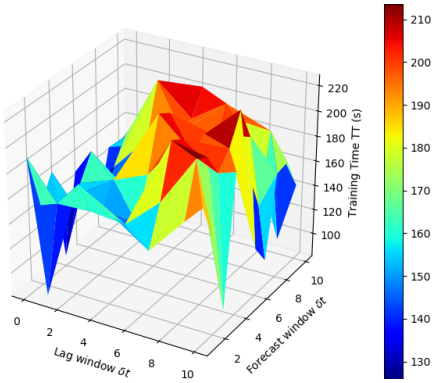
(b) TT for Estimations Based *dataset_RX_power*.



(c) TT for Estimations Based *dataset_RIP*.



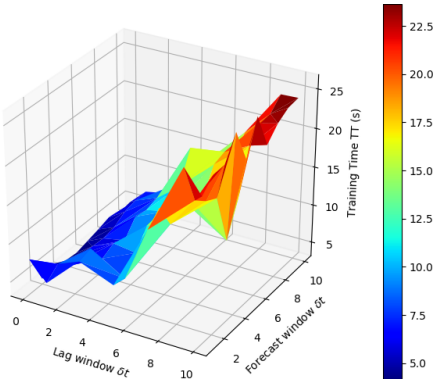
(d) TT for Estimations Based *dataset_All*.



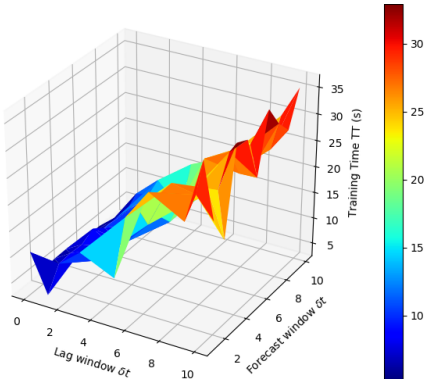
(e) TT for Estimations Based *dataset_SNR*.

Figure A.8 – 3D plot for Training time (s) as a function of lag and forecast windows for estimations based on *dataset_RX_power*, *dataset_RSSI*, *dataset_RIP*, *dataset_All* and *dataset_SNR* with SVR as the underlying MLT.

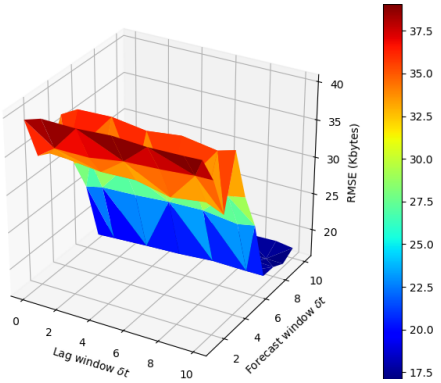
A.3.1.3 RF based Estimations:



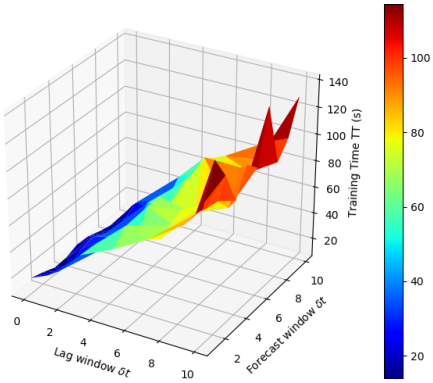
(a) TT for Estimations Based *dataset_RSSI*.



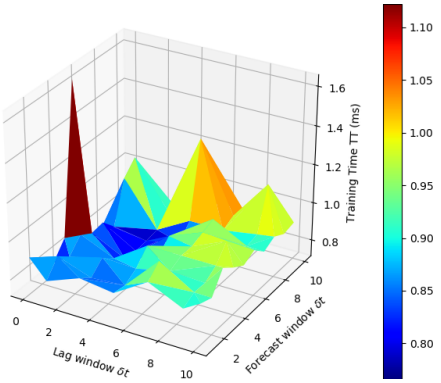
(b) TT for Estimations Based *dataset_RX_power*.



(c) Estimations Based *dataset_RIP*.



(d) TT for Estimations Based *dataset_All*.

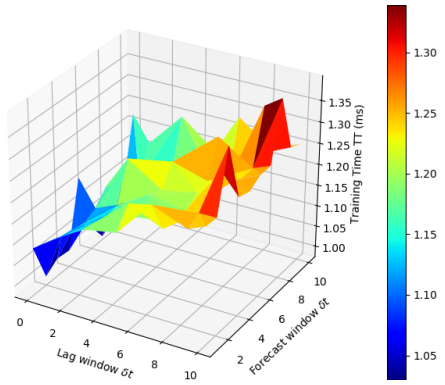


(e) TT for Estimations Based *dataset_SNR*.

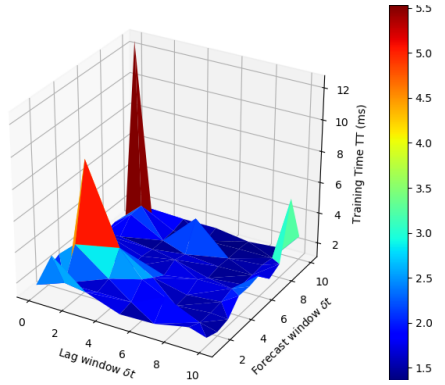
Figure A.9 – 3D plot for Training Time (s) as a function of lag and forecast windows for estimations based on *dataset_RX_power*, *dataset_RSSI*, *dataset_RIP*, *dataset_All* and *dataset_SNR* with RF as the underlying MLT.

A.3.2 Training Time using cross-layers metrics (testbed 2 datasets)

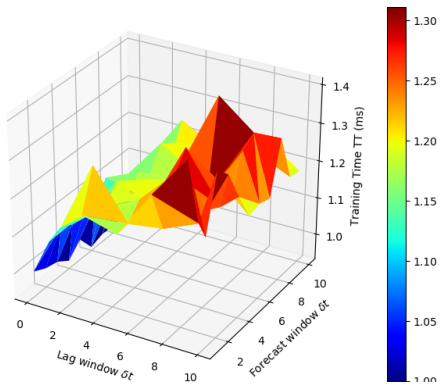
A.3.2.1 LR based Estimations:



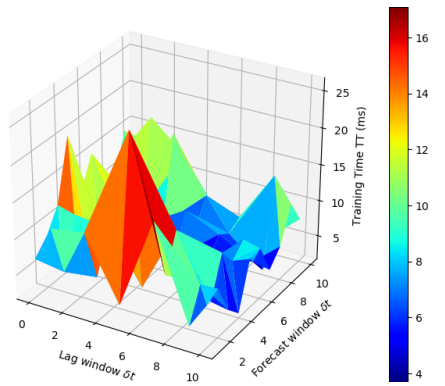
(a) TT for Estimations Based *dataset*_2_1.



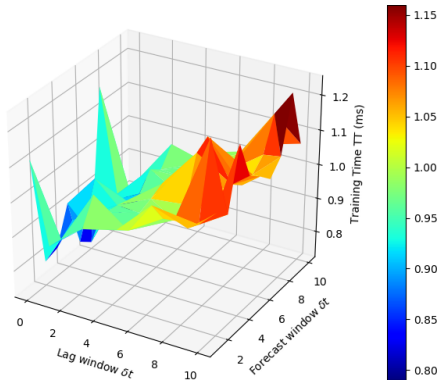
(b) TT for Estimations Based *dataset*_2_2.



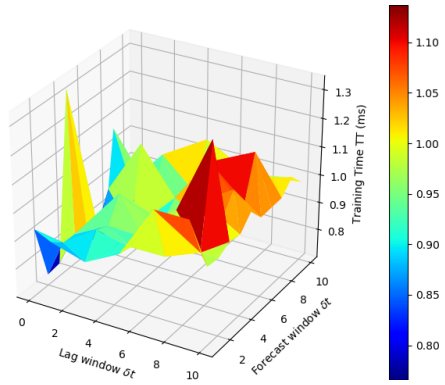
(c) TT for Estimations Based *dataset*_3_1.



(d) TT for Estimations Based *dataset*_3_2.



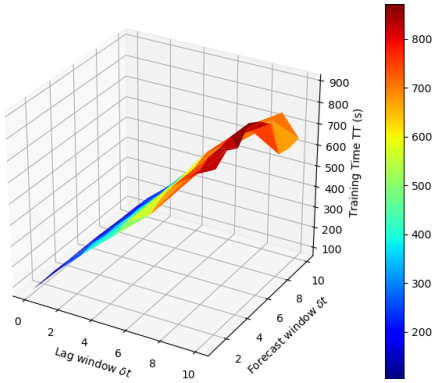
(e) TT for Estimations Based *dataset*_4_1.



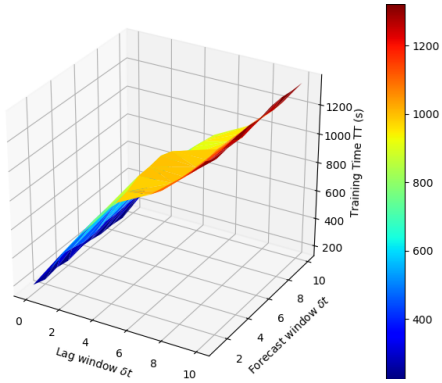
(f) TT for Estimations Based *dataset*_4_2.

Figure A.10 – 3D plot for Training Time (ms) as a function of lag and forecast windows for estimations based on *dataset*_2_1, *dataset*_2_2, *dataset*_3_1, *dataset*_3_2, *dataset*_4_1 and *dataset*_4_2 with LR as the underlying MLT.

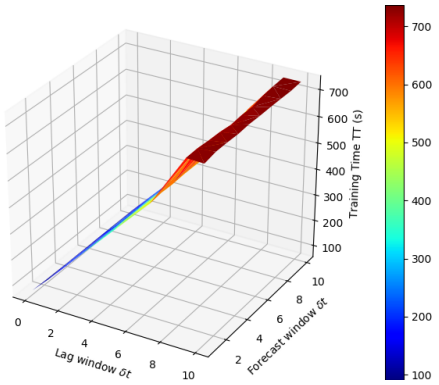
A.3.2.2 SVR based Estimations:



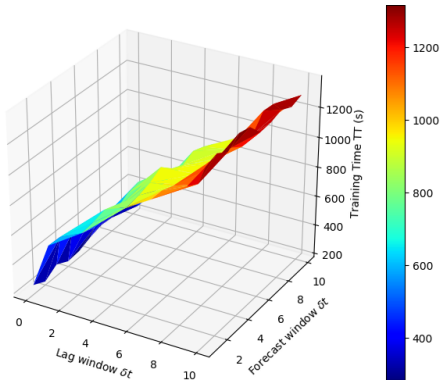
(a) TT for Estimations Based *dataset*_2_1.



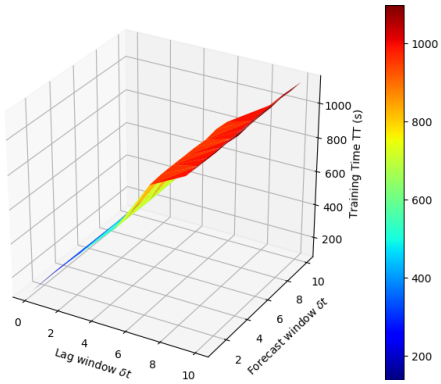
(b) TT for Estimations Based *dataset*_2_2.



