Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik

Optimal Selection and Adaptation of a Flexible Functional Split in 5G Radio Access Networks

Alberto Martínez Alba, M. Sc.

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr. Andreas Herkersdorf
Prüfer der Dissertation:	1.	Prof. Dr.-Ing. Wolfgang Kellerer
	2.	Prof. Dr. Navid Nikaein

Die Dissertation wurde am 28.09.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 28.02.2022 angenommen.

Optimal Selection and Adaptation of a Flexible Functional Split in 5G Radio Access Networks

Alberto Martínez Alba, M. Sc.

September 27, 2021

Abstract

The fifth generation of radio access networks aims at ambitious performance objectives, such as offering low-latency ultra-reliable communication, supporting massive machine-type deployments, and providing high data rates to conventional users. In order to accomplish these objectives, the network operator must procure an efficient management of the available network resources. A promising strategy to improve this efficiency is to increase the density of the deployed cells in the network, which enables the use of high frequency bands and reduces power consumption. However, increasing cell density may lead to aggravated inter-cell interference, thus hindering the achievement of high data rates and compromising network efficiency. In order to reduce the impact of inter-cell interference, it is proposed to split the functions of 5G base stations into two units: a centralized unit and a distributed unit. This functional split facilitates interference coordination among base stations and, in addition, reduces the operating and deployment costs of the network. Nonetheless, there is not a single optimal manner to perform this split. Depending on the location of the users and their instantaneous activities, the functional split that provides the best performance or the lowest cost may change. As a result, it would be clearly beneficial to adapt the optimal split to the instantaneous network situation. In this thesis, we investigate the problem of selecting and dynamically adapting the optimal functional split of a 5G radio access network. In order to do this properly, we present a dedicated cost model for flexible communication networks, which takes into account the cost of operating and changing the network state. Then, we formulate the functional split selection as non-linear mixed-integer optimization problems for various objectives, and evaluate multiple candidate strategies to solve them in a timely manner. In comparison with static approaches, we observe that a dynamically optimized functional split may lead to substantial increases in the user performance and operating cost reductions. Besides the theoretical contribution, we also describe a novel implementation of a radio access network that is able to change its functional split during runtime, with the intention of demonstrating the feasibility of a dynamic functional split adaptation. Finally, we propose and compare several adaptation strategies for the network operator to decide when and how to change the functional split. We conclude that a lively adapted functional split is indeed possible and may result in considerably improved user data rates and lower operating cost with respect to conventional approaches.

Kurzfassung

Die fünfte Generation von Funkzugangsnetzen verfolgt ehrgeizige Leistungsziele, wie zum Beispiel die Bereitstellung einer extrem zuverlässigen Kommunikation mit niedrigen Latenzzeiten, die Unterstützung massiver maschinenartiger Anwendungen und die Bereitstellung hoher Datenraten für konventionelle Nutzer. Um diese Ziele zu erreichen, muss der Netzbetreiber für eine effiziente Verwaltung der verfügbaren Netzressourcen sorgen. Eine vielversprechende Strategie zur Verbesserung dieser Effizienz besteht darin, die Dichte der im Netz eingesetzten Zellen zu erhöhen, was die Nutzung von Hochfrequenzbändern ermöglicht und den Stromverbrauch reduziert. Die Erhöhung der Zelldichte kann jedoch zu einer verstärkten Interferenz zwischen den Zellen führen, was das Erreichen hoher Datenraten behindert und die Netzeffizienz beeinträchtigt. Um die Auswirkungen von Interferenzen zwischen den Zellen zu verringern, wird vorgeschlagen, die Funktionen von 5G-Basisstationen in zwei Einheiten aufzuteilen: eine zentrale Einheit und eine dezentrale Einheit. Diese funktionale Aufteilung erleichtert die Interferenzkoordination zwischen den Basisstationen und senkt darüber hinaus die Betriebs- und Errichtungskosten des Netzes. Es gibt jedoch nicht die eine optimale Art und Weise, diese Aufteilung vorzunehmen. Je nach Standort der Nutzer und ihren momentanen Aktivitäten kann sich die Funktionsaufteilung, die die beste Leistung oder die niedrigsten Kosten bietet, ändern. Daher wäre es eindeutig von Vorteil, die optimale Aufteilung an die momentane Netzsituation anzupassen. In dieser Arbeit untersuchen wir das Problem der Auswahl und der in Echtzeit laufenden Anpassung der optimalen Funktionsaufteilung eines 5G-Funkzugangsnetzes. Zu diesem Zweck stellen wir ein spezielles Kostenmodell für flexible Kommunikationsnetze vor, das die Kosten für den Betrieb und die änderung des Netzzustands berücksichtigt. Anschließend formulieren wir das Problem der Funktionsaufteilung als nichtlineares gemischt-ganzzahliges Optimierungsproblem für verschiedene Ziele und bewerten mehrere Kandidatenstrategien, um sie zeitnah zu lösen. Im Vergleich zu statischen Ansätzen stellen wir fest, dass eine dynamisch optimierte Funktionsaufteilung zu einer erheblichen Steigerung der Nutzerleistung und einer Senkung der Betriebskosten führen kann. Neben diesem theoretischen Beitrag demonstrieren wir die Machbarkeit einer dynamischen Anpassung der Funktionsaufteilung anhand einer neuartigen Implementierung eines Funkzugangsnetzes, welches seine Funktionsaufteilung während der Laufzeit kann. Schließlich schlagen wir verschiedene Anpassungsstrategien für den Netzbetreiber vor und vergleichen diese, um zu entscheiden, wann und wie der Funktionssplit geändert werden soll. Wir kommen zu dem Schluss, dass ein dynamisch angepasster Funktionssplit in der Tat möglich ist und im Vergleich zu konventionellen Ansätzen zu erheblich besseren Nutzerdatenraten und niedrigeren Betriebskosten führen kann.

Contents

1. Introduction

Mobile network operators rely on constant innovation to satisfy the increasingly stringent demands of their ever-growing number of customers. The ubiquity and convenience of mobile communications attract both end users and application providers, between which operators are forced to unceasingly improve their management and infrastructure in order to keep up with the expectations. Indeed, the mobile operators that provided mobile connectivity to 8.8 billion mobile devices in 2018 are foreseen to face a demand of 13.1 billion devices by 2023 [Cis20], of which 1.4 billion devices will be 5G-capable. Moreover, the average download speed of these 5G devices is expected to grow from 76 Mb/s in 2019 to 575 Mb/s by 2023 [Cis20], owing to the emergence of applications requiring high data rates, such as those featuring high-definition video. In fact, while 59% of mobile traffic in 2017 is already video traffic, by 2022 video traffic will be almost 80% of all mobile traffic [Cis19]. These trends are spurring the redesign of 5G radio access networks (RAN) so as to improve their provided service.

Not only mobile operators are compelled to provide increasingly better service to more users, but they also need to do this at relatively lower prices. In fact, the worldwide median price of providing a data service of 1.5 GB per month dropped an average of 10.4% every year from 2013 to 2020, resulting in an accumulated price reduction of around 50% in seven years [Uni21]. This trend is even steeper for developing and least-developed countries, exhibiting yearly price reductions of 13.3% and 18.5%, respectively, which translates in ca. 60% and 75% accumulated price reductions [Uni21]. As direct consequence, the revenue associated with providing these mobile services tends to decrease as well. For example, European mobile operators reported a total revenue of 177 billion Euros in 2006, whereas in 2016 this revenue was only 138 billion Euros, a 22% difference, even though the total number of connected devices increased by 40% [GAO17].

Having to provide better user service at lower prices would be enough motivation for mobile operators to pursue high resource efficiency. Furthermore, nowadays customers explicitly demand this feature to operators, either because of environmental awareness [Eur17; Eur21] or even because of health concerns [Ips20]. Thus, efficiency is not only strictly required to profitably operate a communication network, but it can be also used as a selling feature to attract potential users. Consequently, mobile operators continuously seek strategies to improve the performance-to-cost ratio of their current infrastructure before considering to invest in additional resources, and prefer efficient resource utilization over simpler but more costly operation.

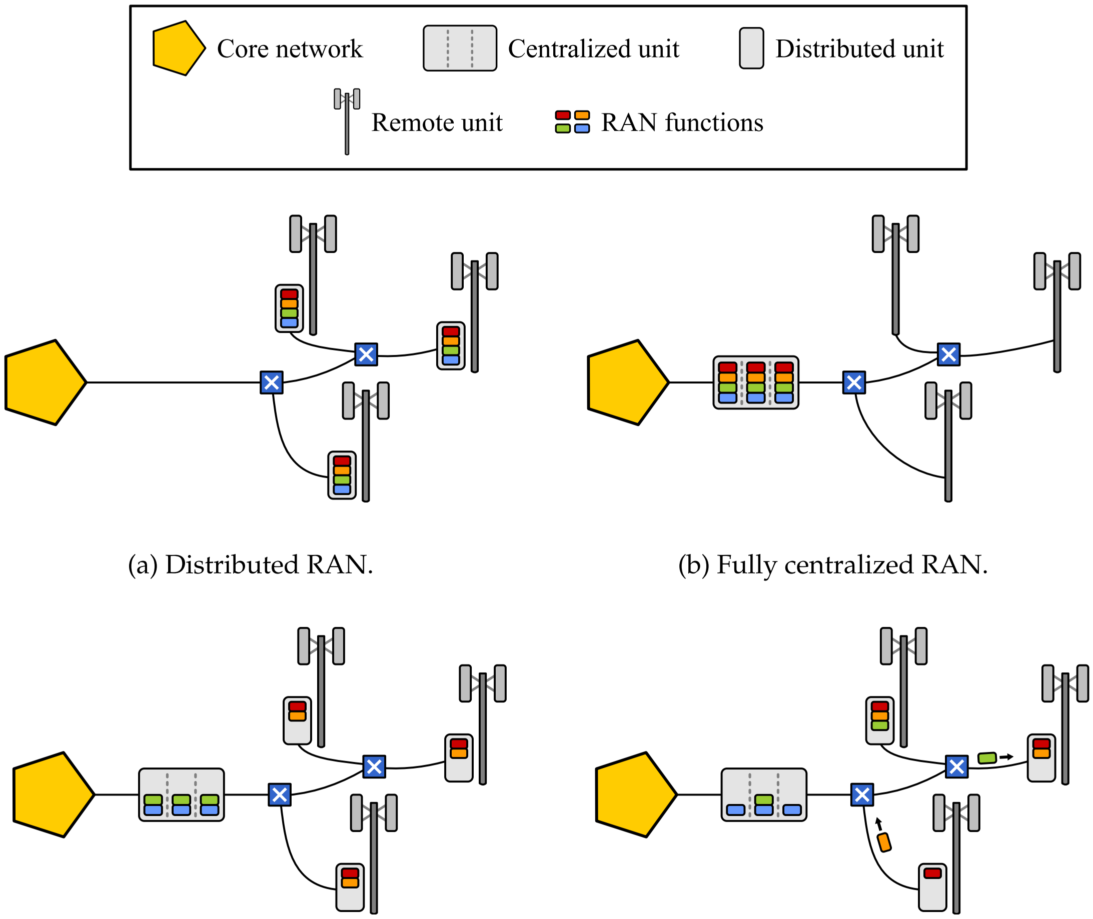
A mandatory step in order to increase resource efficiency of mobile radio access networks is to increase network density, that is, to deploy more base stations per area unit [KS17]. In dense RANs, users are always close to its serving base station, which enables the use of high-frequency bands and increases the received signal power with comparatively low transmission power. In contrast, sparse RANs tend to produce more shadow areas even though the transmission powers of the base stations are higher. However, increased RAN density also leads to increased inter-cell interference, which may jeopardize the overall resource efficiency.

Techniques for dealing with interference exist since the the inception of mobile networks and have evolved over time, ranging from simple frequency allocation [Eng+98] to advanced joint transmission and reception [Sey+16]. Simple techniques can be implemented easily, but their performance is limited and are not sufficient to operate dense deployments. Advanced techniques promise, in theory, powerful interference avoidance or cancellation, but their actual effectiveness may be severely affected by implementation details [Jun+10]. Namely, the more advanced the interference technique, the more and better coordination is needed among involved base stations [Döt+13]. This translates into strict latency and capacity constraints between coordinating base stations, which legacy 3G and 4G networks struggle to satisfy.

In conventional 4G networks, each base station individually receives and processes user data packets, and takes scheduling, encoding, and modulation decisions independently from other cells [3GP21i]. This RAN architecture is depicted in Fig. 1.1a. Although there are logical interfaces designed to provide direct communication between base stations and promote inter-cell coordination [3GP20a], their limitations, owing to the fact that cells may be located far away from each other, often restrict their ability to coordinate to static or semi-static interference mitigation techniques [Kos+12]. However, with the emergence of network softwarization techniques, new network architectures are proposed so as to provide better inter-cell coordination. Namely, if mobile processing functions, which are usually deployed on dedicated hardware, are replaced by software functions, these can be easily relocated into commodity data centers. This leads to a centralized RAN architecture, in which the processing of all base stations is moved to a single central location [Che+14a], as depicted in Fig. 1.1b. As a result, this architecture enables fast communication between all cells, thus allowing the use of advanced interference mitigation techniques, which are required to operate dense networks.

Function centralization also entails lower deployment and operating costs, since the computing resources of off-the-shelf equipment can be pooled to leverage multiplexing gain [Che+14a]. Nonetheless, centralized RAN architectures have also a major disadvantage. Moving all processing functions away from the remote sites, which contain the antennas and radio equipment, requires the presence of high-capacity, low-latency links between them and the data center [Döt+13]. Current mobile operators, however, cannot meet these requirements without re-deploying the whole RAN, which incurs in high additional costs [Che+16a].

The solution proposed by 3GPP to benefit from the advantages of function centralization without compromising the capacity of currently-deployed RANs is to opt for a *partially centralized* architecture [3GP21i], as shown in Fig. 1.1c. With partial centraliza-



(a) Distributed RAN.

(b) Fully centralized RAN.

(c) Partially centralized RAN.

(d) Dynamically centralized RAN.

Figure 1.1.: Architecture of distributed, fully centralized, partially centralized, and dynamically centralized radio access networks.

tion, only a subset of the processing functions is moved to the data center, so that the links between the data center and the remote sites are not overloaded. This may still lead to cost reductions and enhanced interference mitigation techniques, depending on which and how many functions are centralized. In turn, a partially centralized architecture introduces a new problem: selecting which functions should be centralized. Since function centralization affects both the operating cost and the user performance, via interference mitigation, the operator has to use these two performance indicators to decide on the optimal *centralization level*. In addition, as mobile networks are rather dynamic due to the mobility of the users and their time-dependent patterns, the optimal centralization level may vary over time. Hence, being able to adapt the centralization level to the instantaneous network conditions, as depicted in Fig. 1.1d, may result in higher profits with respect to a static configuration [MAK19a]. A dynamic adaptation of the centralization level is, nonetheless, a completely novel feature for a mobile RAN. As such, a careful consideration of all the time-varying cost components, including those related to adapting the centralization level and those reflecting user performance, is strictly required. Furthermore, it is also necessary to show that the current state of technology can support this novel adaptation, hopefully without incurring in service disruptions.

In the light of this, the objectives of this doctoral thesis can be summarized as follows. First, we intend to improve the understanding of the influence of the centralization level on the performance and operating cost of a partially centralized 5G RAN. This also includes modeling cost of operating with an inadequate centralization level, and the cost of updating it when so decided. The second goal is to find a timely and cost-efficient algorithm to find the optimal centralization level, so that the network operator can be aware of its evolution and anticipate the benefits of updating it. Third, we want to show that the set of centralized functions can be dynamically changed in an actual 5G network without severely affecting user performance or increasing operating cost, thus proving that a dynamic reconfiguration of a 5G RAN is technically feasible. Finally, we aim to apply our knowledge about the optimal centralization level and the changing capabilities of a 5G RAN to derive an optimal rule for deciding when to trigger a reconfiguration.

1.1. Research challenges

The modeling, design, and optimization of 5G radio access networks featuring flexible centralization levels entail several research challenges. This section summarizes these challenges, which are discussed in detail in Chapters 3 to 6.

Optimal selection of the RAN centralization level

Being constrained by the capacity of the network connecting the data center with the remote units, the first challenge is to find a feasible subset of centralized RAN functions that maximizes a chosen performance indicator. This can be formulated as an optimization problem, in which the objective function reflects our performance indicator (such as operating cost or user throughput) and the constraints model the capacity of the network. If each centralized function had an associated constant utility and the network constraints could be converted into an equivalent upper limit to the number of centralized functions, we could formulate this problem as an instance of the well-known knapsack problem [MT90; MAK19a]. Nonetheless, in real networks the objective function may be non-linear and the network constraints may impose more complicated restrictions. Therefore, the actual optimization problem has to be formulated from a more complicated network model.

We need to model how the centralization level affects the required capacity on every link and make sure that any obtained solution is within their actual capacity. If the objective function depended only on routing costs, we could achieve this by formulating an instance of the multicommodity flow problem [AMO13]. However, the objective function needs to reflect multiple additional factors. On the one hand, the operator may be interested in minimizing the operating cost associated with each selection of centralized functions. On the other hand, satisfying user demands is also important for the operator, so that we could also focus on centralizing those functions that maximize user utility. For this, we need a model of the mutual interaction between coordinating base stations and its conversion into interference mitigation, then transform interference mitigation into user throughput, and finally relate this throughput to actual utility or quality of experience.

After the optimization problem is formulated, we have to figure out an adequate approach to solve it. We can safely anticipate that this optimization problem is hard to solve, based for example on the fact that a very simplified version of it would result in an instance of the knapsack problem, which is already NP-Hard [MT90]. At the same time, since an actual dense RAN may include hundreds of base stations, it is clear that problem instances may be rather large. Besides, if we intend to dynamically adapt the centralization level, we need to be able to calculate optimal or near-optimal solutions in a timely manner, which may be a difficult feat for large, complex optimization problems. Consequently, we may need to derive approximating approaches or heuristics to find good-quality solutions that converge faster than exact approaches.

Cost-efficient management of a flexible 5G RAN

Once the mobile network operator has a timely and effective algorithm to find the optimal centralization level for a given situation, an important design decision has to be taken. Namely, the operator must decide between a static configuration, in which the set of centralized functions is not modified during operation, and a dynamic operation, in which the centralization level is adapted to the instantaneous situation. Owing to the dynamic nature of mobile networks, the disadvantages of the former are clear: the instantaneous traffic and user distribution may differ substantially from the one at which the static configuration performs optimally. Conversely, a RAN featuring a dynamically-adapting centralization level could track changes in the environment and thus maximize performance and minimize cost over time. Nonetheless, we need to take into account the cost implications of dynamically reconfiguring the centralization level in order to conclude that dynamic adaptation is indeed preferable.

A model of the cost of a dynamically-adapting network has to reflect various components. First, we need to find out how costly it is to operate on suboptimal, obsolete centralization levels with respect to the optimal one. If the cost difference is small, there may be no incentive to perform a change. For this, characterizing the degradation of optimal solutions over time is required. Second, we have to estimate how costly it is to perform a change in the centralization level. There may be multiple factors involved in this estimation: additional resource consumption, compensations for service disruptions, increased end-to-end latency, etc. Finally, we have to ensure that the network can indeed cope with environmental changes fast enough. That is, that the delay between a change in the optimal centralization level and its associated response is not too long, otherwise dynamic adaptation may be pointless.

Realizing a reconfigurable 5G RAN

The ability to dynamically change the centralization level, that is, to modify the set of centralized functions during runtime, is completely novel for mobile radio access networks. Although there are works from the research community investigating this feature [Die+21; DHA20; HR18b], current 5G specifications only describe a static, partially centralized architecture that is identical for all base stations [3GP21d], and at most they consider the possibility of supporting multiple static centralization levels on different base stations in the future [3GP17]. Therefore, in order to show the effectiveness of a dynamically-adapting RAN, we must ensure that this is indeed technically feasible with the current state of technology.

Realizing a software platform that supports standard 5G operation while being able to centralize or distribute functions is a challenging task. Numerous constraints have to be taken into account not to disrupt the user connections, such as ensuring that no packets are lost, end-to-end latency is not severely increased, signaling is still routed and processed correctly, etc. Albeit there are software platforms supporting live migration of virtual network functions [Rup+18], they still may not be ideal at meeting these constraints. As a result, a dedicated migration platform may be required.

Dynamic adaptation of a RAN centralization level

Let us assume that the previous three challenges have been successfully overcome. We have derived a fast and good-quality optimization approach to find the optimal or near-optimal centralization levels for a given traffic and distribution of users. In addition, we have accurately modeled the cost of operating at each possible centralization level, and the cost of changing it. Finally, we have a migration platform that enables such changes in a timely and cost-efficient manner. Even in that case, we still need to have a rule to help us decide *when* to change the centralization level. If the cost of moving RAN functions is larger than the average cost difference between the optimal and recently obsolete centralization levels, it would make sense not to change the centralization level continuously, but to wait until the change is worthwhile. However, waiting too long may be counterproductive, since it would also result in increased operating cost. There must be, hence, an optimal waiting time between both extremes, which is not trivial to find.

Therefore, in order to profitably operate a flexible 5G RAN, we need to devise a way to find an optimal or near-optimal rule to trigger adaptations of the centralization level. The main difficulty of this task stems from the fact that variations in user traffic and mobility are hard to estimate accurately. This may result in inadequate adaptation rules that perform suboptimally.

1.2. Main contributions

In this section, we provide a summary of the main contributions of this doctoral thesis, in relation to the aforementioned research challenges. We can classify these contribution into four major groups: (i) proposal of an optimization framework to find the best centralization level, (ii) derivation of a cost model for flexible networks, (iii) implementation of a proof-of-concept adaptive 5G RAN, and (iv) proposal and evaluation of dynamically adaptive strategies.

The first contribution comprises the network modeling, problem formulation, and approach derivation and assessment required to find the optimal centralization level for any instantaneous network situation. Namely, we model the interference coordination capabilities of all base stations according to their centralization level, how this coordination translates into user throughput, and the final transformation into proportionally fair utility to better reflect the quality of experience of the user. This model is then combined with the network constraints to formulate an optimization problem, which may be used to maximize user throughput in a proportionally fair manner, minimize operating cost, or maximize combined operating revenue, which depends on both performance and operating cost. Furthermore, we relax the original, intractable problem formulation into increasingly simpler and faster approximations that nonetheless yield good quality solutions, with the intention of finding a suitable algorithm to support dynamic optimization. All these formulations are fully independent on the network parameters, and thus they can be employed for an arbitrary number of base stations, users, and centralization options, link capacities, interference-mitigation capabilities, etc. Finally, the convergence time and performance of all considered approaches are thoroughly evaluated under a wide range of network conditions and compared against static approaches.

The second contribution addresses the derivation of a model that reflects all cost components playing a role in the operation of a flexible, dynamically-adapting network. Namely, we define and propose expressions to estimate the cost of operating the network in optimal and obsolete states, as well as the cost of deciding and realizing state changes. We present new functions characterizing the cost degradation of a communication network and base upon a probability-theory framework to derive a complete estimation of the average cost. This estimation is then combined with that of the cost of deciding and realizing state changes, which are analyzed separately, in order to yield a final expression for the total operating cost. This can be used to configure design parameters and decide whether a dynamic network is profitable. In addition to using this cost model for our own use case, we provide example applications of how to employ it on other types of communication networks.

The third contribution is more practical, since it concerns the implementation of a proof-of-concept 5G RAN that features the ability of changing its centralization level without service interruptions. Even though we show via theoretical analysis and simulations that a dynamic centralization level often leads to better performance and lower cost than static solutions, it would be unclear whether these benefits are achievable in practice without an actual implementation, owing to the novelty of the proposed architecture and adaptation approaches. That is the reason why we present the design of a simple 5G RAN that can change between two centralization levels during runtime. We identify the challenges associated with centralizing and distributing running functions, such as rerouting their data and control traffic, transferring the state of transmission and reception buffers, switching on and off processing elements, etc. Then, we propose a migration strategy that provides a seamless transitions between a lower and a higher centralization level, with minimal packet losses and negligible migration latency. In addition, we discuss a strategy to completely remove packet losses at the expense of slightly higher latency, and vice-versa, so that these can be configured to the preference of the operator.

The fourth and last contribution consists in the derivation and evaluation of adaptation strategies, which allow us to identify the best moments to trigger a change in the centralization level. For this, we use the previously derived cost model to characterize the cost of keeping the current the centralization level and compare it against the cost of changing it and switching to the optimal configuration. We realize that the derivation of the optimal rule is not possible unless full information is known about the future evolution of the system. Nonetheless, we propose near-optimal, self-adjusting strategies that approximate the performance of the optimal rule. Finally, we evaluate all considered strategies under a wide range of mobility dynamics and network conditions.

1.3. Outline

A detailed diagram showing the outline of the thesis and summarizing the content of each chapter is depicted in Fig. 1.2. In essence, the remainder of this thesis is organized as follows.

Chapter 2 introduces the problem of optimally selecting the functional split which defines the centralization level of each base station in a 5G RAN. We first describe the architectural options that are considered in 5G and discuss their requirements, their associated interference-mitigation capabilities, and their overall advantages and disadvantages. Then, we introduce the general network model that we use throughout the thesis and use it to formulate static and dynamic optimization problems.

Chapter 3 presents the model that characterizes the cost components of an adaptive communication network. We propose new terminology to denote the distinct adaptation phases, along with the features of user demands and network states, based on a probability-theory framework. We analyze the properties of the discussed components and derive expressions to combine in order to assess the overall network profitability under dynamic conditions. Finally, we apply the cost model into a generic simulation example so as to demonstrate how it can be used.

Chapter 4 addresses the formulation of the optimization problem to select the instantaneously best centralization level according to user performance, operating cost, and combined revenue (also known as *readiness cost*). We tackle mixed-integer non-linear problems by proposing approximations and alternative reformulations to increase their tractability. In addition, we discuss two simple heuristic approaches. Then, we

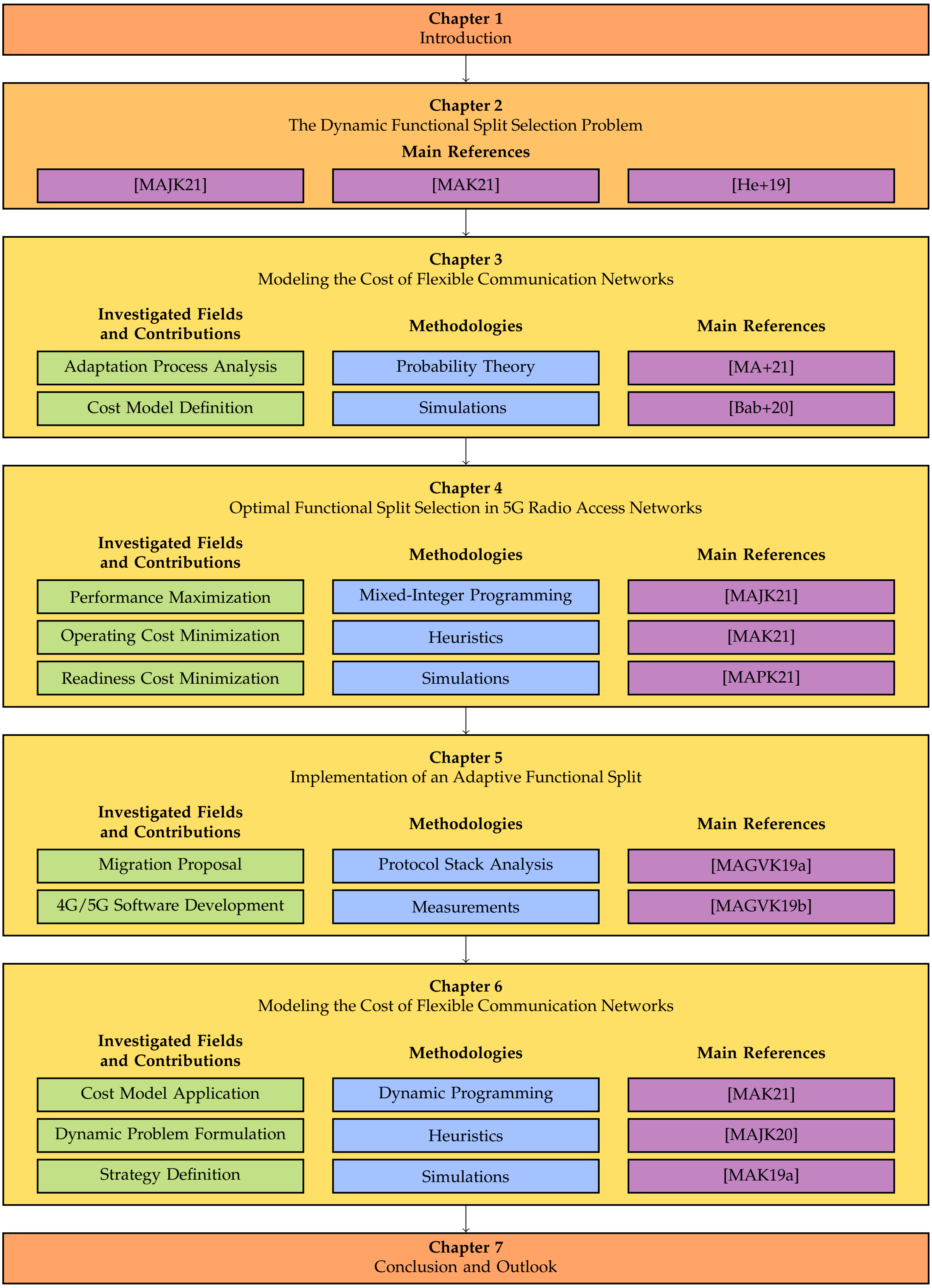

Chapter 1
Introduction
Chapter 2
The Dynamic Functional Split Selection Problem
Main References
[MAJK21]
[MAK21]
[He+19]
Chapter 3
Modeling the Cost of Flexible Communication Networks
Investigated Fields and Contributions
Methodologies
Main References
Adaptation Process Analysis
Probability Theory
[MA+21]
Cost Model Definition
Simulations
[Bab+20]
Chapter 4
Optimal Functional Split Selection in 5G Radio Access Networks
Investigated Fields and Contributions
Methodologies
Main References
Performance Maximization
Mixed-Integer Programming
[MAJK21]
Operating Cost Minimization
Heuristics
[MAK21]
Readiness Cost Minimization
Simulations
[MAPK21]
Chapter 5
Implementation of an Adaptive Functional Split
Investigated Fields and Contributions
Methodologies
Main References
Migration Proposal
Protocol Stack Analysis
[MAGVK19a]
4G/5G Software Development
Measurements
[MAGVK19b]
Chapter 6
Modeling the Cost of Flexible Communication Networks
Investigated Fields and Contributions
Methodologies
Main References
Cost Model Application
Dynamic Programming
[MAK21]
Dynamic Problem Formulation
Heuristics
[MAJK20]
Strategy Definition
Simulations
[MAK19a]
Chapter 7
Conclusion and Outlook

Figure 1.2.: Diagram of the outline of this thesis, including a summary of the contributions, methodologies, and main references of each chapter.

evaluate the convergence times and performance of all formulations by means of a dedicated simulator, which is designed to represent a wide range of network conditions regarding user distribution and network topology.

Chapter 5 discusses the implementation of our proof-of-concept adaptive 5G RAN, which features a dynamically reconfigurable centralization level. This implementation is realized by modifying an underlying 4G/5G software platform. We explain the challenges of achieving this along with our proposed strategies to migrate running RAN functions with minimal end-user impact. Finally, we present a selection of measurement results to back our conclusions.

Chapter 6 explains the application of the cost model presented in Chapter 3 to a dynamic 5G RAN, and extends the problem formulations presented in Chapter 4 to include dynamic optimization. We derive multiple adaptation strategies in order to realize an adaptive optimization, for which we use dynamic programming and heuristic methods. We lay out the problems of dynamically adapting, propose several adaptation strategies, and evaluate them in extensive simulations.

Finally, Chapter 7 concludes this thesis by summarizing our findings and contributions and discusses possible directions for future work.

2. The Dynamic Functional Split Selection Problem

2.1. Introduction

2.1.1. Motivation, scope, and challenges

The evolution from 4G to 5G entails improving a large number of features in legacy radio access networks (RAN). The motivation behind every 5G improvement can be related to two distinct objectives. On the one hand, operators intend to enhance user experience by reducing latency, increasing data rates, or supporting a high number of devices. This is performed in order to attract new users and facilitate the appearance of novel network applications. On the other hand, increasing network efficiency is also of utmost importance, since resources are limited and relative user income tends to decline over time. As a result, mobile network operators always seek to benefit from the latest advances in technology.

A good example of such advances is the emergence of network function virtualization (NFV) and software-defined networking (SDN), which have not gone unnoticed in the field of mobile networks. Indeed, replacing hardware equipment with software functions promises important cost reductions and improved performance, whereas being able to dynamically control the network operation from a centralized point also offers increased flexibility and scalability. One important application of the paradigms of NFV and SDN into the 5G RAN architecture is the definition of the *functional split*, with which the processing of each base station in the network is divided into centralized and distributed portions. Nonetheless, owing to the different advantages and disadvantages of each functional split option, the operator has to carefully decide which functions can be centralized and how to do this properly given the underlying network dynamics.

In this chapter, we provide a detailed background and define the problem of dynamically selecting the optimal functional split in a 5G RAN, which is the central matter of this thesis. In order to do this, we address a series of challenges. We first have to be aware of the objectives, opportunities, and limitations of 5G radio access networks, so as to approach this problem from a realistic perspective. Namely, we need to consider all the possible options for performing functional splits and model their impact on the performance and cost of a 5G RAN. This also implies taking into account the role of all network elements, such as those connecting centralized and distributed units. Moreover, we have to find an appropriate formulation of the functional split selection problem with the objective of finding the best realizable option that optimizes cost and/or performance. Finally, since our ultimate intention is to operate the network in real time, we also need to include the dynamic aspects of the system into the problem.

2.1.2. Key contributions

The main content and contributions of this chapter are based on those presented in [MAJK21], [MAK21], and [He+19], which can be summarized in relation to the aforementioned challenges as follows:

1. We provide a detailed description of the requirements, architectures, and proposed functional splits of 5G radio access networks. We put the emphasis on the impact of interference on the user experience and on how this interference can be mitigated.
2. We propose a complete network model to reflect all relevant parameters that may affect the performance and cost of each functional split option.
3. Based on this model, we formulate the functional split selection problem (FSSP) in a generic manner, only for static scenarios at first, and finally for dynamic scenarios after all the variable components of the network have been presented.

The remaining of this chapter is organized as follows. In Sec. 2.2, we discuss the alternatives for implementing a centralized 5G RAN, based on the 5G requirements. Sec. 2.3 presents our modeling approach and formulates the static functional split selection problem. In Sec. 2.4 we describe the flexibility features that are required in a 5G RAN implementing a dynamically adapting functional split and formulate the dynamic functional split selection problem. Finally, Sec. 2.5 summarizes and concludes the chapter.

2.2. Centralized 5G RAN architectures

Whereas 4G radio access networks usually feature distributed architectures, centralized or partially-centralized architectures have been considered for the 5G RAN since its inception, with the objective of improving user performance and reduce management costs. In this section, we introduce the proposed centralized architectures for 5G radio access networks. We then describe the multiple techniques for interference mitigation that can be leveraged by these architectures. Finally, we present the architectural options that are considered for current 5G and next-generation networks.

2.2.1. 5G use cases and requirements

5G mobile networks address three main use cases [NGM15]: massive machine-type communications (mMTC), ultra-reliable low-latency communications (URLLC), and enhanced mobile broadband (eMBB). The first use case, mMTC, considers networks connecting hundreds or even thousands of machine-type devices, that is, automated devices that operate without direct human intervention. In URLLC, the focus is on providing very reliable and low latency end-to-end connectivity, so as to be able to operate life-critical remote devices through a mobile network. Finally, eMBB, the third use case, is a direct evolution of previous use cases, since its objective is to provide higher data rates to human-type devices in urban or rural scenarios.

The mMTC and URLLC use cases envision a transformation of the traditional utilization of mobile networks as they move away from human-type communication, which has been the main driver for the deployment of these networks in the third and fourth generations. Nonetheless, high-throughput human-type communication is still a major concern for mobile network operators. The reason for this is twofold. On the one hand, owing to the novelty of mMTC and URLLC use cases, their underlying market is not yet consolidated and the potential profit of supporting these use cases is still unclear. Conversely, human-type markets are better known and thus their associated profits are easier to predict. On the other hand, users require ever-increasing data rates as new applications appear, such as those based on high-quality video streaming [Upd21]. As a result, supporting the eMBB use case is crucial for mobile operators. In this work, we address the problem of dynamically modifying the RAN architecture so as to enhance user data rates, and hence we focus on the eMBB use case.

There are three basic strategies to increase data rates in a wireless communication system. The first one is to increase spectral efficiency, that is, to improve modulation, coding, and transmission schemes so that the amount of information transmitted over a fixed channel bandwidth is optimized. In 5G, this is accomplished by using adaptive modulations, ranging from Binary Phase-Shift Keying (BPSK) to 256 Quadrature Amplitude Modulation (256-QAM) [3GP21e], high-efficiency polar and low-density parity-check (LDPC) codes [3GP21c; Bae+19], and advanced MIMO and beamforming techniques [Ali+17]. There are, nonetheless, physical limits to the maximum spectral efficiencies that can be achieved, such as that provided by the Shannon–Hartley theorem [Sha49]. As a result, we cannot rely on this strategy alone in order to arbitrarily increase user data rates.

The second strategy is to allocate additional radio-frequency spectrum, as the channel bandwidth is directly proportional to the achieved data rates. In 5G, this translates into the utilization of new frequency bands within the conventional, sub-6 GHz frequency range (also known as Frequency Range 1, or FR1) [3GP21g], as well as within millimeter-wave bands above 6 GHz (also known as Frequency Range 2, or FR2) [3GP21h]. However, although simple from a theoretical point of view, adding new spectrum entails two major disadvantages. On the one hand, spectrum is a scarce resource that has to be shared among multiple wireless services, such as satellite communications, TV broadcasting, government, emergency, military and professional communications, etc. Consequently, the price of buying new spectrum can be very high for mobile operators. On the other hand, high-frequency communications such as those performed in FR2 often exhibit bad propagation characteristics, since they are heavily attenuated by buildings and other obstacles between the transmission and the receiver [Mac+13]. As a resut, high-frequency communications are often constrained to short-range and/or line-of-sight radio links.

The third strategy consists in increasing cell density, that is, deploying more 5G cells per area unit. This strategy comes out as a consequence of the limitations of the previous two strategies. On the one hand, for fixed bandwidth and spectral efficiency, the only remaining possibility to increase data rates is to enhance the signal-to-interference-and-noise ratio (SINR), which can be accomplished by increasing received signal power. Although theoretically possible, increasing transmitted signal power of sparse deployments is usually not the best approach, since it may lead to shadowy coverage [Mac+13] and even raise health concerns [Lin16]. Hence, the radio equipment needs to be brought closer to the users so that SINR can be improved without higher transmission power. On the other hand, the bad propagation characteristics of high-frequency signals also require abundance of radio transmission points, so as to ensure that there are as few obstacles as possible between users and base stations.

However, increasing cell density is not an ideal solution either. Indeed, when trying to enhance the SINR by bringing cells closer together, we are at risk of worsen the situation because of the additional inter-cell interference. This problem can be addressed with the use of interference mitigation techniques. In the next section, we provide a summarized description of these techniques.

2.2.2. Interference mitigation in 5G networks

We classify the interference mitigation techniques that a 5G RAN may implement according to how fast they perform. Thus, we distinguish between static, semi-static, dynamic, and interference-canceling techniques.

Static interference mitigation

The simplest way of reducing inter-cell interference is to statically assign different frequency bands to each neighboring cell, which is called *frequency reuse*. As a result, interference can be completely avoided, at the expense of reduced maximum data rates, since each cell has only access to a piece of the whole available spectrum. Since this disadvantage may counter the benefits of interference avoidance, more sophisticated techniques such as *fractional frequency reuse* (FFR) have been proposed. With FFR, the whole band is used for UEs camping on the inner region of the cell, which is less affected by interference coming from other cells, whereas those UEs at the cell edge are served with smaller bands that are chosen not to collide with neighbor transmissions [Ham+13]. If the power allocated to these smaller bands is configured with fine granularity, we refer to it as *soft fractional frequency reuse* (SFFR), which allows for a greater control over the interference levels [Ham+13].

The main drawback of these techniques is that the allocation of frequency bands is fixed, regardless of the instantaneous interference situation. That is, in some cases, a cell may benefit from using a larger or a smaller piece of the frequency band if there is less or more inter-cell interference than that expected at network design, respectively [Bou+09]. Nonetheless, an important advantage of static interference mitigation techniques is that they do not require any actual coordination among the cells.

Semi-static interference mitigation

If cells are able to communicate with each other, they can exchange information about the channels that they would like to use to serve a UE, or the channels on which their UEs are experiencing the most interference. Therefore, they can use this information to try to minimize their mutual interference. Namely, they could send periodic reports indicating their preferred frequency bands and use received reports from other cells to agree on the bands that every cell should use. This technique, which is used extensively in 4G networks, is called *inter-cell interference coordination* (ICIC) [Sor+17].