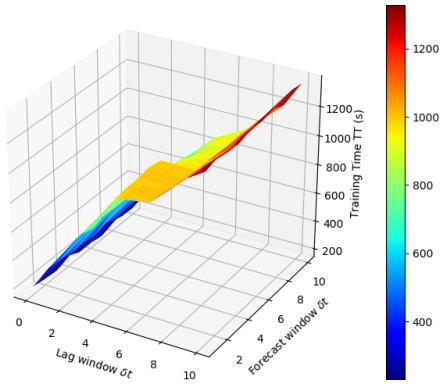
(c) TT for Estimations Based *dataset*_3_1.



(d) TT for Estimations Based *dataset*_3_2.



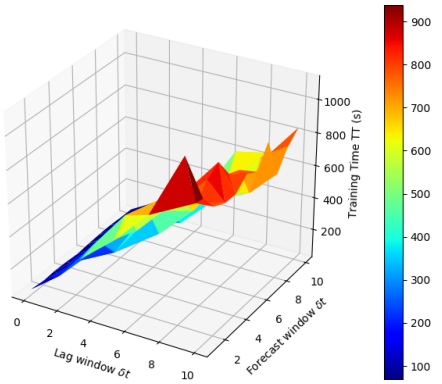
(e) TT for Estimations Based *dataset*_4_1.



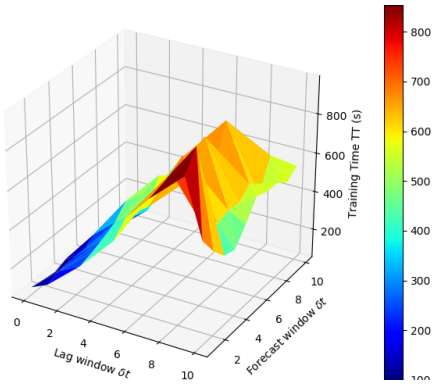
(f) TT for Estimations Based *dataset*_4_2.

Figure A.11 – 3D plot for Training Time (s) as a function of lag and forecast windows for estimations based on *dataset*_2_1, *dataset*_2_2, *dataset*_3_1, *dataset*_3_2, *dataset*_4_1 and *dataset*_4_2 with SVR as the underlying MLT.

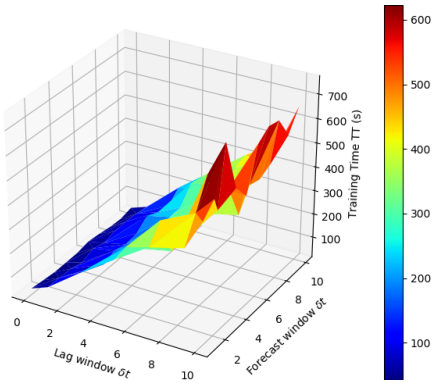
A.3.2.3 RF based Estimations:



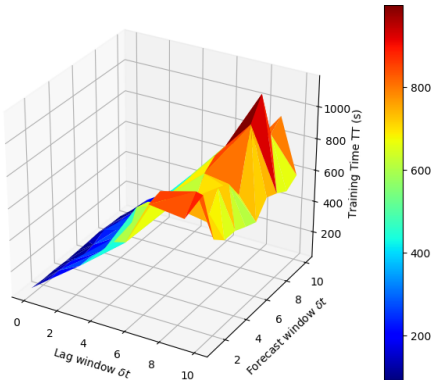
(a) TT for Estimations Based *dataset*_2_1.



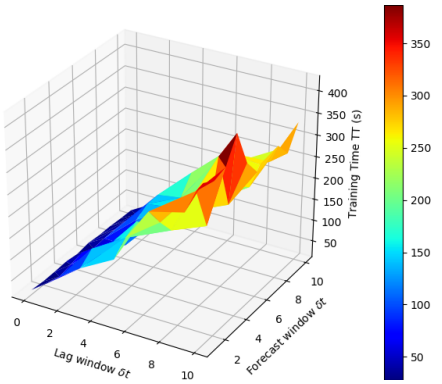
(b) TT for Estimations Based *dataset*_2_2.



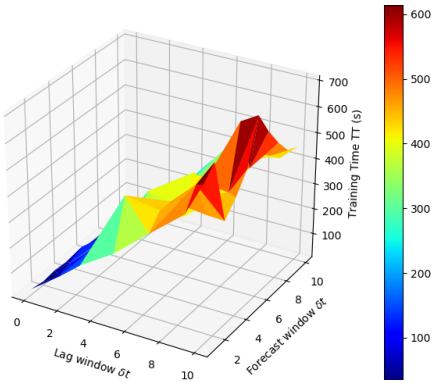
(c) TT for Estimations Based *dataset*_3_1.



(d) TT for Estimations Based *dataset*_3_2.



(e) TT for Estimations Based *dataset*_4_1.



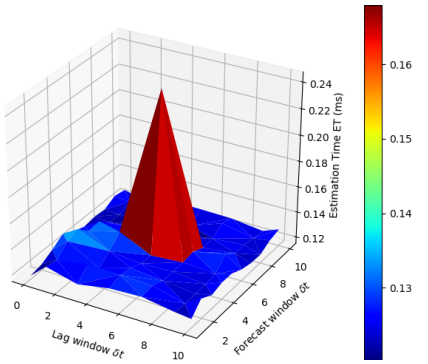
(f) TT for Estimations Based *dataset*_4_2.

Figure A.12 – 3D plot for Training Time (s) as a function of lag and forecast windows for estimations based on both *dataset*_2_1, *dataset*_2_2, *dataset*_3_1, *dataset*_3_2, *dataset*_4_1 and *dataset*_4_2 with RF as the underlying MLT.

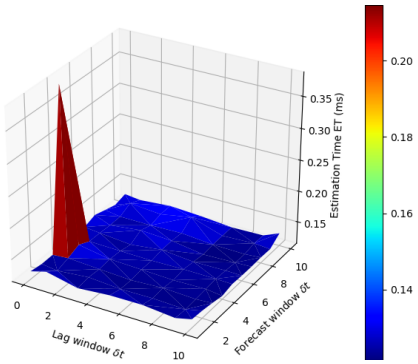
A.4 Estimation Time (ET) for Estimation models

A.4.1 Estimation Time using radio metrics (testbed 1 datasets)

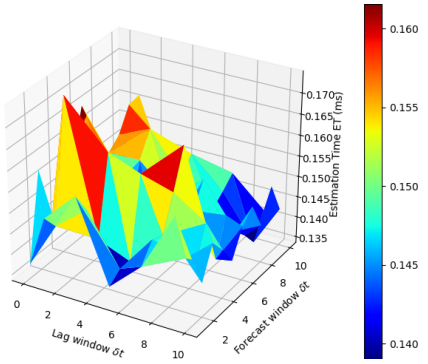
A.4.1.1 LR based Estimations:



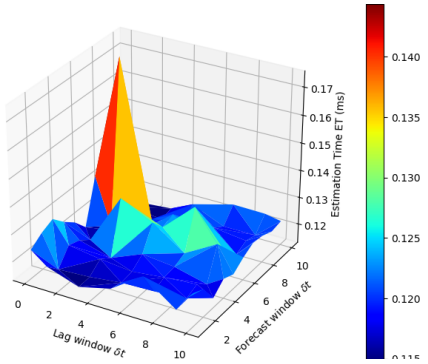
(a) ET for Estimations Based *dataset_RSSI*.



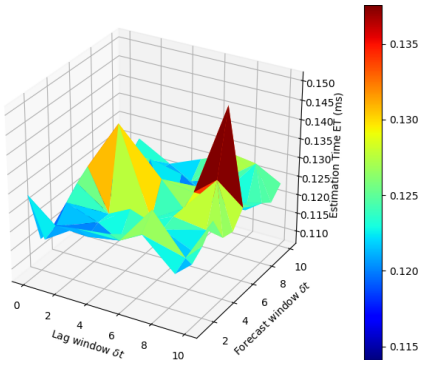
(b) ET for Estimations Based *dataset_RX_power*.



(c) ET for Estimations Based *dataset_RIP*.



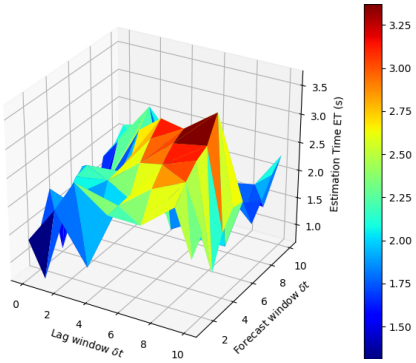
(d) ET for Estimations Based *dataset_All*.



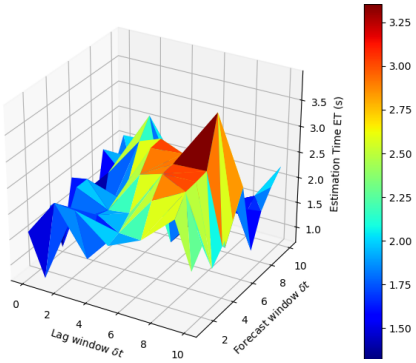
(e) ET for Estimations Based *dataset_SNR*.

Figure A.13 – 3D plot for Estimation Time (ms) as a function of lag and forecast windows for estimations based on *dataset_RX_power*, *dataset_RSSI*, *dataset_RIP*, *dataset_All* and *dataset_SNR* with LR as the underlying MLT.

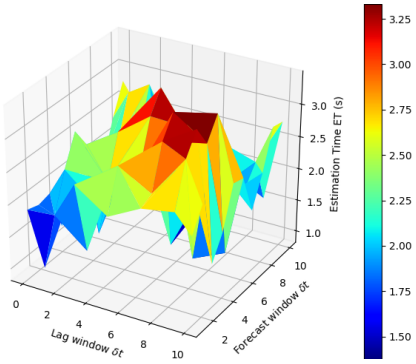
A.4.1.2 SVR based Estimations:



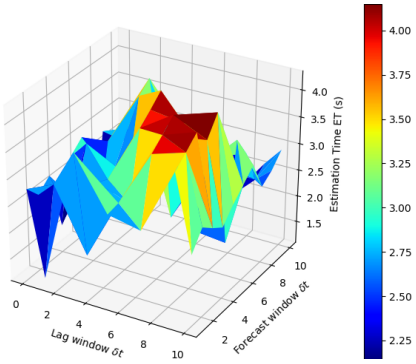
(a) ET for Estimations Based *dataset_RSSI*.



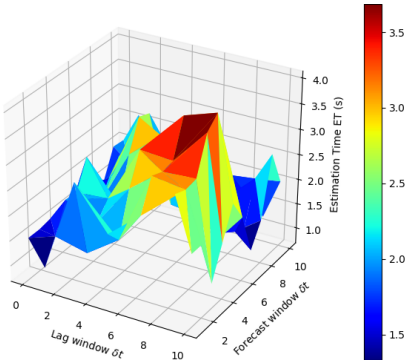
(b) ET for Estimations Based *dataset_RX_power*.