Although more dynamic than FFR or SFFR, ICIC still bases on relatively infrequent coordination between neighbor cells. That is, cells still take their own scheduling decisions, although they are influenced by the interference information sent to and received from neighbor cells. Consequently, ICIC achieves better interference reduction than static techniques [Ham+13], but it is outperformed by faster techniques [Sor+17].

Dynamic interference mitigation

We define *dynamic interference mitigation* as those techniques which use real-time information from a set of cells in order to allocate the best transmission and reception resources for each individual cell. The most simple such technique is called *coordinated link adaptation*, which consists in taking independent time-frequency allocation decisions during scheduling and sharing them with neighbor cells, so that the modulation and coding schemes are adapted preemptively according to the predicted interferences [MA+18].

A more sophisticated approach is *coordinated scheduling*, in which scheduling decisions are taken jointly by all neighbor cells, either by distributed or centralized agreement [Nar+18]. In either case, fast communication between the cells is strictly required, since the scheduling interval in 5G networks ranges from 62.5 μs to 1 ms [3GP21d]. Similarly to coordinated scheduling, *coordinated beamforming* relies on joint agreement to decide on which beams to use to serve UEs, in order to minimize spatial interference [Che+16b].

These dynamic interference-mitigating techniques can be more effective than static or semi-static ones, since they are able to deal with the instantaneous interference situation. Nonetheless, they cannot completely prevent interference, specially when the RAN is congested. In those cases, there may not be enough non-overlapping resources to schedule the users, therefore interference would still be present [Nar+18].

Interference cancellation

The most sophisticated technique to mitigate interference is *interference cancellation*, that is, the removal of interference in the sheer radio or base-band signal. This can be performed in two different ways. For the downlink, if a cell knows the interfering signal coming from a neighbor cell before it is transmitted, it can subtract it in advance to its next transmission so that the received signal at the UE resembles the originally intended signal without interference. This technique is known as *joint transmission*[1] [Jun+14; Dav+13]. For the uplink, cells may share their received signals and compare them against one another in order to extract the individual contribution of interfering UEs. This technique is known as *joint reception* [HP17].

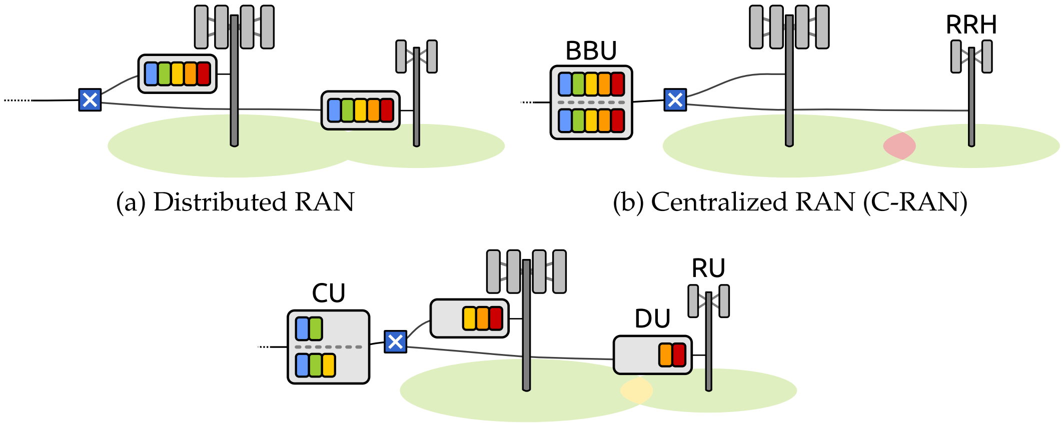
Interference cancellation techniques may be, in theory, the most effective interference mitigation techniques, specially for congested networks. Nonetheless, they require very low-latency and high-throughput communication between cooperating base stations, otherwise their performance in practice may be poor [NMH14]. As a result, centralized architectures are a requirement for the implementation of these techniques, such as those presented in the next section.

2.2.3. The Cloud-RAN architecture

From the previous description of interference techniques, it follows that advanced interference mitigation is only possible when the processing functions of all coordinating base stations can communicate fast with one another. This practically rules out distributed architectures, since the switching, transmission, and propagation delays introduced by medium or large networks may prevent the execution of dynamic coordination techniques. As a result, centralized architectures, in which the processing functions of the RAN are close together are necessary.

The Cloud-RAN architecture [Che+14a], or simply C-RAN, builds upon a simple idea: all the processing performed by all base stations in the RAN, with the only exception of analog RF processing, should be virtualized and moved to a single data center. Virtualization implies that former hardware processing is replaced by software functionality, which is abstracted from the underlying computational platform and therefore can be easily deployed, scaled, and modified. As a result, each base station is divided into two units: a remote radio head (RRH), in charge of filtering, amplifying, and transmitting radio signals; and a baseband unit (BBU) which is in charge of the

[1] The term *joint transmission* can also refer to transmitting the same signal from two different cells simultaneously. However, in this work we henceforth interpret *joint transmission* as previously stated.



(a) Distributed RAN

(b) Centralized RAN (C-RAN)

(c) Partially centralized RAN

Figure 2.1.: Depiction of a simple radio access network featuring a distributed architecture (a), a centralized architecture (b), and a partially centralized architecture implementing a CU/DU functional split (c).

remaining processing, including user- and control-plane signals from both high and low layers. A depiction of the C-RAN architecture is shown in Fig. 2.1b.

The C-RAN architecture features four main advantages to motivate their implementation in actual 5G RANs [Che+14a]:

- **Traffic adaptability and scalability.** It is well-known that mobile traffic exhibit time-varying patterns, owing to the movements and the activity changes of the users during the day or the week. For instance, the traffic load can be 20 to 50 times higher in the mornings and afternoons than at night in most areas [Xu+16]. In addition, the difference in traffic load among area types (residential, business, entertainment, etc.) may also be large depending on the time and day of the week [Xu+16]. If the operator wants to guarantee that the peak traffic load is satisfied the whole time and in all areas, it needs to deploy base stations with large processing capabilities, which will nonetheless be underutilized a large portion of the time. Conversely, if they opt for a more efficient use of the network resources, then they may fail to satisfy user demands in the peak hours. However, if the processing of all base stations of different area types is pooled into a single data center, the operator can take advantage of the multiplexing gain to ensure user satisfaction while requiring less network resources.
- **Energy and cost reductions.** Owing to the multiplexing gain of combining the processing of all base stations into a single location, the operator can also reduce energy consumption. The reason for this is twofold. First, the energy of cooling a single, big data center can be substantially lower than that of cooling individual base stations. Second, function virtualization allows to turn off computing servers if they are not needed. This energy reduction translates into cost savings, which are furthered by the simpler management and installation of centralized computing servers with respect to distributed base stations.

- **Simpler maintenance and upgrades.** Since the whole operation of all base stations is centralized, realizing maintenance tasks or upgrades is faster and less costly than on distributed architectures.
- **Support for advanced interference management.** As mentioned before, the C-RAN architecture allows for fast communication between processing functions, which enables the implementation of advanced interference mitigation techniques, such as the aforementioned coordinated scheduling and beamforming, joint transmission and reception, etc.

Nevertheless, the C-RAN architecture also has important drawbacks [Che+14a]:

- **High-capacity, low-latency transport network.** Since the only distributed function in the C-RAN architecture is the final RF processing, quantized samples of the base-band radio signals need to be sent through the transport network connecting the RRHs and the BBU. The capacity required for this is directly proportional to the channel bandwidth, the number of MIMO layers, the quantization levels, and the number of antenna sectors, hence it can be very large compared to the actual user throughput. For instance, it is estimated that for a 100 MHz channel, 32 antenna ports, and 16-bit quantization, the required capacity for a single antenna sector would be 157.3 Gb/s, even when the actual user data rate does not exceed 4 Gb/s [3GP20b]. If the channel bandwidth is 200 MHz and the number of antenna ports is 256, the required capacity would be 2.56 Tb/s per antenna sector [IT18]. In addition, the transport network must also guarantee very low latency and jitter between RRHs and BBU to enable not only flawless operation, but also the implementation of interference cancellation techniques. Estimated required values for one-way latency and jitter are $250\ \mu$s and $0.5\ \mu$s, respectively [3GP20b; Che+14a].
- **BBU coordination and clustering**. In order to exploit the benefits of function centralization, dedicated procedures and interfaces to coordinated the centralized functions at the BBU need to be designed. The deployment of the BBUs in a data center must be secure, since any security breach would affect the entire network, and the connections with the RRH must be reliable. Moreover, the location of the data center must be carefully selected so as to minimize power consumption and network latency and ensure user satisfaction. Finally, the BBU has to be prepared to cope with potentially high processing loads resulting from the combined operation of multiple base stations, which also depend on the number of antennas, resource blocks, modulations, and coding schemes [Nik15].
- **Virtualization platform**. The whole idea of function centralization relies on being able to replace hardware with software functions, that is, function virtualization. This requires using a software platform to instantiate, manage, and possibly migrate virtual functions over the physical computing servers. Owing to the high throughput, high reliability, and low latency that is required to operate each base station, choosing an appropriate virtualization platform can also be an important challenge.

Although the aforementioned advantages of C-RAN would allow an efficient and cost-effective management of a 5G RAN, its disadvantages, specially those concerning the high capacity of transport network between RRHs and BBU, render it infeasible in many realistic scenarios. Indeed, current fiber-optical deployments of transport networks connecting remote locations with a data center often feature links not exceeding 2 Tb/s in capacity [GS+18a], which severely limit the amount of base stations that can be centralized. In the light of this, the current 3GPP 5G RAN specifications recommend instead a *partially centralized architecture* featuring a *functional split* in the processing function chain of each base station.

2.2.4. Functional Split in 5G networks

If full centralization is not possible, the next best approach is to centralize as many network functions as possible. Namely, for each base station (gNodeB) a subset of functions is deployed at the centralized unit (CU), whereas the remaining functions remain at the distributed unit (DU), which is located close to the remote unit (RU) containing the radio equipment. The CU can be a commodity data center, which hosts the centralized functions of multiple gNodeBs. Moreover, the creation of separate centralized and distributed units leads to the definition of a fronthaul[2] network, which is in charge of carrying user data and all the signaling traffic between centralized and distributed functions.

In theory, we could arbitrarily choose how functions are defined within a base station. In practice, however, functions are usually defined based on the 5G protocol stack [Döt+13], which clearly states a sequence of functional layers. From top to bottom, these layers are:

- **Radio Resource Control (RRC):** This layer terminates the control-plane messages between the base station and the UEs.
- **Service Data Adaptation Protocol (SDAP):** The SDAP layer is in charge of the mapping between a quality-of-service (QoS) flow and data radio bearers.
- **Packet Data Convergence Protocol (PDCP):** This layer performs header compression and implements ciphering and integrity protection.
- **Radio Link Control (RLC):** The main goal of this layer is to ensure reliable and in-sequence delivery of data streams in downlink and uplink.
- **Medium Access Control (MAC):** This layer is in charge of scheduling transmissions, prioritizing and multiplexing logical channels, and controlling the hybrid automatic repeat request (HARQ) function.
- **Physical layer (PHY):** The physical layer comprises the low-level operations

[2] Other authors use the term *fronthaul network* to refer to the network connecting DUs and RUs, whereas that connecting DUs and CU is called *midhaul network*. Since we do not make a functional distinction between DUs and RUs, we prefer the term *fronthaul network* as opposed to the *backhaul network* connecting CUs with the core network.

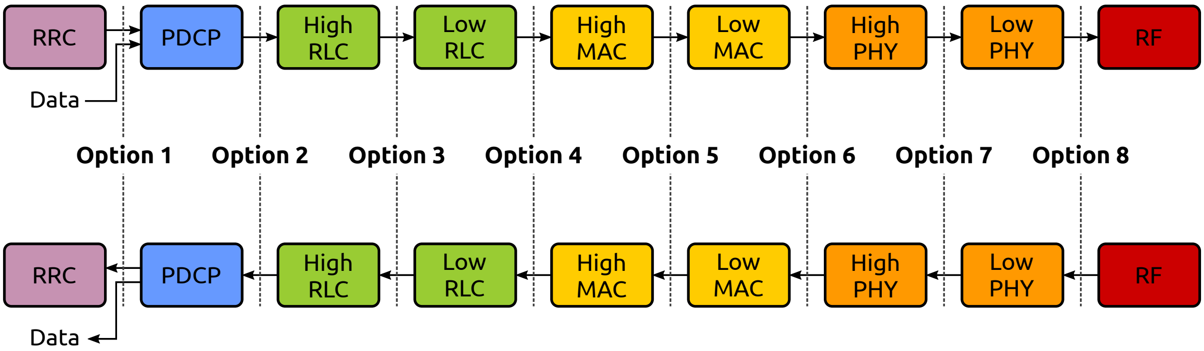


Figure 2.2.: Depiction of the possible functional split options in a 5G network. The upper processing chain belongs to the downlinks, whereas the lower processing chain represents the uplink. The SDAP layer is not depicted, but merged with the PDCP layer.

such as coding, modulation, IFFT/FFT, etc.

Since these functional layers are arranged as a function chain, when deciding which functions to centralize it is sensible to simply perform a "cut" and divide the chain into two subchains so that the interfaces between the functions are not affected. Thus, the subchain containing the topmost layers is centralized, whereas the other subchain is distributed. This "cut" is known as the 5G *functional split*. The resulting division of functions is known as the *centralization level*, which is said to be higher the more functions are centralized.

Deciding on the most appropriate functional split is not a simple task. The higher the centralization level, the more opportunities there are regarding interference mitigation and the lower the operating and capital cost, but also the higher the required capacity at the fronthaul. According to [3GP17], we identify eight main options to implement a functional split, which are described in the following. A depiction of these options is shown in Fig. 2.2.

- **Option 1 (RRC–PDCP or RRC–SDAP split).** With this split, only the RRC function, belonging to the control plane of the access stratum, is centralized, whereas the whole user plane is distributed. This allows for a clear control- and user-plane separation, which may be exploited for edge computing applications where centralized control is beneficial and low latency is required in the user plane. Nonetheless, the very limited centralization level of this split does not allow for advanced interference mitigation techniques nor substantial cost reductions.

- **Option 2 (PDCP–RLC split).** In this split option, both the control-plane RRC function and the user-plane PDCP function are centralized, whereas the remaining functions are deployed in the distributed unit. The main benefit of this split with respect to the previous one is that the user traffic aggregation is done in the centralized unit, thus enabling easier management of the traffic load. From a standardization perspective, the implementation of this split is straightforward, as it is a direct continuation of the interfaces defined for LTE Dual Connectivity [3GP21i]. In fact, this split is already included in the specified architecture for the next-generation RAN (NG-RAN) as the default option for the functional split between CU and DU [3GP21a]. Although it features a higher level of centralization than the RRC-PDCP split, it is still not enough to enable advanced interference mitigation techniques.

- **Option 3 (Intra RLC split).** This split option can be implemented in two different ways, depending on how the high and low sublayers of RLC are defined. Namely, the low RLC sublayer can either refer to the segmentation function or the downlink RLC transmission entities, whereas the high RLC sublayer comprises the remaining RLC functions (automatic repeat request, uplink functions, etc.). In either case, this split benefits from higher flow control and leverages better resource pooling than previous splits without being as sensitive to latency constraints as more centralized options. However, a new interface would need to be defined so as to accommodate all the information exchange between the two sublayers.

- **Option 4 (RLC–MAC split).** In this option, the whole RLC layer is centralized alongside the RRC and PDCP layers, whereas the MAC, PHY, and RF layers are distributed. Although the interface between the RLC and MAC layers is already precisely defined, thus allowing for simple implementation, there are no special benefits of using the RLC–MAC split over the Intra RLC split, apart from a small multiplexing gain due to the higher centralization level.

- **Option 5 (Intra MAC split).** The MAC layer can be divided into a high MAC layer, comprising mainly the scheduler function, and a low MAC layer, which includes the hybrid automatic repeat request (HARQ) function, random access control, and channel measurements, among others. If the high MAC layer is deployed at the CU, the RAN can benefit from enhanced interference coordination and coordinated scheduling or beamforming among all base stations, leading to a potential reduction in the interference levels. Nonetheless, the interface between high and low MAC layers may be complex to define and implement.

- **Option 6 (MAC–PHY split).** In this option, the whole MAC layer is centralized, along with the RRC, PDCP, and RLC layers, whereas only the PHY and RF layers are distributed. Since the MAC layer is centralized, it also enables the use of coordinated scheduling and beamforming, and even simple forms of coordinated transmission. This comes at the price of increased traffic at the fronthaul network [MAGVK19b], since the communication between MAC and PHY layers involves the exchange of a fair amount of signaling commands. The complexity of the interface between MAC and PHY layers is, nonetheless, lower than in the previous option. Indeed, there already exist proposed descriptions of it, such as the 5G nFAPI interface [SCF21].

- **Option 7 (Intra PHY split).** This split also relies on dividing the physical layer into two sublayers, which can be performed in multiple ways. For example, the IFFT/FFT and cyclic-prefix removal/addition could be distributed, whereas the encoder and modulator are centralized. In general, since the PHY layer is computationally-intensive and deals with low-level signals, the main advantage of implementing an Intra PHY split is to leverage cost reductions by resource pooling and enable advanced interference-mitigation techniques. The obvious drawback is that the more functions are centralized, the more capacity is required at the fronthaul network.

- **Option 8 (PHY–RF split or C-RAN).** This last option is equivalent to the original idea of C-RAN: all functions except those dealing with analog RF processing are hosted at the centralized unit. Although, as mentioned before, implementing C-RAN may not be feasible in current 5G networks, there are already interface specifications between the PHY and RF layers that can be used as a reference for future implementations, such as the eCPRI interface [EAN19]. In this section, we describe the considered network and present the concepts required to formulate the functional split selection problem (FSSP).

From the above description of the layers, it is clear that the more centralized functions, the more advanced the interference mitigation techniques that can be applied, but also the higher the capacity required on the fronthaul network. In other words, if the network operator wants to maximize the throughput of the users, it needs to centralize as many functions as possible, but this is constrained by the capacity of the fronthaul network. This trade-off defines the objective and constraints of the *functional split selection problem* (FSSP).

2.3. Functional split selection problem

In this section, we introduce the basic notation and modeling assumptions that we use in the formulation of the FSSP. The purpose of formulating the FSSP is to find the optimal functional split option for every gNodeB in the network such that the overall user experience and/or network revenue is maximized. Solving the FSSP, however, entails a series of challenges, not only due to the limitations imposed by the fronthaul network, but also because of the everchanging nature of mobile networks.

2.3.1. System modeling

General network description

We consider a 5G RAN consisting of G gNBs, including macro and small cells. The operation of each cell is divided into a chain of software functions, which is split into a DU and a CU. The DUs are deployed close to remote units (RU) containing the antennas and radio equipment, whereas the CUs are all located in a single data center, resulting in G different locations for the DUs and a single CU location. Communi-

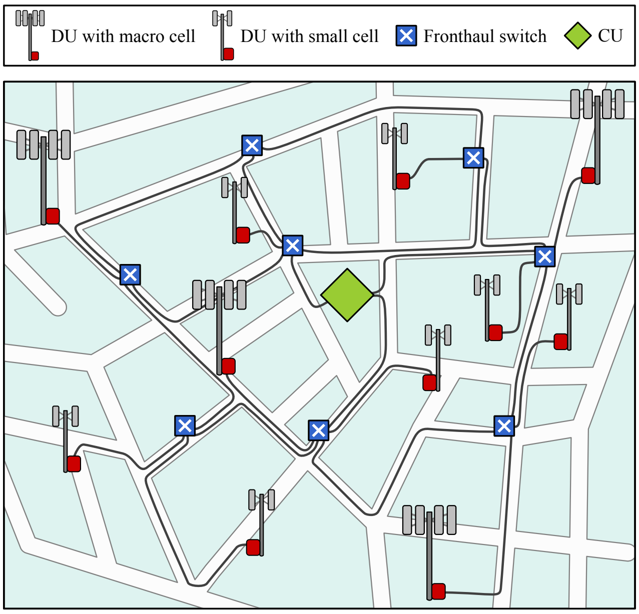


Figure 2.3.: Depiction of an example network with $G = 11$ gNBs (including macro and small cells) and eight fronthaul switches.

cation between CUs and DUs is accomplished by means of a packet-switched fronthaul network [GS+18a], which consists of a set of links and a set of layer-2 or layer-3 switches. We assume that this network is dynamically configurable, which implies that the operator is able to reroute and divide the end-to-end flows during runtime, for instance by utilizing software-defined networking (SDN) techniques.

We model the fronthaul network as a directed graph $\mathcal{D} = \langle \mathbb{N}, \mathbb{E} \rangle$, where $\mathbb{N}$ is the set of network nodes (including switches, DUs and CU) and $\mathbb{E}$ is the set of network links. The total number of nodes and edges is denoted by N and E, respectively. The node corresponding to the CU is referred to as n_0, whereas a DU node is represented by n_g, where $g \in \mathbb{G}$ and $\mathbb{G} \triangleq \{1, ..., G\}$. We show a simple example of a RAN featuring $G = 11$ gNBs and eight fronthaul switches in Fig. 2.3.

The number of *simultaneously active* user equipments (UEs) inside the coverage area of all cells is denoted by U. Each UE $u \in \mathbb{U}$, where $\mathbb{U} \triangleq \{1, ..., U\}$, is connected to a serving gNodeB, whose index is referred to as h_u. We consider a densely-deployed RAN in which all gNodeBs operate in the same frequency bands. Although a more complicated frequency reuse strategy is not precluded, we intend to highlight the feasibility and performance of a network featuring dynamically-enabled interference mitigation techniques. We mainly address downlink communication throughout this work, but an extension of the analysis to include the uplink is straightforward.

Modeling of functional splits

The operation of each gNB can be represented as a software function chain, where functions are usually defined with each of the layers in the RAN protocol stack. These

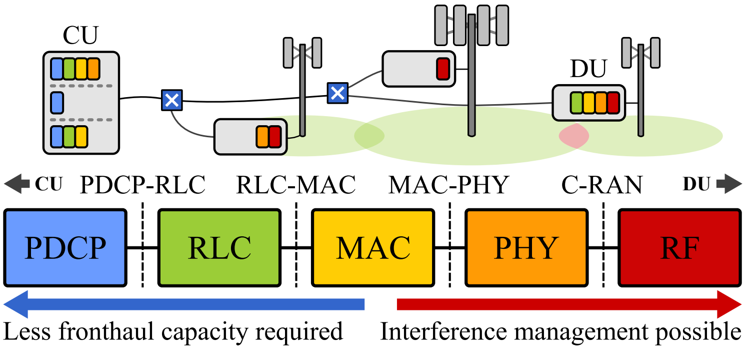

CU
DU
CU
PDCP-RLC
RLC-MAC
MAC-PHY
C-RAN
DU
PDCP
RLC
MAC
PHY
RF
Less fronthaul capacity required
Interference management possible

Figure 2.4.: Representation of the considered network, example gNB functions, and scheme functional splits.

functions can be hosted at either the CU or DU, as long as the overall chain structure is kept. The number of possible functional split options, also referred to as *centralization levels*, is denoted by M. For example, in Fig. 2.4 the network operator can choose between $M = 4$ centralization levels (PDCP-RLC, RLC-MAC, MAC-PHY, and C-RAN) corresponding to five different functions, which are also the layers in the gNB protocol stack. The centralization level implemented by a gNodeB g at any given time is denoted by a_g, where $a_g \in \mathbb{M}$ and $\mathbb{M} \triangleq \{0, ..., M-1\}$. For convenience, we say that $a_g = 0$ represents the lowest centralization level, i. e., the functional split option with the lowest amount of centralized functions. Conversely, $a_g = M - 1$ denotes the highest centralization level, i. e., the functional split with the most centralized functions. For instance, in Fig. 2.4, $a_g = 0$ corresponds to the PDCP-RLC split, whereas $a_g = 3$ corresponds to the C-RAN split. We define the *centralization vector* as the vector containing the centralization levels of all gNBs at any given time, and denote it as $\mathbf{a} \triangleq [a_1 \ \cdots \ a_G]$.

The motivation for choosing a one centralization level over another is twofold. On the one hand, low centralization levels require less fronthaul capacity than high centralization levels, thus they can be implemented more easily without congesting the fronthaul network. On the other hand, high centralization levels enable faster and simpler communication between gNB functions, which can be exploited to enhance transmission and reception coordination and thus to reduce their mutual interference. However, this comes at the price of increased requirements regarding fronthaul capacity [3GP17]. This is trade-off is also depicted in Fig. 2.4.

Modeling of the fronthaul network

We model the capacity required by centralization level a_g implemented by gNB g by means of the function $\nu(a_g)$. The output of this function ranges from a few Gb/s for low centralization levels (such as the PDCP-RLC split) to hundreds of Gb/s for high centralization levels, as previous research has shown [Döt+13; MAGVK19b; 3GP17]. For simplicity, we consider that all gNBs feature the same maximum user data rate, therefore $\nu(a_g)$ does not depend directly on g.

We define ϑ_e as the capacity of each fronthaul link $e \in \mathbb{E}$. For a downlink flow generated by gNB g between its DU and CU, let f_e^g be the fraction of this flow that conveyed by link e. We define the flows-per-link vector as $\mathbf{f}^g \triangleq [f_1^g \ \cdots \ f_E^g] \ \forall g \in \mathbb{G}$ and the flows vector $\mathbf{f} \triangleq \left[\mathbf{f}^1 \ \cdots \ \mathbf{f}^G\right]$.

Modeling of the network state

We often consider the flows vector $\mathbf{f}$ and the centralization vector $\mathbf{a}$ together using the following notation:

$$\mathbf{s} \triangleq \langle \mathbf{a}, \mathbf{f} \rangle \,. \tag{2.1}$$

We refer to s as the *state vector* of the network, since it includes all variables that the network operator is able to change to deal with changes in the environment. In addition, we denote the optimal state, centralization, and flow vectors by $\mathbf{s}^*$, $\mathbf{a}^*$, and $\mathbf{f}^*$, respectively, such that:

$$\mathbf{s}^* \triangleq \langle \mathbf{a}^*, \mathbf{f}^* \rangle \,. \tag{2.2}$$

Modeling of the interference mitigation

The activity of neighboring cells causes downlink interference on nearby UEs, hence reducing user data rates. Proposed techniques to mitigate this interference (such as coordinated scheduling, coordinated beamforming, joint transmission, etc.) require some level of coordination between the involved gNB functions. As result, each interference-mitigation technique requires a minimum centralization level to be applied, depending on which functions need to be centralized for the technique to operate properly. For instance, coordinated scheduling requires the centralization of the MAC layer [Nar+18], whereas joint transmission also requires the centralization of the physical layer [Zha+17].

Based on the analysis shown in [PM17], we model the effectiveness of an interference mitigation technique between two gNBs as a constant factor multiplying their average received interference power. We relate each centralization level a with the interference cancellation factor of the most effective interference mitigation technique that can be applied by means of function $И(a)$. The codomain of function $И(a)$ is $[0, 1]$, that is, it ranges from 0 (full interference cancellation) to 1 (no interference cancellation).

Since centralization levels are defined incrementally along the function chain, the higher the centralization level a, the lower its related interference-cancellation factor $И(a)$. Moreover, an interference-mitigation technique can only be used by two gNBs if both of them are operating at the required centralization level or higher. As a consequence, the resulting interference-cancellation factor between gNBs g and g' is $И(\min(a_g, a_{g'}))$, that is, the gNB with the lowest centralization level is the bottleneck to interference mitigation. Knowing this fact, we can compute the expected total interference power I_u experienced by UE u from all gNBs as:

$$I_u(\mathbf{a}) = \sum_{g=1}^{G} i_{u,g} \cdot \text{И}(\min(a_{h_u}, a_g)), \tag{2.3}$$

where $i_{u,g}$ is the interference power received by UE u from gNB g and $i_{u,h_u} \triangleq 0$, as the UE is not interfered by its serving gNB. Note that I_u is a function of the centralization vector $\mathbf{a}$, hence selecting the right values of $\mathbf{a}$ can be used to reduce overall interference. For all UEs in $\mathbb{U}$, we define the interference vector $\mathbf{I}(\mathbf{a}) \triangleq [I_1(\mathbf{a}) \ \cdots \ I_U(\mathbf{a})]$.

2.3.2. Static FSSP formulation

The objective of the network operator when selecting the functional split is to maximize user data rates, minimize operating costs, or optimize a combination of both. We denote a generic objective function as $\Gamma(\mathbf{a}, \mathbf{I}(\mathbf{a}), \mathbf{f})$, where $\mathbf{a}$ is the centralization vector, $\mathbf{I}(\mathbf{a})$ is the interference vector, and $\mathbf{f}$ is the flow vector. Without loss of generality, we assume that our goal is to maximize the value of $\Gamma(\mathbf{a}, \mathbf{I}(\mathbf{a}), \mathbf{f})$, subject to the fronthaul network constraints. Consequently, we can formulate the generic *functional split selection problem* (FSSP) as:

$$\max_{\mathbf{s}} \quad \Gamma(\mathbf{a}, \mathbf{I}(\mathbf{a}), \mathbf{f}), \tag{P1a}$$

subject to

$$\sum_{e \in \mathbb{E}^+(n)} f_e^g - \sum_{e \in \mathbb{E}^-(n)} f_e^g = \begin{cases} 0 & \forall n \in \mathbb{N} \setminus \{n_0, n_g\} \\ \nu(a_g) & \text{for } n = n_0 \\ -\nu(a_g) & \text{for } n = n_g \end{cases} \qquad \forall g \in \mathbb{G}, \tag{P1b}$$

$$\sum_{g=1}^{G} f_e^g \leq \vartheta_e \qquad \forall e \in \mathbb{E}, \tag{P1c}$$

$$f_e^g \geq 0 \qquad \forall e \in \mathbb{E},\ \forall g \in \mathbb{G}, \tag{P1d}$$

$$\mathbf{s} \in \mathbb{M}^G. \tag{P1e}$$

where $\mathbb{E}^-(n)$ denotes the set of edges entering node n and $\mathbb{E}^+(n)$ is the set of edges leaving node n. Constraint (P1b) is the *flow conservation* constraint, since it guarantees that flow is created at the CU, consumed at the DUs, and conserved in the intermediate nodes. Constraint (P1c) is the *link capacity* constraint, as it ensures that the link capacities are never exceeded. Solving (P1) allows us to obtain the optimal state vector $\mathbf{s}^*$, and thus the optimal centralization vector $\mathbf{a}^*$ and flow vector $\mathbf{f}^*$, for a given objective function $\Gamma(\mathbf{a}, \mathbf{I}(\mathbf{a}), \mathbf{f})$.

Since the objective function $\Gamma(\mathbf{a}, \mathbf{I}(\mathbf{a}), \mathbf{f})$ depends on the interference vector $\mathbf{I}(\mathbf{a})$, every time this vector changes the FSSP (P1) will also change. This means that solving (P1) yields an optimal vector state that is only instantaneously valid. However, finding the optimal state for a given instant is not sufficient to operate a 5G RAN efficiently. This is because changes in the positions and activity of the UEs, as well as variations in the wireless channel, transform the problem over time, so that an optimal solution at the present time may be highly suboptimal in the future. In the next section, we introduce the concepts required to tackle the dynamic FSSP problem.

2.4. Solving the FSSP dynamically

In order to operate profitably, the RAN has to be able to adapt to changes in the environment, such as changes in the interference vector $\mathbf{I}(\mathbf{a})$. This can be realized by leveraging recent advances in network softwarization, which allow for fast and cost-efficient reconfiguration of communication networks. In fact, these new possibilities have spurred the analysis and modeling of the ability of communication networks to adapt to environmental changes, which is known as *network flexibility*.

In this section, we introduce the sofwarization technologies that enable dynamic changes in communication networks, and thus can be applied to a 5G RAN featuring a dynamically-selected functional split. Then, we discuss the new entities that are needed in such a RAN and their implications. Finally, we present the formulation for a dynamic FSSP.

2.4.1. Flexibility in softwarized communication networks

Network softwarization technologies

In the past, changing the network configuration dynamically was difficult to realize, and it was instead preferred to design the RAN to optimize for the average load or to support the peak demand [Li11]. Nowadays, recent advances in network softwarization allow us to replace stiff hardware equipment, which is hard to operate dynamically, with software functions. These functions can be reconfigured, relocated, and scaled with ease, which enables prompt and cost-efficient adaptation to changes in the network environment and demands. Namely, there are three main novel technologies that offer network softwarization:

- **Software defined networking (SDN)** can be described as designing, deploying, and operating communication networks such that the forwarding rules in switches and routers are programmed on a central server [Sta15]. Thus, switching devices are not configured individually, but their operation is defined by their interaction with a central server, which can be reconfigured at will. In practice, this results in the separation of the *control plane*, consisting in the configuration of forwarding decisions, from the *user plane*, dealing with user packet forwarding. The main benefits of SDN are better adaptability, automation, mobility, maintainability, scaling, and security than traditional, distributed approaches for networks facing ever-increasing demands, such as 5G radio access networks [Ope14; BK16].

- **Network functions virtualization (NFV)** is defined as the replacement of network hardware functions by software functions with the intention of running them in virtual environments, such as containers or virtual machines [Sta15]. Often in combination with SDN, it can be used to substitute proprietary hardware platforms by commodity platforms, increase the efficiency of function management, allow for function relocation, and enhance network scaling and maintainability [Sta15].
- **Network virtualization (NV):** This technology consists in creating logically-isolated network slices over a shared physical network, with the goal of enabling the simultaneous use of virtual networks by multiple tenants. Physical network resources can be aggregated or split into single or multiple virtual resources, respectively, so that the virtual network as seen by the tenant is abstracted from the underlying network [IT12]. The main advantages of NV are increased *flexibility*, reduced operational and capital costs, and enhanced scalability [Sta15].

We conclude from the definitions of SDN, NFV, and NV that they can be applied to make the network more adaptable to changes in the demands, that is, more *flexible*. Indeed, these technologies motivate a novel interest in modeling network flexibility, which is presented in the following.

Definition of network flexibility

In [He+19; Bab+20], network flexibility is defined as the *timely and cost-efficient support of changes in the network requirements*. The concept of *network requirements* refers to any environmental or user-related demand that affects network operation or profitability but cannot be directly changed by the network operator. As a result, every time there is a change in the network requirements, which is henceforth referred to as a *request*, the current network configuration may be suboptimal, and thus adaptation may be required.

There are two different manners in which a softwarized network may satisfy requests. First, the network could be designed and deployed with enough resource overhead so that it accommodates requests without reconfiguring itself. In our case, this would imply an over-dimensioned 5G RAN whose fronthaul network supports full centralization. The main drawback of this approach is, however, the potentially high cost of deploying and operating such a network. Second, the network may actively adapt its topology, flows, functions, or resources to match the new network requirements. This may reduce the deployment and operating costs with respect to the first option, although timely adaptation is now of utmost importance. In this work, we investigate whether this second approach is applicable to 5G RANs and whether it is more adequate than non-adaptive implementations.

Table 2.1.: Technical Concepts and their support of flexibility in networks. (✓: main target). This table is extracted from [He+19].

Category	Aspect	SDN	NFV	NV
Adapt configuration	Flow Configuration: flow steering	✓	-	-
	Function Configuration: function programming	-	✓	-
	Parameter Configuration: change function parameters	-	✓	✓
Locate functions	Function Placement: distribution, placement, chaining	-	✓	✓
Scale	Resource and Function Scaling: processing and storage capacity, number of functions	✓	✓	✓
	Topology Adaptation: (virtual) network adaptation	-	-	✓

Flexibility categories and aspects

According to [He+19], a softwarized network can satisfy a request in three different manners: (i) by reconfiguring its current flows, functions, and parameters; (ii) by relocating software functions; and (iii) by scaling resource and functions or modifying its (virtual) topology. These are known as *flexibility categories*, since they can be used to classify a flexible network. Within each category, we identify one or more *flexibility aspects*, which narrow down the specific network feature that is changed during an adaptation. We can use flexibility categories and aspects to identify the softwarization technology that is required to implement a flexible network. This is because we can relate each flexibility aspect with the softwarization technology that enables it, as it is shown in Table 2.1

In order to implement a dynamic functional split, the 5G RAN needs to feature at least three flexibility aspects. First, the RAN needs to be able to move virtual functions in and out CU and the DUs so as to operate in the selected functional split. This can be identified with the *function placement* aspect. Second, the RAN has to scale its computing resources according to the instantaneous load and also the current functional split. This is the *resource and function scaling* aspect. Finally, the flows within the fronthaul network may be rerouted every time the functional split changes, since the capacity required between CU and DUs is different for each functional split. Thus, the *flow configuration* aspect is also required. As a result, we see from Table 2.1 that a 5G RAN featuring a dynamically-adapting functional split requires an SDN fronthaul network

2.4.2. Flexible functional split platform

Owing to both the advantages and disadvantages of function centralization in the RAN, it is clear that being able to change the centralization level in a dynamic manner, according to the instantaneous state of the network, may be beneficial for increasing network performance and reducing operating costs. Nevertheless, implementing a dynamically-configurable functional split is not a trivial task. Indeed, we need to define two new network entities: (i) a *decision-making entity*, which is in charge of monitoring the network and deciding when to change the centralization level, and (ii) a *migration platform*, which realizes changes in the functional split without stopping the operation of the network.

Decision-making entity

The decision-making entity is in charge of implementing a strategy to tackle the dynamic optimization problem of selecting the best functional split. There are abundant previous works regarding dynamic optimization techniques, such as those provided by the field of control theory. This field comprises the study and control of dynamic systems, whose evolution is commonly modeled by differential equations. As classical demonstration of this type of dynamic control problem is the control of an inverted pendulum [FYK91], where the goal is to keep a pendulum in the upright position by controlling a moving base. The theoretical evolution of the movement and position of the pendulum is modeled by well-known equations, which are combined with measurements of the actions and resulting errors to swing up the pendulum. This approach is, nonetheless, difficult to apply to our problem, since the evolution of a mobile network depends on the random roaming of hundreds or even thousands of UEs, which renders a theoretical description of its evolution intractable. However, we may still apply discrete dynamic programming to our problem [Bel66], as long as the future evolution of the system is known or can be accurately predicted. This possibility is discussed in Chapter 6. In any case, predicting the behavior of a large number of UEs is a very challenging task. Furthermore, if the employed dynamic optimization technique relies on mobility predictions, it would be arguable whether they are accurate enough for any conclusion to hold in real networks. Consequently, in this work we do not assume that the network operator has access to predictions about the behavior of the UEs beyond simple statistics, and thus we use the results provided by dynamic programming techniques only as references for other strategies.

An alternative approach is to use evolutionary dynamic optimization techniques [NYB12]. Evolutionary algorithms, such as the genetic algorithm, are often adequate to tackle continuously changing problems, since they can be configured to follow changes in the environment instead of converging. Moreover, previous work uses a genetic algorithm to successfully tackle a simplified formulation of the FSSP [MAK19a]. In this work, however, we address a more complete FSSP formulation that is not as suitable for evolutionary algorithms, due to the stringent constraints that define our solution set. The reason is that the continuous-valued flows vector $\mathbf{f}$ and the discretevalued centralization vector a have to be always in agreement and satisfying the network constraints, rendering a random exploration of the solution set inefficient.

As a consequence, in this work we use simple, yet effective approach for the decision-making entity. First, the decision-making entity periodically monitors the network and solves the static FSSP for each new state, so as to obtain an updated network configuration that would maximize the revenue of the network. Note that, since the FSSP is NP-Hard [MAJK21], we cannot tell whether the network still features an optimal configuration after a change in the problem unless we run the optimization problem again. After a new optimal configuration is known, the decision-making entity chooses whether to perform an adaptation and move to the optimal configuration or stay in the current state. This decision is taken based on a set of rules that are influenced by the cost of operating in the current state, the cost of changing to the optimal state, and the cost of operating at the optimal state. We refer to this set of rules as the *adaptation strategy*, which are discussed in Sec. 6.5. The design and selection of the adaptation strategies are supported by the application of the so-called *cost-of-flexibility framework* [Bab+20; MA+21], which is used to separately model all cost components, decide about the optimal monitoring period, and compare different adaptation strategies.

Migration platform

Regarding the migration platform, the network operator needs to be able to change the functional split without stopping the network operation. There are already NFV platforms in the state of the art featuring such uninterrupted operation. In Chapter 5, we present a framework that can change the functional split of a single base station in less than 20 ms with no packet losses. Alternatively, the operator may employ virtual machines or containers to host the RAN functions, since off-the-shelf frameworks also exist to dynamically migrate between physical hosts [GA18].

2.4.3. Dynamic FSSP formulation

Assuming that the 5G RAN implements an SDN- and NFV-based platform that is able to dynamically change the functional split and flow configuration, we still need to decide when to trigger this change, and which state to move to. From the perspective of a mobile network operator, the objective of performing this dynamic adaptation is to maximize the network revenue, or, equivalently, to minimize the total cost. Clearly, this total cost is a function of the performance experienced by the users, the operating cost of running the network on a fixed state, and the cost of changing the functional split. A thorough analysis of all these components is addressed in the next chapter.

Let us define τ as the time-ordered discrete index representing the time instant corresponding to the end of a monitoring interval. We can use this index to make the time dependence explicit on the state, centralization, and flow vectors as $\mathbf{s}(\tau)$, $\mathbf{a}(\tau)$, $\mathbf{f}(\tau)$, respectively. In addition, we define $C(\tau)$ as the cost associated with changing the state at time τ, if applicable, and $K(\mathbf{s})$ as the instantaneous cost associated with operating on state $\mathbf{s}$. Note that $K(\mathbf{s})$ can be defined similarly to $\Gamma(\mathbf{a}, \mathbf{I}(\mathbf{a}), \mathbf{f})$ if the latter is already a cost function. Finally, the objective of the dynamic functional split selection problem is to find the optimal sequence of states $\mathbf{s}(\tau)$ so that the total RAN cost is maximized without compromising the fronthaul network:

$$\min_{\mathbf{s}(\tau)} \quad \lim_{\tau^{\max}\to\infty} \sum_{\tau=0}^{\tau^{\max}} \Big(K(\mathbf{s}(\tau)) - C(\tau) \Big), \tag{P3a}$$

subject to

$$\sum_{e\in\mathbb{E}^{+}(n)} f_e^g(\tau) \; - \sum_{e\in\mathbb{E}^{-}(n)} f_e^g(\tau) = \begin{cases} 0 & \forall n \in \mathbb{N} \setminus \{n_0, n_g\} \\ \nu(a_g(\tau)) & \text{for } n = n_0 \\ -\nu(a_g(\tau)) & \text{for } n = n_g \end{cases} \qquad \forall g \in \mathbb{G}, \tag{P3b}$$

$$\sum_{g=1}^{G} f_e^g(\tau) \le \vartheta_e \qquad \forall e \in \mathbb{E}, \tag{P3c}$$

$$f_e^g(\tau) \ge 0 \qquad \forall e \in \mathbb{E},\ \forall g \in \mathbb{G}, \tag{P3d}$$

$$\mathbf{s}(\tau) \in \mathbb{M}^G. \tag{P3e}$$

Compared to the static FSSP (P1), (P3) may be considerably more challenging to address, especially if the future evolution of the time-varying parameters cannot be accurately estimated. However, owing to its similarities, the same techniques that are applied to solve (P1) can be used in combination with dynamic optimization approaches for tackling (P3). This is fully addressed in Chapter 6.

2.5. Summary

In Chapter 2, we introduce the problems of optimally and dynamically selecting the functional split for a 5G radio access network. We first discuss the objectives and requirements of 5G networks and delve into the issue of managing inter-cell interference, which may hinder the viability of cell densification. We explore the different RAN architectures that are proposed to cope with this issue and discuss their details, advantages, and disadvantages. We conclude that, whereas a fully centralized architecture is hardly feasible for current networks, partially centralized architectures can be leveraged to both improve user experience (by reducing interference) and limit operating costs.

Consequently, we propose a model for a 5G RAN featuring a configurable functional split, including mathematical descriptions of the overall state, the underlying fronthaul network, and the interference-mitigation capabilities. We then use this model to formulate the static functional split selection problem (FSSP), which aims at finding the optimal functional splits and flow configurations that maximize a generic objective function. In following chapters, this generic function is replaced with concrete definitions reflecting the user data rates, operating cost, and network revenue.

Finally, owing to the variable nature of mobile networks, we discuss the challenges of solving the FSSP dynamically. We present recent advances in the field of network softwarization and flexibility, which may help us to address the dynamic FSSP. After discussing possible options for implementation options, we conclude by proposing a generic formulation of the dynamic FSSP, which addressed in later chapters.

3. Modeling the Cost of Flexible Communication Networks

3.1. Introduction

3.1.1. Motivation, scope, and challenges

In the last chapter, we introduce the problem of dynamically selecting the optimal functional split for a 5G RAN. This problem features two main layers of complexity from a theoretical perspective. On the one hand, we need an efficient algorithm to find an optimal solution for an instantaneous situation, which is not a trivial task. On the other hand, we have to devise a strategy to dynamically decide when and how to reconfigure our network. In other words, our network has to be adaptable enough to support live reconfiguration.

It is indeed a common trend in the design of 5G radio access networks to aim at swift adaptation to traffic patterns and user mobility, in order to achieve high resource efficiency [NTT20; Tal+20]. Nonetheless, mobile RANs are not the only example of communication networks requiring higher flexibility and adaptability. In fact, this field has attracted a substantial amount of research effort in the recent past coming from different angles. This is mainly due to the ever-increasing number of connected devices as well as the emergence of new applications and use cases that many types of networks face. One example is the emergence of the Internet of Things (IoT), where the number of connected autonomous devices is estimated to grow from 12 billion in 2015 to hundreds of billions by 2025 [Cis17]. These IoT-based networks will cover a wide range of use cases, such as smart-city sensors, self-driving cars and platooning, industrial automation, etc [Che+14b]. Furthermore, increasing agility and programmability is also a major concern in data center networks, so as to efficiently deal with new applications [Cas18].

In order to increase their adaptability, virtually every type of communication networks can benefit from network softwarization techniques, namely, from SDN, NFV, and NV [Afo+18]. Exploiting the advantages of network softwarization is, nonetheless, still a greenfield in many regards. Indeed, owing to the novelty of the ability to quickly adapt the network to changes in the demands, even dedicated research has emerged to explore this ability. For example, in [Kel+18] the authors define a *network flexibility measure* with the intention of providing quantitative assessment of possible improvements in the adaptability of the network. The main purpose of this new measure is to evaluate how costly and quickly a softwarized network can adapt to changes in the demand. As a result, network designers and operators can use it to compare the performance and cost of adaptive solutions.

In principle, network flexibility is a desirable feature that network operators would like to maximize. It not only reflects how good the network responds to changes in the demand, but it can also be used to advertise the network over competitors and attract potential users. Moreover, network flexibility may also indicate how likely it is that the network supports changes in future demand trends. Nonetheless, increasing network flexibility also comes at a price. As many other performance metrics, flexibility is affected by a trade-off: the higher its value, the higher the revenue, but also the higher the associated costs. The reason for this is twofold. On the one hand, increasing network flexibility often implies investing in better equipment or consume more energy resources. On the other hand, high flexibility is correlated with better performance: faster demand satisfaction, more supported users, less relevance of link failures, etc. These conflicting trends motivate the formulation of classical network design problems: maximizing performance for a fixed cost, minimizing cost for fixed performance, or find out the best cost-performance combination on the Pareto frontier.

As with most engineering systems, the standard approach to model the cost of a communication network is to divide the total cost of ownership (TCO) into capital expenses (CAPEX) and operating expenses (OPEX) [Bet+17]. If the environment within which the network operates varies slowly, then the OPEX can be estimated from the expected operating states and their corresponding revenues and resource consumption rates [Hue+08; Zan97]. However, modeling the OPEX of a network dealing with frequent changes in its environment is considerably more challenging. On the one hand, estimating the expected operating state may be difficult, since it is the result of a sequence of changing demands. On the other hand, the adaptation itself may incur in additional cost, which has to be then included into the OPEX estimation.

Consequently, in order to characterize a fast-adapting communication network, we require a more powerful model for its operating cost, such that adaptations and lack of adaptations are taken into account. The intention of this model is to accurately predict the total operating cost of a network facing changes in the demands whose durations are in the same order of magnitude as the time required for adapting. That is, the model should be able to estimate the cost of operating a network in "race conditions" between environmental changes and adaptations. As a result, this cost model can be utilized to select the optimal adaptation strategy, test alternatives for profitability in future scenarios, or identify limiting factors.

3.1.2. Key contributions

The main content and contributions presented in this chapter are based on [Bab+20] and specially [MA+21]. In summary, these contributions are as follows:

1. We present a cost model for flexible, dynamically-adaptive communication networks that is based on a probability theory frameworks and designed to capture

the internal trade-offs affecting cost and flexibility. Thus, it provides a more elaborated point of view than conventional TCO analysis.

2. This model can be employed to make cost predictions in adaptive networks facing frequent demand changes.

3. Moreover, this cost model can be used to assist in decisions regarding the network design, such as the selection of the least costly deployment option.

Our main intention when proposing this model is to use it to estimate the cost and select the best adaptation strategy for a 5G RAN featuring a dynamically-adapted functional split, which are the main topics of Chapters 4 and 6. Nonetheless, the model is general and rigorous enough to be used with multiple types of communication networks, as we show in Sec. 3.5 with two application examples.

The rest of this chapter is organized as follows. In Sec. 3.3 we present the system model, including a definition of the network flexibility measure. In Sec. 3.4 we introduce the complete cost model. Sec. 3.5 contains two application examples so as to show how the model can be applied to real networks. Finally, Sec. 3.6 concludes the chapter.

3.2. Related Work

Owing to the promising features of network softwarization technologies, there is considerable interest in investigating their associated deployment and operating costs. Moreover, the ability of softwarized networks to dynamically deal with demand changes also receives dedicated attention. As a result, we classify the relevant related work to this chapter into three categories. First, we discuss *static cost models*, which serve to estimate the cost of deploying or operating a softwarized network without explicitly including the cost of dynamic reconfiguration. Second, we survey *dynamic cost models* in which reconfiguration cost is indeed a main component. Finally, we comment on *flexibility models*, which mainly address the evaluation of reconfiguration abilities along with their associated cost.

3.2.1. Static cost models

As a consequence of their novel architecture and improved features, there are several recent works estimating the TCO of softwarized 5G networks. For instance, in [BNP16] the authors present a detailed cost model for the CAPEX and OPEX required to deploy and operate, respectively, a 5G radio access and core networks featuring SDN and NFV. Nonetheless, this model is intended to help in the selection of deployment alternatives and it does not include dynamic components. Similarly, in [TBK15] the authors address the problem of selecting the optimal location of virtual network functions (VNF) within a 5G core network. They describe a cost model with the objective of minimizing the operating cost, which is estimated from the number of instantiated functions, while also ensuring user satisfaction. Although their approach can be used dynamically, this paper lacks a model for the cost of VNF reconfiguration. In [SRF15], a detailed model of the deployment cost of a C-RAN heterogeneous network is presented. The main intention is to use the model to compare the deployment cost of C-RAN and distributed LTE RANs. Although there are cost components addressing operating and processing costs, the model is intended to represent a static RAN, not a dynamically-adapting one. Finally, in [Bou+18] and [BKM20], the authors present cost models of 5G networks featuring Multiple Input Multiple Output (MIMO), Distributed Antenna System (DAS), Cognitive Radio and a SDN RAN. However, in spite of the complexity of their models, they are not intended to be used for guiding dynamic adaptations, but rather to evaluate alternative deployment options.

Regarding other types of networks, in [Bou+15] a simple cost model is derived for the power consumption, network usage, and license fees of operating an NFV network. The objective is to utilize this model so as to find the optimal location of virtual Deep Packet Inspection (DPI) functions. Once again, albeit this problem may be solved at runtime to trigger dynamic reconfigurations, this aspect is not covered by the cost model. Finally, in [EPP19], the authors study the trade-off between flexibility and cost in the field of SDN optical transport networks. They propose a cost model comprising multiple cost components with the objective of formulating several cost-related optimization problems. Nonetheless, they do not deal directly with dynamic adaptations.

3.2.2. Dynamic cost models

In comparison to static models, literature regarding cost models in which the cost of reconfiguration is taken into account is less abundant. Nevertheless, there are still relevant works in this field. For example, in [Gha+15], the authors propose an approach to tackle the problem of elastically placing VNFs within a cloud network, with the intention of minimizing the operating cost. They model multiple cost components, including not only the cost of running VNFs and carrying traffic, but also the cost of reconfiguring the network and migrating VNFs. Nonetheless, it is assumed that changes in the demands that trigger reconfiguration are sufficiently spaced and that reconfigurations are fast enough not to influence the operating cost. In [SYCP18], an adaptive approach to optimize the monitoring and orchestration processes of a Cloud Management System (CMS) is presented. The authors base on the concept of Quality of Decisions (QoD) [You+16] to propose a dynamic adaptation of the frequency of sample points, when relevant performance parameters are monitored, and decisional points, when the CMS takes management decisions. However, the link between monitoring and orchestration frequencies and actual operating cost is not described in detail. Finally, in [HW10], the authors describe a strategy to dynamically select between push and pull updates in the context of cloud management. This is done with the intention of reducing operating costs, although the actual cost model translating push and pull updates into actual cost is a simple sketch.

3.2.3. Flexibility models

In contrast to static and dynamic cost models, we can also find related work in the literature linking the ability of a softwarized network to reconfigure itself with the cost thereof. For instance, the Chair of Communication Networks at the Technical University of Munich proposes to use the concept of *network flexibility* as a well-defined metric, as presented in [Kel+18]. This definition takes into account the cost that takes to adapt the network to changes in the demands. Moreover, in [Bab+20] we extend this metric and formulate it as a mathematical measure. In addition, multiple types of costs associated with flexible adaptation are presented. Since flexibility and dynamic adaptation is an usual feature of many engineering systems beyond communication networks, we can also find related work in other technological fields. For instance, in [Lin+13] the authors propose a model of the cost of manufacturing flexibility, for which they use dedicated flexibility metrics.

Finally, to the best of our knowledge, [MA+21], upon which we base this chapter, is the first work that addresses in detail the modeling of all cost components in a flexible, dynamically-adapting softwarized network. This includes not only the operating cost in stable conditions, but also the cost of reconfiguring the network and the cost of operating in outdated states.

3.3. System Model

In this section, we introduce the concepts required to derive the cost model of a flexible network.

3.3.1. Network states and demands

We consider a scenario consisting of a softwarized, configurable communication network managed by a *network operator* to achieve a profitable purpose, such as providing connectivity to users, carrying information within a data center, or managing virtual network slices. The instantaneous configuration of the network is referred to as the *state* $\mathrm{s} \in \mathbb{S}$ of the network, where $\mathbb{S}$ is the set of all possible states that can be achieved. For example, the routing tables in the network switches, the location of virtual functions, or the physical resources allocated to a network slice can be used as the state.

The conditions on which the network operates are modeled by the *demand* $\mathbf{d} \in \mathbb{D}$, where $\mathbb{D}$ is the set of all possible demands. The demand includes all parameters that affect the network's profitability but cannot be modified by the network. These parameters can describe the external environment (such as the number of connected users or the requested virtual flows), but they also include any internal configuration that may change out of the network's control (such as the topology graph of active nodes and links in a resilient network).

We say that a demand is *satisfied* if the network state is able to fulfill the expectations that this demand generates. For instance, a demand consisting of a flow request between two network points is satisfied if the intermediate nodes can forward the packets correctly between these points. Those states satisfying a given demand are called *valid* states, whereas any other state is an *invalid* state. We define the function $\mho(\mathbf{d})$ to relate demand $\mathbf{d}$ to its set of valid states:

$$\mho(\mathbf{d}) : \mathbb{D} \mapsto \wp(\mathbb{S}), \tag{1}$$

where $\wp(\mathbb{S})$ is the power set of $\mathbb{S}$. If $\mho(\mathbf{d}) = \emptyset$, we say that the demand $\mathbf{d}$ is *unsatisfiable*.

In a flexible network, demands and states are subject to change over time. From the definition of demand, it follows that the network cannot accurately predict nor prevent demand changes. We model a sequence of demands within time interval $(0, \tau^{\max})$ as a discrete stochastic process $\{\mathcal{D}_j\}_{j\in\mathbb{Z}}$ on the sample space $\mathbb{D}$, where j is an arbitrary time-ordered integer index. We also define the sequence of states $\{\mathcal{S}_j\}_{j\in\mathbb{Z}}$ on the sample space $\mathbb{S}$. We model the duration of each demand $\mathcal{D}_j = \mathbf{d}_j$ by the random variable $\mathcal{T}_j$ on the sample space $\mathbb{R}^+$, and hence we define the stochastic process $\{\mathcal{T}_j\}_{j\in\mathbb{Z}}$ as the sequence of durations of each demand. Assuming that $\{\mathcal{T}_j\}$ is stationary, it can be described by its marginal cumulative distribution function (CDF) $F_{\mathcal{T}}(t)$. Processes $\{\mathcal{D}_j\}$ and $\{\mathcal{T}_j\}$ fully describe the demands over time, as every observed demand $\mathcal{D}_j = \mathbf{d}_j$ is associated with a duration $\mathcal{T}_j = t_j$.

A change in the network state is the result of a conscious network decision, which is taken to address a demand change. Hence, the sequence of states is determined by the sequence of demands. For convenience, we introduce the following notation to represent a demand and a state change, respectively:

$$\widetilde{\mathbf{d}}_j \triangleq \langle \mathbf{d}_j, \mathbf{d}_{j+1} \rangle, \quad \widetilde{\mathbf{s}}_j \triangleq \langle \mathbf{s}_j, \mathbf{s}_{j+1} \rangle. \tag{2}$$

We consider that an *adaptation* consists of a demand change $\widetilde{\mathbf{d}}_j$ and its corresponding state change $\widetilde{\mathbf{s}}_j$. From a modeling point of view, we associate every demand change with a state change, although in practice it may happen that $\mathbf{s}_j = \mathbf{s}_{j+1}$ if there is no effective state change.

3.3.2. The adaptation process

After a noticing a demand change $\widetilde{\mathbf{d}}_j$, a flexible network needs to perform two tasks. First, it needs to run an *adaptation algorithm* to find the most appropriate state $\mathbf{s}_{j+1}$ to satisfy the new demand $\mathbf{d}_{j+1}$. Formally, we model the outcome of this algorithm by means of the *adaptation function*:

$$\chi(d) : \mathbb{D} \mapsto \mathbb{S}, \tag{3}$$

so that $\mathbf{s}_{j+1} = \chi(\mathbf{d}_{j+1})$. Since finding this new state may be computationally hard, we need to account for the time and cost required to do this, as they may impact the

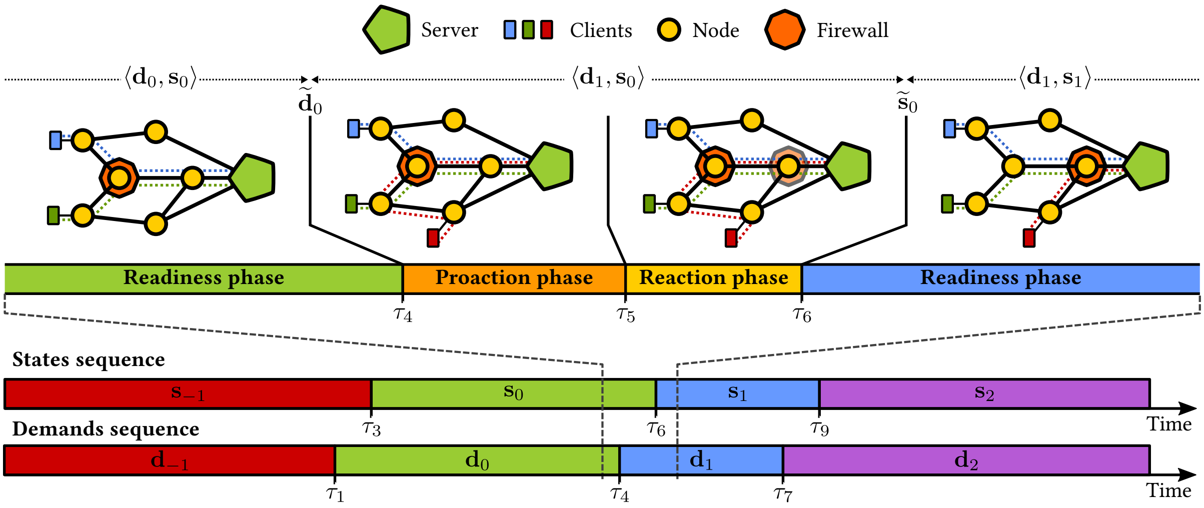


Figure 3.1.: Adaptation phases traversed by a flexible network. The network operator is in charge of providing connectivity between server and clients while ensuring that all packets go through the firewall, whose location can be dynamically chosen.

overall profitability. We refer to the former as *proaction time* z_j^P and to the latter as *proaction cost* c_j^P. In addition, once the network has found the new state, it has to move from the old state to the new one. We refer to the time and cost required to change the state as *reaction time* z_j^R and *reaction cost* c_j^R, respectively.

Overall, the time difference between a demand change $\widetilde{\mathbf{d}}_j$ and its corresponding state change $\widetilde{\mathbf{s}}_j$ is the *action time* $z_j = z_j^P + z_j^R$. As with the sequences of states and demands, we define the discrete stochastic process $\{\mathcal{Z}_j\}_{j\in\mathbb{Z}}$ on the sample space $\mathbb{R}^+$ to model the sequence of action times $\mathcal{Z}_j = z_j$ for every time index i. Assuming stationarity, this process can be characterized by its marginal CDF $F_{\mathcal{Z}}(z)$, i. e., the distribution of the durations of each adaptation. We define in the same manner processes $\{\mathcal{Z}_j^P\}_{j\in\mathbb{Z}}$ and $\{\mathcal{Z}_j^R\}_{j\in\mathbb{Z}}$ for the proaction and reaction times, respectively. Similarly, we denote the *action cost* as $c_j = c_j^P + c_j^R$, which reflects the total effort of addressing demand change $\widetilde{\mathbf{d}}_j$ and realizing state change $\widetilde{\mathbf{s}}_j$, and define the discrete stochastic process $\{\mathcal{C}_j\}_{j\in\mathbb{Z}}$ on the sample space $\mathbb{R}^+$ to model action times $\mathcal{C}_j = c_j$, whose marginal CDF is $F_{\mathcal{C}}(c)$.

If at any time instant τ the network is satisfying the current demand, we say that the network is in the *readiness phase*. The cost per time unit associated with operating the system at this phase is referred to as the *readiness cost* k_j for an arbitrary time index j, which can be expressed as a function of the active demand and state. This is explained in detail in Sec. 3.4.2. The readiness cost is affected by the amount of resources consumed in the current state and the revenue obtained from demand satisfaction. As a result, not being able to satisfy a demand mainly affects this cost component.

In order to clarify the meaning of the aforementioned definitions, we present an exemplary adaptation timeline in Fig. 3.1. This figure shows the observed demands and the states implemented by a network providing connectivity between a server and a set of clients. This connectivity is provided while enforcing a security policy: any packet between the server and the clients must go through a virtual firewall, whose location can be changed during runtime. The optimal firewall location is the one that minimizes the number of links traversed by all packets, thus minimizing latency. Between τ_3 and τ_4, the network is operating in the readiness phase: demand $\mathbf{d}_0$ (clients blue and green) is being satisfied by state $\mathbf{s}_0$ with minimal latency. The revenue obtained from satisfying this demand and the cost of using network resources (links, nodes, CPU, etc.) is reflected by the readiness cost. At time τ_4, the demand changes to $\mathbf{d}_1$: the red client connects to the network. After noticing the demand change, the network realizes that firewall location may not be optimal anymore. As a result, it triggers the adaptation algorithm to find out the optimal state for demand $\mathbf{d}_1$. The time during which the adaptation algorithm is running is the *proaction phase*, and the additional cost associated to it (due to higher resource consumption) is the *proaction cost*. At time τ_5, the adaptation algorithm has converged and returned a state $\mathbf{s}_1 \neq \mathbf{s}_0$ featuring a new firewall location. Therefore, the network starts the procedure to migrate the firewall, hence starting the *reaction phase*, which lasts until the migration is completed at τ_6. Any additional cost associated to this phase is reflected by the *reaction cost*. The union of the proaction and reaction phases is the *action phase*, during which the active state is delayed with respect to the current demand.

3.3.3. Flexibility measure

As mentioned in Sec. 3.2.3, the ability of a network to adapt to a changing environment has been tackled to some extent by previous literature. For instance, in [Klü+19; Bab+20] a mathematical framework for a rigorous definition of network flexibility is provided. In particular, for a given demand sequence, network flexibility $\Phi(z, c)$ is defined as the ratio of satisfied demands within time limit z and cost limit c to the total number of demands. This definition can be easily connected with the present cost model, resulting in a more complete mathematical framework.

The intention of defining network flexibility is to measure the frequency of non-ideal responses to a demand change. Ideally, every demand change should result in a state change leading to a valid state. In real life, adaptation algorithms are not perfect and demands may be unsatisfiable, thus it could happen that the network cannot find a valid state for a new demand. As a result, we can split the sequence of demands $\{\mathcal{D}_j\}$ into two non-overlapping sequences of satisfied demands $\{\mathcal{D}_j^{\in}\}$ and unsatisfied demands $\{\mathcal{D}_j^{\notin}\}$ based on whether $\chi(\mathbf{d}_j) \in \mathbb{V}(\mathbf{d}_j)$ or not. From these sets, we can define the *maximum flexibility* φ of the network as

$$\varphi = \lim_{j \to \infty} \frac{\left|\{\mathcal{D}_j^{\in}\}\right|}{\left|\{\mathcal{D}_j\}\right|}, \tag{4}$$

where the operator $|\cdot|$ yields the total length of a sequence. The maximum flexibility φ is thus the ratio of satisfied demands to total demands, in the absence of the cost and time constraints.

3.4. Cost model of a flexible network

In this section, we analyze the components of the total cost in a flexible network and relate them to the flexibility framework defined in the previous section.

3.4.1. General definitions

As introduced in Sec. 3.3.2, our cost model consists of three independent components: readiness, proaction, and reaction costs. These components can be straightforwardly combined to obtain the total cost of operating a network.

Definition 3.4.1. *The total cost $\overline{Q}$ of operating a flexible network over a long time interval $(0, \tau)$ is*

$$\overline{Q} = \overline{K} + \overline{C}^P + \overline{C}^R, \tag{5}$$

where $\overline{K}$, $\overline{C}^P$, and $\overline{C}^R$ are the mean readiness cost, proaction cost, and reaction cost.

For notation convenience, these components reflect *cost over time* (in arbitrary monetary units per time unit), rather than absolute cost. Hence, the absolute cost of operating a network over interval $(0, \tau)$ is $\overline{Q}\tau$.

In order to achieve a more powerful model, the total cost $\overline{Q}$ includes not only expenses, but also revenue coming from providing service to users. This revenue is modeled as negative cost, hence we say that the network is *profitable* over interval $(0, \tau)$ if and only if $\overline{Q} < 0$. Although a network provider could charge users when they specifically request a service, nowadays a subscription-based revenue, in which users pay a flat rate for a service, is the dominant strategy [ZWW14; KDV08]. Thus, we model revenue as a part of the readiness cost, resulting in $\overline{C}^P > 0$, $\overline{C}^R > 0$ and $\overline{K} < 0$ in a profitable network.

3.4.2. Readiness cost

The instantaneous readiness cost $K(\mathbf{s}, \mathbf{d})$ is the cost of operating a network in state s under demand $\mathbf{d}$. Formally:

$$K(\mathbf{s}, \mathbf{d}) : \mathbb{S} \times \mathbb{D} \mapsto \mathbb{R}^+. \tag{6}$$

In words, $K(\mathbf{s}, \mathbf{d})$ reflects how well the network is satisfying demand $\mathbf{d}$. It includes both cost and revenue of operating a state: resource consumption, user payment via subscriptions, penalizations for unsatisfied demands, etc. Thus, it is the only cost component that can take negative values, which implies that the network operator obtains a profit from operating in the current state. This fact leads to the definition of the *optimal adaptation function* $\chi^*(\mathbf{d}_j)$, which returns the valid state that minimizes the readiness cost for demand $\mathbf{d}_j$:

$$\chi^*(\mathbf{d}_j) = \arg\min_{s} K(\mathbf{s}, \mathbf{d}_j), \tag{7a}$$

$$\text{s.t.} \quad s \in \mathbb{V}(\mathbf{d}_j). \tag{7b}$$

In real scenarios, however, finding the optimal solution to this problem may be too time consuming. We thus consider a more general definition of the adaptation function $\chi(\mathbf{d}_j)$, which approximates $\chi^*(\mathbf{d}_j)$ but may return suboptimal states or may fail to find a valid state. The ability of the adaptation function to return a (possibly suboptimal) valid state is captured by the maximum flexibility φ as defined in (4) in Sec. 3.3.3.

When the network adapts to a sequence of demands $\{\mathcal{D}_j\}$ via a sequence of states $\{\mathcal{S}_j\}$, this results in a sequence of readiness costs that can be modeled as the stochastic process $\{\mathcal{K}_j\}_{j\in\mathbb{Z}}$ for every different demand-state pair. Since $\{\mathcal{D}_j\}$ and $\{\mathcal{S}_j\}$ are stationary, it follows that $\{\mathcal{K}_j\}$ is also stationary. Therefore, the mean readiness cost can be defined as the expected value of this sequence:

$$\overline{K} \triangleq \mathrm{E}\{\mathcal{K}\}. \tag{8}$$

Note that we use a different variable to index the elements of process $\{\mathcal{K}_j\}$ with respect to processes $\{\mathcal{D}_j\}$ and $\{\mathcal{S}_j\}$. This is due to the possible presence of multiple demand-state combinations that result in different readiness costs. To explain this, let us consider a system facing demand $\mathbf{d}_j$ by implementing state $\mathbf{s}_j = \chi(\mathbf{d}_j)$. The resulting readiness cost in this situation is $k_j = K(\mathbf{s}_j, \mathbf{d}_j)$ (with slight abuse of notation). When a new demand $\mathbf{d}_{j+1}$ is requested, the readiness cost changes to $k_{j+1} = K(\mathbf{s}_j, \mathbf{d}_{j+1})$, as state $\mathbf{s}_j$ may not satisfy the new demand, leading to degraded performance and higher cost. At this point, there are multiple possibilities for the next readiness cost value. It could happen that the network finds a valid state $\mathbf{s}_{j+1} = \chi(\mathbf{d}_{j+1})$ before the demand changes again, leading to $k_{j+2} = K(\mathbf{s}_{j+1}, \mathbf{d}_{j+1})$ after state change $\widetilde{\mathrm{s}}_j$. Conversely, the system may be unable to find a valid state or a new demand may appear before the new state is implemented, leading to a new readiness cost value of $k_{j+2} = K(\mathbf{s}_j, \mathbf{d}_{j+2})$.

In order to model the cost resulting from the offset between demands and valid states, we define the *state delay* $\delta(\tau)$ at time instant τ as the index difference between current demand $\mathbf{d}(\tau) = \mathbf{d}_j$ and current state $\mathbf{s}(\tau) = \mathbf{s}_j$, such that $\mathbf{s}_j = \chi(\mathbf{d}_{j-\delta(\tau)})$. Similarly to $\{\mathcal{S}_j\}$ and $\{\mathcal{D}_j\}$, the sequence of state delays can be modeled by the discrete stochastic process $\{\Delta_j\}_{j\in\mathbb{Z}}$. The instantaneous state delay resulting from a sequence of demands and states is shown in Fig. 3.2. We denote the marginal pmf of $\{\Delta_j\}$ as $f_\Delta(\delta)$, which yields the overall probability of the network operating with state delay x.

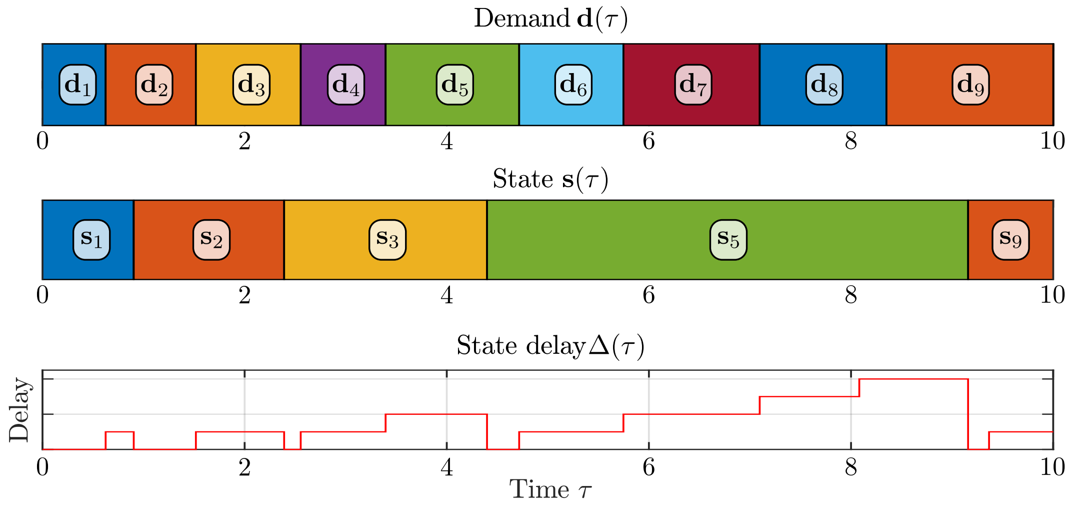


Figure 3.2.: Demands, states, and state delays experienced by an action-interrupting flexible network. The state delay is the instantaneous difference between demand and state indices (numbers in circles).

We define the *readiness degradation function*:

$$K_\Delta(\delta) \triangleq \mathrm{E}\{\mathcal{K}|\Delta = \delta\} \tag{9}$$

as the mean readiness cost when the state delay is $\Delta = \delta$. This function characterizes the performance of the network when dealing with delayed states. An example of such a function is shown in Sec. 3.5.2. In a properly designed network, the mean readiness cost is lowest when state delay is $\Delta = 0$, that is, when the network implements a valid state. Moreover, the cost of a state should monotonically grow with the state delay, reflecting that the demand becomes, on average, increasingly different from the last satisfied demand. Formally, this implies that $K_\Delta(\delta_2) \geq K_\Delta(\delta_1)$ if and only if $\delta_2 \geq \delta_1$. This leads to the following conclusion.

Lemma 3.4.2. *A necessary condition for a network to be profitable is* $K_\Delta(0) < 0$.

The proof for Lemma 3.4.2 is trivial, as it implies that a profitable network ($\overline{Q} < 0$) requires at least that operating in valid states is profitable. Knowing this fact, the following lemma provides a method to compute the mean readiness cost.

Lemma 3.4.3. *The mean readiness cost* $\overline{K}$ *of a flexible network can be calculated in terms of* $K_\Delta(\delta)$ *as:*

$$\overline{K} = \sum_{\delta=0}^{\infty} K_\Delta(\delta) f_\Delta(\delta). \tag{10}$$

Proof. Eq. (10) follows directly from the application of the law of total expectation [Bil95]. □

As mentioned before, $K_\Delta(\delta)$ is a characteristic function of the analyzed network and has to be measured, simulated, or theoretically derived for each case. The pmf $f_\Delta(\delta)$ can be obtained from $F_\mathcal{T}(t)$ and $F_\mathcal{Z}(z)$, that is, from the distributions of demand duration and action time. Nonetheless, the resulting expression for $f_\Delta(\delta)$ is affected by the behavior of the network when a new demand change appears during the action phase, that is, while looking for or moving to a new state.

We refer to a network as *action-persistent* if its action phases are not interrupted by a new change in the demand. That is, an action-persistent network carries on with the action phase of the demand change that originated it, and eventually realizes the associated state change, regardless of any new demand change that may occur in the meantime. As a result, the new state may not be optimal from the start if there are other demand changes after the first demand change. Conversely, an *action-interrupting* network stops and resets its action phase if a new demand appears during the action phase. As a consequence, an action-interrupting network only realizes state changes leading to valid, non-delayed states. Fig. 3.2 is an example of an action-interrupting network behavior. We can observe, for instance, that the action phase associated with demand change $\widetilde{\mathbf{d}}_3$ is interrupted by $\widetilde{\mathbf{d}}_4$, which starts a new action phase that eventually leads to the implementation of $\mathbf{s}_5^*$ and optimally satisfies $\mathbf{d}_5$. We argue that action-interrupting networks are the best option for operators who prefer to implement valid states rather than not to interrupt adaptations. Therefore, in this thesis we only show the derivation of $f_\Delta(\delta)$ for action-interrupting networks, leaving the analysis of action-persistent networks for future work.

In our path to calculate $f_\Delta(\delta)$, we define the random variable $\mathcal{R}$ as the time difference between any instant in the considered interval $(0, \tau^{\max})$ and the most recent demand change. The probability density function (pdf) of this variable is provided in the following lemma.

Lemma 3.4.4. *The pdf $f_\mathcal{R}(r)$ of $\mathcal{R}$ is*

$$f_\mathcal{R}(r) = \frac{1 - F_\mathcal{T}(r)}{\overline{T}}, \tag{11}$$

where $\overline{T} \triangleq E\{\mathcal{T}\}$.

Proof. We introduce the intermediate random variable $\mathcal{T}'$ to model the duration of the active demand at any uniformly-selected instant. By the law of the total expectation:

$$f_\mathcal{R}(r) = \int_0^\infty f_{\mathcal{R}|\mathcal{T}'}(r|t) f_{\mathcal{T}'}(t) dt, \tag{12}$$

where $f_{\mathcal{T}'}(t)$ is the pdf of $\mathcal{T}'$ and $f_{\mathcal{R}|\mathcal{T}'}(r|t)$ is the conditional pdf of $\mathcal{R}$ when the most recent demand is known. The probability of randomly selecting a demand is directly proportional to its duration. From this fact and the law of total probability it follows that

$$f_{\mathcal{T}'}(t) = \frac{t \cdot f_{\mathcal{T}}(t)}{\int_0^\infty \xi \cdot f_{\mathcal{T}}(\xi) d\xi} = \frac{t \cdot f_{\mathcal{T}}(t)}{\overline{T}}. \tag{13}$$

The conditional pdf $f_{\mathcal{R}|\mathcal{T}'}(r|t_j)$ yields the probability density of selecting an instant that is r time units after the start of demand $\mathbf{d}_j$, given that the active demand is $\mathbf{d}_j$. Since there must be no bias when selecting these instants, it is clear that $f_{\mathcal{R}|\mathbb{D}'}(r|t_j) = \frac{1}{t_j}$ if $0 \leq r < t_j$ and $f_{\mathcal{R}|\mathbb{D}'}(r|\mathbf{d}_j) = 0$ otherwise. Combining (13) with this fact results in:

$$f_{\mathcal{R}}(r) = \int_r^\infty \frac{f_{\mathcal{T}}(t)}{\overline{T}} dt, \tag{14}$$

which directly leads to (11). □

Using Lemma 3.4.4 we can directly calculate an expression for $f_\Delta(\delta)$, as shown in the following lemma.

Lemma 3.4.5. *The pmf $f_\Delta(\delta)$ of the state delay of an action-interrupting network is*

$$f_\Delta(\delta) = \begin{cases} \alpha\varphi & \text{if } \delta = 0, \\ (1-\alpha\varphi)(1-\beta\varphi)^{\delta-1}\beta\varphi & \text{if } \delta > 0, \end{cases} \tag{15}$$

where

$$\alpha \triangleq \frac{1}{\overline{T}} \int_0^\infty F_{\mathcal{Z}}(t) \left(1 - F_{\mathcal{T}}(t)\right) dt \tag{16}$$

and

$$\beta \triangleq \int_0^\infty F_{\mathcal{Z}}(t) f_{\mathcal{T}}(t) dt. \tag{17}$$

Proof. An action-interrupting network resets its action phase every time the demand changes. Hence, for any instant, the probability of operating with state delay $\Delta = 0$ is the probability of being able to find a valid state and surpassing the action time for the most recent demand. The probability of the former event is given by the maximum flexibility φ, whereas the latter event is derived as follows [Dur10]:

$$f_\Delta(0) = \varphi \Pr\{\mathcal{Z} \leq \mathcal{R}\} = \varphi \int_0^\infty F_{\mathcal{Z}}(r) f_{\mathcal{R}}(r) dr, \tag{18}$$

which leads to the first case of (15) after substituting (11). If this is not the case, with probability $(1-\alpha\varphi)$, the probability of reaching a state delay $\delta > 0$ is the probability of being able to find a valid solution after x unsuccessful attempts. This event follows a geometric distribution of parameter p:

$$p = \varphi \Pr\{\mathcal{Z} \leq \mathcal{T}\} = \varphi \int_0^\infty F_{\mathcal{Z}}(r) f_{\mathcal{T}}(r) dr, \tag{19}$$

which is the probability of obtaining a valid solution within an action phase that is shorter than the duration of a demand. Given the pmf of a geometrically-distributed random variable as $p(1-p)^{\delta-1}$ (for $\delta > 0$), (15) is finally obtained. □

With an expression for $f_\Delta(\delta)$, we calculate the resulting mean readiness cost in the following theorem.

Theorem 3.4.6. *The mean readiness cost $\overline{K}$ of an action-interrupting network is*

$$\overline{K} = \alpha\varphi K_\Delta(0) + (1-\alpha\varphi)\beta\varphi\widehat{K}_\beta, \tag{20}$$

where

$$\widehat{K}_\beta \triangleq \sum_{\delta=1}^{\infty} K_\Delta(\delta)(1-\beta\varphi)^{\delta-1}. \tag{21}$$

Proof. Eq. (20) is the result of combining (10) and (15). □

The expression in Theorem 3.4.6 allows us to calculate the mean readiness cost of an adaptive system from its demand duration distribution, action time distribution, readiness degradation function, and maximum flexibility. In order to find out if a network is profitable, the following corollary can be used.