(c) ET for Estimations Based *dataset_RIP*.



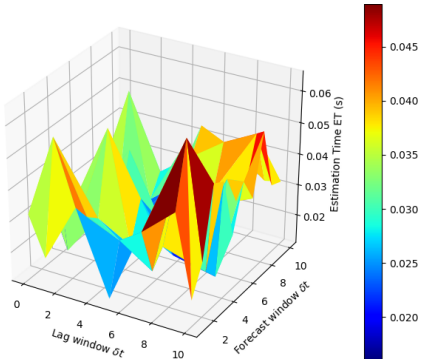
(d) ET for Estimations Based *dataset_All*.



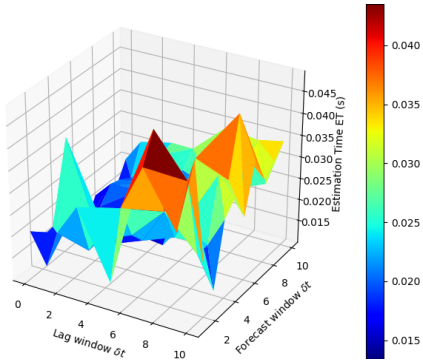
(e) ET for Estimations Based *dataset_SNR*.

Figure A.14 – 3D plot for Estimation time (s) as a function of lag and forecast windows for estimations based on *dataset_RX_power*, *dataset_RSSI*, *dataset_RIP*, *dataset_All* and *dataset_SNR* with SVR as the underlying MLT.

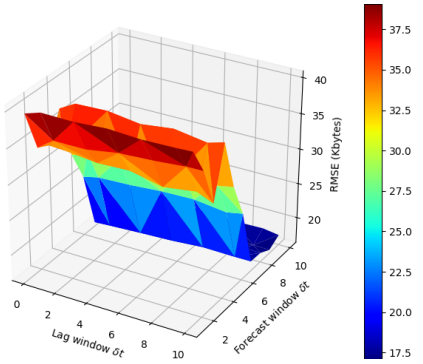
A.4.1.3 RF based Estimations:



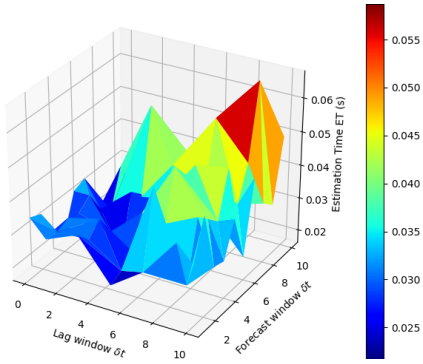
(a) ET for Estimations Based *dataset_RSSI*.



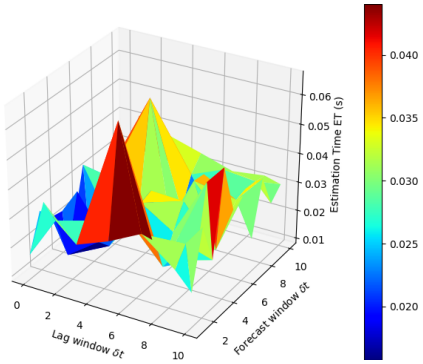
(b) ET for Estimations Based *dataset_RX_power*.



(c) Estimations Based *dataset_RIP*.



(d) ET for Estimations Based *dataset_All*.

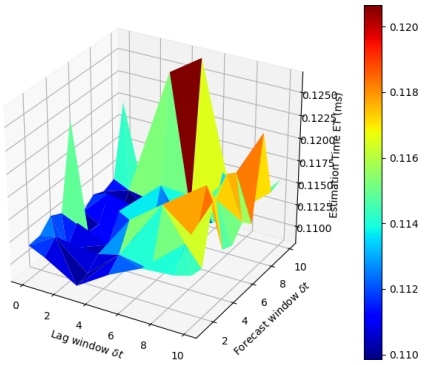


(e) ET for Estimations Based *dataset_SNR*.

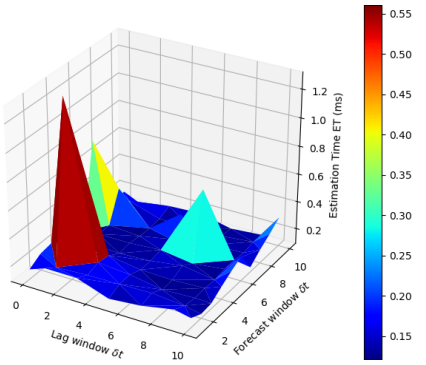
Figure A.15 – 3D plot for Estimation Time (s) as a function of lag and forecast windows for estimations based on *dataset_RX_power*, *dataset_RSSI*, *dataset_RIP*, *dataset_All* and *dataset_SNR* with RF as the underlying MLT.

A.4.2 Estimation Time using cross-layers metrics (testbed 2 datasets)

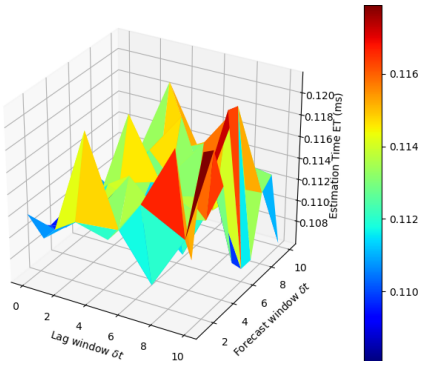
A.4.2.1 LR based Estimations:



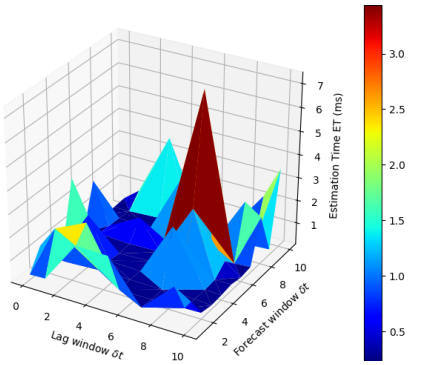
(a) ET for Estimations Based *dataset*_2_1.



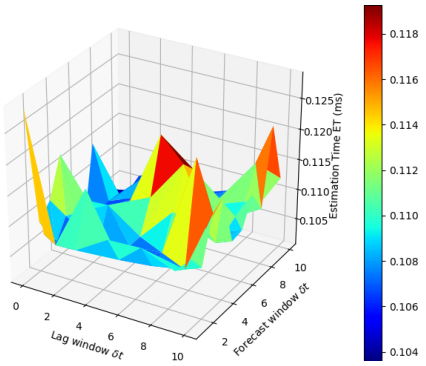
(b) ET for Estimations Based *dataset*_2_2.



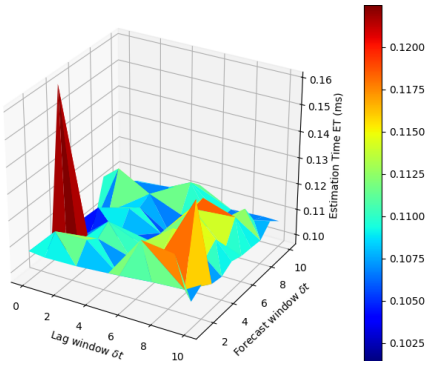
(c) ET for Estimations Based *dataset*_3_1.



(d) ET for Estimations Based *dataset*_3_2.



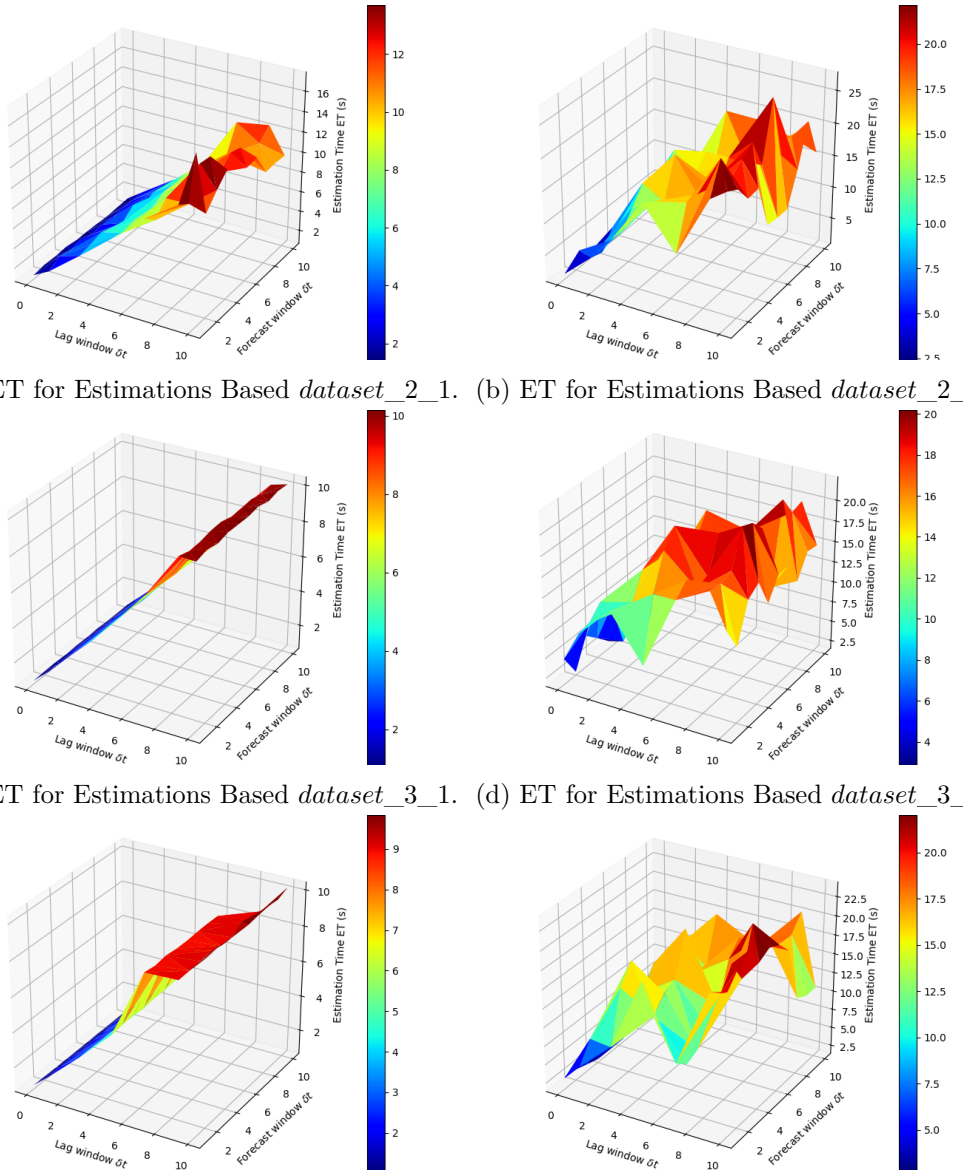
(e) ET for Estimations Based *dataset*_4_1.



(f) ET for Estimations Based *dataset*_4_2.