Corollary 3.4.6.1. *A necessary condition for a flexible network to be profitable is*

$$\frac{\alpha}{(\alpha\varphi-1)\beta} < \frac{\widehat{K}_\beta}{K_\Delta(0)}. \tag{22}$$

given that $K_\Delta(0) < 0$.

Proof. A profitable network must fulfill $\overline{K} < 0$, which implies that $K_\Delta(0) < 0$ (Lemma 3.4.2). After applying these relations in (20), we reach (22). □

Corollary 3.4.6.1 provides us with a simple method to rule out non-profitable network configurations. The left side of (22) reflects the frequency of demand changes and the swiftness and effectiveness of the adaptation, whereas the right side is influenced by the quality of the solutions. As a result, it delimits a border for finding profitable configurations within the speed-quality tradeoff. An example application of Theorem 3.4.6 and Corollary 3.4.6.1 is presented in Sec. 3.5.2.

Once we have a closed-form expression of $\overline{K}$, we can derive some interesting properties that may be useful when analyzing flexible communication networks. In order to highlight the dependence of $\overline{K}$ on the maximum flexibility φ, we use the notation $\overline{K}(\varphi)$ in the following.

Property 3.4.7. *As $\varphi \to 0$, the value of the mean readiness cost $\overline{K}(\varphi)$ tends to the limit of the RDF $K_\Delta(\delta)$ when $\delta \to \infty$. That is:*

$$\lim_{\varphi\to 0} \overline{K}(\varphi) = \lim_{\delta\to\infty} K_\Delta(\delta) \tag{23}$$

Proof. (23) is a consequence of the final value theorem. This theorem states the following equivalence [Opp+97]:

$$\lim_{\delta \to \infty} K_\Delta(\delta) = \lim_{\mathfrak{z} \to 1} (\mathfrak{z} - 1) \sum_{\delta=0}^{\infty} K_\Delta(\delta) \mathfrak{z}^{-\delta}, \tag{24}$$

where the right-hand term is the Z-transform (using $\mathfrak{z}$ as variable) of $K_\Delta(\delta)$. We perform the variable change $\mathfrak{z} = (1 - \beta\varphi)^{-1}$, which leads to:

$$\begin{aligned}
\lim_{\delta \to \infty} K_\Delta(\delta) &= \lim_{\mathfrak{z} \to 1} (\mathfrak{z} - 1) \sum_{\delta=0}^{\infty} K_\Delta(\delta) \mathfrak{z}^{-\delta} & (25) \\
&= \lim_{\varphi \to 0} \beta\varphi \sum_{\delta=0}^{\infty} K_\Delta(\delta)(1 - \beta\varphi)^{\delta-1} & (26) \\
&= \lim_{\varphi \to 0} \beta\varphi \widehat{K}_\beta(\varphi) & (27) \\
&= \lim_{\varphi \to 0} \alpha\varphi K_\Delta(0) + (1 - \alpha\varphi)\beta\varphi \widehat{K}_\beta(\varphi) & (28) \\
&= \lim_{\varphi \to 0} \overline{K}(\varphi), & (29)
\end{aligned}$$

which is the identity stated in (23). □

The implications of Property 3.4.7 are those intuitively expected: as the maximum flexibility decreases, the mean readiness cost in the network approaches that corresponding to an infinite state delay. Thus, in a real network, $K_\Delta(\delta)$ converges to the mean cost of operating in a state that is totally uncorrelated with the demand as $\varphi \to \infty$. The next property deals with the opposite extreme case.

Property 3.4.8. *As $\varphi \to 1$, the value of the mean readiness cost $\overline{K}(\varphi)$ is given by:*

$$\lim_{\varphi \to 1} \overline{K}(\varphi) = \alpha \overline{K}_\Delta(0) + (1 - \alpha)\beta \sum_{\delta=1}^{\infty} K_\Delta(\delta)(1 - \beta)^{\delta-1} \tag{30}$$

Proof. This identity can be obtained by simply introducing $\varphi = 1$ into $\overline{K}(\varphi)$ as presented in (20). □

By comparing Properties 3.4.7 and 3.4.8, we observe that, although $\varphi = 0$ guarantees operating at the worst possible readiness cost, $\varphi = 1$ is not sufficient to operate at the minimum readiness cost. Indeed, the RDF $K_\Delta(\delta)$ and the distributions of the durations of demands and states, by means of parameters α and β, define the minimum achievable cost.

Once we know the mean readiness cost for extreme values of φ, the following properties address the shape of $\overline{K}(\varphi)$.

Property 3.4.9. *For an increasing RDF $K_{\Delta}(\delta)$, the mean readiness cost $\overline{K}(\varphi)$ is a decreasing function of φ. That is:*

$$\overline{K}(\varphi_1) \leq \overline{K}(\varphi_2) \iff \varphi_1 \geq \varphi_2, \quad \forall \varphi_1, \varphi_2 \in [0,1]. \tag{31}$$

Proof. We prove Property 3.4.9 by showing that $\frac{d\overline{K}(\varphi)}{d\varphi} \leq 0 \; \forall \varphi \in [0,1]$. This is done in Appendix A.1. □

Property 3.4.10. *For a bounded and increasing RDF$K_{\Delta}(\delta)$, the mean readiness cost $\overline{K}(\varphi)$ is a convex function of φ. That is:*

$$\overline{K}\left(\mathfrak{s}\varphi_1 + (1-\mathfrak{s})\varphi_2\right) \leq \mathfrak{s}\overline{K}\left(\varphi_1\right) + (1-\mathfrak{s})\overline{K}\left(\varphi_2\right) \quad \forall \mathfrak{s}, \varphi_1, \varphi_2 \in [0,1] \tag{32}$$

Proof. We prove Property 3.4.10 by showing that $\frac{d^2\overline{K}(\varphi)}{d\varphi^2} \geq 0 \; \forall \varphi \in [0,1]$. This is done in Appendix A.2. □

Properties 3.4.9 and 3.4.10 lead to important implications in the relationship between the mean readiness cost $\overline{K}(\varphi)$ and the maximum flexibility φ. Namely, they allow us to conclude not only that the mean readiness cost increases as the maximum flexibility decreases, but also that the lower the maximum flexibility φ, the faster the growing rate of $\overline{K}(\varphi)$, regardless of any other parameter. In addition, the convexity of $\overline{K}(\varphi)$ will allow us to identify the value of φ that leads to the global minimum total cost $\overline{Q}$, since the action cost is a linear function of φ, as we show in Sec. 3.4.4.

The shape of four exemplary $\overline{K}(\varphi)$ functions for multiple RDF shapes is depicted in Fig. 3.3 and Fig. 3.3. In Fig. 3.3, we show RDFs with the same range, $-10 \leq K_{\Delta}(\delta) \leq 10$, in four different shapes: step, hyperbolic, linear, and logistic growth. Their corresponding mean readiness cost functions $\overline{K}(\varphi)$ are shown in Fig. 3.3, alongside experimental results that are obtained after simulating a system with those RDFs, exponentially-distributed demand durations of mean 1, i. e., $\mathcal{T} \sim Exp(1)$, and uniformly-distributed action times between 0 and 0.5, i. e., $\mathcal{Z} \sim Unif(0, 0.5)$, for a total simulated time of 20,000 demand durations. We observe that the shape of $\overline{K}(\varphi)$ is always convex and decreasing, regardless of the shape of its associated RDF, which may be non-convex and non-concave. We also see that, as expected from Properties 3.4.7 and 3.4.8, $\overline{K}(0) \approx 10$ but $\overline{K}(1) \neq -10$. Finally, we conclude that that the model and the simulation results are an almost perfect match.

3.4.3. Proaction cost

The proaction cost $C^P(\widetilde{\mathbf{d}}_j)$ of a flexible network represents the cost of consuming computational resources so as to find the new state after a demand change. This cost component can be formally defined as a function mapping a demand change to a cost value:

$$C^P(\widetilde{\mathbf{d}}_j) : \mathbb{D}^2 \mapsto \mathbb{R}^+. \tag{33}$$

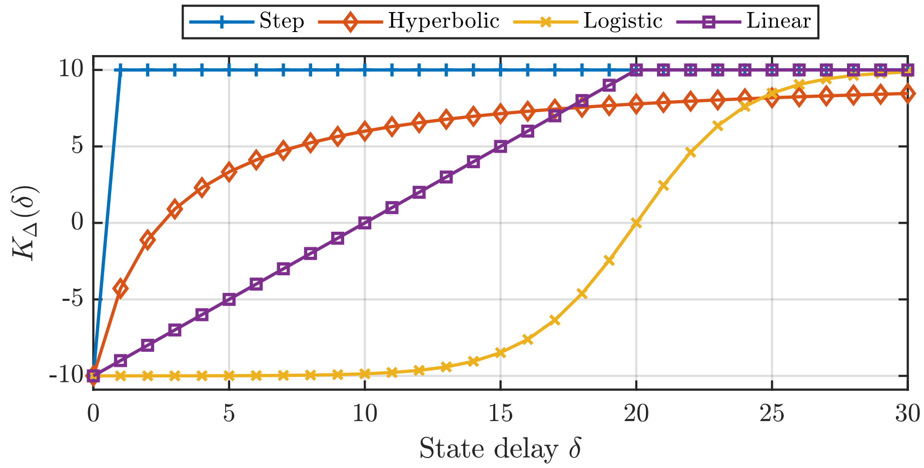


Figure 3.3.: Four exemplary readiness degradation functions (RDFs) featuring the same range but different shapes.

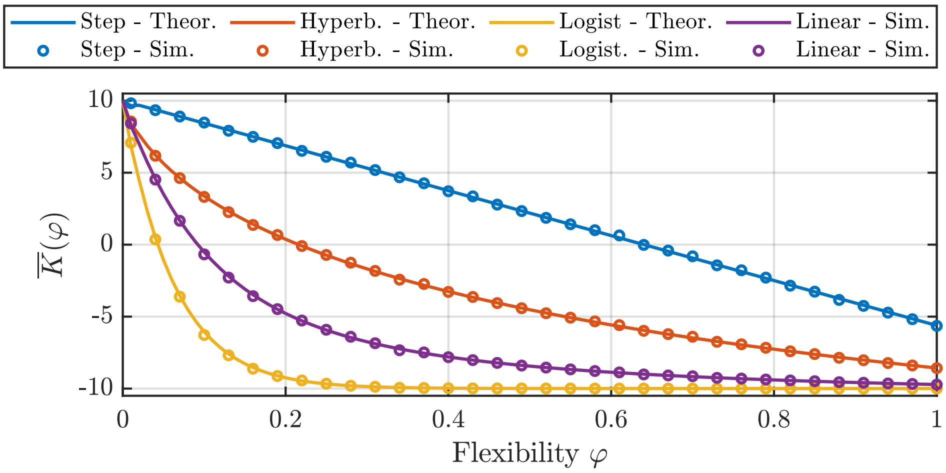


Figure 3.4.: Mean readiness cost $\overline{K}(\varphi)$ against maximum flexibility φ corresponding to the four RDFs described above, for exponentially-distributed demand durations of mean 1, and uniformly-distributed action times between 0 and 0.5. Theoretical and simulated values.

The exclusive dependency on $\widetilde{\mathbf{d}}_j$ implies that this cost is present every time there is a demand change, even if there is no eventual state change. We identify two factors contributing to the proaction cost. On the one hand, a demand change may imply an instantaneous time-independent cost C_0^P, for instance resulting from the activation of new capabilities to solve the adaptation problem. On the other hand, while the adaptation problem is being solved, additional resources (CPU, memory, etc.) are consumed during the proaction phase, incurring in a cost of C_z^P monetary units per time unit. As a result, we can express the proaction cost as a function of the proaction time of an action-interrupting network as follows:

$$C^P(\widetilde{\mathbf{d}}_j) \triangleq \frac{1}{t_{j+1}} \left(C_0^P + C_z^P \cdot \min(z_j^P, t_{j+1}) \right), \tag{34}$$

where the minimum operator guarantees that the proaction phase is stopped if the demand changes and the term $\frac{1}{t_{j+1}}$ normalizes the cost to the duration of the demand. From (34) and the sequence of proaction times $\{\mathcal{Z}_j^P\}$ we define a new stochastic process $\{\mathcal{C}_j^P\}_{j\in\mathbb{Z}}$ to model the sequence of proaction costs, such that:

$$\mathcal{C}_j^P = \frac{1}{\mathcal{T}_{j+1}} \left(C_0^P + C_z^P \cdot \min(\mathcal{Z}_j^P, \mathcal{T}_{j+1}) \right). \tag{35}$$

The mean of the variable above is presented in the following.

Theorem 3.4.11. *The mean proaction cost C^P of an action-interrupting network is:*

$$\overline{C}^P = \frac{C_0^P}{\overline{T}} + \left(\frac{\beta^P \overline{Z}^P}{\overline{T}} + 1 - \beta^P \right) C_z^P, \tag{36}$$

where

$$\beta^P \triangleq \int_0^\infty F_{\mathcal{Z}^P}(t) f_{\mathcal{T}}(t) dt, \tag{37}$$

and $\overline{Z}^P \triangleq E\{\mathcal{Z}^P\}$.

Proof. After applying the expectation operator to (35), we need to calculate

$$\mathrm{E}\{\min(\mathcal{Z}_j^P, \mathcal{T}_{j+1})\}.$$

The random variable within the brackets takes the same values as $\mathcal{Z}_j^P$ when $\mathcal{Z}_j^P \leq \mathcal{T}_{j+1}$. From stationarity and the law of total expectation:

$$\mathrm{E}\{\min(\mathcal{Z}^P, \mathcal{T})\} = \Pr\{\mathcal{Z}^P \leq \mathcal{T}\} \cdot \overline{Z}^P + (1 - \Pr\{\mathcal{Z}^P \leq \mathcal{T}\}) \cdot \overline{T} \tag{38}$$

The probability $\Pr\{\mathcal{Z}^P \leq \mathcal{T}\}$ is derived in the same way as (17) to yield (37). □

In Sec. 3.5.3, we show an example network where we apply Theorem 3.4.11 to find out the optimal number of CPU cores to be used during the proaction phase.

Corollary 3.4.11.1. *A necessary condition for an action-interrupting network to be profitable is*

$$C_z^P < \frac{\overline{TK} + C_0^P}{(\beta^P - 1)\overline{T} - \beta^P \overline{Z}^P}. \tag{39}$$

Proof. This relation follows directly from the fact that a profitable network must fulfill $\overline{K} + \overline{C}^P < 0$. □

Corollary 3.4.11.1 provides us with an upper bound on the maximum number of additional resources that a flexible network is allowed to utilize in order to cope with demand changes before it turns unprofitable.

3.4.4. Reaction cost

The reaction cost $C^R(\widetilde{\mathbf{s}}_j)$ reflects the effort of performing the state change required for an adaptation, after this has been selected in the proaction phase. Therefore, we define it as a function of the state change $\widetilde{\mathbf{s}}_j$:

$$C^R(\widetilde{\mathbf{s}}_j) : \mathbb{S}^2 \mapsto \mathbb{R}^+. \tag{40}$$

Note that $C^R(\widetilde{\mathbf{s}}_j) = 0$ if the demand change $\widetilde{\mathbf{d}}_j$ results in no adaptation, that is, if $\mathbf{s}_{j+1} = \mathbf{s}_j$. This can happen either if $\mathbf{s}_j$ is already optimal, or if the adaptation algorithm could not find a better state which satisfying $\mathbf{d}_j$.

Based on the same rationale as with the proaction cost, we identify two factors contributing to the reaction cost. First, an instantaneous time-independent cost C_0^R models the activation of state-changing procedures (such as memory allocation for virtual migrations [MAGVK19a], for instance). Second, we define a constant rate of C_z^R monetary units per time unit to characterize the usage of additional resources when changing the network state. Consequently, we formulate the reaction cost as:

$$C^R(\widetilde{\mathbf{s}}_j) \triangleq \frac{1}{t_{j+1}} \left(C_0^R + C_z^R \cdot \min(z_j^R, t_{j+1} - z_j^P) \right) \tag{41}$$

whenever $t_{j+1} \leq z_j^P$ and the demand is satisfiable. Otherwise, $C^R(\widetilde{\mathbf{s}}_j) = 0$ as no new state has been generated. We define the stochastic process $\{\mathcal{C}_j^R\}_{j\in\mathbb{Z}}$ to model a time-ordered sequence of reaction costs as:

$$\mathcal{C}_j^R = \frac{1}{\mathcal{T}_{j+1}} \left(C_0^R + C_z^R \cdot \min(\mathcal{Z}_j^R, \mathcal{T}_{j+1} - \mathcal{Z}_j^P) \right), \tag{42}$$

for every index i whenever $\mathcal{T}_{j+1} \leq \mathcal{Z}_j^P$ and $\mathcal{C}_j^R = 0$ otherwise.

Theorem 3.4.12. *The mean reaction cost $\overline{C}^R$ of an action-interrupting network is:*

$$\overline{C}^R = \frac{\varphi \beta^P}{T} \left(C_0^R + C_z^R \left(\beta Z (1-\beta) T - Z^P \right) \right). \tag{43}$$

Proof. The equality (42) occurs when $\mathcal{Z}_j^P \leq \mathcal{T}_{j+1}$ with probability β^P. By the law of total expectation, we just need to figure out the value of $\mathrm{E}\{\min(\mathcal{Z}_j^R, \mathcal{T}_{j+1} - \mathcal{Z}_j^P)\}$. The random variable within the brackets takes the same values as $\mathcal{Z}_j^R$ when $\mathcal{Z}_j \leq \mathcal{T}_{j+1}$, that is, with probability β. After some straightforward algebra, (43) is obtained. □

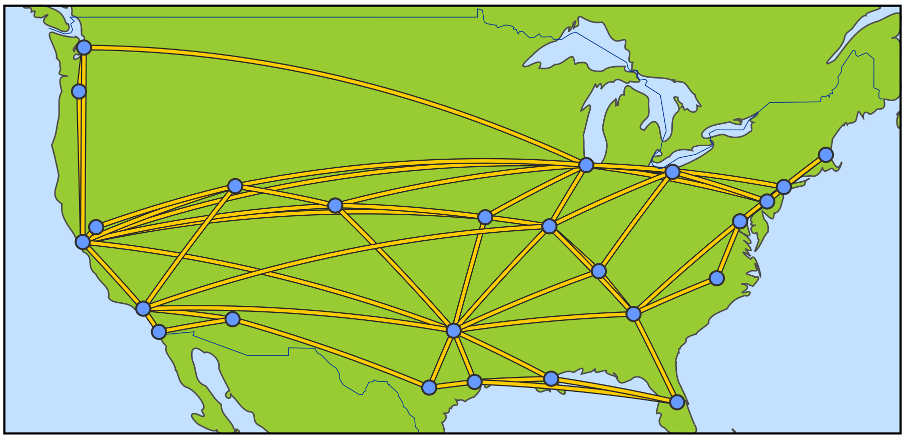


Figure 3.5.: Topology of the example network implementing random flow requests. The topology is that of AT&T North America [ATT] taken from the Internet Topology Zoo [Thea].

Corollary 3.4.12.1. *The optimal maximum flexibility φ^* that minimizes the total cost $\overline{Q}$ is that satisfying the following equation:*

$$\alpha K_\Delta(0) + \beta \sum_{\delta=1}^{\infty} K_\Delta(\delta) \left[1 - 2\alpha\varphi^* - \beta\varphi^* \frac{1-\alpha\varphi^*}{1-\beta\varphi^*}(\delta - 1)\right] (1-\beta\varphi^*)^{\delta-1} = \\ -\frac{\beta^P}{T}\left(C_0^R + C_z^R\left(\beta Z(1-\beta)T - Z^P\right)\right). \quad (44)$$

Proof. The total cost $\overline{Q}$ can be computed as a function of φ as:

$$\overline{Q}(\varphi) = \overline{K}(\varphi) + \overline{C}^P + \overline{C}^R(\varphi). \quad (45)$$

From Property 3.4.10 and Theorem 43, we know that $\overline{K}(\varphi)$ and $\overline{C}^R(\varphi)$ are convex over φ. Thus, the value of φ that globally minimizes $\overline{K}(\varphi)$ satisfies:

$$\left.\frac{d\overline{Q}(\varphi)}{d\varphi}\right|_{\varphi=\varphi^*} = 0. \quad (46)$$

After differentiating (20) and (43) and some straightforward algebra, (44) is obtained. □

3.5. Application example

The main goal of the cost model presented in this chapter is to help in the definition of the FSSP, as well as in solving it dynamically. These issues are addressed in detail in Chapters 4 and 6. However, this cost model is also general enough to be used for a wide variety of use cases. Owing to this, in this section we show an example application to a different problem. Namely, we present the problem of designing a flow-embedding network facing frequent requests. After proposing several alternatives to solve it, and use the derived cost model to select the best implementation options.

3.5.1. Network description

We consider a communication network with the topology shown in Fig. 3.5. The objective of this network is to provide virtual connectivity between pairs of nodes, that is, to embed flow requests between any two nodes. These flows have to be implemented on single paths, i. e., there can be no fractional flows. This can be accomplished by dynamically reserving links and reconfiguring the routing tables in the intermediate nodes. Providing this service yields a revenue for the network operator, but managing and reconfiguring the network is costly. As a result, the operator wants to serve users with the minimum operational cost, so that the net revenue is maximized.

Each flow request consists of a source-destination pair and a required throughput. For simplicity, we assume that all nodes have the same probability of becoming source or destination, regardless of their geographical position, so that every source-destination pair is equally likely. We define the instantaneous demand $\mathbf{d}$ as the set of all active flow requests. As a consequence, every time a new flow request arrives, or a flow is not needed any more, the demand $\mathbf{d}$ changes. The instantaneous network state s is defined as a vector containing the state of each link (active or inactive) and the list of flows that are assigned to it, if any. Inactive links cannot carry any flows, but they consume less power. When a link is active, multiple flows can be assigned to it as long as the capacity of the link is not exceeded. Without loss of generality, we normalize all link capacities to 1. Moreover, we model the throughput requested by each flow with a random uniform distribution between 0 and 1. We deem that a demand is satisfied if the current network state allows the embedding of all flows contained in the demand without exceeding any link capacity.

The duration of the demands $\mathcal{T}$ (in seconds) follows a Pareto Type II distribution, also called Lomax distribution [Lom54]:

$$F_{\mathcal{T}}(t) = 1 - \left(1 + \frac{t}{\lambda}\right)^{\sigma}, \tag{47}$$

for $t \geq 0$. We set $\lambda = 10$ and $\sigma = 2.25$ so that the mean demand duration is $T = \frac{\lambda}{\sigma - 1} = 8$ seconds. This distribution is selected since it is commonly observed in the interarrival time between internet bursts, file sizes, transfer times, etc [Dow05]. Moreover, we choose a mean demand duration of 8 s so that it is comparable to the action time of reconfiguring the network. Hence, it is unclear whether a network can operate profitably in this situation when using conventional cost models. Note that this selection of parameters is due to illustrative purposes, but other values or distributions can be used without affecting the effectiveness of the model.

The problem of providing the intended connectivity can be formulated as an instance of the *integer* min-cost multicommodity flow problem [Tom66]. This problem is known to be NP-Hard [EIS75], thus the network operator relies on an approximation algorithm rather than on an exact approach. In our example, the operator uses a genetic algorithm as the adaptation algorithm [FBB19]. In a nutshell, the operation of the genetic algorithm is as follows [MTK01]. First, a certain number of random solutions, called the *population size* $\mathfrak{p}$, are generated. Each solution contains a flag per link indicating if this link is active or not. Then each solution is evaluated to assess how close it is to satisfy the current demand and how many links it uses. After all solutions have been evaluated, the worst ones are discarded whereas the best ones are kept for the next generation. These are then combined to each other and randomly modified to produce the next generation. These steps are repeated until a convergence criterion is satisfied, which in our case is the absence of improvements after 25 generations.

3.5.2. Selection of a profitable population size

Given the high number of parameters in a genetic algorithm, the operator is interested in finding the right ones so that the network is profitable. For example, they want to select the population size $\mathfrak{p}$. Selecting the right $\mathfrak{p}$ is not trivial, since it affects the speed-accuracy trade-off of the algorithm. On the one hand, a large $\mathfrak{p}$ increases the probability of eventually finding an optimal solution with minimum readiness cost. On the other hand, the higher $\mathfrak{p}$, the more solutions have to be evaluated, which increases proaction time and cost.

The readiness cost k_j at any point is the combination of three factors. First, subscribed users provide a constant revenue of 71 normalized cost units (ncu). Second, active links have a cost of 11 ncu/s, whereas inactive links do not cost anything. Finally, whenever a requested flow is not being satisfied, the operator has to pay a compensation of 10 ncu/s to each affected user. Thus, the average readiness cost of state s and demand $\mathbf{d}$ is:

$$K(\mathbf{s}, \mathbf{d}) = -71 + 11\mathfrak{l}(\mathbf{s}) + 10\mathfrak{v}(\mathbf{s}, \mathbf{d}) \text{ ncu}, \tag{48}$$

where $\mathfrak{l}(\mathbf{s})$ is the number of used links in state s and $\mathfrak{v}(\mathbf{s}, \mathbf{d})$ is the number of unsatisfied flows for demand $\mathbf{d}$ and state s.

The operator is considering to use a population size in the set:

$$\mathfrak{P} = \{250, 750, 1250, 1750, 2250, 2750\}.$$

After evaluating the performance of the genetic algorithm in a dedicated simulator, we observe that the action time can be modeled by a uniform distribution such that $\mathcal{Z} \sim \mathcal{U}(0, \widehat{Z})$, where $\widehat{Z} = 0.016 \cdot \mathfrak{p}$. We also measure the maximum flexibilities

$$\Phi = \{0.35, 0.55, 0.64, 0.7, 0.73, 0.75\}$$

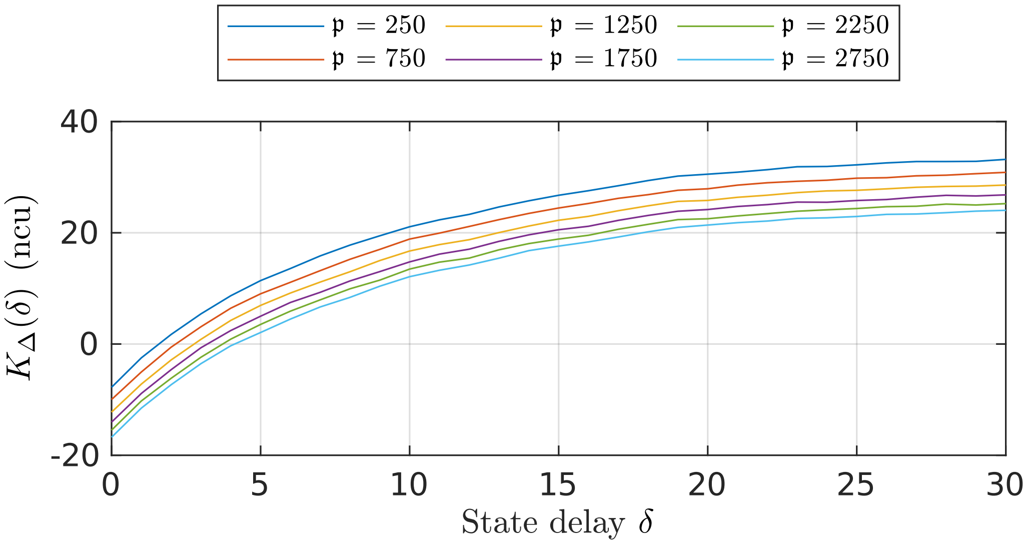


Figure 3.6.: Readiness degradation function $K_\Delta(\delta)$ of a network employing a genetic algorithm to implement integer multicommodity flows for six different population sizes.

for each $\mathfrak{p}$ in the same order as Π, representing the frequency of demand-satisfying solutions. Finally, we measure the readiness degradation function $K_\Delta(\delta)$ for state delays $0 \leq \delta \leq 30$ for each population size $\mathfrak{p}$, which is shown in Fig. 3.6. We observe that the mean readiness cost steadily increases with the state delay, as a result of higher compensations due to unsatisfied flows and unnecessarily active links. It is also clear that the evolution of the cost for the different population sizes is very similar, although low population sizes lead to higher link usage, which increases the cost.

From the above measurements and the distributions of $\mathcal{T}$ and $\mathcal{Z}$, we are able to compute parameters α and β as shown in (16) and (17). With this and the readiness degradation function $K_\Delta(\delta)$, we can apply Corollary 3.4.6.1 to figure out if any of the considered population sizes may lead to a profitable network. A graphical representation of the result is shown in Fig. 3.7, where left and right sides of inequality (22) are depicted as horizontal and vertical axes, respectively. We observe that three of the considered population sizes, $\mathfrak{p} \in \{1250, 1750, 2250\}$, lie on the profitable region. An interesting behavior is captured by the model, as those populations that are lower than 1250 or greater than 2250 lead to unprofitable networks. The explanation is that, when the population is small, the quality of the states yielded by the genetic algorithm is not good enough to properly address the demands. Conversely, when the population is large, the action time is so high that the network cannot properly cope with frequent demand changes.

3.5.3. Optimal parallelization level

Let us now consider that the network operator has the ability to dedicate multiple CPU cores to solving the adaptation problem in the proaction phase. Increasing the

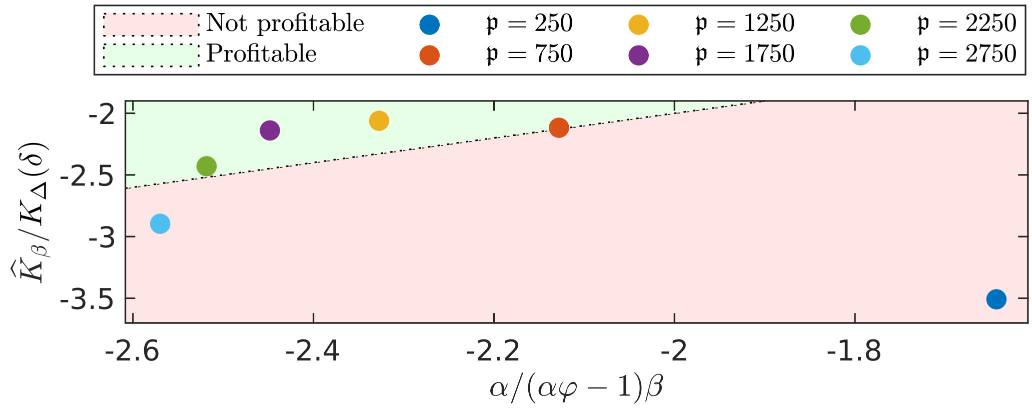


Figure 3.7.: Graphical representation of inequality (22) for testing the profitability of different population sizes.

number of cores reduces the proaction time, thus decreasing the state delay, which may lead to higher revenue. Nevertheless, utilizing more cores also increases the proaction cost, which may counter the revenue increase and result in higher total cost. In order to find out the optimal level of parallelization, we can combine Theorems 3.4.6 and 3.4.11 to predict the evolution of readiness and proaction costs as the number of cores assigned to the adaptation algorithm increases.

Let us denote the parallelization level, i. e., the number of additional CPU cores used in the proaction phase, as $\mathfrak{g}$. For simplicity, let us assume that the reaction cost and time are negligible so that $\mathcal{Z}^{\mathcal{P}} \approx \mathcal{Z}$, hence $\mathcal{Z}^{\mathcal{P}} \sim \mathcal{U}(0, \widehat{Z})$. Clearly, the value of $\widehat{Z}$ is a decreasing function of $\mathfrak{g}$. Namely, we define it as $\widehat{Z} = \frac{\widehat{Z}_0}{\Upsilon(\mathfrak{g})}$, where $\widehat{Z}_0$ is the maximum proaction time with a single core and $\Upsilon(\mathfrak{g})$ is the time reduction factor for $\mathfrak{g}$ cores. Ideally, $\Upsilon(\mathfrak{g}) = \mathfrak{g}$ if the load can be perfectly shared among all cores. Nonetheless, in real scenarios it is observed that $\Upsilon(\mathfrak{g})$ grows linearly at first, but eventually saturates due to imperfections in load division [Glo15; Ber+14]. To capture this, we use the $\Upsilon(\mathfrak{g})$ depicted in Fig. 3.8.

Regarding the proaction cost components, let $C_0^P = 0$ (no additional cost for starting the proaction cost) and $C_z^P = 1 \cdot \mathfrak{g}$ ncu/s, where $\mathfrak{g}$ is the number of assigned CPU cores, that is, the parallelization level. This value of C_z^P means that the network consumes 1 ncu/s per used CPU core during the proaction phase, in addition to the readiness cost. Finally, based on the previous results, we select a population size of $\mathfrak{p} = 1750$, which implies a maximum flexibility of $\varphi = 0.7$ and $\widehat{Z}_0 = 28$.

We can calculate the relationship between the mean proaction cost C^P and the parallelization level $\mathfrak{g}$ by feeding the aforementioned expressions into (36), in Theorem 3.4.11. The result is shown in Fig. 3.9, where an interesting behavior can be observed. Up to around $\mathfrak{g} = 8$, the proaction cost increases rapidly, since the duration of the proaction phase is limited by the demand duration. Indeed, when $\mathfrak{g} = 1$, the average demand duration is $T = 8$ s, whereas the mean proaction time is, $Z^P = \frac{\widehat{Z}_0}{2} = 14$ s. As a consequence, the network is almost always in the proaction phase, and thus increas-

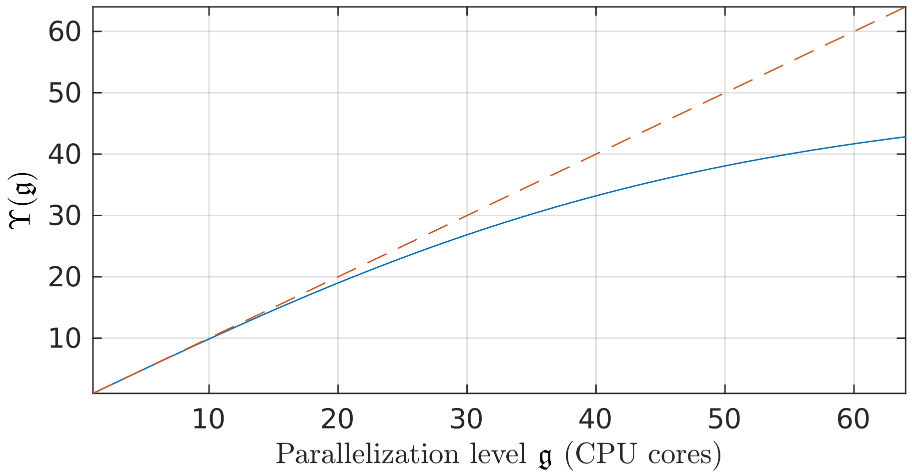


Figure 3.8.: Reduction factor of the proaction time for different parallelization levels. The dashed line has unit slope.

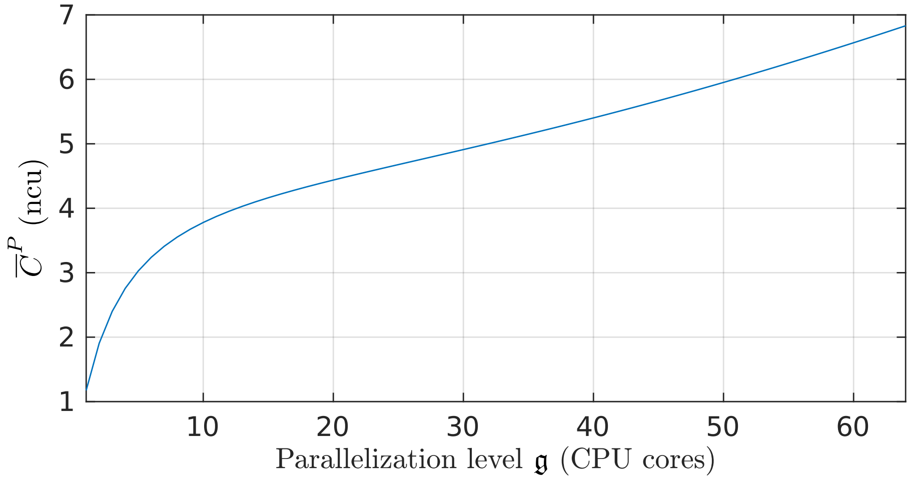


Figure 3.9.: Mean proaction cost $\overline{C}^P$ as a function of the parallelization level.

ing $\mathfrak{g}$ only increases the proaction cost without affecting the duration of the proaction phases. Nevertheless, as $\mathfrak{g}$ grows, eventually the proaction time becomes lower than the demand duration, thus allowing the network to leave the proaction phase and leading to a less steep cost increase.

Finally, in Fig. 3.10 we show the combined readiness and proaction cost for this scenario, which can be achieved via Theorems 3.4.6 and 3.4.11. We clearly observe a minimum point at $\mathfrak{g} = 7$ cores, which is thus the optimal parallelization level. Before this value the proaction cost is lower, but the network cannot cope with demand changes fast enough, resulting high readiness cost due to large state delays. For $\mathfrak{g} > 7$ the readiness approaches its minimum value but the proaction cost increases, resulting in higher combined cost.

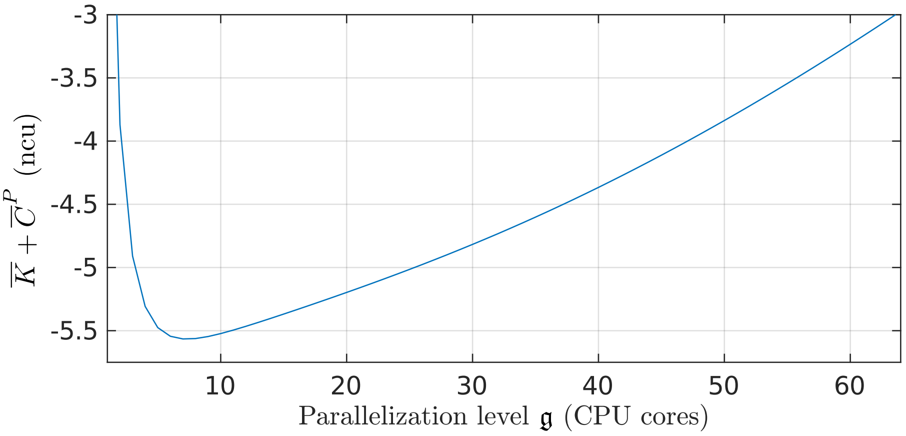


Figure 3.10.: Combined mean readiness and proaction costs $\overline{K} + \overline{C}^P$ as a function of the parallelization level.

3.6. Summary

In this chapter, we propose a comprehensive cost model for a flexible, dynamically-adapting network. We use probability-theory tools to gain a deep understanding of the multiple phases and cost components that a flexible network may feature. Namely, we identify three main adaptation phases (readiness, proaction, and reaction phases) and three corresponding cost components (readiness, proaction, and reaction cost). These cost components can be combined in order to calculate the total operating cost. In addition, we provide several expressions and relationships that can be used to accurately predict cost and take design decisions. Finally, we apply the cost model to realistic examples to show how this can be done in a generic scenario.

4. Optimal Functional Split Selection in 5G Radio Access Networks

4.1. Introduction

4.1.1. Motivation, scope, and challenges

In Chapter 2, we discuss the advantages of a centralized 5G RAN architecture, which are mainly twofold. On the one hand, function centralization enables improved function coordination, which allows the implementation of advanced techniques for mitigating interference, such as interference coordination [Sor+17], or joint transmission and reception [Jun+14]. This increases the signal quality at both the UEs and the gNodeBs, thus improving user data rates. On the other hand, centralization entails resource pooling and the use of large-scale computing servers, which reduces the operating cost of the whole network. Owing to these two advantages, we conclude that the mobile network operator should aim at the highest possible level of function centralization.

Nonetheless, centralization is usually constrained by the fronthaul network connecting CU and DUs, since the capacity required to support high centralization levels may exceed that of currently deployed networks [IT18; GS+18a]. That is, the operator may only be able to centralize a subset of processing functions, but not all of them. The selection of these centralized functions needs to be done carefully so as to improve user performance and reduce the operating cost as much as possible without surpassing the capacity limits.

However, the performance and the operating cost of a partially centralized RAN architecture depend not only on the network state as configured by the operator, but on instantaneous network conditions, such as the geographical distribution and activity of the users. Intuitively, if UEs are more concentrated or active on a certain area, we would expect that centralizing the processing functions of the gNodeBs located in that area is more beneficial that a homogeneous centralization of functions over the whole network. As a result, we need to take these instantaneous conditions into account in order to find the optimal centralization and flow vectors for any instant.

In our path to dynamically operate an adaptive 5G RAN, the goal of this chapter is to find the network state that optimizes user data rates, operating cost, or network revenue for a fixed time instant. That is, we start by considering a single “snapshot” of the RAN in which all variables (such as UE locations, channel quality, etc.) are fixed. We refer to this as *instantaneous optimization*, since the network state is selected to optimize performance for a single instant. For this, we build upon the static FSSP formulation that is presented in Sec. 2.3.2. In Chapter 6, we extend this instantaneous formulation so as to include time variation and support dynamic adaptation to changes in the network demands.