Figure A.16 – 3D plot for Estimation Time (ms) as a function of lag and forecast windows for estimations based on *dataset*_2_1, *dataset*_2_2, *dataset*_3_1, *dataset*_3_2, *dataset*_4_1 and *dataset*_4_2 with LR as the underlying MLT.

A.4.2.2 SVR based Estimations:



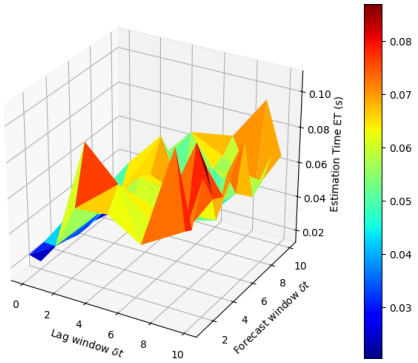
(a) ET for Estimations Based *dataset*_2_1. (b) ET for Estimations Based *dataset*_2_2.

(c) ET for Estimations Based *dataset*_3_1. (d) ET for Estimations Based *dataset*_3_2.

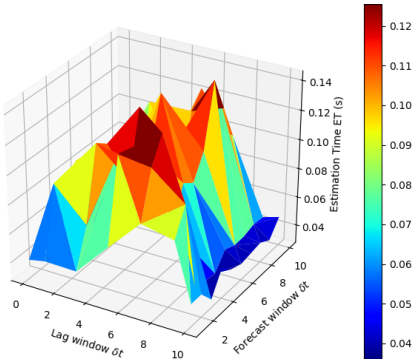
(e) ET for Estimations Based *dataset*_4_1. (f) ET for Estimations Based *dataset*_4_2.

Figure A.17 – 3D plot for Estimation Time (s) as a function of lag and forecast windows for estimations based on *dataset*_2_1, *dataset*_2_2, *dataset*_3_1, *dataset*_3_2, *dataset*_4_1 and *dataset*_4_2 with SVR as the underlying MLT.

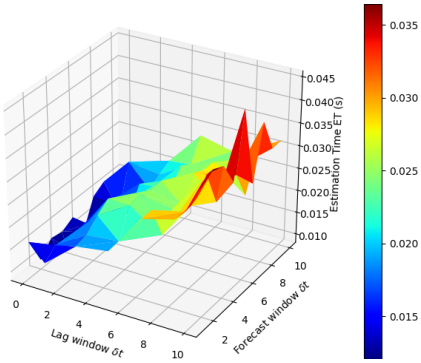
A.4.2.3 RF based Estimations:



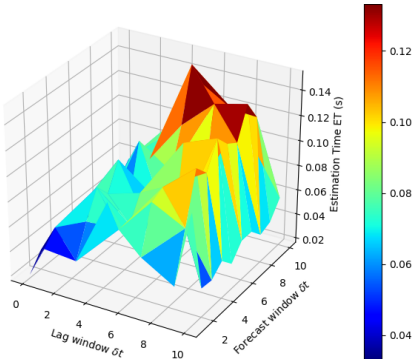
(a) ET for Estimations Based *dataset*_2_1.



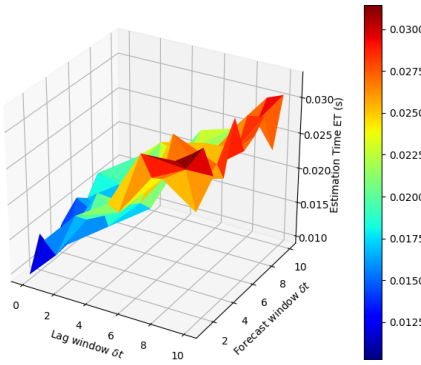
(b) ET for Estimations Based *dataset*_2_2.



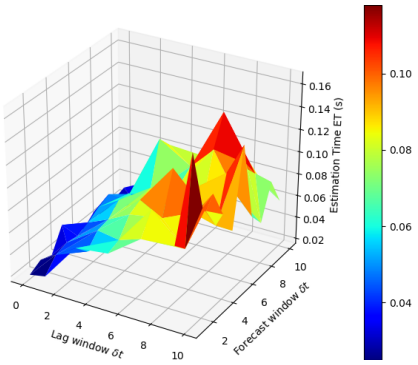
(c) ET for Estimations Based *dataset*_3_1.



(d) ET for Estimations Based *dataset*_3_2.



(e) ET for Estimations Based *dataset*_4_1.



(f) ET for Estimations Based *dataset*_4_2.

Figure A.18 – 3D plot for Estimation Time (s) as a function of lag and forecast windows for estimations based on both *dataset*_2_1, *dataset*_2_2, *dataset*_3_1, *dataset*_3_2, *dataset*_4_1 and *dataset*_4_2 with RF as the underlying MLT.

Appendix B

RAN slicing performance evaluation

The optimal model scores details are shown in this annex. Also, the performance of the three heuristics HMF, HSF and IMA are exhibited with varying B and S sizes. The performance metrics include the convergence time (s), the total tied sRBs TTR and the largest space score.

B.1 ESRP-v1 statistics

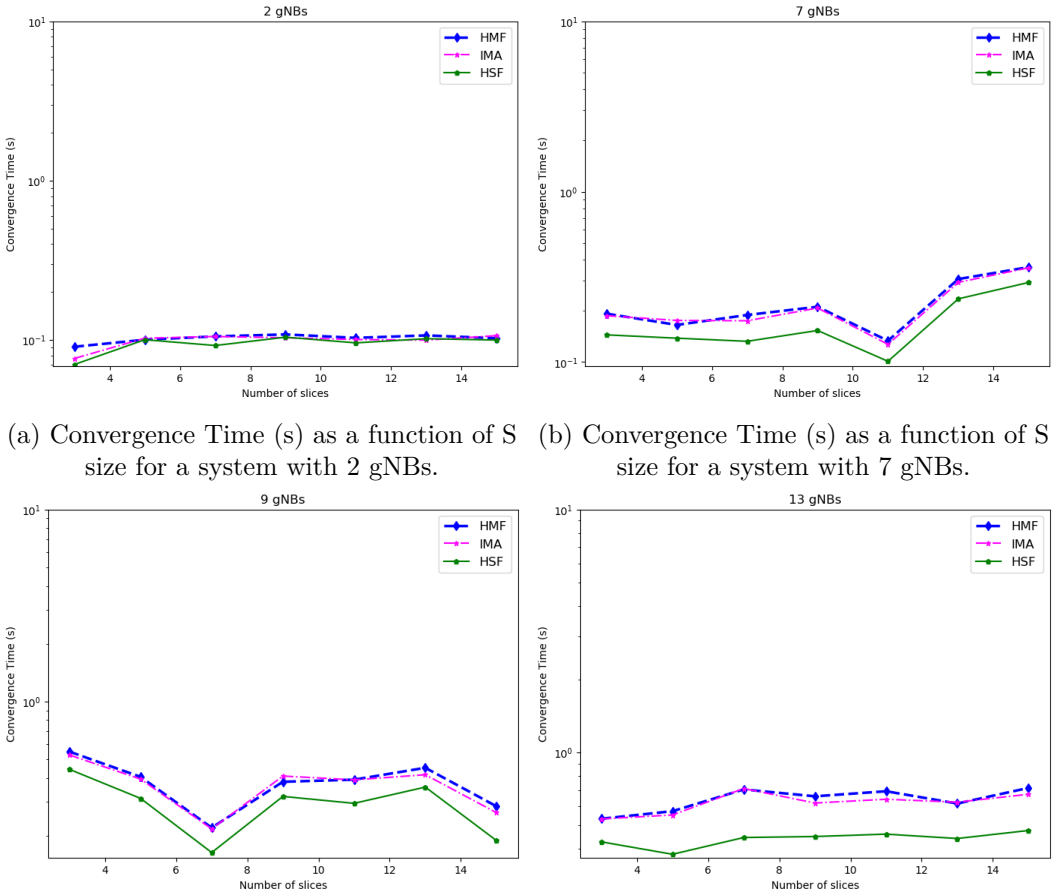
The statistics of the total number of simulations run where the ESRP-v1 model converges at the time limit of 600 s are shown on fig B.1.

S size \ B size	2	3	5	7	9	13
3	87	61	55	58	51	45
5	63	35	14	9	11	13
7	38	14	2	1		
9	22	4				
11	14	2		1		
13	11	3				
15	3					

Figure B.1 – ESRP-v1 optimal score achievements. For each B and S size, the number of simulations runs reaching the optimal score before the time limit of 600 s is shown.

B.2 Convergence Time (CT) performance

B.2.0.1 Convergence Time with variation of S size



(a) Convergence Time (s) as a function of S size for a system with 2 gNBs.

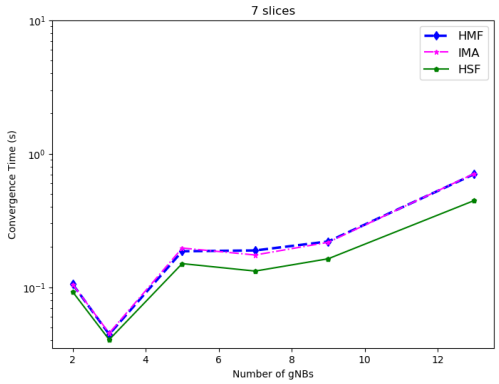
(b) Convergence Time (s) as a function of S size for a system with 7 gNBs.

(c) Convergence Time (s) as a function of S size for a system with 9 gNBs.

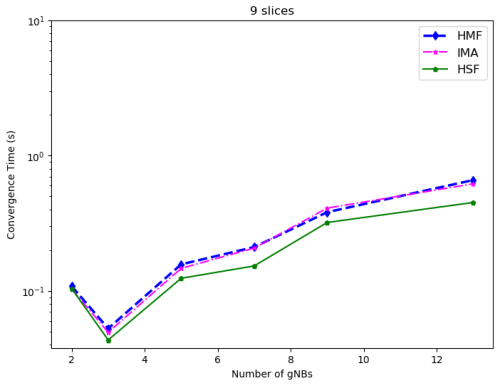
(d) Convergence Time (s) as a function of S size for a system with 13 gNBs.

Figure B.2 – Convergence Time (s) for the three heuristics HMF, HSF and IMA as a function of S size for a system with different B sizes.

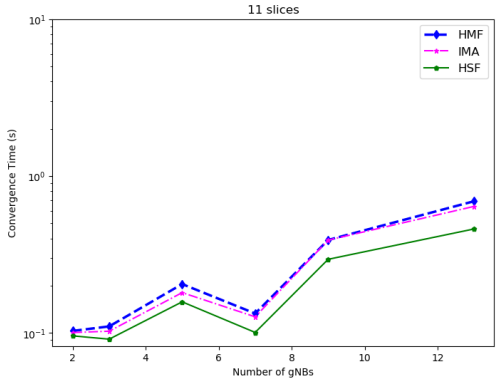
B.2.0.2 Convergence Time with variation of B size



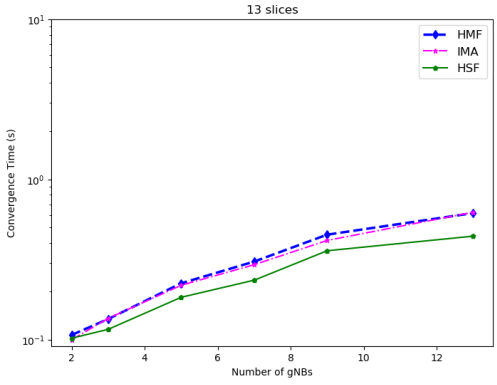
(a) Convergence Time (s) as a function of B size for a system with 7 slices.