Finding the instantaneous optimal RAN state entails a series of challenges. The first one is the modeling the user performance, operating cost, and readiness cost as a function of the functional split of each gNodeB. This modeling includes the effect of interference coordination on the user throughput and the impact of resource pooling. In addition, we need to formulate an optimization problem that reflects the instantaneous distribution and activity of the UEs along with the fronthaul network constraints. Solving this problem may be also difficult, since it may involve non-linear components and problem instances may be rather large, since a complete RAN may contain hundreds of gNodeBs and thousands of UEs. In spite of this, it is not sufficient to find an approach that yields optimal or near-optimal state vectors, but this has to be achieved in a very limited time interval. Indeed, since the problem is only fully valid for a single time instant and our ultimate intention is to be able to adapt the network state dynamically, the optimization approach must converge quickly to a solution. In this chapter, all these challenges are addressed and our proposed approaches are evaluated.

4.1.2. Key contributions

In relation to the aforementioned challenges, this chapter bases on our previous works [MAJK21], [MAK21], and [MAPK21] to feature the following contributions:

1. We model a 5G RAN featuring a dynamically-adapting functional split and propose three different objective functions to find the optimal network state. The first function represents the proportionally-fair user throughput, which is estimated after modeling the effect of the functional split on the interaction between pairs of gNodeBs. The second functions reflects the operating cost associated with each centralization and flow vectors, which is based on the network model presented in Chapter 3.
2. Based on these objective functions, we formulate three main optimization problems to find the instantaneously optimal network state, building upon the generic formulation shown in Chapter 2.
3. We propose multiple reformulations and approximations to these main optimization problems, with the intention of finding an optimization approach that yields good-quality solutions in a reasonable time and can be tackled by off-the-shelf solvers.
4. We simulate a city-sized dense 5G RAN so as to evaluate the achieved spectral efficiencies, operating cost, readiness cost, and convergence times of our proposed approaches. We also compare these approaches to static architectures and show that they lead to substantial performance improvement and cost reduc-

tion.

The remainder of this chapter is structured as follows. In Section 4.2 we present the related work on this topic. Section 4.3 briefly summarizes the system model. In Section 4.4, we formulate the performance maximizing version of the functional split selection problem and derive multiple approximations to tackle it efficiently. Section 4.5 describes the derivation of the operating-cost minimizing version of the functional split selection problem. In Section 4.6 we discuss our performance-to-revenue functions and present the readiness-cost minimizing version of the functional split selection problem. Section 4.7 presents the experimental evaluation of these approaches regarding convergence time, spectral efficiency, operating cost, and readiness cost. Finally, Section 4.8 concludes the chapter.

4.2. Related work

Previous work already tackles, to some extent, the problem of statically selecting the optimal functional split of each gNB. This previous research can be divided into two categories: those dealing mainly with theoretical aspects and those focusing on the implementation. In the former category, [Mae+14] is one of the first works to propose that the functional split could be selected differently for each gNB, basing this reasoning on the limitations of the fronthaul network. The authors argue that function centralization is desirable to reduce interference and cost, but limited by the fronthaul capacity. A similar idea is developed further in [Sab+13], where a more complete framework is presented. Nevertheless, neither work proposes an actual optimization problem to find the best RAN configuration, but they both limit to describe the reasons why the functional split should not be homogeneous over a 5G RAN and describe a high-level platform supporting this feature.

Continuing with this idea, in [GS+18a] the authors formulate the problem of selecting the optimal functional split for the deployment phase. Their objective is to minimize network and computing costs while centralizing as many functions as possible. In order to estimate the required fronthaul capacity, the expected average traffic of each gNB is used. The authors of [DGA19] face a similar problem with a different objective: minimizing traffic delay. In [HR18b] the idea of dynamically changing the functional split is introduced with the intention of allocating new slices within a virtual RAN framework. Inter-cell interference reduction and fronthaul bandwidth minimization are the main objectives when selecting the functional split, although this selection is not updated once the slice is implemented. Finally, our earlier work [MAK19a] present for the first time the idea of adapting the functional split dynamically to cope with the instantaneous interference situation. Nonetheless, they tackle a simplified version of the problem and focus on confirming that the network changes slowly enough so that dynamic adaptation is possible, without providing a detailed strategy on how to select the functional split.

Regarding the implementation aspects, there are two main works that focus on realiz-

Variable	Definition
G	Number of gNodeBs
$\mathbb{G}$	Set of gNodeB indices
U	Number of UEs
$\mathbb{U}$	Set of UE indices
h_u	Index of gNodeB serving UE $u \in \mathbb{U}$
N	Number of fronthaul network nodes
$\mathbb{N}$	Set of fronthaul network nodes
n_0	Node index of CU, $n_0 \in \mathbb{N}$
n_g	Node index of gNodeB $g \in \mathbb{G}$, $n_g \in \mathbb{N}$
E	Number of fronthaul network links
$\mathbb{E}$	Set of fronthaul network links
ϑ_e	Capacity of link $e \in \mathbb{E}$
M	Number of functional split option (centralization levels)
$\nu(m)$	Required link capacity of centralization level $m \in \mathbb{M}$
$\mathbb{M}$	Set of functional split indices
$\mathbf{a}$	Vector of functional split indices for each gNodeB, $\mathbf{a} \in \mathbb{M}^G$
$\mathbf{f}$	Vector of allocation of network flows
$\mathbf{s}$	State vector of centralization and flow vectors, $\mathbf{s} \triangleq \langle \mathbf{a}, \mathbf{f} \rangle$
$\text{И}(m)$	Interference cancellation factor of centralization level $m \in \mathbb{M}$
$i_{u,g}$	Interference power received by UE $u \in \mathbb{U}$ from gNodeB $g \in \mathbb{G}$
$I_u(\mathbf{a})$	Total interference power received by UE $u \in \mathbb{U}$, $\mathbf{a} \in \mathbb{M}^G$

Table 4.1.: System modeling variables.

ing a flexible functional split. In [Cha+17b], a comprehensive description of a platform supporting multiple functional splits is presented, although the capability of changing during runtime is not included. Conversely, in [MAGVK19a] we present a pioneer framework that enables to change the functional split of a gNB without stopping its operation or dropping packets. However, the motivation to trigger such a change is not studied.

To the best of our knowledge, our work [MAJK21], on which this chapter is based, is the first work to address in detail the problem of finding the instantaneously optimal functional splits of all gNBs based on its experienced interference. In addition, [MAK21] is the first work that addresses this problem while trying to maximize the total network revenue.

4.3. System model

In Chapter 2, Sec. 2.3.1, we present a detailed description of the system model that is used throughout this thesis. For the sake of clarity, in this section we briefly summarize those concepts required to formulate the dynamic functional split selection problem. We describe the network components, explain the considered functional split, and present the adaptation framework. In Table 4.1, we show a list of the most relevant variables used in this chapter.

4.3.1. Network description

The considered network consists of G gNBs, whose operation is divided into a DU and a CU. The CUs of all gNBs are deployed in a single data center, whereas the DUs are located close to the radio equipment of the cells. As a result, there are G different DU locations and a single CU location. CUs and DUs are connected by means of a packet-switched fronthaul network that includes layer-2 or layer-3 switches. We assume that these switches are able to steer and divide the incoming flows as configured by a central controller at the CU, thus allowing for reconfigurable fractional flows. We model this fronthaul network via a directed graph $\mathcal{D} = \langle \mathbb{N}, \mathbb{E} \rangle$, where $\mathbb{N}$ is the set of network nodes (DUs, switches, and CU) and $\mathbb{E}$ is the set of network links. We denote by n_0 the node corresponding to the CU and by n_g the node corresponding to DU g, such that $g \in \mathbb{G} \triangleq \{1, ..., G\}$.

There are U simultaneously active UEs within the coverage area of all cells. Each UE $u \in \mathbb{U} \triangleq \{1, ..., U\}$ is connected to a serving gNodeB, which is denoted by h_u. Throughout this chapter, we focus on the downlink data rates and on the downlink interference as perceived by the UEs. Nonetheless, an extension of the analysis to include the uplink is straightforward. Finally, we assume that all gNodeBs operate in the same frequency bands, that is, a frequency reuse factor of 1. This is done to highlight the performance of the interference mitigation enabled by the dynamic functional split, although using a different reuse factor is not precluded.

4.3.2. Functional splits

The processing chain of every gNB can be divided into functions, which are often identified with the layer or sublayers of the RAN protocol stack [Döt+13]. For each pair of consecutive functions we define a functional split option. We denote by M the number of possible functional split options, also referred to as *centralization levels*. The instantaneous centralization level of a gNodeB g is denoted by a_g, such that $x_g \in \mathbb{M} \triangleq \{0, ..., M-1\}$. We consider that $a_g = 0$ denotes the lowest centralization level, that is, the functional split option for which the least amount of functions are centralized. Conversely, $a_g = M - 1$ denotes the highest centralization level. The *centralization vector* of all centralization levels is defined as $\mathbf{a} \triangleq [a_1 \cdots a_G]$. Low centralization levels require less fronthaul capacity, but their interference-mitigation capabilities are limited. Conversely, gNBs implementing high centralization levels are able to coordinate with each other to reduce the interference they cause to each other, at the expense of requiring higher fronthaul capacity [3GP17].

4.3.3. Interference mitigation

The mutual inter-cell interference that base stations cause to each other can be mitigated by using well-known techniques, such as coordinated scheduling, coordinated beamforming, joint transmission, etc. The application of these techniques results in improved user data rates, but they all require some level of coordination between the involved gNB functions. Owing to potentially high processing, switching, and propagation latency offered by the fronthaul network, sophisticated coordination is often deemed infeasible for distributed functions. For example, in order to employ interference cancellation, gNBs need to generate, communicate, and apply the interference cancellation algorithm to their transmission slots in less than the duration of a 5G time slot [FLF19], which can be as short as $62.5\ \mu$s when using high numerologies [3GP21e]. Consequently, in order to implement an interference-mitigation technique we need to guarantee a minimum centralization level for coordinating gNBs. For example, joint transmission also requires the centralization up to the physical layer [Zha+17], whereas coordinated scheduling only requires the centralization of the MAC layer [Nar+18].

Based on the results shown in [PM17], we model the effect of an interference-mitigation technique as a constant factor multiplying the average received interference power between two coordinating gNBs. We define $И(a)$ as the function relating centralization level a with the associated interference-mitigation factor of the most effective technique that it can support. The value of function $И(a)$ ranges from 0 (full interference cancellation) to 1 (no interference cancellation). Function $И(a)$ is a decreasing function of a, since the higher the centralization level a, the more effective the interference mitigation. Using $И(a)$, the expected total interference power I_u experienced by UE u from all gNBs can be computed as:

$$I_u(\mathbf{a}) = \sum_{g=1}^{G} i_{u,g} \cdot И(\min(a_{h_u}, a_g)), \tag{1}$$

where $i_{u,g}$ is the interference power received by UE u from gNB g and $i_{u,h_u} \triangleq 0$, as the UE is not interfered by its serving gNB. Note that the $\min(\,\cdot\,)$ operator ensures that an interference-mitigation technique can only be used by two gNBs if both of them are operating at the required centralization level.

4.3.4. Fronthaul network

The capacity required for a fronthaul link connecting the DU and CU of a gNB depends on its centralization level, that is, on its functional split. Namely, previous research has shown that high centralization levels, such as the Intra-PHY split or full centralization, require large link capacities (in the order of hundreds of Gb/s), whereas low centralization levels (such as the PDCP-RLC split) require capacities barely larger than the user data rate (in the order of a few Gb/s) [Döt+13; MAGVK19b;

3GP17]. Formally, we model the capacity required by gNB g with centralization level a_g as the function $\nu(a_g)$. For the sake of simplicity, we assume that all gNBs offer the same maximum user data rate, hence $\nu(a_g)$ does not depend explicitly on g. If required, extending $\nu(a_g)$ to include this dependency is straightforward.

Finally, we define ϑ_e as the capacity of each fronthaul link $e \in \mathbb{E}$. For each gNB g producing a downlink flow between its DU and the CU, we denote by f_e^g the fraction of this flow that is carried over link e. For notation convenience, we also define $\mathbf{f}^g \triangleq [f_1^g \cdots f_E^g]\ \forall g \in \mathbb{G}$ and $\mathbf{f} \triangleq \left[\mathbf{f}^1 \cdots \mathbf{f}^G\right]$ as the vectors of the flow generated by gNB g and all flows, respectively.

4.4. Performance-maximizing FSSP

The main objective of this chapter is to derive a fast approach to find good-quality solutions to the instantaneous functional split selection problem (FSSP), in which all problem parameters are fixed. For this, we need to decide on the objective function that we want to optimize by changing the centralization and flow vectors. In Chapter 3, we argue that a mobile operator is interested in minimizing the total cost of managing the network, which comprises readiness and action cost. Action cost is, nonetheless, only present when the network state changes, so that it is only useful for optimizing sequential adaptations over time. As a result, in this chapter we focus on minimizing the readiness cost, whereas action cost is fully taken into account in Chapter 6.

We can intuitively identify two components contributing to the instantaneous readiness cost $K(\mathbf{s})$ of any communication network resulting from operating in state s: the operating cost $K_{\text{oper}}(\mathbf{s})$ and the revenue $K_{\text{rev}}(\mathbf{s})$ associated to providing service to the users. As a result:

$$K(\mathbf{s}) = K_{\text{oper}}(\mathbf{s}) + K_{\text{rev}}(\mathbf{s}) \tag{2}$$

Revenue $K_{\text{rev}}(\mathbf{s})$ is modeled as negative cost, therefore $K_{\text{rev}}(\mathbf{s}) \leq 0\ \forall$s and $K(\mathbf{s}) < 0$ if and only if the revenue is higher than the operating cost. Note that, since we are dealing with instantaneous cost, we can drop the dependency of the readiness cost on the demand $\mathbf{d}$, as this can be interpreted as fixed problem parameters instead of input variables. Thus, in this chapter we use simply $K(\mathbf{s})$, $K_{\text{oper}}(\mathbf{s})$, and $K_{\text{rev}}(\mathbf{s})$ instead of $K(\mathbf{s}, \mathbf{d})$, $K_{\text{oper}}(\mathbf{s}, \mathbf{d})$, and $K_{\text{rev}}(\mathbf{s}, \mathbf{d})$. In general, in this chapter we focus on the influence of state s in the RAN cost and performance. In Chapter 6, we recover the explicit dependency on demand $\mathbf{d}$ to address the problem of time-dependent functional split adaptation.

The operating cost can be computed in a relatively straightforward manner, since there are already models in the literature for such a task, as we discuss in Sec. 4.5. Modeling the revenue as a function of the network state is, however, considerably more challenging, since operators rarely disclose any details regarding this internal information. Nonetheless, we can safely assume that the revenue should be an increasing function of the *user performance*, which reflect the utility that users obtain from a given collection of user throughputs. Consequently, we decouple the problem of selecting the optimal functional split into three stages, since they exhibit very distinct features, as we show in the following. First, we tackle the FSSP with the intention of maximizing this user performance, which is addressed in this section. Second, we solve the FSSP so as to minimize the operating cost, which is the matter of Sec. 4.5. Finally, we combine both approaches into a single formulation to minimize the total readiness cost, which is shown in Sec. 4.6.

4.4.1. Proportionally-fair formulation

From the point of view of user performance, we could set the objective of selecting the optimal functional split to maximizing the data rate of all users, which can be accomplished by smartly reducing interference. The data rate $\rho_u(\mathbf{a})$ achieved by UE u for a centralization vector $\mathbf{a}$ can be calculated as:

$$\rho_u(\mathbf{a}) = \mu_u \eta_u(\mathbf{a}), \tag{3}$$

where μ_u is the bandwidth allocated to UE u and $\eta_u(\mathbf{a})$ denotes its downlink spectral efficiency. We can use Shannon's formula to estimate the latter as follows [Sha49]:

$$\eta_u(\mathbf{a}) = \log_2 \left(1 + \frac{p_u}{\varsigma + I_u(\mathbf{a})} \right), \tag{4}$$

where p_u is the signal power received by UE u from its serving gNB h_u, $I_u(\mathbf{a})$ is the experienced interference as defined in (1), and ς is thermal noise power (assumed constant over all UEs). Using (4), we could formulate an optimization problem to find the centralization vector $\mathbf{a}^*$ that maximizes the sum of user data rates:

$$\mathbf{a}^* = \arg\max_{\mathbf{a} \in \mathbb{A}} \sum_{u=1}^{U} \rho_u(\mathbf{a}), \tag{5}$$

where $\mathbb{A}$ is the set of vectors $\mathbf{a}$ whose required link capacities are supported by the fronthaul network. From the operator's point of view, obtaining $\mathbf{a}^*$ as defined in (5) would lead to an efficient use of the network resources, since it guarantees working at maximum capacity. However, (5) may lead to unfair situations, since data rates of users with good signal-to-interference-and-noise ratio (SINR) may be prioritized over those with poor SINRs [SY14]. In order to prevent that, we have to maximize the sum of the individual *utilities* associated with each data rate:

$$\mathbf{a}^* = \arg\max_{\mathbf{a} \in \mathbb{A}} \sum_{u=1}^{U} \mathfrak{U}(\rho_u(\mathbf{a})), \tag{6}$$

where $\mathfrak{U}(\,\cdot\,)$ is an arbitrary rate utility function. This function should reflect some level of risk aversion, that is, small data rate increments are more valuable when the initial data rate is low than when the data rate is high, thus leading to a concave shape. This risk aversion can be also interpreted as *fairness*, since it translates into giving higher priority to those UEs facing bad interference situations.

There are several manners of achieving fairness in communication networks. Two of the more usual ones are max-min fairness, in which the minimum data rate is maximized [BGH92], and proportional fairness, which maximizes the sum of relative data rate increments with respect to the capabilities of each UE [MMD91]. In practice, max-min fairness is more strict than proportional fairness, since it may give absolute priority to those UEs featuring lower rates [SL05]. Therefore, in this work we select proportional fairness to define our rate utility function, which is indeed common in problems dealing with dynamic rate allocation, such as in time-frequency scheduling [KH05]. According to [KMT98], a rate vector $\boldsymbol{\rho} = [\rho_1 \ \cdots \ \rho_U]$ is a proportionally-fair allocation of data rates if and only if

$$\sum_{u=1}^{U} \frac{\rho'_u - \rho_u}{\rho_u} \leq 0 \tag{7}$$

for any other feasible rate vector $\boldsymbol{\rho}' = [\rho'_1 \ \cdots \ \rho'_U]$. This requirement can be easily fulfilled by using a logarithmic utility function $\mathfrak{U}(\rho) = \log(\rho)$, as it is proven in [SL05].

Thus, we define the proportionally-fair optimal centralization vector $\mathbf{a}^*$ as:

$$\begin{aligned} \mathbf{a}^* &= \arg\max_{\mathbf{a}\in\mathbb{A}} \sum_{u=1}^{U} \log\left(\rho_u(\mathbf{a})\right) && (8) \\ &= \arg\max_{\mathbf{a}\in\mathbb{A}} \sum_{u=1}^{U} \log\left(\mu_u \eta_u(\mathbf{a})\right) && (9) \\ &= \arg\max_{\mathbf{a}\in\mathbb{A}} \sum_{u=1}^{U} \log(\eta_u(\mathbf{a}))\,. && (10) \end{aligned}$$

Note that we can remove μ_u from the formulation since it becomes an adding term that does not depend on a. We refer to the problem of finding the centralization vector $\mathbf{a}^*$ as defined in (10) as the *proportionally-fair FSSP*.

The objective function in (10) is directly related to the *geometric mean* $\widetilde{\eta}(\mathbf{a})$ of the spectral efficiency over all UEs, which is defined as:

$$\widetilde{\eta}(\mathbf{a}) \triangleq \left(\prod_{u=1}^{U} \eta_u(\mathbf{a})\right)^{\frac{1}{U}} = \exp\left(\frac{1}{U}\sum_{u=1}^{U} \log\left(\eta_u(\mathbf{a})\right)\right). \tag{11}$$

Therefore, since the exponential function is monotonically increasing, an equivalent definition of the optimal, proportionally-fair centralization vector is:

$$\mathbf{a}^* = \arg\max_{\mathbf{a}\in\mathbb{A}} \widetilde{\eta}(\mathbf{a}). \tag{12}$$

This equivalence allows us to use $\widetilde{\eta}(\mathbf{a})$ as performance indicator when comparing alternative solutions using the same units as $\eta_u(\mathbf{a})$.

From (4) and the definition of $\mathbf{a}^*$ in (10), we can obtain a closed-form expression for the objective function of the FSSP. Regarding the constraints of the FSSP, it is clear that the validity of a solution $\mathbf{a}$ is limited by the topology and capacity of the fronthaul network. In other words, a solution $\mathbf{a}$ is valid if and only if there exists a vector of flows $\mathbf{f}$ that satisfies the flow requirements for every gNB, as mentioned in Sec. 2.3.1, and can be implemented on the fronthaul network without exceeding the capacity of any link. As a result, we can formulate the FSSP as follows:

$$\max_{\mathbf{a},\mathbf{f}} \sum_{u=1}^{U} \log\left(\log_2\left(1+\frac{p_u}{\varsigma+\sum_{g=1}^{G} i_{u,g}\cdot \text{И}(\min(a_{h_u},a_g))}\right)\right), \tag{P4a}$$

subject to

$$\sum_{e\in\mathbb{E}^+(n)} f_e^g - \sum_{e\in\mathbb{E}^-(n)} f_e^g = \begin{cases} 0 & \forall n\in\mathbb{N}\setminus\{n_0,n_g\} \\ \nu(a_g) & \text{for } n=n_0 \\ -\nu(a_g) & \text{for } n=n_g \end{cases} \qquad \forall g\in\mathbb{G}, \tag{P1b}$$

$$\sum_{g=1}^{G} f_e^g \leq \vartheta_e \qquad \forall e\in\mathbb{E}, \tag{P1c}$$

$$f_e^g \geq 0 \qquad \forall e\in\mathbb{E},\ \forall g\in\mathbb{G}, \tag{P1d}$$

$$\mathbf{s}\in\mathbb{M}^G. \tag{P1e}$$

where $\mathbb{E}^+(n)$ is the set of edges leaving node n, and $\mathbb{E}^-(n)$ is the set of edges entering node n. As introduced in Sec. 2.3.2, constraint (P1b) is the *flow conservation* constraint, which ensures that the flow leaving the CU and entering the DU is $\nu(a_g)$ for each gNB g. In addition, (P1c) enforces the *link capacity* constraint for each link e.

The FSSP as formulated in (P4) is a mixed integer non-linear problem (MINLP), which are, in general, NP-Hard. Moreover, the non-standard expression of the objective function (P4a) prevents the direct utilization of state-of-the-art techniques. In order to make it more tractable, we present two reformulations that simplify the problem structure at the expense of introducing additional variables.

We start with the following variable change, which allows us to replace the discrete functions $\nu(\cdot)$ and $\text{И}(\cdot)$ by 0-1 polynomial expressions:

$$b_g^m \triangleq \begin{cases} 0 & \text{if } a_g < m \\ 1 & \text{if } a_g \geq m \end{cases} \qquad \forall m\in\widehat{\mathbb{M}}, \tag{13}$$

where $\widehat{\mathbb{M}} \triangleq \{1, ..., M-1\}$ For compactness, we define

$$\mathbf{b}_g \triangleq \left[b_g^1 \; \cdots \; b_g^{M-1}\right] \qquad \forall g \in \mathbb{G} \tag{14}$$

and

$$\mathbf{b} \triangleq \left[\mathbf{b}_1 \; \cdots \; \mathbf{b}_G\right]. \tag{15}$$

For example, we can use (13) to convert $a_g = 2$ into $\mathbf{b}_g = [1, 1, 0]$ when $M = 4$. There is a bijection between $\mathbf{b}_g$ and a_g $\forall g \in \mathbb{G}$, and thus a bijection between $\mathbf{b}$ and $\mathbf{a}$, by means of the following reciprocal conversion:

$$a_g = \sum_{m=1}^{M-1} b_g^m. \tag{16}$$

As a result, $\mathbf{b}$ is an alternative, equivalent representation of centralization vector $\mathbf{a}$, which can be also expressed as:

$$\mathbf{a} = \mathbf{b} \cdot (\mathbf{I}_G \otimes \mathbf{1}_{M-1}), \tag{17}$$

where $\mathbf{I}_g$ is the identity matrix of size g, $\mathbf{1}_m$ is an all-ones column vector of size m, and $\otimes$ is the Kronecker product. The purpose of this variable change is to convert the integer variables a_g into binary variables in a particularly useful fashion. In fact, (13) is not the conventional manner of performing an integer-to-binary conversion in integer programming, which usually consists in the binary representation of the numbers from 0 to M and thus requires $\lceil \log_2(M) \rceil$ new variables per original variable. Instead, conversion (13) requires $M-1$ new variables, but this increment in the number of additional variables is very limited (since $M \leq 8$ in real deployments [Döt+13]), and it is compensated by its useful implications. Namely, since it follows that $b_g^m \geq b_g^{m'}$ if and only if $m \leq m'$, the minimum operator between a variables can be converted into a polynomial function of y variables:

$$\min(a_g, a_k) = \sum_{m=1}^{M-1} b_g^m b_k^m. \tag{18}$$

This property can be exploited to rewrite the integer non-polynomial function $И(\min(a_g, a_k))$ as a 0-1 polynomial:

$$И(\min(a_g, a_k)) = И(1) - \sum_{m=1}^{M-1} \varrho(m) b_g^m b_k^m, \tag{19}$$

where $\varrho(m) \triangleq И(m-1) - И(m)$. As a result, we can reformulate the FSSP as:

$$\max_{\mathbf{b},\mathbf{f}} \sum_{u=1}^{U} \log\left(\log_2\left(1 + \frac{p_u}{\varsigma + I_u - \sum_{g=1}^{G} i_{u,g}\left(\sum_{m=1}^{M-1} \varrho(m) b_g^m b_{h_u}^m\right)}\right)\right), \tag{P5a}$$

subject to

$$\sum_{e\in\mathbb{E}^+(n)} f_e^g - \sum_{e\in\mathbb{E}^-(n)} f_e^g = \begin{cases} 0 & \forall n \in \mathbb{N}\backslash\{n_0, n_g\} \\ \widehat{\nu}(\mathbf{b}_g) & \text{for } n = n_0 \\ -\widehat{\nu}(\mathbf{b}_g) & \text{for } n = n_g \end{cases} \qquad \forall g \in \mathbb{G}, \tag{P5b}$$

$$b_g^1 \geq b_g^2 \geq \ldots \geq b_g^{M-1} \qquad \forall g \in \mathbb{G}, \tag{P5c}$$

$$\mathbf{b} \in \{0,1\}^G, \tag{P5d}$$

(P1c), and (P1d),

where $I_u = \sum_{g=1}^{G} \text{И}(1) i_{u,g}$ is the interference power received by UE u when the lowest centralization level is in operation on its serving gNodeB, and function $\widehat{\nu}(\,\cdot\,)$ is defined as follows:

$$\widehat{\nu}(\mathbf{b}_g) \triangleq \nu(1) - \sum_{m=1}^{M-1} \left(\nu(m-1) - \nu(m)\right) b_g^m, \tag{20}$$

such that $\widehat{\nu}(\mathbf{b}_g) = \nu(a_g)$.

Formulation (P5) replaces the integer variables and discrete functions $\nu(\,\cdot\,)$ and $\text{И}(\,\cdot\,)$ of (P4) by binary variables and polynomial functions. As a result, linearization techniques can be now applied to improve the tractability of the FSSP. Namely, the product of two b_g^m variables can be linearized via the following variable change:

$$v_{g,k}^m \triangleq b_g^m b_k^m, \tag{21}$$

which can be enforced with additional linear inequalities [GW74]. This leads to the following reformulation:

$$\max_{\mathbf{b},\mathbf{v},\mathbf{f}} \sum_{u=1}^{U} \log\left(\log_2\left(1 + \frac{p_u}{\varsigma + I_u - \sum_{g=1}^{G} i_{u,g}\left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^m\right)}\right)\right) \tag{P6a}$$

subject to

$$2v_{g,k}^m \leq b_g^m + b_k^m \qquad \forall m \in \widehat{\mathbb{M}},\ \forall k \in \mathbb{G},\ \forall g < k,\ g \in \mathbb{G}, \tag{P6b}$$

$$1 + v_{g,k}^m \geq b_g^m + b_k^m \qquad \forall m \in \widehat{\mathbb{M}},\ \forall k \in \mathbb{G},\ \forall g < k,\ g \in \mathbb{G}, \tag{P6c}$$

$$\mathbf{v} \in \{0,1\}^{(M-1)\binom{G}{2}}, \tag{P6d}$$

(P1c), (P1d) and (P5b) – (P5d),

where

$$\mathbf{v} \triangleq \left[v_{g,k}^m\right] \quad \forall m \in \widehat{\mathbb{M}},\ \forall k \in \mathbb{G},\ \forall g < k,\ g \in \mathbb{G}. \tag{22}$$

Note that constraints (P6b)–(P6d) enforce (21), so that we can replace $\mathbf{b}$, and thus the centralization vector $\mathbf{a}$, with $\mathbf{v}$. The number of additional $\mathbf{v}$ variables is $(M-1)\binom{G}{2} = O(G^2)$, as one additional variable is required for every pair of gNBs and consecutive splits. In Section 4.4.2, we exploit the characteristics of the network to reduce the number of these additional variables.

Formulation (P6) is still a MINLP, but its simpler objective function admits further analysis. Indeed, we observe that the continuous relaxation of the objective function is convex on $\mathbf{v}$, but since the FSSP is a maximization problem, this implies that we are in the realm of concave optimization. Thus, there may be multiple local maxima, making the problem hard to tackle. It is still possible to use exact global optimization techniques for concave MINLPs, such as those presented in [Hor86], but these mainly consist on applying branch-and-bound or branch-and-cut algorithms, whose convergence time may be high. Since the FSSP is a real-time problem, we opt instead for deriving increasingly simpler approximations to the original FSSP, until a suitable approach within the speed-quality trade-off is found.

4.4.2. Fractional approximations

The main obstacles when tackling formulations (P4)–(P6) are the logarithmic functions, which prevent the application of simplifying reformulations. Fortunately, we can exploit the slow growth rate of these functions, as its combination can be very well approximated by a rational function:

$$\log(\log_2(1+\sigma)) \approx \frac{\aleph_{\text{rat}}}{\beth_{\text{rat}}+\sigma} + \gimel_{\text{rat}}. \tag{23}$$

Parameters $\aleph_{\text{rat}}$, $\beth_{\text{rat}}$ and $\gimel_{\text{rat}}$ can be obtained from rational fitting within the desired interval. In our case, we choose the interval $0.1 \leq \sigma \leq 100$, that is, an SINR ranging from -10 dB to 20 dB. After applying (23) to (P6) and simplifying the expression, we obtain the following reformulation:

$$\operatorname*{arg\,max}_{\mathbf{b},\mathbf{v},\mathbf{f}} \sum_{u=1}^{U} \log\left(\log_2\left(1+\frac{p_u}{\varsigma + I_u - \sum_{g=1}^{G} i_{u,g}\left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)}\right)\right) \tag{P6a}$$

$$\approx \operatorname*{arg\,max}_{\mathbf{b},\mathbf{v},\mathbf{f}} \sum_{u=1}^{U} \frac{\aleph_{\text{rat}}}{\beth_{\text{rat}} + \frac{p_u}{\varsigma+I_u-\sum_{g=1}^{G} i_{u,g}\left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)}} + \gimel_{\text{rat}}. \tag{24}$$

$$= \operatorname*{arg\,max}_{\mathbf{b},\mathbf{v},\mathbf{f}} \sum_{u=1}^{U} \frac{\aleph_{\text{rat}}\left(\varsigma + I_u - \sum_{g=1}^{G} i_{u,g}\left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)\right)}{\beth_{\text{rat}}\left(\varsigma + I_u\right) + p_u - \beth_{\text{rat}} \sum_{g=1}^{G} i_{u,g}\left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)} \tag{25}$$

$$= \operatorname*{arg\,max}_{\mathbf{b},\mathbf{v},\mathbf{f}} \sum_{u=1}^{U} \frac{\beth_{\text{rat}}\left(\varsigma + I_u - \sum_{g=1}^{G} i_{u,g}\left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)\right) + p_u - p_u}{\beth_{\text{rat}}\left(\varsigma + I_u\right) + p_u - \beth_{\text{rat}} \sum_{g=1}^{G} i_{u,g}\left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)} \tag{26}$$

$$= \arg\min_{\mathbf{b},\mathbf{v},\mathbf{f}} \sum_{u=1}^{U} \frac{p_u}{\beth_{\text{rat}}\left(\varsigma + I_u\right) + p_u - \beth_{\text{rat}} \sum_{g=1}^{G} i_{u,g} \left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)} \tag{27}$$

Note that in (26), constant $\aleph_{\text{rat}}$ is replaced with $\beth_{\text{rat}}$ so as to eventually remove the polynomial from the numerator, since this replacement does not affect the location of the optimal point. As a result, we formulate the *fractional* FSSP as an approximation to the original proportionally-fair FSSP as:

$$\min_{\mathbf{b},\mathbf{v},\mathbf{f}} \sum_{u=1}^{U} \frac{p_u}{\beth_{\text{rat}}\left(\varsigma + I_u\right) + p_u - \beth_{\text{rat}} \sum_{g=1}^{G} i_{u,g} \left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)} \tag{P7a}$$

subject to

$$\text{(P1c)},\ \text{(P1d)},\ \text{(P5b)} - \text{(P5d)},\ \text{and (P6b)} - \text{(P6d)}.$$

Problem (P7) is now a multiple-ratio fractional mixed-integer optimization problem, which can be tackled with state-of-the-art techniques. Indeed, since the continuous variables f do not appear on the objective function, we can directly apply existing techniques to reformulate it into an MILP. We present two such techniques: the Li-Wu-Tawarmalani transformation and the Borrero-Gillen-Prokopyev transformation.

Li-Wu-Tawarmalani transformation

The Li-Wu-Tawarmalani transformation (in short, LWT transformation) reformulates a 0-1 multiple-ratio fractional program into an MILP by introducing continuous variables via the following variable changes [Li94; Wu97; TAS02]:

$$w_u \triangleq \frac{1}{\pi_u - \beth_{\text{rat}} \sum_{g=1}^{G} i_{u,g} \left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)} \qquad \forall u \in \mathbb{U}, \tag{28}$$

$$x_{u,g}^{m} \triangleq w_u v_{g,h_u}^{m} \qquad \forall m \in \widehat{\mathbb{M}},\ \forall u \in \mathbb{U},\ \forall g \in \mathbb{G}, \tag{29}$$

where $\pi_u \triangleq \beth_{\text{rat}}\left(\varsigma + I_u\right) + p_u$. For compactness, we define the following vectors to contain the new LWT variables:

$$\mathbf{w} \triangleq \left[w_u\right] \qquad \forall u \in \mathbb{U}, \tag{30}$$

$$\mathbf{x} \triangleq \left[x_{u,g}^{m}\right] \quad \forall m \in \widehat{\mathbb{M}}. \tag{31}$$

Identities (28) and (29) can be enforced by additional constraints, resulting in the following equivalent formulation:

$$\min_{\mathbf{b},\mathbf{v},\mathbf{w},\mathbf{x},\mathbf{f}} \sum_{u=1}^{U} p_u w_u \tag{P8a}$$

subject to

$$\begin{aligned}
&\pi_u w_u - \daleth_{\text{rat}} \sum_{g=1}^{G} i_{u,g} \left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^m \right) = 1 && \forall u \in \mathbb{U}, && \text{(P8b)}\\
&W_u^- v_{g,h_u}^m \le x_{u,g}^m \le W_u^+ v_{g,h_u}^m && \forall m \in \widehat{\mathbb{M}},\ \forall g \in \mathbb{G},\ \forall u \in \mathbb{U}, && \text{(P8c)}\\
&x_{u,g}^m \le w_u + W_u^- \left(v_{g,h_u}^m - 1 \right) && \forall m \in \widehat{\mathbb{M}},\ \forall g \in \mathbb{G},\ \forall u \in \mathbb{U}, && \text{(P8d)}\\
&x_{u,g}^m \ge w_u + W_u^+ \left(v_{g,h_u}^m - 1 \right) && \forall m \in \widehat{\mathbb{M}},\ \forall g \in \mathbb{G},\ \forall u \in \mathbb{U}, && \text{(P8e)}\\
&\text{(P1c), (P1d), (P5b) – (P5d), and (P6b) – (P6d),}
\end{aligned}$$

where W_u^- and W_u^+ are lower and upper bounds for w_u, respectively, which can be obtained straightforwardly by setting the $\mathbf{v}$ variables in (28) to all zeros and all ones:

$$\begin{aligned}
W_u^- &= \frac{1}{\pi_u} && \forall u \in \mathbb{U}, && (32)\\
W_u^+ &= \frac{1}{\pi_u - \daleth_{\text{rat}} \sum_{g=1}^{G} i_{u,g} \left(\sum_{m=1}^{M-1} \varrho(m) \right)} && \forall u \in \mathbb{U}. && (33)
\end{aligned}$$

Formulation (P8) is an MILP that requires $U + 4(M-1)GU$ new constraints, U additional $\mathbf{w}$ variables and $(M-1)(G-1)U$ additional $\mathbf{x}$ variables with respect to (P7). Note that, in the general case, the number of required $\mathbf{x}$ variables would be $(M-1)\binom{G}{2}U$ as these variables originate from the product of $\mathbf{v}$ and $\mathbf{w}$ variables. However, in our case it is clear that the interference coefficient corresponding to a triple $\langle u, g, k \rangle$, $u \in \mathbb{U}$, $g, k \in \mathbb{G}$ is zero unless $g = h_u$ or $k = h_u$ and $g \neq k$, hence we can remove the variables indexed by those triples. As a result, the number of variables of this reformulation grows with $\mathcal{O}(G^2)$, assuming that U scales linearly with G [3GP20b], instead of $\mathcal{O}(G^3)$.

Borrero-Gillen-Prokopyev transformation

The Borrero-Gillen-Prokopyev transformation (BGP transformation) is a recent improvement on the LWT transformation, which aims at reducing the number of required variables and constraints by approximating all coefficients in the objective function with integers [BGP16]. This is accomplished by introducing the three new sets of variables with respect to (P7). In order to avoid notational clutter, we reuse and redefine variables $\mathbf{w}$ and $\mathbf{x}$, which are firstly defined for the LWT transformation, since BGP and LWT transformations are independent approaches that cannot be combined.

Variables $\mathbf{w} \triangleq [w_u]\ \forall u \in \mathbb{U}$ are redefined as:

$$w_u \triangleq \frac{\eth}{o_{u,0} + \sum_{g=1}^{G} \sum_{m=1}^{M} o_{u,g}^{q} v_{g,h_u}^{m}} \qquad \forall u \in \mathbb{U}, \tag{34}$$

where

$$\begin{aligned} o_{u,0} &= \eth + \left\lfloor \frac{\eth \beth_{\text{rat}}(\varsigma + I_u)}{p_u} \right\rceil, \\ o_{u,g}^{q} &= - \left\lfloor \frac{\beth_{\text{rat}} \eth i_{u,g} \varrho(m)}{p_u} \right\rceil, \end{aligned} \tag{35}$$

$\eth$ is a constant factor used to scale the integer coefficients into the desired range, and $\lfloor \cdot \rceil$ represents the rounding operation to the nearest integer. In addition, new binary variables $\mathbf{x} \triangleq [x_{u,l}]$, $x_{u,l} \in \{0,1\}$ are defined via the following identity:

$$\sum_{l=1}^{L_u} 2^{l-1} x_{u,l} = O_u + \sum_{g=1}^{G} \sum_{m=1}^{M} o_{u,g}^{q} v_{g,h_u}^{m} \quad \forall u \in \mathbb{U}, \forall l \in \{1, ..., L_u\}, \tag{36}$$

where

$$O_u \triangleq - \sum_{g=1}^{G} \sum_{m=1}^{M} o_{u,g}^{q} \tag{37}$$

and

$$L_u \triangleq \lfloor \log_2 (O_u) \rfloor + 1. \tag{38}$$

Finally, new variables $\mathbf{y} \triangleq [y_{u,l}]$ are defined as:

$$y_{u,l} \triangleq x_{u,l} w_u \qquad \forall u \in \mathbb{U}, \forall l \in \{1, ..., L_u\}. \tag{39}$$

The resulting reformulation, including additional constraints to enforce (34)–(39), is the following [BGP16]:

$$\min_{\mathbf{b},\mathbf{v},\mathbf{w},\mathbf{x},\mathbf{y},\mathbf{f}} \sum_{u=1}^{U} w_u \tag{P9a}$$

subject to

$$(o_{u0} - O_u) w_u + \sum_{l=1}^{L_u} 2^{l-1} x_{u,l} = -\eth \qquad \forall u \in \mathbb{U}, \tag{P9b}$$

$$\sum_{g=1}^{G} \sum_{m=1}^{M} o_{u,g} v_{g,h_u}^{m} - \sum_{l=1}^{L_u} 2^{l-1} x_{u,l} = -O_u \qquad \forall u \in \mathbb{U}, \tag{P9c}$$

$$W_u^{-} x_{u,l} \leq y_{u,l} \leq W_u^{+} x_{u,l} \qquad \forall u \in \mathbb{U},\ \forall l \in \{1, ..., L_u\}, \tag{P9d}$$

$$
\begin{aligned}
& y_{u,l} - w_u \leq W_u^{-} x_{u,l} - W_u^{-} && \forall u \in \mathbb{U},\ \forall l \in \{1, ..., L_u\}, \qquad \text{(P9e)} \\
& y_{u,l} - w_u \geq W_u^{+} x_{u,l} - W_u^{+} && \forall u \in \mathbb{U},\ \forall l \in \{1, ..., L_u\}, \qquad \text{(P9f)} \\
& \text{(P1c)},\ \text{(P1d)},\ \text{(P5b)} - \text{(P5d)},\ \text{and (P6b)} - \text{(P6d)},
\end{aligned}
$$

where W_u^- and W_u^+ are lower and upper bounds to w_u, respectively, whose value can be computed as:

$$W_u^- = \frac{-\eth}{o_{u,0}} \tag{40}$$

$$W_u^+ = \frac{-\eth}{o_{u,0} - O_u}. \tag{41}$$

Formulation (P9) is an MILP that requires $2U+4\sum_{u\in\mathbb{U}} L_u$ new constraints, $U+\sum_{u\in\mathbb{U}} L_u$ additional continuous variables (vectors $\mathbf{w}$ and $\mathbf{y}$) and $\sum_{u\in\mathbb{U}} L_u$ additional binary variables (vector $\mathbf{x}$) with respect to (P7). Since $\sum_{u\in\mathbb{U}} L_u = \mathcal{O}(U \log(G))$ [BGP16] and assuming again $U = \mathcal{O}(G)$, this implies that the number of additional variables and constraints compared to (P7) grows with $\mathcal{O}(G \log(G))$, at the expense of losing accuracy in the problem coefficients. Nonetheless, the overall size of the problem instances still grows with $\mathcal{O}(G^2)$, due to the presence of variables $\mathbf{v}$.

Punctured transformations

The fractional reformulation (P7) relies on the addition of $\mathbf{v}$ variables to be tractable by the LWT and the BGP transformations. These variables replace the product of $\mathbf{b}$ variables by single binary variables, which eventually enables these MILP reformulations. As there must be a $v_{g,k}$ variable for each pair of gNBs $\langle g, k \rangle$, their number grows quadratically with the number of gNBs G.

However, in our problem not every pair of gNBs is worth considering. The interference between two gNBs that are far apart is negligible, so any variable modeling it contributes little to the overall solution. Knowing this fact, we can remove unnecessary variables so that the problem size is reduced without noticeably affecting the optimal solution. To do so, we define $\hat{\imath}_{g,k}$ as the combined interference caused by gNBs g and k:

$$\hat{\imath}_{g,k} = \sum_{u\in\mathbb{H}_g} i_{u,k} + \sum_{u\in\mathbb{H}_k} i_{u,g} \tag{42}$$

where

$$\mathbb{H}_g = \{u \in \mathbb{U} \mid h_u = g,\ h_u, g \in \mathbb{G}\} \tag{43}$$

is the set of the UE indices served by gNB g. Now we sort coefficients $\hat{\imath}_{g,k}$ and remove those gNB pairs $\langle g, k \rangle$ whose combined interference is below a configurable threshold. For instante, in our experiments, we remove those gNB pairs with the smallest $\hat{\imath}_{g,k}$ such that their addition contributes less than 5% to the total interference. This removes the related $v_{g,k}$ variables and all additional variables and constraints that are defined from them, hence simplifying the problem. Since removing these variables may impact the performance of the obtained solutions, we evaluate them separately for the LWT and BGP transformations, and refer to them as *punctured* LWT and BGP transformations, respectively.

4.4.3. Quadratic formulation

Instead of the approximation shown in (23), we can consider a simpler fractional approximation of the $\log(\log_2(1+\sigma))$ function:

$$\log(\log_2(1+\sigma)) \approx \aleph_{\text{sim}} - \frac{\beth_{\text{sim}}}{\sigma}. \tag{44}$$

This approximation is less tight than (23), but in return it produces a much simpler problem formulation. Indeed, after combining (44) with (P5), we arrive at the following equivalent problem:

$$\max_{\mathbf{b},\mathbf{f}} \sum_{u=1}^{U} \sum_{g=1}^{G} \frac{i_{u,g}}{p_u} \left(\sum_{m=1}^{M-1} \varrho(m) b_g^m b_{h_u}^m \right), \tag{P10a}$$

subject to

$$\text{(P1c)},\ \text{(P1d) and (P5b)} - \text{(P5d)},$$

As this reformulation is a quadratic integer problem (QIP), we refer to it as the *quadratic formulation*. Its standard form enables the use of off-the-shelf solvers to tackle it. Nonetheless, its structure can be further exploited to reformulate it into a simple MILP. For this, we first introduce these new coefficients:

$$\ell_{g,k}^m \triangleq \begin{cases} \varrho(m) \left(\sum\limits_{u \in \mathbb{H}_g} \frac{i_{u,k}}{p_u} + \sum\limits_{u \in \mathbb{H}_k} \frac{i_{u,g}}{p_u} \right) & \text{if } g \neq k, \\ 0 & \text{if } g = k, \end{cases} \qquad \forall m \in \widehat{\mathbb{M}},\ \forall g,k \in \mathbb{G} \tag{45}$$

These coefficients allow for the following alternative reformulation of (P10):

$$\max_{\mathbf{b},\mathbf{f}} \sum_{m=1}^{M} \sum_{g=1}^{G} b_g^q \left(\sum_{k=1}^{G} \ell_{g,k}^m b_k^m \right), \tag{P11a}$$

subject to

$$\text{(P1c)},\ \text{(P1d) and (P5b)} - \text{(P5d)}.$$

Finally, we define a new set of variables

$$\mathbf{r} \triangleq \left[r_g^m\right] \quad \forall g \in \mathbb{G},\ \forall m \in \widehat{\mathbb{M}}, \tag{46}$$

by means of the following variable change:

$$r_g^m \triangleq y_g^m \left(\sum_{k=1}^{G} \ell_{g,k}^m y_k^m\right) \qquad \forall m \in \widehat{\mathbb{M}}, g \in \mathbb{G}, \tag{47}$$

which can be enforced into our optimization problem by adding three new sets of constraints, as follows [Glo75]:

$$\max_{\mathbf{b},\mathbf{r},\mathbf{f}} \sum_{m=1}^{M-1} \sum_{g=1}^{G} r_g^m \tag{P12a}$$

subject to

$$0 \leq r_g^m \leq R_g^m b_g^m \qquad \forall m \in \widehat{\mathbb{M}}, \forall g \in \mathbb{G}, \tag{P12b}$$

$$r_g^m \geq \sum_{k=1}^{G} \ell_{g,k}^m b_g^m - (1 - b_g^m) R_g^m \qquad \forall m \in \widehat{\mathbb{M}}, \forall g \in \mathbb{G}, \tag{P12c}$$

$$r_g^m \leq \sum_{k} \ell_{g,k}^m b_g^m \qquad \forall m \in \widehat{\mathbb{M}}, \forall g \in \mathbb{G}, \tag{P12d}$$

$$\text{(P1c)},\ \text{(P1d)},\ \text{and (P5b)} - \text{(P5d)}.$$

where $R_g^m \triangleq \sum_{k=1}^{G} \ell_{g,k}^m$.

Formulation (P12) requires $(M-1)G$ additional variables and $4GM$ additional constraints with respect to formulation (P5). As a result, problem instances grow with $\mathcal{O}(G)$, leading to substantially smaller problems when compared to the previous MILP formulations, which grow with $\mathcal{O}(G^2)$. The drawback of this approach is that the approximation (44) is less tight than (23), which may impact the quality of the solutions.

4.4.4. Heuristic approaches

The previous reformulations approximate the proportionally-fair FSSP into MILPs, which can be tackled by dedicated off-the-shelf software. Nonetheless MILPs are still NP-Hard, thus exact techniques to solve them may be slow, exhibiting exponential worst-case performance. This is actually shown in Sec. 4.7.2, along with other related measurements. Since we are interested in solving the problem as fast as possible so as to dynamically adapt to changes in the interference situation, we propose two heuristics with the intention of finding good-quality solutions with low convergence time. The first technique exploits the correlation between the interference caused by a gNB and its centralization level. The second technique is a local-search approach to improve the solutions of the quadratic reformulation.

Heuristic 1

As the objective of centralizing gNBs is to mitigate interference, it intuitively follows that, given an optimal solution $\mathbf{a}^*$, the gNBs with the highest centralization levels may tend to be those causing the most interference. From this, we can derive a heuristic rule that assigns the centralization level of a gNB g based on the total interference $\widehat{I}_g$ that it causes to all UEs in the absence of interference-mitigation:

$$\widehat{I}_g \triangleq \sum_{u=1}^{U} i_{u,g}. \tag{48}$$

Such a heuristic must not only produce a solution that is proportional to $\widehat{I}_g$ $\forall g \in \mathbb{G}$, but it must also satisfy constraints (P1b)–(P1d) and be as centralized as possible.

We propose the following method to accomplish these objectives. First, we define the *accumulated centralization level* $\widehat{A}$ given a (possibly infeasible) solution $\mathbf{a}$ as:

$$\widehat{A} \triangleq \sum_{g=1}^{G} a_g. \tag{49}$$

An upper bound $\widehat{A}^+ \geq \widehat{A}$ for all feasible $\mathbf{a}$ can be obtained by solving the following MILP, which maximizes the accumulated centralization level without taking into account the resulting cost or performance:

$$\widehat{A}^+ = \max_{\mathbf{a},\mathbf{f}} \sum_{g=1}^{G} a_g \tag{P13a}$$

subject to

$$\text{(P1b)} - \text{(P1d)}.$$

Note that (P13a) only depends on the fronthaul network configuration via constraints (P1b)–(P1d), therefore it can be solved offline before the network is put into operation. In addition, a lower bound $\widehat{A}^- \leq \widehat{A}$ for all feasible $\mathbf{a}$ can be also easily found, for example as:

$$\widehat{A}^- = \left\lfloor \frac{\min_{e\in\mathbb{E}}(\vartheta_e)}{И(M)} \right\rfloor, \tag{50}$$

which is the maximum number of fully-centralized gNBs that can be supported by the weakest link.

Now, given an initial guess of $\widehat{A}$ within these bounds, we need a method to assign a value to each a_g $\forall g \in \mathbb{G}$ in accordance to the values of $\widehat{I}_g$. We use the Webster/Sainte-

Laguë method (WSL method) to do this, an algorithm originally designed to proportionally allocate seats in party-list voting systems [Gal91]. This method is selected since it preserves the proportionality of the original interference levels better than alternative approaches [Sch+03], yet it is simple to implement. In a nutshell, the WSL method starts from $\mathbf{a} = \mathbf{0}$, where $\mathbf{0}$ is an all-zeros vector, finds the index

$$g^* = \arg\max_{g \in \mathbb{G}} \widehat{I}_g \tag{51}$$

belonging to the gNodeB the maximum interference $\widehat{I}_g$, increments a_g by 1 to increase its centralization level, updates $\widehat{I}_g$ to $\frac{I_g}{2a_g+1}$ (or to $-\infty$ if $a_g = M$) and repeats the process until the desired value of $\widehat{A}$ is achieved.

After running the WSL method, we have a candidate centralization vector $\mathbf{a}$ for the desired $\widehat{A}$. At this point, we can solve the following minimization problem to find the corresponding flow vector $\mathbf{f}$:

$$\min_{\mathbf{f}} \sum_{g=1}^{G} \sum_{e=1}^{E} f_e^g \tag{P14a}$$

subject to

$$\text{(P1b)} - \text{(P1d)}.$$

Note that vector $\mathbf{a}$ is not present in the objective function, but only in constraint (P1b). This problem is a linear program (LP) with E variables, and thus it can be tackled very efficiently by modern solvers. However, it may happen that our initial guess of $\widehat{A}$ yields a vector $\mathbf{a}$ such that (P14) is infeasible. In that case, we need to find a different value of $\widehat{A}$ and try again until a feasible value of $\widehat{A}$ is found. This process can be performed efficiently by exploiting the properties of our objective function and the WSL method via a binary search of the largest feasible $\widehat{A}$.

As $\mathit{И}(\cdot)$ is a monotonically decreasing function, it can be trivially proven that $\widetilde{\eta}(\mathbf{a}) \geq \widetilde{\eta}(\mathbf{a}')$ whenever $a_g \geq a'_g \; \forall g \in \mathbb{G}$, and vice-versa. In words, this means that increasing (decreasing) the centralization level of any gNB can only increase (decrease), on average, the mean spectral efficiency, as intuitively expected. Similarly, given constant $\widehat{I}_g$ $\forall g \in \mathbb{G}$ and two accumulated centralization levels $\widehat{A}$ and $\widehat{A}'$ such that $\widehat{A} > \widehat{A}'$, it can be easily shown that the resulting centralization vectors $\mathbf{a}$ and $\mathbf{a}'$ yielded by the WSL algorithm fulfill $a_g \geq a'_g \; \forall g \in \mathbb{G}$. Finally, it is also clear that if $\mathbf{a}$ is feasible, then $\mathbf{a}'$ must be as well. We conclude that there is a single maximum value of $\widehat{A}$ such that for all $\widehat{A}' < \widehat{A}$ the WSL method returns feasible but lower-performance solutions, and for all $\widehat{A}'' > \widehat{A}$ the WSL method returns infeasible solutions. In the light of the above, we can implement a binary search of the highest feasible value of $\widehat{A}$.

The summarized operation of Heuristic 1 is shown in Algorithm 1. Since this algorithm deals with the centralization vector $\mathbf{a}$ and the flow vector $\mathbf{f}$ directly, its complexity grows with $\mathcal{O}(G)$.

Algorithm 1: Heuristic 1 (Webster/Saint-Laguë method and binary search).

```
Input: $\widehat{A}^{+},\ \widehat{A}^{-},\ \widehat{I}_g\ \forall g \in \mathbb{G}$
Output: $\mathbf{a}, \mathbf{f}$
$a_g \leftarrow 0 \quad \forall g \in \mathbb{G}$
$\widehat{A} \leftarrow \widehat{A}^{+}$
repeat                                              // Binary search
    repeat                                          // WSL assignment
        $g^* \leftarrow \arg\max_g \{\widehat{I}_g \,|\, g \in \mathbb{G}\}$
        $a_{g^*} \leftarrow a_{g^*} + 1$
        if $a_{g^*} < M$ then
            $I_{g^*} \leftarrow \frac{I_{g^*}}{2x_{g^*}+1}$
        else
            $I_{g^*} \leftarrow -\infty$
        end
    $\sum_{g=1}^{G} a_g = \widehat{A}$
    if (P14) *feasible* then                         // Feasibility check
        $\widehat{A}^{-} \leftarrow \widehat{A}$
        $\mathbf{f} \leftarrow \mathbf{f}^*(\mathbf{a})$ as in (P14).
    else
        $\widehat{A}^{+} \leftarrow \widehat{A} - 1$
    end
$\widehat{A}^{+} = \widehat{A}^{-}$
```

Heuristic 2

This heuristic exploits the properties of the FSSP by using local search so as to improve solutions provided by a previous algorithm. We start with an initial solution $\mathrm{s} = \langle \mathbf{a}, \mathbf{f} \rangle$ provided by the quadratic approach (P12) and we compute the following parameters from it:

$$\overline{I}_m = \frac{1}{|\mathbb{G}_m|} \sum_{g \in \mathbb{G}_m} \widehat{I}_g \tag{52}$$

where $\mathbb{G}_m = \{g \in \mathbb{G} \,|\, a_g = m\}\ \forall m \in \mathbb{M}$. The value of $\overline{I}_m$ is the average interference power caused by those gNBs whose centralization level is m. We then calculate deviations $\Delta\widehat{I}_g = \widehat{I}_g - \overline{I}_{a_g}\ \forall g \in \mathbb{G}$, which represent how far the total interference caused by gNB g is from the average value among those with the same centralization level. Based on these deviations, we can identify two types of gNBs: those producing relatively high interference (which may benefit from a higher centralization level) and those producing relatively low interference (which might accept a lower centralization). Consequently, we select pairs of gNBs $[k, k'] \in \{1, ..., M-1\} \times \{2, ..., M\}$ such that k belongs to the former type and k' belongs to the latter, and generate a new candidate solution $\mathbf{a}'$ whose elements are as follows:

$$a'_g = \begin{cases} a_g + 1 & \text{if } g = k, \\ a_g - 1 & \text{if } g = k', \\ a_g & \text{otherwise.} \end{cases} \tag{53}$$

Then, the mean spectral efficiency $\widetilde{\eta}(\mathbf{a}')$ of the solution is evaluated. If $\widetilde{\eta}(\mathbf{a}') > \widetilde{\eta}(\mathbf{a})$, the feasibility of $\mathbf{a}'$ is evaluated with (P14). If $\mathbf{a}'$ is both feasible and its spectral efficiency is higher than that of $\mathbf{a}$, it is taken as new initial solution and the procedure repeats until no better solution is found.

4.5. Operating-cost-minimizing FSSP

In this section, we formulate the FSSP that focuses on minimizing the cost of operating the network in a given state, regardless of the performance experienced by the UEs. This problem is, thus, the counterpart to the performance-maximizing FSSP, both of which are eventually combined in Sec. 4.6.

We base on the model presented in [GS+18b] to characterize the operating cost $K_{\text{oper}}(\mathbf{s})$ of a mobile RAN featuring a configurable functional split. This operating cost can be divided into three components:

$$K_{\text{oper}}(\mathbf{s}) = K_{\text{inst}} + K_{\text{comp}}(\mathbf{a}) + K_{\text{rout}}(\mathbf{f}). \tag{54}$$

The first component K_{inst} is the cost of instantiating functions at the DUs and CU, which can be calculated as:

$$K_{\text{inst}} = \left(K_{\text{inst,CU}} + K_{\text{inst,DU}}\right) G, \tag{55}$$

where $K_{\text{inst,CU}}$ is the cost of instantiating the gNB functions at the CU and $K_{\text{inst,DU}}$ is the cost of instantiating the gNB functions at the DU. The second component, the computational cost $K_{\text{comp}}(\mathbf{a})$ of running centralized and decentralized functions, can be expressed as a non-linear function of the state vector $\mathbf{a}$ as follows:

$$K_{\text{comp}}(\mathbf{a}) = \sum_{g=1}^{G} \left(K_{\text{comp,CU}}(a_g)\gamma_{\text{CU}} + K_{\text{comp,DU}}(a_g)\gamma_{\text{DU}}\right) \rho_g, \tag{56}$$

where $K_{\text{comp,CU}}(a)$ and $K_{\text{comp,DU}}(a)$ denote the computational effort in *reference cores–seconds* per Gb/s (RC · s/Gb/s) required to process traffic at the CU and DU (respectively) with centralization level a, γ_{CU} and γ_{DU} are the cost per computational effort at the CU and DU (respectively) in ncu per RC · s, and ρ_g is the downlink mobile traffic at gNB g in Gb/s. In order to be consistent with previous research, a RC · s is defined as utilizing a single core of an Intel i7-4770 at 100% load during 1 second [GS+18b]. Finally, the third component, the cost $K_{\text{rout}}(\mathbf{f})$ of routing the resulting flows is calculated as:

$$K_{\text{rout}}(\mathbf{f}) = \sum_{e \in \mathbb{E}^+(n_g)} \omega f_e^g \tag{57}$$

where ω is the average cost in ncu per Gb/s of carrying traffic on a link and $\mathbb{E}^+(n)$ is the set of edges leaving node n.

Therefore, the operating-cost-minimizing FSSP can be alternatively formulated as:

$$\min_{\mathbf{a},\mathbf{f}} \; K_{\text{inst}} + K_{\text{comp}}(\mathbf{a}) + K_{\text{rout}}(\mathbf{f}) \tag{P15a}$$

subject to

$$\text{(P1b)} - \text{(P1d)}.$$

Problem (P15) is non-linear owing to the non-linearity of $K_{\text{comp}}(\mathbf{a})$, as it is evident from (56). Nonetheless, we can replace again variable vector $\mathbf{a}$ with variable vector $\mathbf{b}$ as shown in (13)–(16) to obtain a linear function of $\mathbf{b}$. In order to do this we define the following cost function:

$$\widehat{K}_{\text{comp}}(\mathbf{b}) \triangleq \widehat{K}_{\text{comp},0} + \sum_{g=1}^{G} \sum_{m=1}^{M-1} \left(\widehat{K}_{\text{comp},\text{CU}}(m)\gamma_{\text{CU}} + \widehat{K}_{\text{comp},\text{DU}}(m)\gamma_{\text{DU}} \right) \rho_g b_g^m, \tag{58}$$

where

$$\widehat{K}_{\text{comp},0} \triangleq \sum_{g=1}^{G} \left(K_{\text{comp},\text{CU}}(0)\gamma_{\text{CU}} + K_{\text{comp},\text{DU}}(0)\gamma_{\text{DU}} \right) \rho_g, \tag{59}$$

and

$$\widehat{K}_{\text{comp},\text{CU}}(m) \triangleq K_{\text{comp},\text{CU}}(m) - K_{\text{comp},\text{CU}}(m-1), \tag{60}$$

$$\widehat{K}_{\text{comp},\text{DU}}(m) \triangleq K_{\text{comp},\text{DU}}(m) - K_{\text{comp},\text{DU}}(m-1). \tag{61}$$

It can be easily shown that $\widehat{K}_{\text{comp}}(\mathbf{b}) = K_{\text{comp}}(\mathbf{a})$ as long as (13) holds, but $\widehat{K}_{\text{comp}}(\mathbf{b})$ is now a linear function of $\mathbf{b}$.

$$K_{\text{oper}}(\mathbf{s}) = K_{\text{inst}} + \widehat{K}_{\text{comp}}(\mathbf{b}) + K_{\text{rout}}(\mathbf{f}). \tag{62}$$

Therefore, the cost-minimizing FSSP can be alternatively formulated as:

$$\min_{\mathbf{b},\mathbf{f}} \; K_{\text{inst}} + \widehat{K}_{\text{comp}}(\mathbf{b}) + K_{\text{rout}}(\mathbf{f}) \tag{P16a}$$

subject to

$$\text{(P1c)},\; \text{(P1d) and (P5b)} - \text{(P5d)},$$

Problem (P16) is equivalent to that presented in [GS+18b], using an edge formulation instead of a path formulation for flow modeling. Note that (P16) is an MILP that utilizes the same variables as (P12), except for the absence of auxiliary $\mathbf{w}$ variables. As a result, it not only can be tackled directly by off-the-shelf solvers, but its complexity grows with $\mathcal{O}(G)$, leading to reasonably smaller problems than those resulting from the LWT and BGP reformulations of the performance-maximizing FSSP.

4.6. Readiness-cost-minimizing FSSP

In this section, we combine the performance-maximizing and operating-cost-minimizing approaches shown in previous sections into a single formulation that minimizes the average readiness cost. That is, the objective now is to minimize the whole readiness function, not its individual components:

$$\min_{\mathbf{s}} \; K(\mathbf{s}) = \min_{\mathbf{s}} \; K_{\text{oper}}(\mathbf{s}) + K_{\text{rev}}(\mathbf{s}). \tag{P17a}$$

subject to

$$\text{(P1b)} - \text{(P1d)}.$$

From (62) we know that we can express the operating cost $K_{\text{oper}}(\mathbf{s})$ as a linear function of the centralization vector $\mathbf{b}$ and flow vector $\mathbf{f}$. The average revenue $K_{\text{rev}}(\mathbf{s})$, however, may lead to more complex expressions.

We can assume that $K_{\text{rev}}(\mathbf{s})$ is an increasing function of the proportionally-fair objective function, $\log(\eta_u(\mathbf{a}))$. The reason is that, if this is the case, maximizing this function would also imply maximizing the revenue, which seems a reasonable approach for any operator to follow. Formally, we can express this relationship as:

$$K_{\text{rev}}(\mathbf{s}) = \widehat{K}_{\text{rev}}\left(\sum_{u=1}^{U} \log(\eta_u(\mathbf{a}))\right), \tag{63}$$

where $\widehat{K}_{\text{rev}}(\eta)$ is an increasing function of η. For notational consistency, we define this alternative version of the spectral efficiency function using $\mathbf{b}$ variables instead of $\mathbf{a}$ variables:

$$\widehat{\eta}_u(\mathbf{b}) \triangleq \eta_u(\mathbf{a}) \quad \Longleftrightarrow \quad (13). \tag{64}$$

We can then use (64) to formulate $K_{\text{rev}}(\mathbf{s})$ as a function of the alternative centralization vector $\mathbf{b}$:

$$K_{\text{rev}}(\mathbf{s}) = \widehat{K}_{\text{rev}}\left(\sum_{u=1}^{U} \log(\widehat{\eta}_u(\mathbf{b}))\right), \tag{65}$$

After combining (P17), (P16), and (65), we formulate the following optimization problem:

$$\min_{\mathbf{b},\mathbf{f}} \; K_{\text{inst}} + \widehat{K}_{\text{comp}}(\mathbf{b}) + K_{\text{rout}}(\mathbf{f}) + \widehat{K}_{\text{rev}}\left(\sum_{u=1}^{U} \log(\widehat{\eta}_u(\mathbf{b}))\right) \tag{P18a}$$

subject to

$$\text{(P1c)},\ \text{(P1d) and (P5b)} - \text{(P5d)}.$$

Problem (P18) is, in general, a mixed-integer non-linear problem (MINLP), owing to the potential non-linearity of $\widehat{K}_{\text{rev}}(\,\cdot\,)$. Nonetheless, if $\widehat{K}_{\text{rev}}(\,\cdot\,)$ were indeed linear, then (P18) could be approximated as an MILP using the same techniques presented in Sec. 4.4. In order to appreciate this more clearly, we can derive the following approximation of $\log(\widehat{\eta}_u(\mathbf{b}))$ after combining (4) and (23):

$$\begin{aligned}\sum_{u=1}^{U} \log(\widehat{\eta}_u(\mathbf{b})) \approx{}& \left(\gimel_{\text{rat}} + \frac{\aleph_{\text{rat}}}{\beth_{\text{rat}}}\right) U \\ &- \frac{\aleph_{\text{rat}}}{\beth_{\text{rat}}} \sum_{u=1}^{U} \frac{p_u}{\beth_{\text{rat}}\left(\varsigma + I_u\right) + p_u - \beth_{\text{rat}} \sum_{g=1}^{G} i_{u,g} \left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)}.\end{aligned} \tag{66}$$

Note that (66) is just a linear transformation of the objective function (P7a), and thus if $\widehat{K}_{\text{rev}}(\,\cdot\,)$ were linear we could transform (P18) into an MILP by using the LWT or BGP transformations. Alternatively, if we use (44) instead to approximate $\log(\widehat{\eta}_u(\mathbf{b}))$, the following expression arises:

$$\sum_{u=1}^{U} \log(\widehat{\eta}_u(\mathbf{b})) \approx \aleph_{\text{sim}} U - \sum_{u=1}^{U} \frac{\varsigma + I_u}{p_u} \beth_{\text{sim}} + \frac{\beth_{\text{sim}}}{2} \sum_{m=1}^{M-1} \sum_{g=1}^{G} r_g^m. \tag{67}$$

Once again (67) is a linear transformation of the objective function (P12a), which already belongs to an MILP. As a result, we conclude that if we model $\widehat{K}_{\text{rev}}(\,\cdot\,)$ as a linear function, this would result in a readiness-cost-optimizing MILP of a size similar to those presented in Sec. 4.4, thus it could be solved in the same manner. Unfortunately, in general we cannot guarantee that this conversion is linear, since the relationship between user performance and revenue characterizes a central part of the operator business, which is rarely disclosed.

Nevertheless, there are arguments to justify that $\widehat{K}_{\text{rev}}(\,\cdot\,)$ may be indeed a quasi-linear function. In order to show this, we first base on the following identity:

$$\sum_{u=1}^{U} \log\left(\eta_u\left(\mathbf{a}\right)\right) = U \log(\widetilde{\eta}(\mathbf{a})), \tag{68}$$

where $\widetilde{\eta}(\mathbf{a})$ is the geometric mean of the spectral efficiency over all UEs as defined in (11). Previous research has shown that the value of $\widetilde{\eta}(\mathbf{a})$ ranges from $\widetilde{\eta}(\mathbf{a}) \approx 2$ b/s/Hz in RANs whose functional split is chosen to optimize performance, to $\widetilde{\eta}(\mathbf{a}) \approx 1$ b/s/Hz in non-optimized RANs [MAJK21]. This very limited range translates into an almost linear relationship between $\check{\eta}(\mathbf{a})$ and $\widetilde{\eta}(\mathbf{a})$, thus we can formulate the fol-

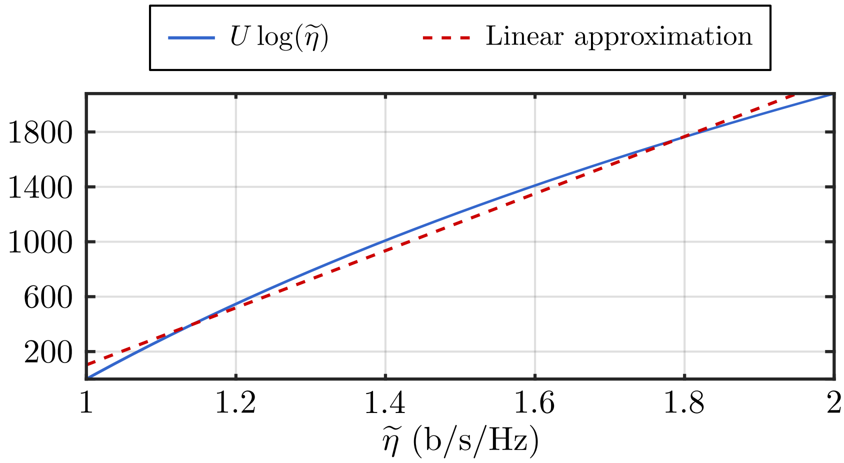


Figure 4.1.: Comparison between function $U \log(\widetilde{\eta})$ and its best linear approximation for $U = 3000$ UEs and $1 \leq \widetilde{\eta} \leq 2$.

lowing approximation:

$$\widetilde{\eta}(\mathbf{a}) \approx \aleph_{\text{lin}} + \daleth_{\text{lin}} \sum_{u=1}^{U} \log\left(\eta_u\left(\mathbf{a}\right)\right). \tag{69}$$

Parameters $\aleph_{\text{lin}}$ and $\daleth_{\text{lin}}$ can be obtained by linear fit, for instance by least-squares minimization. A depiction of this approximation for $U = 3000$ UEs is shown in Fig. 4.1. Furthermore, the limited range of $\widetilde{\eta}(\mathbf{a})$ also implies that the average UE data rates experience a roughly twofold increase in performance-optimizing RANs with respect to non-optimizing RANs. Consequently, there is little room for severe non-linearities in $\widehat{K}_{\text{rev}}(\,\cdot\,)$ for actual networks.

In any case, even if we assume that $\widehat{K}_{\text{rev}}(\,\cdot\,)$ is linear, we cannot provide a single function that fits all networks. Indeed, operators may utilize multiple commercial strategies and these may change over time. Therefore, even if we knew the function converting performance into revenue for one operator within a time period, it would be difficult to extrapolate this information to other operators and time periods. The approach we follow to overcome this modeling difficulty is to provide not a single function but instead a family of performance-to-revenue functions, which are characterized by two parameters. The first parameter is the reference revenue-to-operating-cost ratio ξ:

$$\xi = -\frac{\widetilde{K}_{\text{rev}}(\widetilde{\eta}_{\text{ref}})}{K_{\text{oper}}^{\text{ref}}}, \tag{70}$$

where

$$\widetilde{K}_{\text{rev}}(\widetilde{\eta}) \triangleq \widehat{K}_{\text{rev}}\left(U \log(\widetilde{\eta})\right) \approx \widehat{K}_{\text{rev}}\left(\frac{\widetilde{\eta} - \aleph_{\text{lin}}}{\daleth_{\text{lin}}}\right) \tag{71}$$

is a reformulation of the performance-to-revenue function that takes the geometric mean of the spectral efficiency $\widetilde{\eta}$ as input, $\widetilde{\eta}_{\text{ref}}$ is the geometric mean of the spectral efficiency achieved in a reference scenario, and $K_{\text{oper}}^{\text{ref}}$ is the operating cost at a reference scenario. We select the scenario where all UEs are uniformly distributed over the covered area as the reference scenario. The reason is that, in such a scenario, the operating cost and performance achieved by all optimization approaches are approximately the same, regardless of whether they are operating-cost-minimizing or performance-maximizing [MAJK21]. This can be observed in the results shown in Sec. 4.7.3. In addition, the spectral efficiency achieved in this scenario is a lower bound to that of clustered scenarios. As a result, the revenue of all optimization approaches is also similar, yielding a consistent reference point. For our experiments, we consider milestone values of $\xi \in \{0.5, 1, 2\}$, meaning that the revenue at the reference scenario may be half, the same, or twice as much as the operating cost.

The second parameter υ is the revenue growth rate, defined as the ratio between the revenue when the spectral efficiency is twice that of the reference scenario and the reference revenue. Formally, we formulate it as follows:

$$\upsilon = \frac{\widetilde{K}_{\text{rev}}(2\widetilde{\eta}_{\text{ref}})}{\widetilde{K}_{\text{rev}}(\widetilde{\eta}_{\text{ref}})}, \tag{72}$$

Parameter υ reflects how the revenue grows as the spectral efficiency doubles, thus characterizing the slope of the performance-to-cost function. We select example values $\upsilon \in \{1.5, 2, 3\}$, reflecting that the revenue increases 50%, 100%, and 200% with respect to the reference revenue when the current spectral efficiency is twice the reference spectral efficiency.

As a result, we can formulate the performance-to-revenue conversion function $\widetilde{K}_{\text{rev}}(\widetilde{\eta})$ between the geometric average of the spectral efficiency $\widetilde{\eta}$ and the revenue as:

$$\widetilde{K}_{\text{rev}}(\widetilde{\eta}) \approx \xi K_{\text{oper}}^{\text{ref}} \left[2 - \upsilon + (\upsilon - 1)\frac{\widetilde{\eta}}{\widetilde{\eta}_{\text{ref}}}\right]. \tag{73}$$

In Fig. 4.2, we show the shape of these functions for the selected range of parameters. It is clear that, even though we assume that our performance indicator and its associated revenue are linearly related, we provide multiple options of such linear relationships so as to cover a wide range of cases. If the actual relationship happens to be non-linear, our proposed conversion functions can still be used as either upper or lower bounds.

Finally, after applying the performance-to-cost function (73) and (66) to (P18) and some straightforward simplifications, we obtain the following readiness-cost-mini-

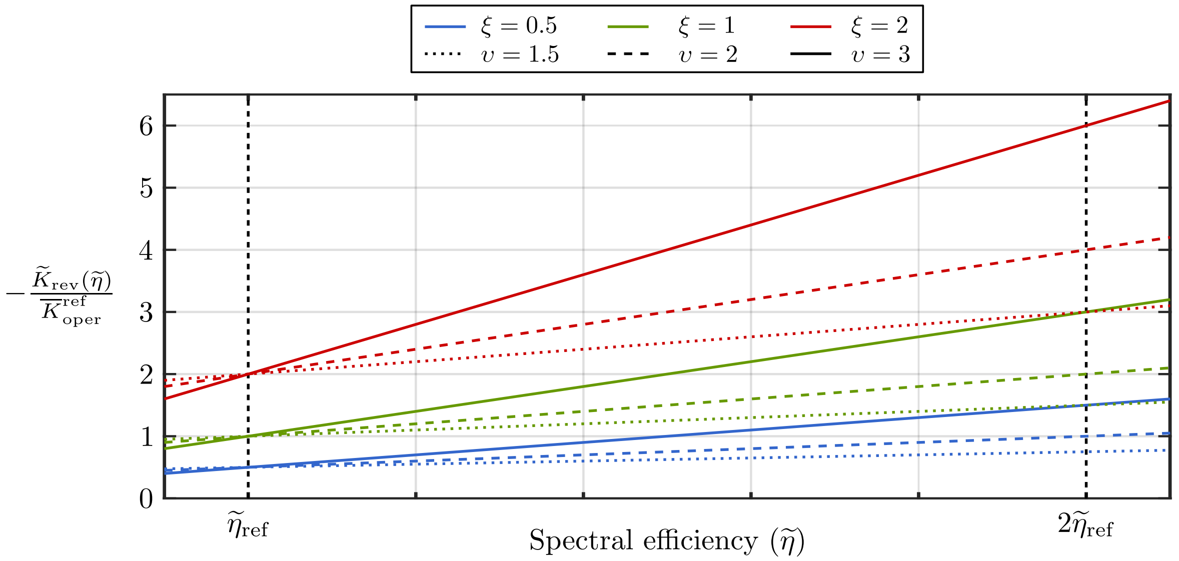


Figure 4.2.: Comparison between nine performance-to-revenue functions $\widetilde{K}_{\mathrm{rev}}(\widetilde{\eta})$ for revenue-operating cost ratios $\xi \in \{0.5, 1, 2\}$ and revenue growth rates $\upsilon \in \{1.5, 2, 3\}$.

mizing FSSP that can be tackled by means of LWT and BGP formulations:

$$\begin{aligned}\min_{\mathbf{b},\mathbf{v},\mathbf{f}} \; & \widehat{K}_{\mathrm{comp}}(\mathbf{b}) + K_{\mathrm{rout}}(\mathbf{f}) \\ & - \xi K_{\mathrm{oper}}^{\mathrm{ref}}(\upsilon - 1)\frac{\beth_{\mathrm{lin}}\aleph_{\mathrm{rat}}}{\widetilde{\eta}_{\mathrm{ref}}\beth_{\mathrm{rat}}} \sum_{u=1}^{U} \frac{p_u}{\beth_{\mathrm{rat}}\left(\varsigma + I_u\right) + p_u - \beth_{\mathrm{rat}} \sum_{g=1}^{G} i_{u,g} \left(\sum_{m=1}^{M-1} \varrho(m) v_{g,h_u}^{m}\right)}.\end{aligned} \tag{P19a}$$

subject to

$$\text{(P1c)},\ \text{(P1d)},\ \text{(P5b)} - \text{(P5d)},\ \text{and (P6b)} - \text{(P6d)}. \tag{P19b}$$

If we instead combine (73) and (P18) with the quadratic approximation shown in (67), we arrive to this alternative formulation of the readiness-cost-minimizing FSSP, which can be tackled by the quadratic approach described in Sec. 4.4.3:

$$\min_{\mathbf{b},\mathbf{r},\mathbf{f}} \; \widehat{K}_{\mathrm{comp}}(\mathbf{b}) + K_{\mathrm{rout}}(\mathbf{f}) + \xi K_{\mathrm{oper}}^{\mathrm{ref}}(\upsilon - 1)\frac{\beth_{\mathrm{lin}}\beth_{\mathrm{sim}}}{2\widetilde{\eta}_{\mathrm{ref}}} \sum_{m=1}^{M-1} \sum_{g=1}^{G} r_g^m \tag{P20a}$$

subject to

$$\text{(P1c)},\ \text{(P1d)},\ \text{(P5b)} - \text{(P5d)},\ \text{and (P12b)} - \text{(P12d)}.$$

As we show in the following sections, the quadratic approach usually outperforms the LWT and BGP reformulations. Therefore, we take (P20) as the final, reference MILP whose solving yields the optimal centralization vector $\mathbf{a}^*$ (after the appropriate conversion from vectors $\mathbf{b}^*$ and $\mathbf{v}^*$) and flows vector $\mathbf{f}^*$ that minimizes $K(\mathbf{s}^*) = K(\mathbf{a}^*, \mathbf{f}^*)$.