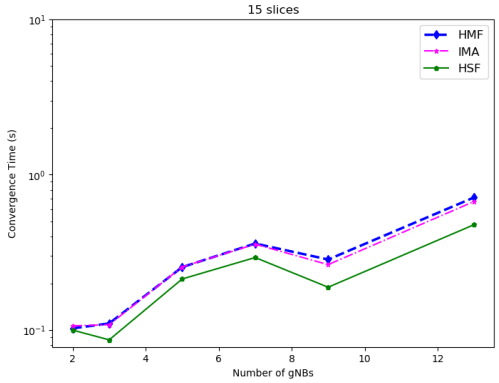
(b) Convergence Time (s) as a function of B size for a system with 9 slices.



(c) Convergence Time (s) as a function of B size for a system with 11 slices.



(d) Convergence Time (s) as a function of B size for a system with 13 slices.

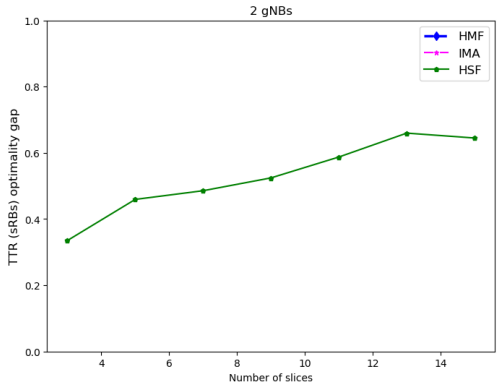


(e) Convergence Time (s) as a function of B size for a system with 15 slices.

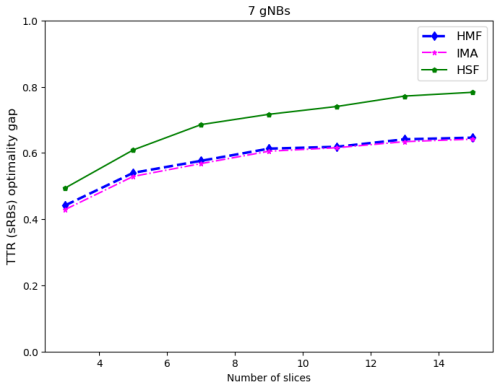
Figure B.3 – Convergence Time (s) for the three heuristics HMF, HSF and IMA as a function of B size for a system with different S sizes.

B.3 Total Tied sRBs (TTR) performance

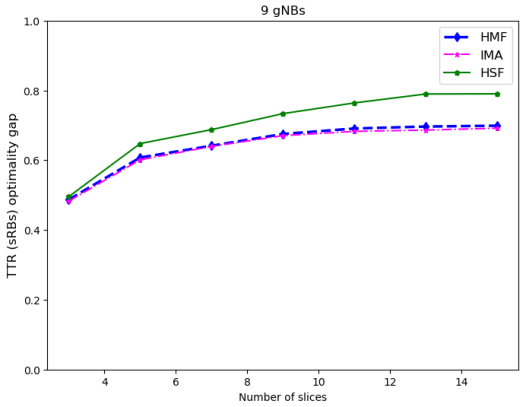
B.3.0.1 Total Tied sRBs (TTR) with variation of S size



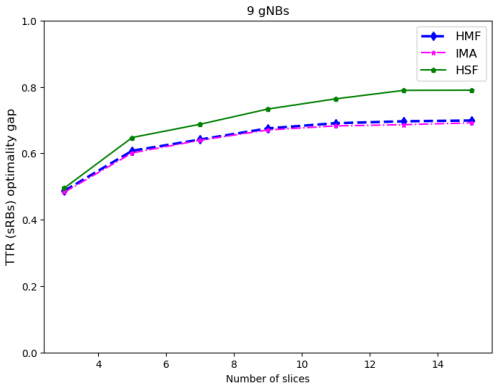
(a) Optimality gap of the achieved TTR (sRBs) as a function of S size for a system with 2 gNBs.



(b) Optimality gap of the achieved TTR (sRBs) as a function of S size for a system with 7 gNBs.



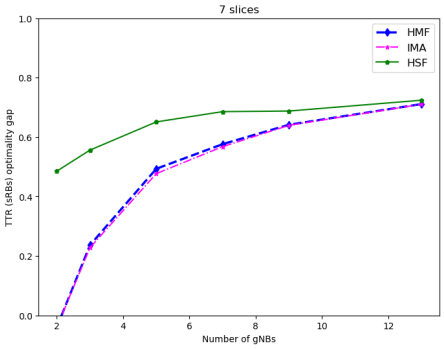
(c) Optimality gap of the achieved TTR (sRBs) as a function of S size for a system with 9 gNBs.



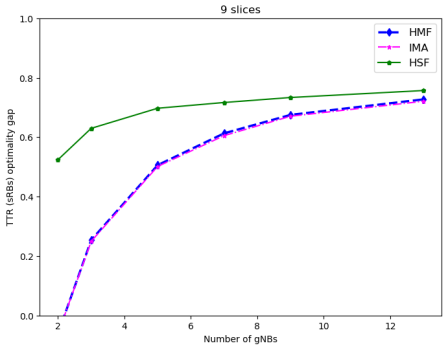
(d) Optimality gap of the achieved TTR (sRBs) as a function of S size for a system with 13 gNBs.

Figure B.4 – Optimality gap of the achieved TTR (sRBs) for the three heuristics HMF, HSF and IMA as a function of S size for a system with different B sizes.

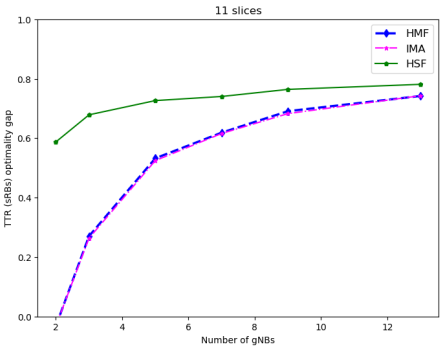
B.3.0.2 Total Tied sRBs (TTR) with variation of B size



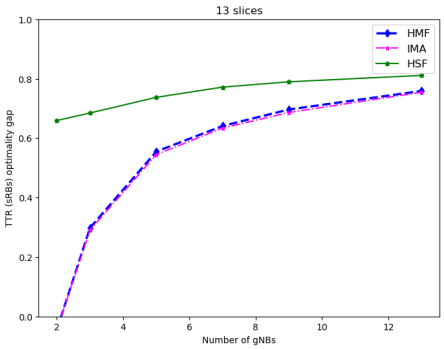
(a) Optimality gap of the achieved TTR (sRBs) as a function of B size serving 7 slices.



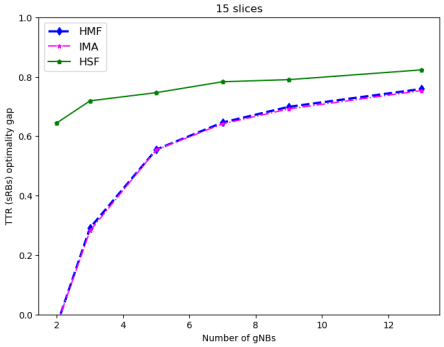
(b) Optimality gap of the achieved TTR (sRBs) as a function of B size for a system serving 9 slices.



(c) Optimality gap of the achieved TTR (sRBs) as a function of B size for a system serving 11 slices.



(d) Optimality gap of the achieved TTR (sRBs) as a function of B size for a system serving 13 slices.

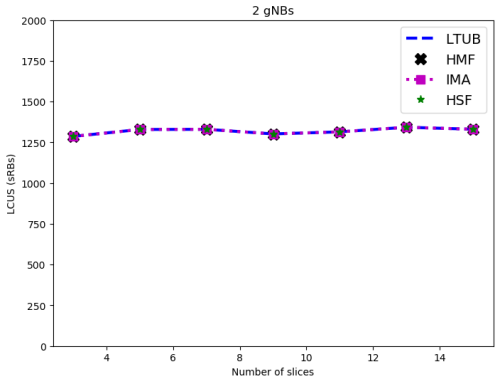


(e) Optimality gap of the achieved TTR (sRBs) as a function of B size for a system serving 15 slices.

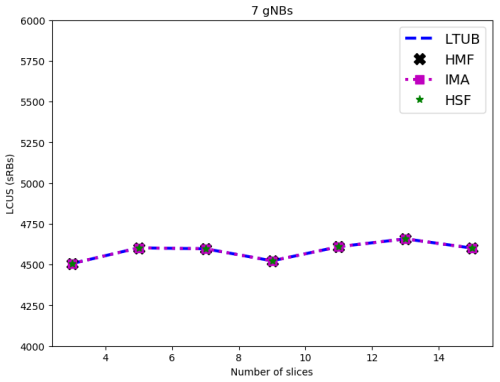
Figure B.5 – Optimality gap of the achieved TTR (sRBs) for the three heuristics HMF, HSF and IMA as a function of B size for a system with different S sizes.

B.4 Largest Continuous Unallocated Space (LCUS) performance

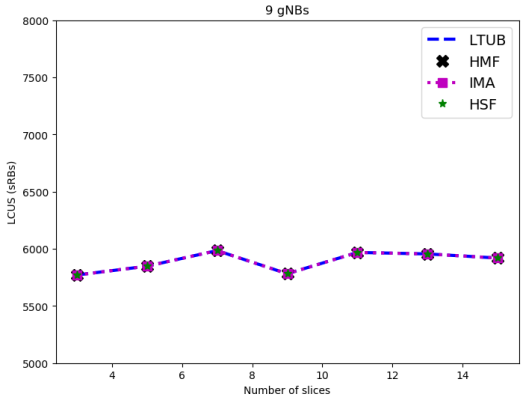
B.4.0.1 Largest Continuous Unallocated Space with variation of S size



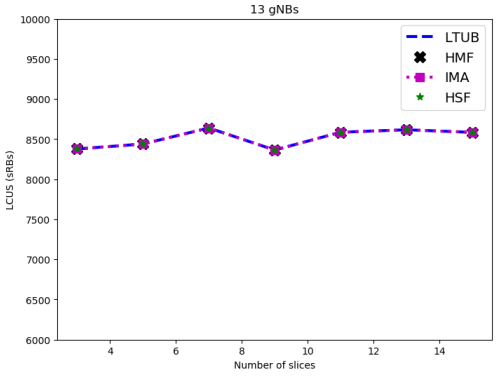
(a) LCUS (sRBs) as a function of S size for a system with 2 gNBs.



(b) LCUS (sRBs) as a function of S size for a system with 7 gNBs.