4.7. Experimental evaluation

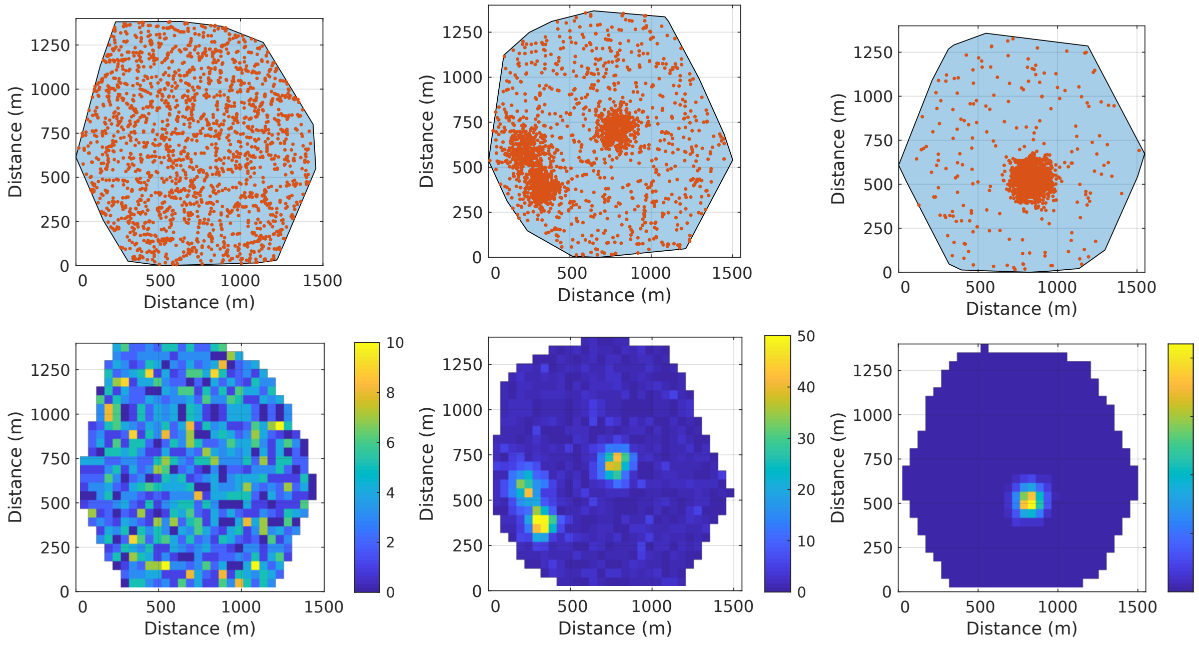
After having presented three different interpretations of the FSSP, including the final formulation minimizing the average readiness cost, in this section we evaluate the performance of these approaches. We first focus on evaluating the convergence times of the alternative formulations, which allow us to preliminarily discard those formulations that take an excessive amount of time. Then, we assess the spectral efficiency, operating cost, and readiness cost that all approaches achieve. This allows us to conclude that the final readiness-cost-minimizing formulation is indeed adequate and also observe the differences with the approaches that only focus on either user performance or operating cost.

We use a MATLAB simulator to generate the interference coefficients $i_{u,g}$, required by all presented approaches, based on simulated UE and gNB positions. Then, we evaluate the algorithms on operator-grade hardware consisting of six computing servers and 48 CPU cores [Bas+17], by using Gurobi [Gur], a commercial optimization solver. We depict observed values of convergence time and spectral efficiency by means of boxplots, with the intention of representing their distribution. In order to provide as much information as possible in little space, we use a compact version of standard boxplots, whose interpretation is as follows: the central dot represents the median, the box contains the data between the first and third quartiles, and the whiskers extend to the lowest and highest values contained in 1.5 times the inter-quartile range. Occasionally, we show the (arithmetic) mean of the distribution as a horizontal bar. For the sake of clarity, outliers are not shown.

4.7.1. Simulation setup

Simulated mobile coverage

In order to produce realistic instances of the FSSP, we follow the recommendations for simulating dense urban scenarios as described in 3GPP TS38.193 [3GP20b]. According to this specification, gNBs are divided into two layers: macro and micro layer. The macro layer follows an hexagonal layout with an inter-site distance of 200 m. This results in a density of around 29 macro gNBs per square kilometer. In addition, the number of micro gNBs should be three times that of the macro gNBs. As a result, the average cell density is roughly 115 gNBs/km^2. This implies that a RAN with $G = 300$ gNBs covers an area of 2.6 km^2, which approximately corresponds to the center of a medium-sized or large city. For reference, the area of the City of London is approximately 2.9 km^2.



(a) $\theta = 0.5$ (b) $\theta = 0.75$ (c) $\theta = 0.95$

Figure 4.3.: Visualization of UE concentration index for 2000 UEs. The red dots on the left figure represent the positions of each UE on the considered area, which is shown in light blue. The right figure is a color map of the 2-dimensional empirical distribution of the UEs, showing the UE density of each 50×50 m square bin.

Regarding the UEs, 3GPP recommends to consider 10 UEs per gNB on average when simulating 5G dense urban scenarios [3GP20b], which results in 1150 UEs/km^2. This means, for instance, that a RAN with $G = 300$ gNBs serves, on average, 3000 simultaneously active UEs. Since the UE distribution may impact the performance of the algorithms, we derive a dedicate metric to model it. We divide the considered network area into square bins of side 50 m. Then, we count the number of UEs in each bin to compute its 2-dimensional empirical distribution. From this distribution, we compute its Gini coefficient and use it as the *UE concentration index* θ. It is observed that $\theta = 0.5$ corresponds to a uniformely-distributed random distribution of UEs, as it can be seen in Fig. 4.3a. Higher values correspond to higher UE clustering, (Fig. 4.3b and Fig. 4.3c), with a maximum value of $\theta = 1$ when all UEs are within the same bin. Once the position of the gNBs and the UEs is generated, we compute the received signal and interference power by using the log-distance path model for urban scenarios [SS10].

Fronthaul network

For our performance evaluation, we set the number of functional split options to $M = 4$ (such as C-RAN, Intra-PHY, MAC-PHY, PDCP-RLC). Based on the analytical and

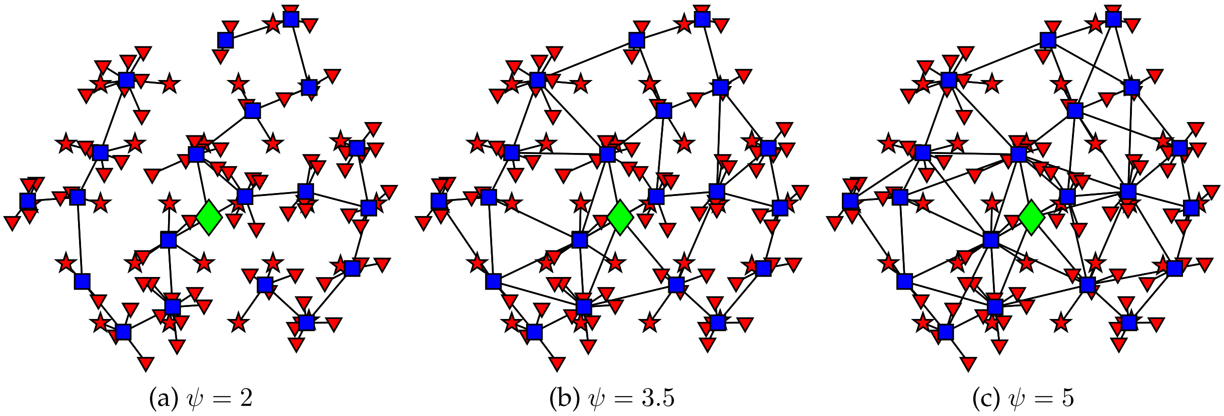


Figure 4.4.: Visualization of the average fronthaul network degree for the same gNB distribution. Black lines represent links, red stars represent macro DUs, red triangles are micro DUs, blue square are fronthaul switches, and green diamonds are CUs.

experimental results shown in [PM17], we use the following interference-cancellation vector: $\mathbf{c} = \langle 1, 0.6, 0.2, 0.01 \rangle$, such that there is no interference mitigation when using the lowest centralization level (as with PDCP-RLC) and a cancellation factor of 20 dB when using the highest centralization level. Note, however, that other values or split options are also possible, since they do not affect the problem formulation.

Regarding the fronthaul capacity vector, we use the values $\mathbf{r} = \langle 4, 8, 80, 160 \rangle$ Gb/s, as provided in [3GP17]. In order to simulate realistic fronthaul network layouts, we follow the descriptions provided in [GS+18a], which are based on real mobile networks on Italy, Romania, and Switzerland. Accordingly, we set the maximum link capacities to 0.5, 1 or 2 Tb/s (the higher value that makes full centralization infeasible, to prevent trivial results). Furthermore, we simulate several types of fronthaul networks by controlling the *average fronthaul network degree* ψ, defined as the ratio of links to fronthaul switches. According to [GS+18a], the average degree of a typical fronthaul network ranges from $\psi = 2$ (a tree graph) to $\psi = 5$. In Fig. 4.4 we show some exemplary layouts resulting from varying these parameters.

Computing platform

After creating the interference coefficients and the fronthaul network with the MATLAB simulator, we have all the required components to run our adaptation algorithms. Since the convergence time of this algorithms is very relevant to decide on their viability, we use an operator-grade hardware platform consisting of 48 Intel Xeon E5 cores distributed over six computing servers [Bas+17]. As optimization solver, we use the commercial Gurobi software [Gur]. This software is able to divide large MILP instances and efficiently process them in parallel.

Parameter	Value	Units	Source
$K_{\text{inst,CU}}$	1	ncu	[GS+18b, Table I]
$K_{\text{inst,DU}}$	0.5	ncu	[GS+18b, Table I]
$K_{\text{comp,CU}}(a)$	[1 1.8 3.4 5]	RC · s/Gb/s	[GS+18b, Table I]
$K_{\text{comp,DU}}(a)$	[4 3.2 1.6 0]	RC · s/Gb/s	[GS+18b, Table I]
γ_{CU}	0.017	ncu/RC	[GS+18b, Table I]
γ_{DU}	1	ncu/RC	[GS+18b, Table I]
ω_{DU}	Variable	ncu/Gb/s	[GS+18b, Table I]

Table 4.2.: Summary of operating cost parameters.

Operating cost model

In order to apply the operating cost model presented in Sec. 4.5, we need to select appropriate values to represent realistic 5G RAN deployments. With the intention of providing comparable results, we take the values presented in [GS+18b] as our main reference, which are summarized in Table 4.2.

4.7.2. Convergence time

The applicability of the FSSP approaches presented in this chapter is heavily influenced by their convergence times, that is, the time required for the approaches to reach a (near-)optimal solution. This is due to the fact that our ultimate intention is to dynamically solve the FSSP during runtime, which is addressed in Chapter 6. Consequently, it is possible that the solutions yielded by the algorithms are outdated if their convergence time is too long. Previous work has shown that mobile traffic is highly variable, and may often sharply deviate from average patterns [TGD19]. In addition, in certain region types (such as entertainment or transport areas), even the average patterns may be fast-changing [Xu+16]. In fact, the analysis of recent mobile traffic traces show that the user traffic experienced by a 5G RAN may abruptly change in few minutes [MAK19b; PL15]. As a result, approaches with long convergence times are probably not useful for dynamic adaptation. For our experiments, we configure the solver to allow for a maximum convergence time of 15 minutes, since solutions taking longer are unlikely to be usable. Indeed, in Chapter 6 we show that the convergence time should actually be in the order of a few seconds for dynamic adaptation to be feasible.

We first focus on the convergence times of the MILP reformulations of the performance-maximizing FSSP. In Fig. 4.5 we show the distributions of the convergence time of the five MILP reformulations (quadratic, LWT, BGP, and punctured LWT and BGP formulations) as a function of the number of gNBs, which in turn reflects the total size of the RAN. The solver is configured for a maximum running time of 15 minutes with a relative gap tolerance of 0.01% and each boxplot represents at least 100 runs.

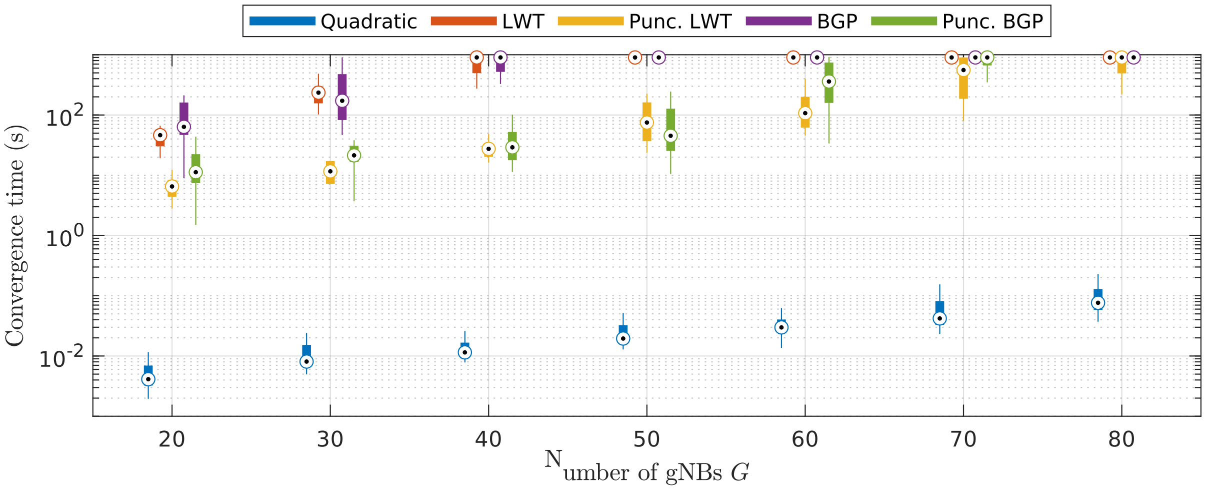


Figure 4.5.: Convergence time of the MILP reformulations of the performance-maximizing FSSP.

We observe that the convergence times of unpunctured LWT and BGP formulations reach the 15-minutes limit with only 40 gNBs, and with 50 gNBs all instances take longer or equal than this limit. Assuming a cell density of 115 gNBs/km^2, this implies that they may be suitable only for areas smaller than 0.4 km^2. The running time of the punctured LWT and BGP formulations is noticeably smaller, although the 15-minute limit is once again reached with only 60 or 70 gNBs, corresponding to an area of ca. 0.6 km^2. As a result, both punctured and unpunctured LWT and BGP formulations can only be applied to very small deployments. Moreover, since they only offer a very slight performance improvement with respect to the quadratic approach (as we show in Sec. 4.7.3), these approaches are not suitable for actual RAN optimization.

In contrast, the convergence time of the quadratic approach remains below 1 s for areas with $G = 80$ gNBs or less. In fact, in most runs the quadratic approach yields a solution in less than 0.1 s even when $G = 80$. In Fig. 4.6 we show an expanded range of gNBs and include Heuristics 1 and 2 in the comparison. We can see that the almost all instances of the quadratic approach take less than 10 seconds to converge for the whole considered interval. This includes the case of $G = 400$, which translates into a covered area of 3.5 km^2 and 4000 simultaneously active UEs, which is enough to cover the densely populated centers of most cities. We can also observe that the convergence time of Heuristic 2 is at first similar to that of the quadratic approach, but when $G \geq 300$ this heuristic takes over ten times longer to converge than the quadratic approach. This is due to the fact that Heuristic 2 is a local search approach, and thus it may fail to converge if the explored area is large and no better local minima are found. Finally, all instances of Heuristic 1 converge in less than 1 s for the whole shown range, being thus the fastest algorithm to solve the performance-maximizing version of the FSSP. We conclude, thus, that the most promising algorithms in terms of convergence time are the quadratic approach and Heuristic 1.

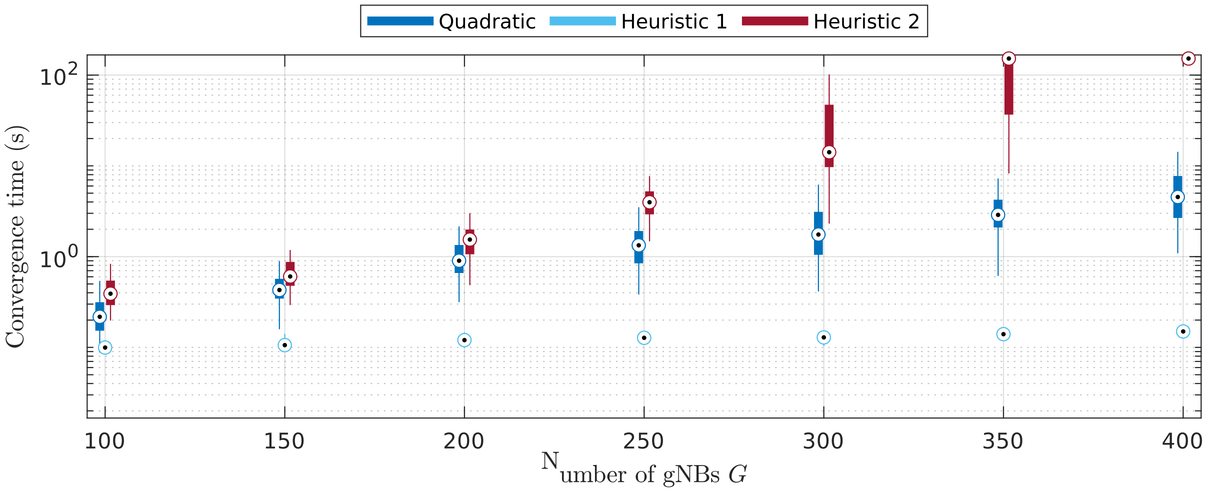


Figure 4.6.: Convergence time of the quadratic reformulation and Heuristics 1 and 2 for the performance-maximizing FSSP..

Owing to its good convergence time, we select the quadratic approach to tackle the readiness-cost-minimizing version of the FSSP, that is, formulation (P20). In Fig. 4.7 we show at last a direct comparison of the convergence times of the quadratic approach to the performance-maximizing FSSP, the operating-cost-minimizing FSSP, and the quadratic approach of the readiness-cost-minimizing FSSP. We observe that the operating-cost-minimizing FSSP, as formulated in (P16), is the fastest of the three approaches, converging in less than 0.1 s on average for $G \leq 450$ gNBs. The second fastest is, somehow counter-intuitively, the readiness-cost-minimizing approach (P20), whose convergence times are consistently lower than those of the performance-maximizing approach for the whole interval, although the objective function of the former contains that of the latter. The reason is for this is the influence of the operating cost component in the (P20), which leads to formulations that are easier to solve. As a result, we conclude that with operator-grade equipment, the readiness-cost-minimizing FSSP can be solved in less than 2 s on average when $G \leq 300$ gNBs, and in less than 3 s for $G \leq 450$ gNBs.

4.7.3. Spectral efficiency and cost evaluation

After evaluating the convergence time of the approaches, we measure their performance in terms of the achieved geometric mean of the spectral efficiency $\widetilde{\eta}(\mathbf{a})$, as it is defined in (11), as well as their achieved operating and readiness cost. We first evaluate $\widetilde{\eta}(\mathbf{a})$ for all considered performance-maximizing approaches over a set of extreme cases so as to select the best approach out of them. Then, we analyze the impact of the configuration of the fronthaul network and the user concentration on the most promising approaches. Finally, we assess the spectral efficiency, operating cost, and readiness cost of all three interpretations of the FSSP.

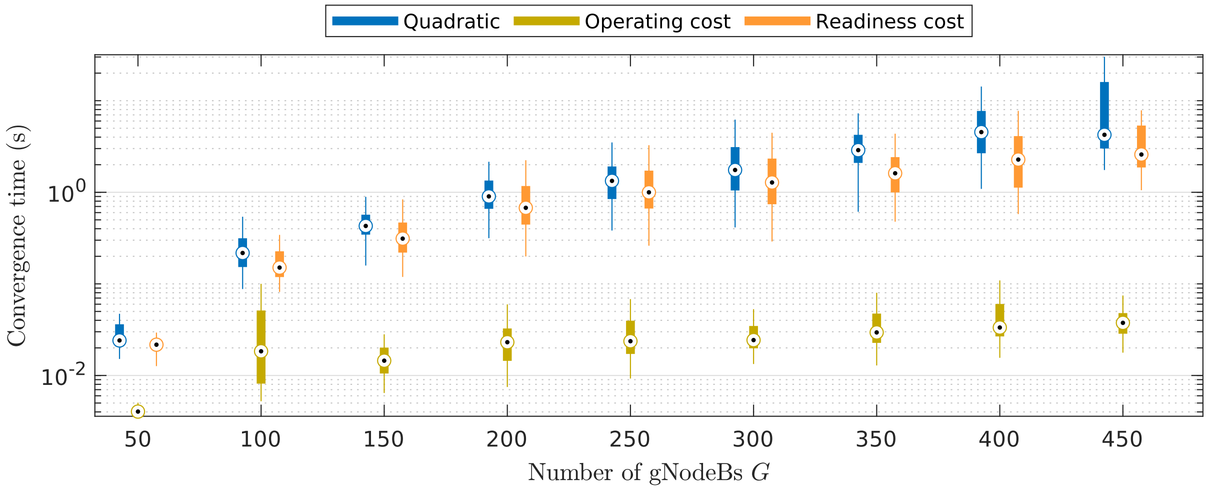


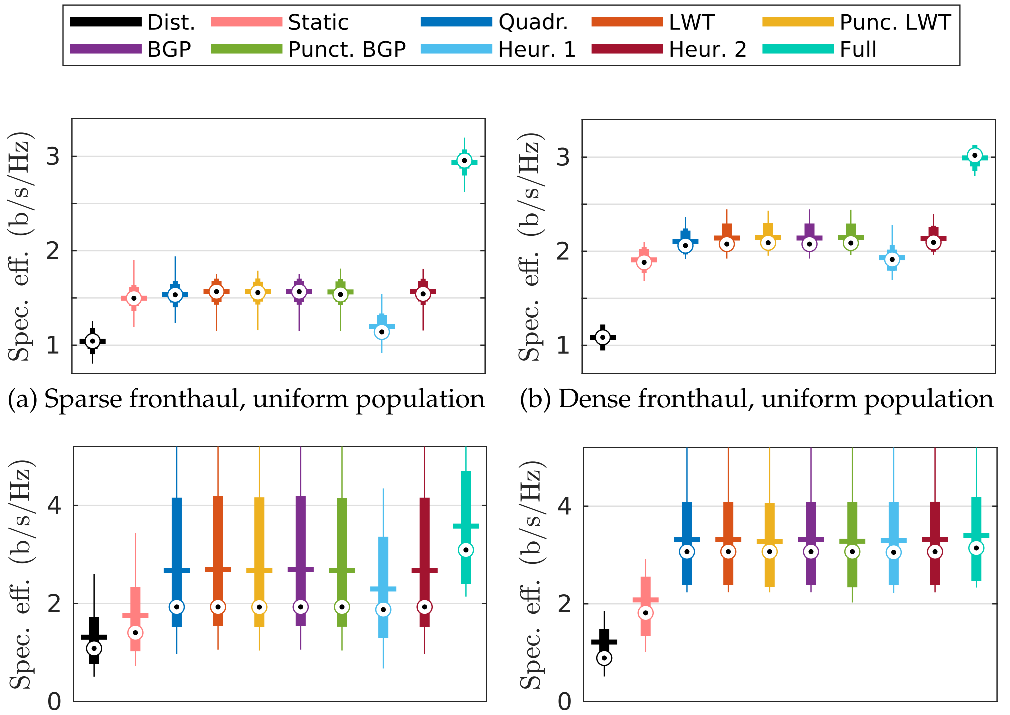
Figure 4.7.: Convergence time of the quadratic approach to the performance-maximizing FSSP, the operating-cost-minimizing FSSP, and the quadratic approach of the readiness-cost-minimizing FSSP.

Selection of best performance-maximizing approach

In order to compare the performance of all performance-maximizing approaches, we evaluate their spectral efficiency $\widetilde{\eta}(\mathbf{a})$ on the same scenario. Since the LWT and BGP formulations are only applicable to small networks, we choose $G = 50$ gNBs for this comparison, corresponding to $U = 500$ UEs and an area of approximately $0.4\,\mathrm{km}^2$. We now generate four types of scenarios to cover a wide range of interference cases: (i) a sparse fronthaul ($\psi = 2$) with uniform population ($\theta = 0.5$); (ii) a dense fronthaul ($\psi = 5$), with uniform population ($\theta = 0.5$); (iii) a sparse fronthaul ($\psi = 2$), with concentrated population ($\theta = 0.95$); and (iv) a dense fronthaul ($\psi = 5$), with concentrated population ($\theta = 0.95$).

The simulation results after 200 runs are shown in Fig. 4.8, with scenarios (i)–(iv) being used in Fig. 4.8a–4.8d, respectively. Apart from the proposed approaches, we also include in the comparison the spectral efficiencies of a fully distributed solution ($a_g = 0\ \forall g \in \mathbb{G}$), a static solution (in which the optimal s is precomputed assuming a perfectly uniform population), and a fully centralized solution ($a_g = M\ \forall g \in \mathbb{G}$). The fully distributed solution represents a network in which all processing is distributed, thus providing a lower bound to all solutions. The static solution is the one proposed in works such as [GS+18a; HR18b], in which the functional split of every gNB is calculated for the average traffic and not adapted to the instantaneous interference situation. The fully centralized solution, in which all gNBs are centralized, is infeasible in all cases, but it serves as a upper bound for the other approaches.

From these results we can observe two general trends. First, the denser the fronthaul network, the higher the spectral efficiency of all solutions, but also the better the performance of our proposed solutions with respect to the static solutions. Second, the more concentrated the population, the higher the variance and the mean spectral effi-



(a) Sparse fronthaul, uniform population (b) Dense fronthaul, uniform population

(c) Sparse fronthaul, concentrated population (d) Dense fronthaul, concentrated population

Figure 4.8.: Distributions of the mean spectral efficiency $\widetilde{\eta}(\mathbf{a})$ achieved by the performance-maximizing approaches, a fully distributed approach, a static approach, and a fully centralized approach in four extreme scenarios.

ciency achieved by our solutions with respect to the static solutions. As a result, when the fronthaul is dense and the population is concentrated, our approaches achieve similar spectral efficiencies to that achieved by full centralization.

Apart from these general trends, we can compare the performance of our proposed performance-maximizing approaches. We can conclude that, with the exception of Heuristic 1, all these approaches perform very similarly in all cases. Under close examination, we can see that the best performance is achieved by LWT and BGP transformations and Heuristic 2, followed closely by the quadratic approach and the punctured LWT and BGP transformations. Nonetheless, the maximum difference in their average spectral efficiency is less than 2% in all scenarios.

Given its good-quality solutions and its low convergence time, we conclude that the quadratic formulation is the most efficient approach, and hence the most suitable to use in combination with the operating-cost-minimizing approach to optimize the average readiness cost. Nonetheless, Heuristic 1 may be still adequate for large networks ($G \geq 400$) in which convergence time needs to be very short, in the sub-second range. Similarly, the punctured and unpunctured LWT and BGP transformations may be applicable to small networks ($G \leq 40$) in which maximizing the spectral efficiency is of utmost importance, such as industrial or ultra-reliable networks.

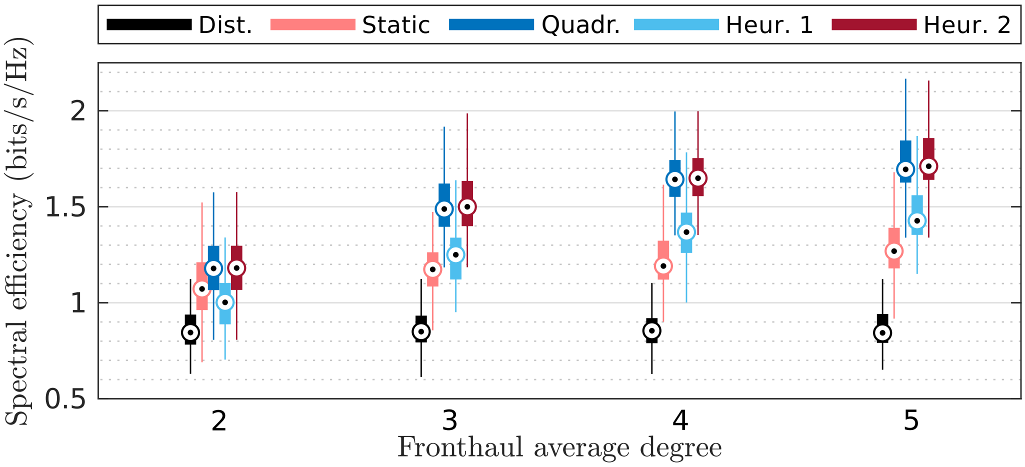


Figure 4.9.: Average spectral efficiency achieved by the quadratic approach, Heuristics 1 and 2, a static and a fully distributed network for $G = 300$ and a UE concentration index of 0.75 as the fronthaul average degree varies.

In Fig. 4.8 we can see how the density of the fronthaul network affects the quality of the solutions achieved by all approaches for a deployment with $G = 50$ gNBs. When using a sparse fronthaul network of $\psi = 2$, few centralization vectors are feasible because of the limited number of paths, which leads to mean optimal spectral efficiencies of around 1.57 b/s/Hz for dispersed UEs and 2.14 b/s/Hz for concentrated UEs. With a denser fronthaul network of $\psi = 5$, the average spectral efficiency grows up to around 2.7 b/s/Hz for dispersed UEs and 3.31 b/s/Hz for concentrated UEs. With the intention of observing this trend more clearly, we perform another experiment on a larger network ($G = 300$, 2.6 km^2), with partially concentrated UEs ($\theta = 0.75$) and let the density of the fronthaul vary from $\psi = 2$ to $\psi = 5$. The results are shown in Fig. 4.9. We conclude that with sparse, tree networks ($\psi = 2$) the benefits of implementing a performance-optimized solution are marginal, improving only from 1.07 b/s/Hz (static solution) to 1.18 b/s/Hz (quadratic approach), a 10% improvement. With an average degree of $\psi = 3$ this improvement increases up to 28% (quadratic approach), and with $\psi = 5$ it reaches 36% (quadratic approach).

Implementing an adaptive functional split is specially beneficial when the UEs tend to be clustered in time-varying clusters. This can be observed once again in the experiment shown in Fig. 4.8: as the UE concentration index changes from $\theta = 0.5$ to $\theta = 0.95$, the average spectral efficiency increases from 1.57 b/s/Hz to 2.7 b/s/Hz for $\psi = 2$, and from 2.14 b/s/Hz to 3.31 b/s/Hz for $\psi = 5$. Conversely, the average spectral efficiency of the static solution barely changes. This is due to the fact that, when UEs are concentrated around the same spots, an adaptive network can mitigate their interference more efficiently than when they are spread apart. In order to evaluate the effect of UE concentration in more detail, we perform another experiment on a larger network ($G = 300$, 2.6 km^2) with a constant fronthaul average degree of $\psi = 3.5$ and let the UE concentration index vary from $\theta = 0.5$ to $\theta = 0.95$. The results are shown in Fig. 4.10. We conclude that an adaptive approach may achieve substantially

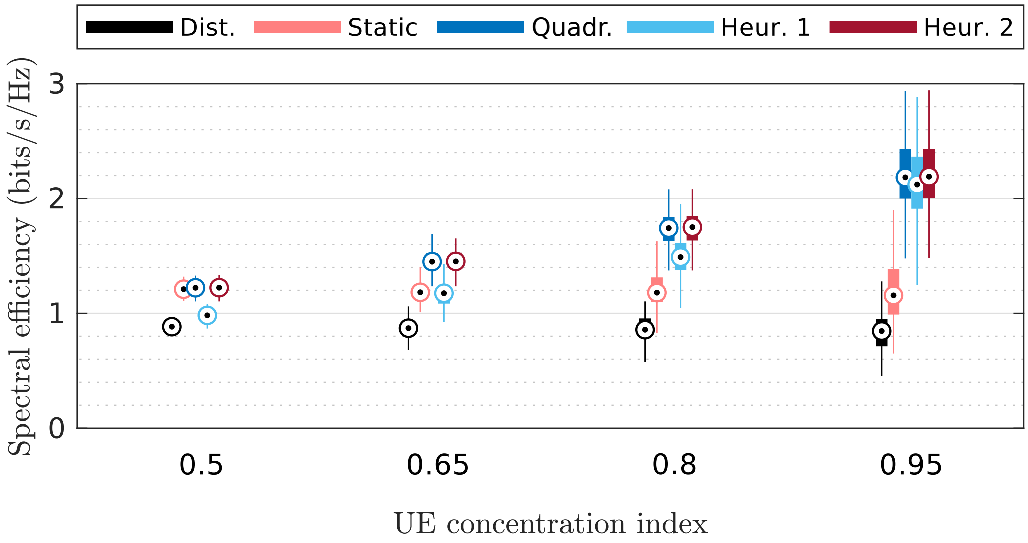


Figure 4.10.: Average spectral efficiency achieved by the quadratic approach, Heuristics 1 and 2, a static and a fully distributed network for $G = 300$ and a fronthaul average degree of 3.5 the fronthaul average degree varies as the UE concentration index varies.

better spectral efficiency when UEs are concentrated with respect to a static solution. Indeed, when $\theta = 0.8$, the mean spectral efficiency can be improved from 1.18 b/s/Hz to 1.75 b/s/Hz, a 48% improvement. For $\theta = 0.95$, this improvement reaches almost 90%. Interestingly, the mean spectral efficiency of a static solution is barely affected by the UE concentration, although its variance does increase. This is because clusters form at any point of the covered area with equal probability, combined with the fact that the static solution explicitly optimizes for the average UE position, regardless of the instantaneous UE concentration.

From these evaluations of the convergence times and achieved spectral efficiencies, we conclude once again that the quadratic approach as formulated in (P12) is the most promising option for solving the performance-maximizing FSSP. Consequently, we also use this approach, in combination with the operating-cost-minimizing formulation, to find the best state vector s that minimizes the readiness cost. That is, we opt for formulation (P20) when dealing with the readiness-cost-minimizing FSSP. Nonetheless, since the heuristic approaches yield good quality solutions in a relatively short time, we still include them in subsequent comparisons.

Comparison between performance-maximizing and operating-cost-minimizing approaches

Although the performance-maximizing and the operating-cost-minimizing FSSPs have different objectives, their solutions may not be radically different. Indeed, there may be a positive correlation between operating cost minimization and spectral efficiency maximization as a result of the influence of computational costs $\widehat{K}_{\text{comp}}(\mathbf{b})$. Since the

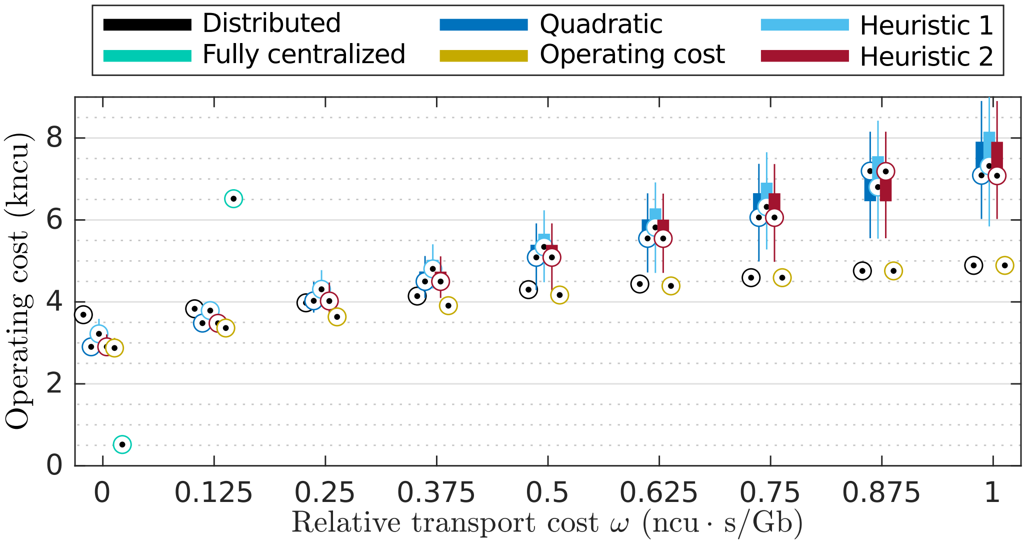


Figure 4.11.: Relative operating cost achieved by fully distributed, fully centralized, quadratic, operating-cost-minimizing, and heuristic approaches when $G = 300$, average fronthaul degree $\psi = 3.5$, and UE concentration index $\theta = 0.75$.

cost of running mobile functions at the CU is much smaller than at the DU ($\gamma_{\text{CU}} = 0.017$ ncu/cycle vs. $\gamma_{\text{DU}} = 1$ ncu/cycle, according to [GS+18b]), minimizing this cost component entails centralizing as many functions as possible, which is also desired by the performance-maximization approach. Nonetheless, the routing costs $K_{\text{rout}}(\mathbf{f})$ have the opposite effect, since function centralization increases network usage. As a result, the value of ω, the transport cost of carrying traffic on a link, greatly influences how exceedingly costly a performance-maximizing solution is, and how much throughput is lost by a operating-cost-minimizing solution.

Fig. 4.11 shows the impact of ω on the operating cost of solutions provided by the performance-maximizing quadratic approach and Heuristics 1 and 2, along with that of the operating-cost-minimizing solution from (P16). For reference, we also include the operating cost of fully distributed and fully centralized networks, although the latter is always infeasible. We observe that when $\omega = 0$, the operating cost of our proposed approaches is very similar to that of the operating-cost-minimizing approach. In fact, the average cost difference between the operating-cost-minimizing and quadratic approaches is less than 1%. As ω grows, the operating-cost-minimizing solution converges to the distributed solution, since centralization incurs in high routing costs. Conversely, the operating cost of our proposed approaches increases linearly with ω, as this parameter is not taken into account for solution selection. As a consequence, at $\omega = 0.5$, the average operating cost difference between the operating-cost-minimizing and quadratic approaches is 22%, and if $\omega = 1$ this difference increases up to 45%.

In Fig. 4.12 we show the influence of the routing costs ω on the spectral efficiencies achieved by the same approaches as in Fig. 4.11. We observe that the operating-cost-minimizing approach always achieves noticeably lower spectral efficiencies than per-

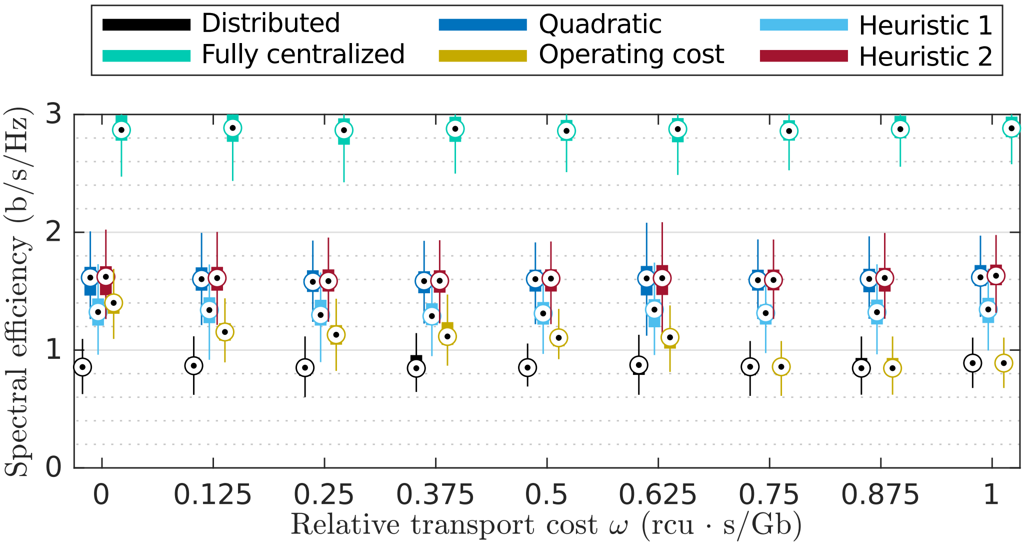
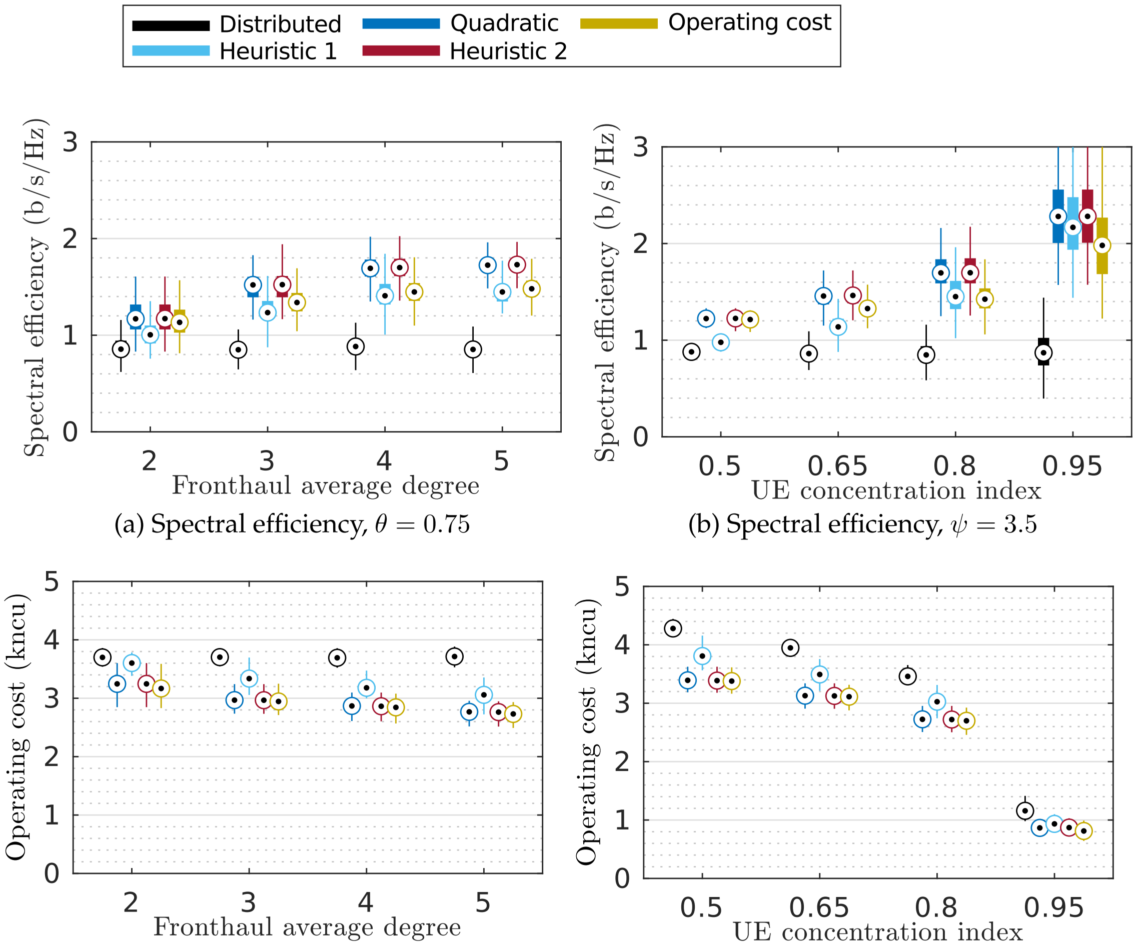


Figure 4.12.: Average spectral efficiency achieved by fully distributed, fully centralized, quadratic, operating-cost-minimizing, and heuristic approaches when $G = 300$, average fronthaul degree $\psi = 3.5$, and UE concentration index $\theta = 0.75$.

formance-maximizing approaches, as we might expect. At $\omega = 0$, our quadratic approach achieves a 15% higher spectral efficiency than the operating-cost-minimizing approach, this increases to 45% at $\omega = 0.5$, and finally to 86% for $\omega \geq 0.75$, as the operating-cost-minimizing approach converges to the distributed solution. We conclude that the additional cost of performance-maximizing approaches translates into proportionally higher improvements in the spectral efficiency. If the operator is able to profit from these improvements, then performance-maximizing approaches may lead to a higher revenue with respect to static or operating-cost-minimizing approaches.