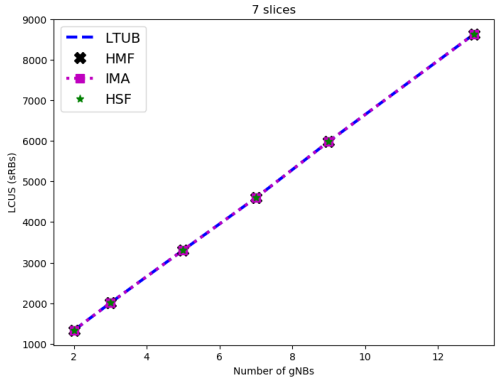
(c) Optimality gap of the achieved TTR (sRBs) as a function of S size for a system with 9 gNBs.



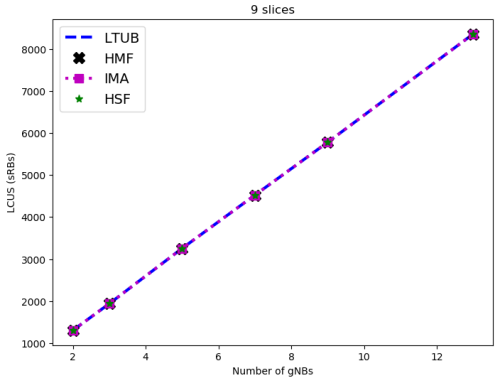
(d) LCUS (sRBs) as a function of S size for a system with 13 gNBs.

Figure B.6 – LCUS (sRBs) for the three heuristics HMF, HSF and IMA as a function of S size for a system with different B sizes.

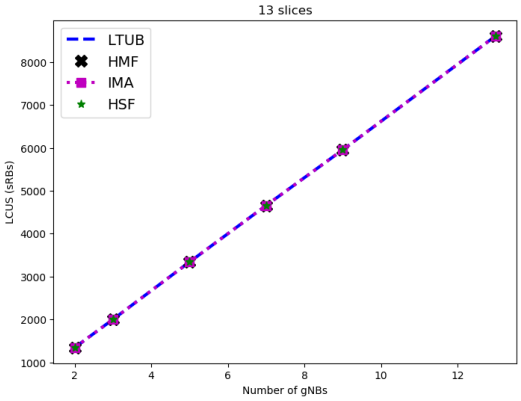
B.4.0.2 Largest Continuous Unallocated Space with variation of B size



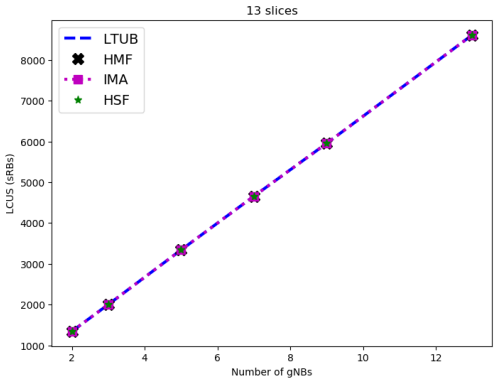
(a) LCUS (sRBs) as a function of B size serving 7 slices.



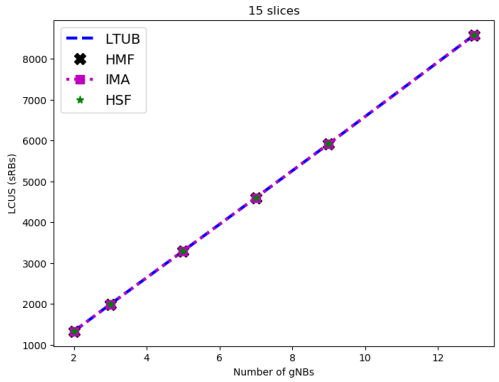
(b) LCUS (sRBs) as a function of B size for a system serving 9 slices.



(c) LCUS (sRBs) as a function of B size for a system serving 11 slices.



(d) LCUS (sRBs) as a function of B size for a system serving 13 slices.



(e) LCUS (sRBs) as a function of B size for a system serving 15 slices.

Figure B.7 – LCUS (sRBs) for the three heuristics HMF, HSF and IMA as a function of B size for a system with different S sizes.

Bibliography

[Abu-Ali 2014] N. Abu-Ali, A.-E. M. Taha, M. Salah and H. Hassanein. *"Uplink Scheduling in LTE and LTE-Advanced: Tutorial, Survey and Evaluation Framework," IEEE Communications Surveys & Tutorials, vol.16, no.3, pp.1239-1265.* 2014. (Cited in page 18.)

[Akyildiz 2015] Ian F. Akyildiz, Pu Wang and Shih-Chun Lin. *SoftAir: A software defined networking architecture for 5G wireless systems.* Computer Networks, vol. 85, page 1–18, Jul 2015. (Cited in page 70.)

[Andersen 1995] J.B. Andersen, T.S. Rappaport and S. Yoshida. *Propagation measurements and models for wireless communications channels.* IEEE Communications Magazine, vol. 33, no. 1, page 42–49, Jan 1995. (Cited in page 21.)

[B. Han 2018] L. Ji B. Han and H. D. Schotten. *"Slice as an evolutionary service: Ge-netic optimization for inter-slice resource management in5G networks,"IEEE Access, vol. 6, no. 1.* p, vol. 33, page 137, 2018. (Cited in page 75.)

[Bang 2008] H. Bang, T. Ekman and D. Gesbert. *"Channel predictive proportional fair scheduling," IEEE Transactions on Wireless Communications, vol. 7, no. 2.* p, vol. 482, February 2008. (Cited in page 17.)

[Bansal 2004] Nikhil Bansal and Maxim Sviridenko. *New Approximability and Inapproximability Results for 2-Dimensional Bin Packing.* In Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '04, page 196–203, USA, 2004. Society for Industrial and Applied Mathematics. (Cited in page 90.)

[Bergstra 2011] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. The Learning Workshop(Snowbird), 2011. (Cited in page 31.)

[Bergstra 2012] James Bergstra and Yoshua Bengio. *Random search for hyper-parameter optimization.* Journal of Machine Learning Research, page 281–305, Feb 2012. (Cited in page 31.)

[Bolat 2003] E. Bolat. Study of ofdm performance over awgnn channels. B. Sc. Project, Department of Electrical and Electronic Engineering, Eastern Mediterranean University, 2003. (Cited in page 43.)

[Bramer 2013] Max Bramer. Avoiding overfitting of decision trees, pages 121–136. Springer London, London, 2013. (Cited in page 29.)

[Caballero 2017] Pablo Caballero, Albert Banchs, Gustavo de Veciana and Xavier Costa-Perez. *Network slicing games: Enabling customization in multi-tenant*

networks. In IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, page 1–9. IEEE, May 2017. (Cited in page 80.)

[Chakraborty 2013] A. Chakraborty, V. Navda, V. N. Padmanabhan and R. Ramjee. *Coordinating cellular background transfers using LoadSense*. In Proc. of ACM MobiCom. pages 63–74, 2013. (Cited in page 13.)

[Chang 2018] C.-Y. Chang, N. Nikaein and T. Spyropoulos. *"Radio access network resource slicing for flexible service execution*. In " in IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE pp. 668–673, 2018. (Cited in page 76.)

[Demel 2015] Johannes Demel, Sebastian Koslowski and A Lte Friedrich K. Jondral. *Receiver Framework Using GNU Radio*. Journal of Signal Processing Systems, vol. 78, page 3, 2015. (Cited in page 40.)

[Devlic 2017] Alisa Devlic, Ali Hamidian, Deng Liang, Mats Eriksson, Antonio Consoli and Jonas Lundstedt. *NESMO: Network slicing management and orchestration framework*. In 2017 IEEE International Conference on Communications Workshops (ICC Workshops), page 1202–1208. IEEE, May 2017. (Cited in page 75.)

[Di Francescomarino 2018] Chiara Di Francescomarino, Marlon Dumas, Marco Federici, Chiara Ghidini, Fabrizio Maria Maggi, Williams Rizzi and Luca Simonetto. *Genetic algorithms for hyperparameter optimization in predictive business process monitoring*. Information Systems, vol. 74, page 67–83, May 2018. (Cited in page 31.)

[D'Oro 2019] Salvatore D'Oro, Francesco Restuccia, Alessandro Talamonti and Tommaso Melodia. *The Slice Is Served: Enforcing Radio Access Network Slicing in Virtualized 5G Systems*. In IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, page 442–450. IEEE, Apr 2019. (Cited in pages 77 and 80.)

[Elayoubi 2019] Salah Eddine Elayoubi, Sana Ben Jemaa, Zwi Altman and Ana Galindo-Serrano. *5G RAN Slicing for Verticals: Enablers and Challenges*. IEEE Communications Magazine, vol. 57, no. 1, page 28–34, Jan 2019. (Cited in page 72.)

[Ferro 2005] E. Ferro and F. Potorti. *Bluetooth and wi-fi wireless protocols: a survey and a comparison*. IEEE Wireless Communications, vol. 12, no. 1, page 12–26, Feb 2005. (Cited in page 1.)

[Ferrus 2018] Ramon Ferrus, Oriol Sallent, Jordi Perez-Romero and Ramon Agusti. *On 5G Radio Access Network Slicing: Radio Interface Protocol Features and Configuration*. IEEE Communications Magazine, vol. 56, no. 5, page 184–192, May 2018. (Cited in pages 75 and 80.)

[Foukas 2017] Xenofon Foukas, Mahesh K. Marina and Kimon Kontovasilis. *Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture*. In Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking - MobiCom '17, page 127–140. ACM Press, 2017. (Cited in pages 69 and 80.)

[Friedman 2001] J. Friedman, T. Hastie and R. Tibshirani. *The Elements of Statistical Learning, Springer Series in Statistics*. 2001. (Cited in page 32.)

[G. 2015] Omojokun G. *A Survey of ZigBee Wireless Sensor Network Technology: Topology, Applications and Challenges*. International Journal of Computer Applications, vol. 130, no. 9, page 47–55, Nov 2015. (Cited in page 1.)

[Garey 2009] Michael R. Garey and David S. Johnson. Computers and intractability: a guide to the theory of np-completeness. A series of books in the mathematical sciences. Freeman, 27. print edition, 2009. (Cited in page 90.)

[Gebremariam 2018] Anteneh A. Gebremariam, Mainak Chowdhury, Muhammad Usman, Andrea Goldsmith and Fabrizio Granelli. *SoftSLICE: Policy-Based Dynamic Spectrum Slicing in 5G Cellular Networks*. In 2018 IEEE International Conference on Communications (ICC), page 1–6. IEEE, May 2018. (Cited in page 80.)

[Geisser 1975] S. Geisser. *The Predictive Sample Reuse Method with Applications*. Journal of the American Statistical Association, vol. 70, no. 350, pages 320–328, 1975. (Cited in page 32.)

[Gomez-Miguelez 2016] Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D. Sutton, Pablo Serrano, Cristina Cano and Doug J. Leith. *srsLTE: an open-source platform for LTE evolution and experimentation*. In Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization - WiNTECH '16, page 25–32. ACM Press, 2016. (Cited in page 40.)

[Gringoli 2018] Francesco Gringoli, Paul Patras, Carlos Donato, Pablo Serrano and Yan Grunenberger. *Performance Assessment of Open Software Platforms for 5G Prototyping*. IEEE Wireless Communications, vol. 25, no. 5, page 10–15, Oct 2018. (Cited in page 40.)

[Guo 2008] X.C. Guo, J.H. Yang, C.G. Wu, C.Y. Wang and Y.C. Liang. *A novel LS-SVMs hyper-parameter selection based on particle swarm optimization*. Neurocomputing, vol. 71, no. 16–18, page 3211–3215, Oct 2008. (Cited in page 31.)

[He 2015] Jun He and Wei Song. *AppRAN: Application-oriented radio access network sharing in mobile networks*. In 2015 IEEE International Conference on Communications (ICC), page 3788–3794. IEEE, Jun 2015. (Cited in page 76.)

[He 2017] Lifeng He, Xiwei Ren, Qihang Gao, Xiao Zhao, Bin Yao and Yuyan Chao. *The connected-component labeling problem: A review of state-of-the-art algorithms.* Pattern Recognition, vol. 70, page 25–43, Oct 2017. (Cited in page 93.)

[Hebrard 2017] Emmanuel Hebrard, Marie-José Huguet, Daniel Veysseire, Ludivine Boche Sauvan and Bertrand Cabon. *Constraint programming for planning test campaigns of communications satellites.* Constraints, vol. 22, no. 1, page 73–89, Jan 2017. (Cited in page 84.)

[Heegard 1999] Chris Heegard and Stephen B. Wicker. Turbo coding. Springer US, 1999. (Cited in page 9.)

[Heinrich 2017] Stephan Heinrich. *Flash Memory in the emerging age of autonomy.* In Lucid Motors, 2017. (Cited in page 3.)

[Hossain 2014] Ekram Hossain, Mehdi Rasti, Hina Tabassum and Amr Abdelnasser. *Evolution toward 5G multi-tier cellular wireless networks: An interference management perspective.* IEEE Wireless Communications, vol. 21, no. 3, page 118–127, Jun 2014. (Cited in page 73.)

[Hutchison 2008] David Hutchison, Ju¨rgen Branke, Kalyanmoy Deb, Takeo Kanade, Josef Kittler, Jon M Kleinberg, Friedemann Mattern, Kaisa Miettinen, John C Mitchell, Moni Naor and et al. Multiobjective optimization: Interactive and evolutionary approaches. Springer Berlin Heidelberg, 2008. (Cited in page 83.)

[Ikuno 2010] J. Ikuno, M. Wrulich and M. Rupp. *"System level simulation of LTE networks.* In " in Proc. 71st Vehicular Technology Conference VTC2010-Spring, 2010. (Cited in page 39.)

[Jia 2018] Yang Jia, Hui Tian, Shaoshuai Fan, Pengtao Zhao and Kun Zhao. *Bankruptcy game based resource allocation algorithm for 5G Cloud-RAN slicing.* In 2018 IEEE Wireless Communications and Networking Conference (WCNC), page 1–6. IEEE, Apr 2018. (Cited in page 80.)

[Jolliffe 1986] I. T. Jolliffe. Principal component analysis. Springer Series in Statistics. Springer New York, 1986. (Cited in page 53.)

[Jylänki 2010] Jukka Jylänki. *A thousand ways to pack the bin – a practical approach to two-dimensional rectangle bin packing*, 2010. (Cited in page 90.)

[Karp 1972] R. Karp. *Reducibility among combinatorial problems.* In R. Miller and J. Thatcher, editors, Complexity of Computer Computations, pages 85–103. Plenum Press, 1972. (Cited in page 90.)

[Kiatmanaroj 2016] Kata Kiatmanaroj, Christian Artigues and Laurent Houssin. *On scheduling models for the frequency interval assignment problem with*

cumulative interferences. Engineering Optimization, vol. 48, no. 5, page 740–755, May 2016. (Cited in page 86.)

[Kohavi 1995] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In IJCAI, 1995. (Cited in page 32.)

[Kokku 2012] Ravi Kokku, Rajesh Mahindra, Honghai Zhang and Sampath Rangarajan. *NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks*. IEEE/ACM Transactions on Networking, vol. 20, no. 5, page 1333–1346, Oct 2012. (Cited in page 76.)

[Koopman 2004] P. Koopman and T. Chakravarty. *Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks*. In 2004 International Conference on Dependable Systems and Networks, page 145, Los Alamitos, CA, USA, jul 2004. IEEE Computer Society. (Cited in page 10.)

[Koutsonikolas 2011] Dimitrios Koutsonikolas and Y. Charlie Hu. *On the feasibility of bandwidth estimation in wireless access networks*. Wireless Networks, vol. 17, no. 6, page 1561–1580, Aug 2011. (Cited in page 12.)

[Ksentini 2017] Adlen Ksentini and Navid Nikaein. *Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction*. IEEE Communications Magazine, vol. 55, no. 6, page 102–108, 2017. (Cited in page 75.)

[Kushner 2004] H. J. Kushner and P. A. Whiting. *"Convergence of proportional-fair sharing algorithms under general conditions," IEEE Transactions on Wireless Communications, vol. 3*. p, vol. 1250, July 2004. (Cited in page 17.)

[Lee 2012] Daewon Lee, Hanbyul Seo, Bruno Clerckx, Eric Hardouin, David Mazzarese, Satoshi Nagata and Krishna Sayana. *Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges*. IEEE Communications Magazine, vol. 50, no. 2, page 148–155, Feb 2012. (Cited in page 73.)

[Li 2012] Yunxin Li. *An Overview of the DSRC/WAVE Technology*. In Xi Zhang and Daji Qiao, editors, Quality, Reliability, Security and Robustness in Heterogeneous Networks, volume 74, pages 544–558. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. Series Title: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. (Cited in page 2.)

[Li 2016] Yuanjie Li, Chunyi Peng, Zengwen Yuan, Jiayao Li, Haotian Deng and Tao Wang. *Mobileinsight: Extracting and Analyzing Cellular Network Information on Smartphones*. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, MobiCom '16, pages 202–215, New York, NY, USA, 2016. ACM. event-place: New York City, New York. (Cited in page 38.)

[Liaw 2002] Andy Liaw and Matthew Wiener. *Classification and Regression by randomForest.* R News, vol. 2, no. 3, pages 18–22, 2002. (Cited in page 26.)

[Liu 2008] Xin Liu, Ashwin Sridharan, Sridhar Machiraju, Mukund Seshadri and Hui Zang. *Experiences in a 3G network: interplay between the wireless channel and applications.* In Proceedings of the 14th ACM international conference on Mobile computing and networking - MobiCom '08, page 211. ACM Press, 2008. (Cited in page 12.)

[Lu 2015] Feng Lu, Hao Du, Ankur Jain, Geoffrey M. Voelker, Alex C. Snoeren and Andreas Terzis. *CQIC: Revisiting Cross-Layer Congestion Control for Cellular Networks.* In Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications - HotMobile '15, page 45–50. ACM Press, 2015. (Cited in pages 4, 13, and 17.)

[Luoto 2016] Petri Luoto, Mehdi Bennis, Pekka Pirinen, Sumudu Samarakoon, Kari Horneman and Matti Latva-aho. *System Level Performance Evaluation of LTE-V2X Network.* ArXiv, vol. abs/1604.08734, 2016. (Cited in page 12.)

[Mahindra 2013] Rajesh Mahindra, Mohammad A. Khojastepour, Honghai Zhang and Sampath Rangarajan. *Radio Access Network sharing in cellular networks.* In 2013 21st IEEE International Conference on Network Protocols (ICNP), page 1–10, Oct 2013. (Cited in page 76.)

[Mandelli 2019] Silvio Mandelli, Matthew Andrews, Sem Borst and Siegfried Klein. *Satisfying Network Slicing Constraints via 5G MAC Scheduling.* In IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, page 2332–2340. IEEE, Apr 2019. (Cited in page 75.)

[Mantovani 2015] Rafael G. Mantovani, Andre L. D. Rossi, Joaquin Vanschoren, Bernd Bischl and Andre C. P. L. F. de Carvalho. *Effectiveness of Random Search in SVM hyper-parameter tuning.* In 2015 International Joint Conference on Neural Networks (IJCNN), page 1–8. IEEE, Jul 2015. (Cited in page 31.)

[Margolies 2014] Robert Margolies, Ashwin Sridharan, Vaneet Aggarwal, Rittwik Jana, N. K. Shankaranarayanan, Vinay A. Vaishampayan and Gil Zussman. *Exploiting mobility in proportional fair cellular scheduling: Measurements and algorithms.* In IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, page 1339–1347. IEEE, Apr 2014. (Cited in pages 13 and 17.)

[Mehlfuhrer 2009] C. Mehlfuhrer, M. Wrulich, J. C. Ikuno, D. Bosanska and M. Rupp. *"Simulating the Long Term Evolution Physical Layer.* In " in Proc. 17th European Signal Processing Conference EUSIPCO 2009, Scotland, August 2009. Glasgow. (Cited in page 38.)

[Mekki 2019] Kais Mekki, Eddy Bajic, Frederic Chaxel and Fernand Meyer. *A comparative study of LPWAN technologies for large-scale IoT deployment.* ICT Express, vol. 5, no. 1, page 1–7, Mar 2019. (Cited in page 1.)

[Mitola 2000] J. Mitola. "cognitive radio: An integrated agent architecture for software defined radio," doctor of technology, royal inst. Technol. (KTH), Stockholm, Sweden, 2000. (Cited in page 39.)

[Nikaein 2014] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp and Christian Bonnet. *Openairinterface: A flexible platform for 5g research, SIGCOMM Comput.* Commun. Rev, vol. 44, page 5, October 2014. (Cited in page 40.)

[Nshimiyimana 2016] Arcade Nshimiyimana, Deepak Agrawal and Wasim Arif. *Comprehensive survey of V2V communication for 4G mobile and wireless technology.* In 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pages 1722–1726, Chennai, India, March 2016. IEEE. (Cited in page 2.)

[Ordonez-Lucena 2017] Jose Ordonez-Lucena, Pablo Ameigeiras, Diego Lopez, Juan J. Ramos-Munoz, Javier Lorca and Jesus Folgueira. *Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges.* IEEE Communications Magazine, vol. 55, no. 5, page 80–87, May 2017. (Cited in page 75.)

[Papa 2019] Arled Papa, Markus Klugel, Leonardo Goratti, Tinku Rasheed and Wolfgang Kellerer. *Optimizing Dynamic RAN Slicing in Programmable 5G Networks.* In ICC 2019 - 2019 IEEE International Conference on Communications (ICC), page 1–7. IEEE, May 2019. (Cited in page 75.)

[Parmentelat 2018] Thierry Parmentelat, Thierry Turletti, Walid Dabbous, Mohamed Naoufal Mahfoudi and Francesco Bronzino. *nepi-ng: an efficient experiment control tool in R2lab.* ACM WiNTECH, vol. 2018, pages 1–8, November 2018. (Cited in page 44.)

[Pedersen 2016] Klaus I. Pedersen, Gilberto Berardinelli, Frank Frederiksen, Preben Mogensen and Agnieszka Szufarska. *A flexible 5G frame structure design for frequency-division duplex cases.* IEEE Communications Magazine, vol. 54, no. 3, page 53–59, Mar 2016. (Cited in page 72.)

[Phan 2011] M.-A. Phan, R. Rembarz and S. Sories. *capacity analysis for the transmission of event and cooperative awareness messages in LTE networks.* The World Congress on Intelligent Transport Systems, Orlando, USA, vol. 18, October 2011. (Cited in page 12.)

[Piro 2011] G. Piro, L. Grieco, G. Boggia, F. Capozzi and P. Camarda. *"Simulating LTE Cellular Systems: An Open-Source Framework," IEEE Trans.* Veh, vol. 60, page 2, February 2011. (Cited in page 39.)

[Qazi 2017] Zafar Ayyub Qazi, Melvin Walls, Aurojit Panda, Vyas Sekar, Sylvia Ratnasamy and Scott Shenker. *A High Performance Packet Core for Next Generation Cellular Networks.* In Proceedings of the Conference of the ACM Special Interest Group on Data Communication - SIGCOMM '17, page 348–361. ACM Press, 2017. (Cited in page 69.)

[Raca 2018] D. Raca, J. J. Quinlan, A. H. Zahran and C. J. Sreenan. *Beyond Throughput: a 4G LTE Dataset with Channel and Context Metrics.* In Proceedings of ACM Multimedia Systems Conference (MMSys, vol. 2018, pages 12–15, June 2018. (Cited in page 38.)

[Richter 2005] Andreas Richter, Reiner Thomä, Martin Haardt and Ernst Bonek. Estimation of radio channel parameters: models and algorithms. ISLE, Ilmenau, 2005. OCLC: 254771575. (Cited in page 3.)

[Rondeau 2014] Tom Rondeau. Liblte github repository, last access 17/03/2020 `https://github.com/trondeau/libLTE.` 2014. (Cited in page 40.)

[Rost 2016] Peter Rost, Albert Banchs, Ignacio Berberana, Markus Breitbach, Mark Doll, Heinz Droste, Christian Mannweiler, Miguel A. Puente, Konstantinos Samdanis and Bessem Sayadi. *Mobile network architecture evolution toward 5G.* IEEE Communications Magazine, vol. 54, no. 5, page 84–91, May 2016. (Cited in page 70.)

[Rost 2017] Peter Rost, Christian Mannweiler, Diomidis S. Michalopoulos, Cinzia Sartori, Vincenzo Sciancalepore, Nishanth Sastry, Oliver Holland, Shreya Tayade, Bin Han, Dario Bega and et al. *Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks.* IEEE Communications Magazine, vol. 55, no. 5, page 72–79, May 2017. (Cited in page 75.)

[Sallent 2017] Oriol Sallent, Jordi Perez-Romero, Ramon Ferrus and Ramon Agusti. *On Radio Access Network Slicing from a Radio Resource Management Perspective.* IEEE Wireless Communications, vol. 24, no. 5, page 166–174, Oct 2017. (Cited in page 77.)

[Samba 2017] A. Samba, Y. Busnel, A. Blanc, P. Dooze and G. Simon. *Instantaneous throughput prediction in cellular networks: Which information is needed?* In IFIP/IEEE International Symposium on Integrated Network Management (IM, May 2017. (Cited in pages 13, 14, 18, and 19.)

[Sapankevych 2009] N. Sapankevych and R. Sankar. *Time series prediction using support vector machines: A survey.* IEEE Computational Intelligence Magazine, vol. 4, page 2, 2009. (Cited in page 26.)

[Sato 2009] Kengo Sato, Yutaka Saito and Yasubumi Sakakibara. *Gradient-Based Optimization of Hyperparameters for Base-Pairing Profile Local Alignment Kernels.* In Genome Informatics 2009, page 128–138. Imperial College Press, Oct 2009. (Cited in page 31.)

[Sawahashi 2010] Mamoru Sawahashi, Yoshihisa Kishiyama, Akihito Morimoto, Daisuke Nishikawa and Motohiro Tanno. *Coordinated multipoint transmission/reception techniques for LTE-advanced [Coordinated and Distributed MIMO.* IEEE Wireless Communications, vol. 17, no. 3, page 26–34, Jun 2010. (Cited in page 73.)

[Sciancalepore 2017] Vincenzo Sciancalepore, Konstantinos Samdanis, Xavier Costa-Perez, Dario Bega, Marco Gramaglia and Albert Banchs. *Mobile traffic forecasting for maximizing 5G network slicing resource utilization.* In IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, page 1–9. IEEE, May 2017. (Cited in page 80.)

[Smola 2004] A. J. Smola and B.: A Sch¨olkopf. *tutorial on support vector regression.* Statistics and Computing, vol. 14, page 3, August 2004. (Cited in page 26.)

[Sánchez A 2003] V.David Sánchez A. *Advanced support vector machines and kernel methods.* Neurocomputing, vol. 55, no. 1–2, page 5–20, Sep 2003. (Cited in page 28.)

[Trichias 2011] K. Trichias. Modeling and evaluation of lte in intelligent transportation systems. University of Twente and TNO, Enschede, Netherlands, 2011. (Cited in page 12.)

[Trivedi 2014] R. Trivedi and M. Patel. *Comparison of Different Scheduling Algorithm for LTE.* International Journal of Emerging Technology and Advanced Engineering", vol. 4, no. 5, pages 334–339, May 2014. (Cited in page 17.)

[Varma 2006] S. Varma and R. Simon. *Bias in error estimation when using cross-validation for model selection.* BMC Bioinformatics, vol. 7, 2006. (Cited in page 33.)

[Villa 2012] Tania Villa, Ruben Merz and Raymond Knopp. "adaptive modulation and coding with hybrid-arq for latency-constrained networks". in the proceedings of IEEE European Wireless Conference (EW2012) Poznan, Poland, 2012. (Cited in page 8.)

[Vinel 2012] A. Vinel. *"3GPP LTE Versus IEEE 802.11p/WAVE: Which Technology Is Able to Support Cooperative Vehicular Safety Applications," IEEE Commun.* Letters, vol. 1, no. 2, pages 125–28, April 2012. (Cited in page 12.)

[Virdis 2016] A. Virdis, G. Stea and G. Nardini. *Simulating LTE/LTE-Advanced Networks with SimuLTE.* In DOI 10.1007/-319-26470-7_5, in: Advances in Intelligent Systems and Computing, Vol 402, pp. 83-105, , 15, pages 978–3. 2016. (Cited in page 39.)

[Wallace 1996] Mark Wallace. *Practical applications of constraint programming.* Constraints, vol. 1, no. 1–2, page 139–168, Sep 1996. (Cited in page 84.)

[Weisberg 1980] S. Weisberg. Applied linear regression. Wiley, New York.YE, J. (1998), 1980. (Cited in page 26.)

[Winstein 2013] K. Winstein, A. Sivaraman and H. Balakrishnan. *Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks.* In Proc. of Networked Systems Design & Implementation (NSDI). pages 459–472, 2013. (Cited in page 12.)

[Wu 2007] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand and D. Steinberg. *Top 10 algorithms in data mining.* Knowl. Inf, vol. 14, page 1, December 2007. (Cited in page 26.)

[Xu 2013] Q. Xu, S. Mehrotra, Z. Mao and J. Li. *"PROTEUS: Network Performance Forecast for Real-time, Interactive Mobile Applications," in ACM MobiSys.* 2013. (Cited in page 13.)

[Xu 2017] Zhigang Xu, Xiaochi Li, Xiangmo Zhao, Michael H. Zhang and Zhongren Wang. *DSRC versus 4G-LTE for Connected Vehicle Applications: A Study on Field Experiments of Vehicular Communication Performance.* Journal of Advanced Transportation, vol. 2017, page 2750452, August 2017. Publisher: Hindawi. (Cited in page 2.)

[Yan 2019] Mu Yan, Gang Feng, Jianhong Zhou, Yao Sun and Ying-Chang Liang. *Intelligent Resource Scheduling for 5G Radio Access Network Slicing.* IEEE Transactions on Vehicular Technology, vol. 68, no. 8, page 7691–7703, Aug 2019. (Cited in page 75.)

[Yin 2015] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar and Bruno Sinopoli. *A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP.* In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication - SIGCOMM '15, page 325–338. ACM Press, 2015. (Cited in page 3.)

[Yue 2017] C. Yue, R. Jin, K. Suh, Y. Qin, B. Wang and W. Wei. *LinkForecast: Cellular Link Bandwidth Prediction in LTE Networks- IEEE Transactions on Mobile Computing.* 2017. (Cited in pages 14, 18, and 19.)

[Zekić-Sušac 2014] Marijana Zekić-Sušac, Sanja Pfeifer and Nataša Šarlija. *A Comparison of Machine Learning Methods in a High-Dimensional Classification Problem.* Business Systems Research Journal, vol. 5, no. 3, 2014. (Cited in page 26.)

[Zhou 2016] Xuan Zhou, Rongpeng Li, Tao Chen and Honggang Zhang. *Network slicing as a service: enabling enterprises' own software-defined cellular networks.* IEEE Communications Magazine, vol. 54, no. 7, page 146–153, Jul 2016. (Cited in page 70.)

Abstract: The proliferation of sophisticated applications and services comes with diverse performance requirements as well as an exponential traffic growth for both upload and download. The cellular networks such as 4G and 5G are advocated to support this diverse and huge amount of data. This thesis work targets the enforcement of advanced cellular network supervision and management techniques taking the traffic explosion and diversity as two main challenges in these networks. The first contribution tackles the intelligence integration in cellular networks through the estimation of users uplink instantaneous throughput at small time granularities. A real time 4G testbed is deployed for such aim with an exhaustive metrics benchmark. Accurate estimations are achieved. The second contribution enforces the real time 5G slicing from radio resources perspective in a multi-cell system. For that, two exact optimization models are proposed. Due to their high convergence time, heuristics are developed and evaluated with the optimal models. Results are promising, as two heuristics are highly enforcing the real time RAN slicing.

Keywords: Cellular Network, 4G, 5G, Machine Learning, Metrology, RAN Slicing, Optimization.

Résumé : La prolifération d'applications et de services sophistiqués s'accompagne de diverses exigences de performances, ainsi que d'une croissance exponentielle du trafic pour le lien montant (uplink) et descendant (downlink). Les réseaux cellulaires tels que 4G et 5G évoluent pour prendre en charge cette quantité diversifiée et énorme de données. Le travail de cette thèse vise le renforcement de techniques avancées de gestion et supervision des réseaux cellulaires prenant l'explosion du trafic et sa diversité comme deux des principaux défis dans ces réseaux. La première contribution aborde l'intégration de l'intelligence dans les réseaux cellulaires via l'estimation du débit instantané sur le lien montant pour de petites granularités temporelles. Un banc d'essai 4G temps réel est déployé dans ce but de fournir un benchmark exhaustif des métriques de l'eNB. Des estimations précises sont ainsi obtenues. La deuxième contribution renforce le découpage 5G en temps réel au niveau des ressources radio dans un système multicellulaire. Pour cela, deux modèles d'optimisation ont été proposés. Du fait de leurs temps d'exécution trop long, des heuristiques ont été developées et évaluées en comparaisons des modèles optimaux. Les résultats sont prometteurs, les deux heuristiques renforçant fortement le découpage du RAN en temps réel.

Mots-clés : Réseaux Cellulaires, 4G, 5G, Apprentissage Automatique, Métrologie, Découpage RAN, Optimisation.