With the intention of providing a detailed cost analysis, we also study how the fronthaul network density and the UE concentration influence the trade off between spectral efficiency and cost in two scenarios. First, we investigate the case where routing costs are negligible ($\omega = 0$), and thus the operator is motivated to centralize as many functions as possible, either pursuing minimal operating cost or maximum spectral efficiency. The results for this scenario are depicted in Fig. 4.13. We observe that, in this case, higher fronthaul densities lead to higher spectral efficiencies (Fig. 4.13a) and lower costs (Fig. 4.13c) for all approaches, since more functions can be maximized. Nevertheless, our proposed performance-maximizing approaches take more advantage of dense fronthaul networks, achieving better spectral efficiency than the operating-cost-minimizing approach while having comparable cost. Indeed, when $\psi = 5$, the quadratic approach achieves a 16.5% higher spectral efficiency while being only 1.3% more costly than the operating-cost-minimizing approach. A similar trend can be observed as the UEs become more concentrated: if $\theta = 0.95$, the quadratic approach achieves a 15.1% higher spectral efficiency while being only 6.3% more costly.

Finally, we examine a scenario where routing costs are not negligible, but still low



(a) Spectral efficiency, $\theta = 0.75$

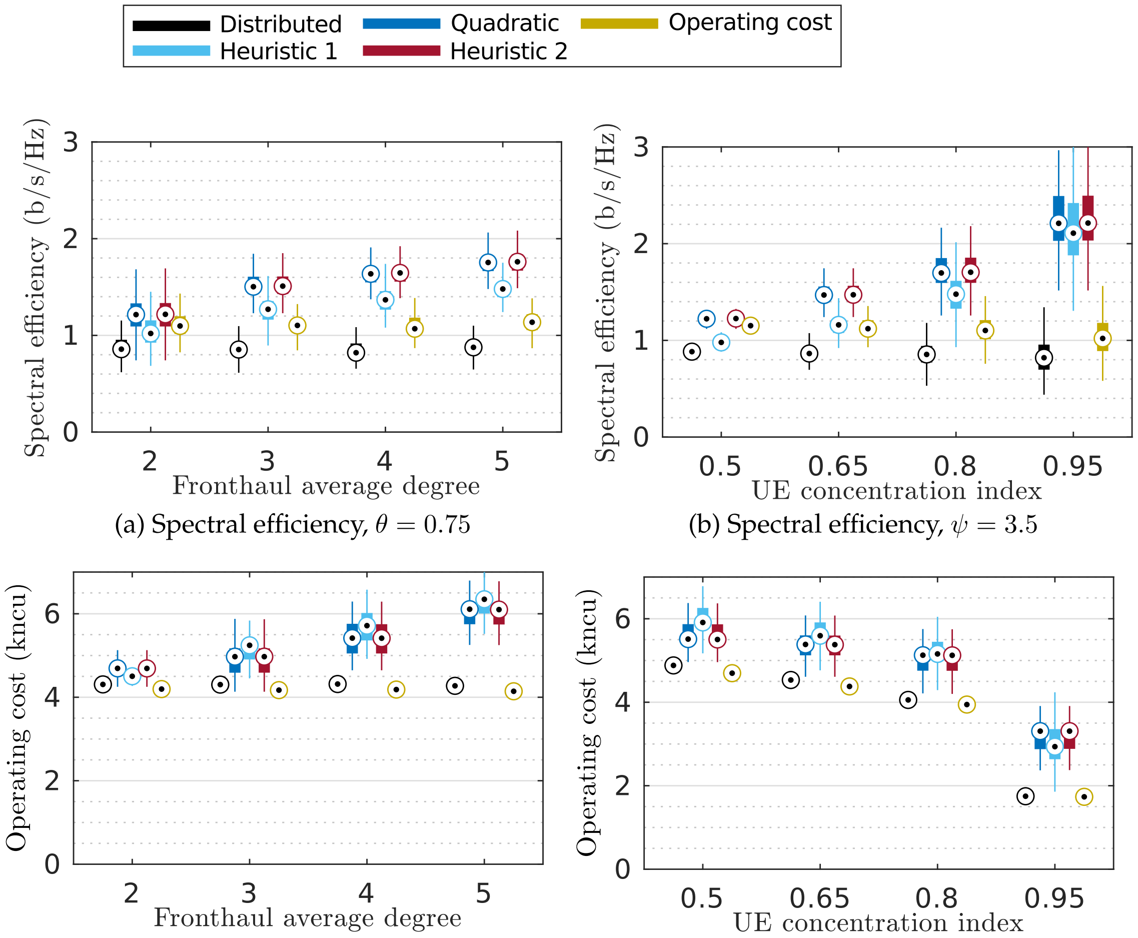
(b) Spectral efficiency, $\psi = 3.5$

(c) Relative operating cost, $\theta = 0.75$

(d) Relative operating cost, $\psi = 3.5$

Figure 4.13.: Spectral efficiency and relative operating cost achieved by fully distributed, quadratic, operating-cost-minimizing and heuristic approaches when $G = 300$ and $\omega = 0$ ncu · s/Gb.

enough so that a fully distributed configuration is not the least costly option. We set the routing cost to $\omega = 0.5$ ncu · s/Gb to illustrate this scenario, which is the midpoint between the value where the performance-maximizing approach becomes more costly than the fully distributed configuration ($\omega \approx 0.25$ ncu · s/Gb), and the value where the cost optimal approach converges to the fully distributed configuration ($\omega \approx 0.75$ ncu · s/Gb). The results are depicted in Fig. 4.14. For this scenario, we observe that the cost and spectral efficiency gaps between the operating-cost-minimizing and performance-maximizing approaches are noticeably larger than in the previous scenario, since the former is now very limited by the routing costs. When $\psi = 5$, the quadratic approach achieves a 54.4% higher spectral efficiency (Fig. 4.14a) while being 47.5% more costly than the operating-cost-minimizing approach (Fig. 4.14c). A similar trend can be observed as the UEs become more concentrated: if $\theta = 0.95$, the quadratic approach achieves a 116.7% higher spectral efficiency (Fig. 4.14b) while being 90.7% more costly (Fig. 4.14d). These results suggest that, when the routing costs



(a) Spectral efficiency, $\theta = 0.75$

(b) Spectral efficiency, $\psi = 3.5$

(c) Relative operating cost, $\theta = 0.75$

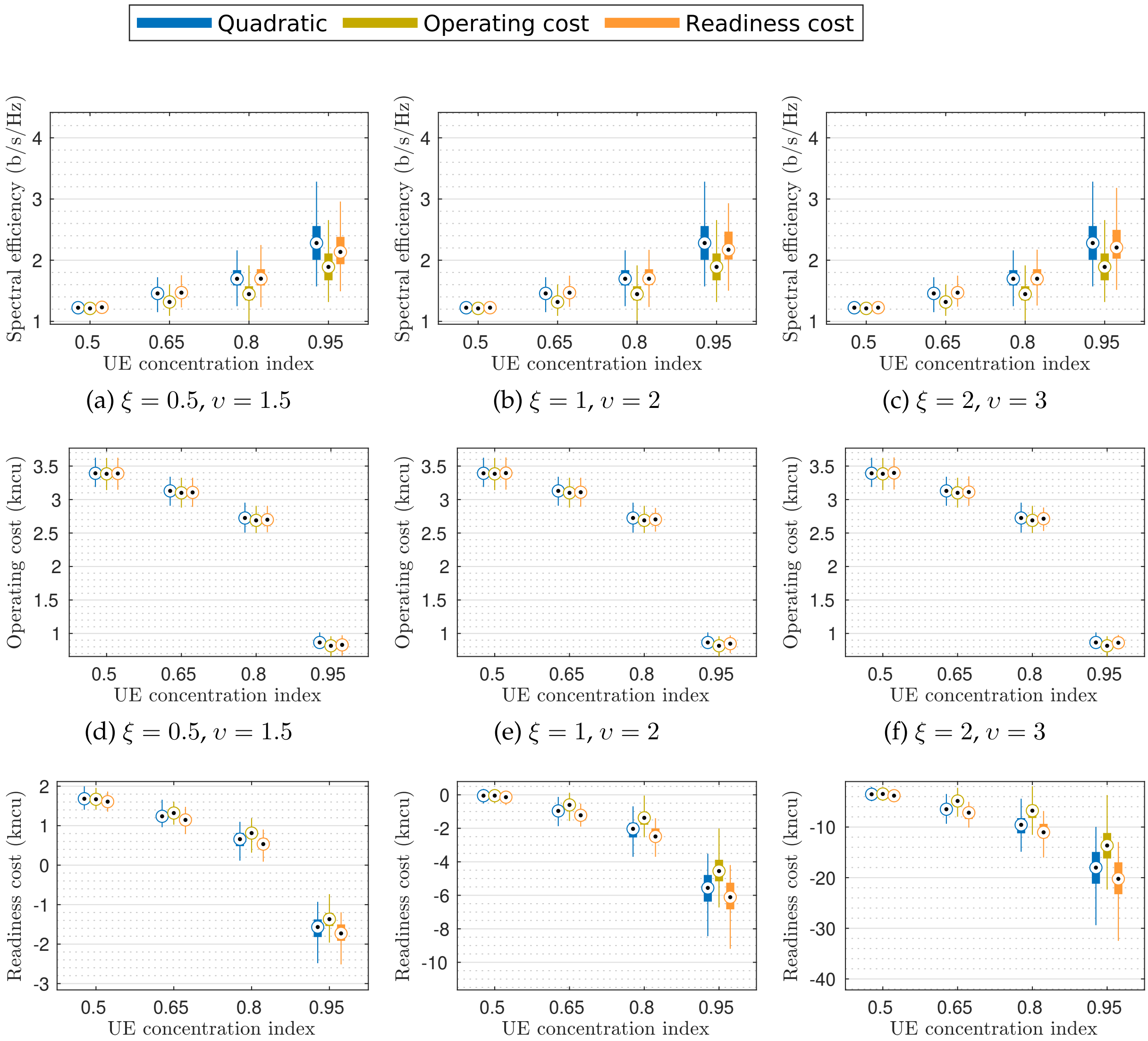
(d) Relative operating cost, $\psi = 3.5$

Figure 4.14.: Spectral efficiency and relative operating cost achieved by fully distributed, quadratic, operating-cost-minimizing and heuristic approaches when $G = 300$ and $\omega = 0.5$ ncu · s/Gb.

are high, performance-maximizing approaches can only compete with the operating-cost-minimizing one as long as the higher spectral efficiency translates into higher profit.

Readiness cost evaluation

Finally we compare the spectral efficiency, operating cost, and readiness cost achieved by the performance-maximizing, operating-cost-minimizing, and readiness-cost-minimizing interpretations of the FSSP. In Fig. 4.15 we show this comparison for $G = 300$, $\psi = 3.5$, negligible routing costs ($\omega = 0$ ncu · s/Gb), and three selected performance-to-revenue functions: a low-intercept, small-slope relationship $\langle \xi, \upsilon \rangle = \langle 0.5, 1.5 \rangle$; a medium-intercept, medium-slope relationship $\langle \xi, \upsilon \rangle = \langle 1, 2 \rangle$; and ; a large-intercept, large-slope relationship $\langle \xi, \upsilon \rangle = \langle 2, 3 \rangle$. For all cases, we observe that the spectral efficiency achieved by the readiness-cost-minimizing approach is similar to that of the



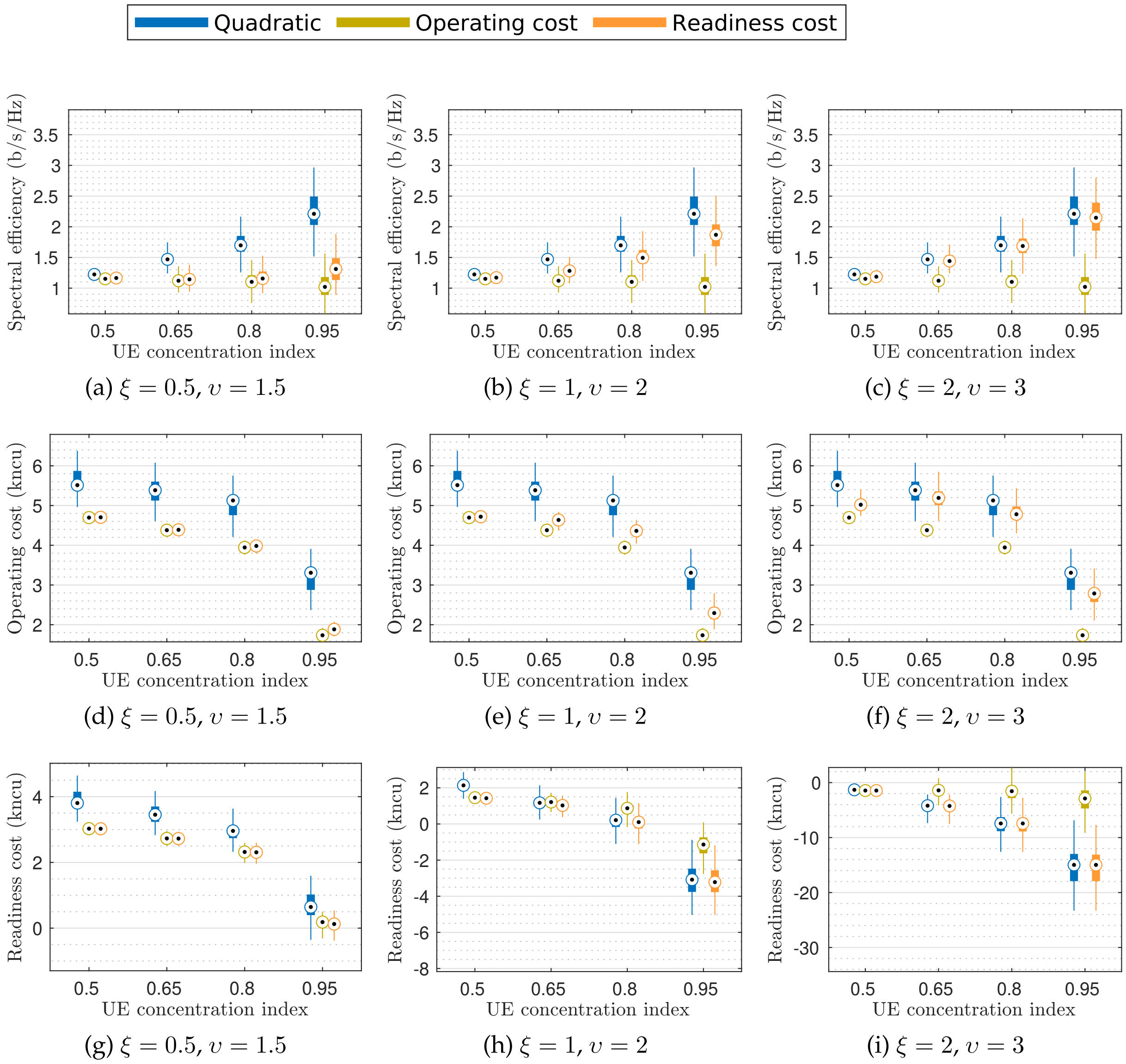
(a) $\xi = 0.5$, $\upsilon = 1.5$ (b) $\xi = 1$, $\upsilon = 2$ (c) $\xi = 2$, $\upsilon = 3$

(d) $\xi = 0.5$, $\upsilon = 1.5$ (e) $\xi = 1$, $\upsilon = 2$ (f) $\xi = 2$, $\upsilon = 3$

(g) $\xi = 0.5$, $\upsilon = 1.5$ (h) $\xi = 1$, $\upsilon = 2$ (i) $\xi = 2$, $\upsilon = 3$

Figure 4.15.: Spectral efficiency, operating cost, and readiness cost achieved by performance-maximizing (quadratic), operating-cost-minimizing, and readiness-cost-minimizing approaches when $G = 300$, $\psi = 3.5$, and $\omega = 0\,\mathrm{ncu} \cdot \mathrm{s/Gb}$.

performance-maximizing approach when the UE concentration index θ is small or moderate, and up to 5% lower on average for high UE concentrations. This implies that the achieved spectral efficiency is consistently higher than that of the operating-cost-minimizing approach. In addition, the operating costs of all approaches is very similar in all cases. Indeed, the maximum average difference between the operating cost of the readiness-cost-minimizing approach and the operating-cost-minimizing approach is less than 2%. Finally, we conclude that, when $\omega = 0$, the readiness cost of the performance-maximizing and readiness-cost-minimizing approaches is rather similar, although the latter achieves a 3% to 7% lower readiness cost when the UEs



(a) $\xi = 0.5, \upsilon = 1.5$ (b) $\xi = 1, \upsilon = 2$ (c) $\xi = 2, \upsilon = 3$

(d) $\xi = 0.5, \upsilon = 1.5$ (e) $\xi = 1, \upsilon = 2$ (f) $\xi = 2, \upsilon = 3$

(g) $\xi = 0.5, \upsilon = 1.5$ (h) $\xi = 1, \upsilon = 2$ (i) $\xi = 2, \upsilon = 3$

Figure 4.16.: Spectral efficiency, operating cost, and readiness cost achieved by performance-maximizing (quadratic), operating-cost-minimizing, and readiness-cost-minimizing approaches when $G = 300$, $\psi = 3.5$, and $\omega = 0.5$ ncu · s/Gb.

are highly concentrated. Overall, it is a remarkable result that, for negligible routing costs, the readiness-cost-minimizing approach behaves similarly to the performance-maximizing approach regardless of which performance-to-revenue function is in use (see Fig. 4.2).

In Fig. 4.16, we repeat the same experiments as in Fig. 4.15 but for non-negligible routing costs, that is, we set $\omega = 0.5$ ncu · s/Gb, as previously discussed. With respect to the previous experiments, we notice two main differences. First, the spectral efficiency, operating cost, and readiness cost of the readiness-cost-minimizing approach is not consistently similar to those of the performance-approach anymore. Second, the shape of the performance-to-revenue function does play a role in the behavior of the readiness-cost-minimizing approach in this case. Namely, for a low-intercept, small-slope performance-to-revenue function $\langle \xi, \upsilon \rangle = \langle 0.5, 1.5 \rangle$, the readiness-cost-minimizing approach resembles the operating-minimizing-approach more than the performance-maximizing approach. This is consistent with our previous observation that, when $\omega = 0.5$, the performance and cost gaps between the performance-maximizing and operating-minimizing-approaches are more similar than when $\omega = 0$. Conversely, for large-intercept, large-slope performance-to-revenue function $\langle \xi, \upsilon \rangle = \langle 2, 3 \rangle$, the improved performance plays a more relevant role in the total revenue, and thus in that case the readiness-cost-minimizing approach resembles the performance-maximizing approach. In conclusion, this confirms that our readiness-cost-minimizing approach can indeed adapt to a wide range of network conditions and yield lower readiness cost than any alternative approach.

4.8. Summary

In this chapter, we tackle the problem of finding the optimal functional split that optimizes an instantaneous network scenario, building upon the static FSSP formulation presented in Chapter 2. We provide three different optimization objectives, which result in three FSSP formulations: a performance-maximizing FSSP, an operating-cost-minimizing FSSP, and a readiness-cost-minimizing FSSP that combines the other two.

We first tackle the performance-maximizing FSSP, which leads to a non-linear optimization problem. We present four fractional reformulations, which can be converted into MILPs: the punctured and unpuntured versions of the LWT and BGP transformations. In addition, we propose a simpler reformulation into a quadratic program, which is then linearized too. We also provide two heuristic approaches: Heuristic 1, which exploits the correlation between inter-cell interference and centralization level, and Heuristic 2, which improves upon the quadratic approach.

The operating-cost-minimizing FSSP is simpler to address, since it can be directly expressed as an MILP. However, for the readiness-cost-minimizing FSSP we need to translate network performance into revenue. In order to cover a wide range of cases, we use a family of linear functions for this translation, whose intercept and slope can be configured. After this, we combine previous performance-maximizing and operating-cost-minimizing MILP approaches to obtain the readiness-cost-minimizing FSSP.

In order to assess the viability and performance of the proposed formulations and solving approaches, we simulate multiple realistic scenarios using the size of the network G, average fronthaul degree ψ, and the UE concentration θ as control variables. We first focus on evaluating the trade-off between achieved spectral efficiency and convergence time for each approach, since our ultimate intention is to extend these approaches and use them for dynamic adaptation. We observe that the fractional reformulations of the performance-maximizing approach lead to the best spectral efficiency, but this is less than a 2% improvement over the quadratic approach in all scenarios. Moreover, they may take 15 minutes or more to converge for networks with more than $G = 80$ gNodeBs, thus rendering these approaches unsuitable for dynamic adaptation. Conversely, Heuristic 1 converges in less than 200 ms even for large networks ($G = 300$), but the spectral efficiency achieved by this approach can be up to a 20% smaller than the quadratic approach, specially when the fronthaul network is sparse ($\psi = 2$). As a result, the quadratic approach seems specially promising since it often requires less than 5 s to converge, even when $G = 300$, and provides a spectral efficiency comparable to that of fractional approaches. Regarding the other FSSP formulations, the operating-cost-minimizing formulation converges in less than 30 ms on average, whereas the readiness-cost-minimizing approach takes between 0.7 to 2.5 seconds to converge when using the quadratic approach.

We conclude that for sparse, uniformly-populated networks ($\psi = 2$, $\theta = 0.5$) the benefits of implementing a performance-optimized solution are marginal, since the spectral efficiency improvement of the quadratic approach over the static reference is only ca. 10%. Nonetheless, for dense fronthaul networks ($\psi = 5$) this improvement reaches 36%. If the routing costs are moderate ($\omega < 1$), we observe that performance-maximizing approaches can leverage the benefits of fronthaul density better than both static and operating-cost-minimizing approaches, since they achieve better spectral efficiency than both of them while requiring an operating cost comparable to that of the operating-cost-minimizing approach. However, when the routing costs are high ($\omega \geq 1$), performance-maximizing approaches can only compete with the operating-cost-minimizing one as long as the higher spectral efficiency translates into higher profit. Something similar happens with the concentration of UEs. We conclude that a performance-maximizing approach may achieve substantially better spectral efficiency when UEs are concentrated with respect to a static solution, reaching a 48% improvement when $\theta = 0.8$ and a 90% improvement for $\theta = 0.95$.

Finally, we observe that the additional cost of performance-maximizing approaches over the operating-cost-minimizing approach translates into proportionally higher improvements in the spectral efficiency. As a result, if the operator is able to profit from these improvements, then performance-maximizing approaches may lead to a higher revenue with respect to static or operating-cost-minimizing approaches. This is confirmed after evaluating the performance and operating cost of the readiness-cost-minimizing approach. We conclude that, when the routing costs are negligible, the readiness costs of the performance-maximizing and readiness-cost-minimizing approaches are rather similar, although the latter achieves a 3% to 7% lower readiness cost when the UEs are highly concentrated. For non-negligible routing costs, the readiness-cost-minimizing approach may still resemble the performance-maximizing approach if the revenue function is steep, although it converges to the operating-cost-minimizing approach if this is not the case. In conclusion, this confirms that our readiness-cost-minimizing approach can indeed adapt to a wide range of network conditions and yield lower readiness cost than any alternative approach, while still requiring a very low convergence time. Therefore, the readiness-cost-minimizing approach is a perfect candidate for being used in dynamic optimization.

5. Implementation of an Adaptive Functional Split

5.1. Introduction

5.1.1. Motivation, scope, and challenges

In Chapters 2, 3, and 4 we motivate, propose, formulate, and present strategies to solve the problem of dynamically selecting the optimal functional split for a instantaneous network configuration, with the objective of maximizing the experience of the users without overloading the fronthaul network. We show the relevance of this problem, observe that there are efficient strategies to tackle it, and conclude that a network featuring an optimally adaptive functional split outperforms non-adaptive deployments in terms of experienced user data rates and may even lead to lower operating costs. Nonetheless, any advantages that an adaptive functional split may have depend on a critical feature: changing the functional split during runtime must be feasible, fast, and cost-efficient. That is, if changing the functional split were too expensive, took too long, or were technically infeasible, it would be impossible to benefit from dynamically adapting the functional split.

Being specific to 5G and late 4G networks, the concept of a functional split in the RAN is, on its own, rather novel. Indeed, in Sec. 2.2.4 we mention that, although it is generally agreed that a functional split should be beneficial in terms of user performance and operating cost, there is relatively little standardization work regarding how to realize the interfaces between separated functions. Consequently, with the current state of the art, it would be intractable to assess the feasibility of adapting from and to every possible functional split, since this would require defining new, dedicated protocols modeling all interactions between RAN functions. Nonetheless, we can indeed focus on the most promising functional split options, which receive the most attention from the standardization and research communities, and build upon previous work to propose a proof-of-concept RAN that can switch between two distinct functional split options. This proof-of-concept RAN can be used to show that changing the functional split during runtime is actually feasible and also to identify potential problems in more complicated systems, thus paving the way for advanced functional split adaptations.

Even if we only consider a small subset of functional split options in our proof-of-concept, assessing the feasibility of being able to implement several of these options and also supporting live changes in the centralization level without severely disturbing normal operation is a very challenging task. Namely, we first have to devise a manner to support two or more functional split options so that they can work independently without interfering with each other. Second, we must design a procedure to migrate from one functional split option to the other during runtime. Finally, we need to ensure that there is no noticeable downtime, and packet losses and additional latency are minimized during the functional split migration. In this chapter, we present a pioneer implementation of a proof-of-concept 5G RAN that is able to switch between the PDCP-RLC and MAC-PHY splits during runtime with minimal impact on the network operation.

5.1.2. Key contributions

In order to address the aforementioned challenges, this chapter bases on our previous works [MAGVK19a] and, to a lesser extent, [MAGVK19b] to feature the following contributions:

1. We evaluate the possibilities to implement a proof-of-concept RAN featuring a dynamically adaptive functional split, and select the most promising options. Then, we combine and extend previous works to allow for a simultaneous implementation of multiple functional split options within a single gNodeB.
2. We identify and discuss the objectives, challenges, and possible strategies to change the functional split during runtime with minimal impact on normal operation.
3. We explain a detailed procedure to migrate between the selected functional split options that overcomes the identified challenges by defining and transferring the instantaneous state of the base station from the CU to the DU, and vice versa. A description of the additional required functions and their impact on the migration procedure is also included.
4. We implement the proposed migration procedure on an actual 5G RAN testbed and provide measurements of the internal operation of the buffers during a migration, as well as the impact of migrating on end-user latency and packet losses, so as to ensure that the migration does not severely affect the operation of the RAN.

The rest of this chapter is organized as follows. Sec. 5.2 summarizes the current state of the art on the topic, including academic research and existing 4G/5G RAN implementations, upon which we build our proof-of-concept RAN. In Sec. 5.3, we briefly discuss the functional split options and select those that are used in our proof-of-concept. Sec. 5.4 presents the objectives and challenges of implementing an adaptive functional split, as well as a detailed description of our proposed migration procedure. In Sec. 5.5, we describe our implementation of the proof-of-concept RAN and provide measurement results. Finally, Section 5.6 summarizes the contributions and findings of this chapter.

5.2. Related work

In this section, we divide previous works related to the content of this chapter into two categories. On the one hand, we present a summary of relevant academic research that considers or discusses 5G RAN designs whose functional split can be changed, either from a theoretical or a practical point of view. On the other hand, since this chapter focuses on implementation details, we describe the most relevant software platforms that can be currently used to implement a 5G RAN, since our proof-of-concept RAN actually builds upon one such platform.

5.2.1. Academic research

The advantages and challenges of functional splits in the architecture of the 5G RAN have triggered the research on their optimal definition and selection. As mentioned in previous chapters, we can find abundant work on the characteristics of diverse functional splits, such as [Döt+13], [LCC18], [Bar+15], or [Val+16]. Owing to their differences, the authors of these works suggest that the functional split should be adapted to the external conditions that the network experiences. To this end, [Mae+16] presents a high-level overview of a flexible 5G RAN, which includes the ability to support multiple functional splits, in order to adapt the RAN configuration to the expected user traffic. Building upon this, in [Cha+17a] the authors propose a RAN architecture that simultaneously supports different functional splits for each DU in the network. However, none of these works explicitly consider changing the functional split on-the-fly, but rely on a priori statistics of the network to select the optimal, static functional split.

Conversely, there are works which do tackle, to a greater or lesser extent, the change of RAN functions at runtime. For instance, in FlexRAN [Fou+16], a novel implementation of a software-defined RAN is presented, featuring a virtualized RAN controller whose location and configuration can be dynamically changed. In FlexCRAN [Cha+17b] the authors propose a framework for a partially centralized RAN that supports on-the-fly changes of the functional split, although they do not elaborate on this feature. In [HR18a], the problem of dynamically selecting the functional split for each cell is addressed. The authors present an algorithm that allows virtual mobile operators to change the functional splits every time a new virtual network is added. Similarly, [Mar+18] proposes an architecture for a 5G RAN implementing an algorithm to dynamically select the functional split of a cell. This work puts special emphasis on enabling changes at runtime. Nonetheless, it only covers the optimal selection of the functional split, regardless of the feasibility of the required changes. Finally, in [Alf+18] the authors present a pioneer platform that can switch between functional splits at runtime. However, the options are limited to low-layer, intra-physical splits, and the details about performing the switching are not explained in detail.

In the light of these previous works, we conclude that our proof-of-concept RAN is, to the best of our knowledge, the first work addressing the implementation of a network that can adapt its functional split at runtime. As a result, we also pioneer the discussion of objectives and challenges, and the provision of actual measurement results.

5.2.2. 4G/5G Software Platforms

Implementing a mobile RAN whose functional split can be changed during runtime requires an underlying 4G/5G software platform to realize the remaining elements of the mobile network. Fortunately, in recent years, some remarkable initiatives have emerged with the intention of providing full-stack implementations of 3GPP mobile networks, such as OpenAirInterface [Oped], srsRAN [SRSa], OpenLTE [Ben], and OpenBTS [Ran]. In this section, we briefly introduce the two most relevant ones: OpenAirInterface and srsRAN.

OpenAirInterface

OpenAirInterface [Nik+14; Oped] is an open-source software platform that implements a complete 3GPP mobile network, including radio access and core networks, which can be used on commercial off-the-shelf equipment, such as standard PCs, software-defined radios (SDRs), smartphones and LTE dongles, etc. It provides a complete set of C/C++ libraries and binaries that emulate the operation the most relevant functions, protocols, and units of a mobile network, so that a full mobile network can be replicated and experimented with in standard laboratory conditions. For instance, by using OpenAirInterface it is possible to operate two conventional PCs and SDRs as if they were a UE and a 3GPP base station, with support of the most important internal protocols and interfaces. This allows to evaluate potential modifications in the operation of any component in the mobile network with ease and without requiring to invest in expensive, dedicated equipment.

OpenAirInterface is developed by the OpenAirInterface Software Alliance (OSA), which was founded in 2014 by EURECOM and comprises more than 85 members, including mobile operators (such as Orange), hardware manufacturers (such as Qualcomm, Xilinx, and Fujitsu), research companies (such as Nokia Bell Labs) and universities (such as the Sorbonne University and the Technical University of Munich) [Opef]. The OSA is responsible for deciding on the development roadmap and milestones of OpenAirInterface, providing quality control, and promoting the produced software in both academic and industrial communities.

The OpenAirInterface project is divided into two sub-projects: one focusing on the core network and another focusing on the RAN. Early versions of the OpenAirInterface core network implement the legacy LTE components of the Evolved Packet Core (EPC) as software functions, including the Mobility Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (S-GW), and PDN Gateway (P-GW). Standard interfaces connecting these functions, such as the S11 interface between the MME and S-GW, the S6a interface between the HSS and MME, and the S5 interface between S-GW and P-GW are also supported, thus facilitating development and providing interoperability with other EPC implementations. Moreover, in September 2020, the OSA presented a new, open implementation of the service-based 5G core architecture, with the intention of eventually supporting a stand-alone deployment of a 5G network. This implementation of the 5G core comprises the Access and Mobility Management Function (AMF), the Authentication Server Management Function (AUSF), the Network Repository Function (NRF), the Session Management Function (SMF), the Unified Data Management function (UDM), the Unified Data Repository (UDR) and the User Plane Function (UPF). Although this implementation is, by September 2021, still under development, it already provides the majority of the core functions required to run a simple 5G core network [Opec; Opee].

The OpenAirInterface RAN is structured into three parts, which roughly follow the protocol stack separation in 4G and 5G networks. The first part complies with the physical layer specification described in 3GPP LTE Release 8.6 and includes some features of Release 10. This translates into a maximum of 20 MHz channel bandwidth, two transmission and reception antennas, and a throughput of 70 Mb/s [Opea]. In addition, this first part also includes the required code to operate off-the-shelf SDRs, such as the Eurecom EXMIMO II, Ettus B210/X300, BladeRF, and LimeSDR. The second part includes MAC, RLC, PDCP, RRC layers as described in 3GPP LTE Releases 8.6, 9.3, 10.1, and 14.3, respectively. The MAC layer comprises a scheduler function implementing a proportionally fair allocation of radio resources and fully supports hybrid ARQ (HARQ), link adaptation, and power control [Opea]. The RLC layer implements all three transmission modes (transparent, unacknowledged, and acknowledged), whereas the PDCP layer supports packet sequencing, integrity check, and encryption using AES or Snow3G algorithms, but not header compression [Opea]. The RRC layer supports generation and broadcast of system information blocks 1, 2, 3, and 13 as well as standard RRC connection establishment, reconfiguration, release, and re-establishment procedures, but handover and paging procedures are still in development [Opea]. Finally, the third part deals with implementing the S1 application protocol between the base station and the MME as well as the user-plane Non-Access Stratum (NAS) GPRS Tunneling Protocol (GTPV1-U) for both base stations and UEs as specified in 3GPP LTE Release 10.

The OpenAirInterface RAN currently supports splitting a base station into three units: the Radio-Cloud Center (which can be identified with the CU in 5G networks, since they share the same objective), the Radio-Access Unit (equivalent to the DU), and the Remote Radio Unit (similar to the RU). A partial support for the MAC-PHY split between the Radio-Access Unit and the Remote Radio Unit is included, based on the nFAPI interface proposed by the Small Cell Forum [Sma17]. It is also planned to implement PDCP-RLC functional split between the Radio-Cloud Center and the Radio-Access Unit as specified in interface F1 of 3GPP Release 15 [3GP21j], although this feature is not fully supported yet [Opea].

Since the research community is currently interested in testing and developing fifth-generation features, it is planned that by end of 2021 OpenAirInterface will support a completely stand-alone implementation of a 5G RAN [Opeb]. This will entail improvements and additions to the physical layer in order to support new radio (NR) transmission and reception, as well as modifications to the MAC and RLC layers so as to support bandwidth parts, among other novelties. Finally, the implementation of the nFAPI interface enabling the MAC-PHY split will be completed so as to allow for combined exchange control and data information, and also the support of improved MIMO communications.

srsLTE/srsRAN

srsRAN [SRSa], formerly known as srsLTE, is an open-source software suite which, as OpenAirInterface, intends to provide a full-stack implementation of a 4G/5G mobile network, including core network, RAN, and UEs for off-the-shelf hardware equipment. The srsRAN suite contains a collection of libraries, modules, and tools written in C/C++ that can be compiled on standard Linux PCs to use alone or in combination with third-party 4G/5G software. Like OpenAirInterface, srsRAN can be used to replicate a complete mobile network using only conventional PCs and SDRs.

srsRAN is developed by Software Radio Systems (SRS) and started in 2015 as a full-stack UE application [SRSc]. In 2017 and 2018, SRS developed eNodeB and EPC implementations, respectively, to complement the UE application, thus producing a complete implementation of a 4G RAN. Since then, srsRAN has incorporated several advanced features, such as MIMO, mobility management, and carrier aggregation. Although software development is mainly performed by SRS, in contrast to the multiple contributors to OpenAirInterface, the srsRAN suite is actively used and improved by third-party organizations, such as National Instruments, Analog Devices, Nokia, and the Massachusetts Institute of Technology [SRSa].

The srsRAN suite consists of three main components: srsUE (implementing the UE protocol stack), srseNB/srsgNB (implementing the base station), and srsEPC (implementing core network functions). The first component, srsUE, runs as a Linux PC application to replicate the behavior and internal operation of a 3GPP LTE Release 10 UE, while also including some features of a 3GPP NR Release 15 UE. As OpenAirInterface, srsUE supports a maximum channel bandwidth of 20 MHz for a single carrier, but it is also able to use carrier aggregation to combine the capacity of up to two carriers. It also includes support for evolved multimedia broadcast and multicast service (eMBMS) [SRSb].

The second component, srseNB/srsgNB, implements a base station in accordance to the specifications of 3GPP LTE Release 10. In contrast to OpenAirInterface, its support of carrier aggregation enables a downlink throughput of up to 150 Mb/s, albeit it also features some limitations like implementing frequency-division duplex (FDD) but not time-division duplex (TDD) [SRSb]. Although it does not include out-of-the-box support for functional splits, in srsRAN the interfaces between PHY, MAC, RLC, PDCP, and RRC layers are well-defined and implemented as dedicated classes in the source code, which facilitates the experimentation with custom functional splits.

The third component, srsEPC, is a lightweight implementation of an LTE EPC, including MME, HSS, S-GW, and P-GW. Whereas in OpenAirInterface each function requires a dedicated binary, in srsEPC all functions are implemented as a single binary [SRSb]. This allows for a simpler operation of the core functions while still providing clear distinction among functions in the source code.

5.3. Considered functional splits

As mentioned in Chapter 2, the 3GPP envisions up to eight different options for the functional split in a 5G RAN [3GP17]. These correspond to the interfaces of the five layers of the protocol stack (RRC-PDCP, PDCP-RLC, RLC-MAC, MAC-PHY, and PHY-RF), plus three internal splits within the RLC, MAC, and PHY layers. However, as mentioned in Sec. 2.2.4, not all options are equally beneficial in actual implementations. For instance, the RLC-MAC split is deemed almost useless, as it is more complex than PDCP-RLC but does not bring any additional benefit [3GP17]. Similarly, the RRC-PDCP or RRC-SDAP splits have almost identical requirements to PDCP-RLC without providing any major advantage. The Intra-RLC and Intra-MAC splits may, conversely, outperform the PDCP-RLC split since they would feature higher flow control and better resource pooling, and enable interference coordination. Nevertheless, currently there are no well-established initiatives to standardize these split options. Finally, the Intra-PHY and C-RAN splits promise good performance and count with standardizing initiatives [EAN19], but their timing and capacity constraints are so high that render practical experimentation very difficult. As a result, in this chapter, we focus on an adaptive 5G RAN that can switch between PDCP-RLC and MAC-PHY splits. The selection of these two splits is based on their advantages regarding low fronthaul utilization and good interference-coordination capabilities, respectively, their relative simplicity, and the fact that there are already initiatives to standardize their required interfaces, which simplifies development [Mak+17]. In the following sections, we present them and summarize their features.

5.3.1. PDCP-RLC

In this split, the PDCP function is centralized in the CU, whereas the RLC, MAC, PHY, and RF functions are located in the DU. This has three main advantages. First, it reduces the operating cost with respect to a distributed architecture, since the PDCP function is in charge of ciphering, which may be a computationally intensive task. Second, the fronthaul traffic is very similar to the user traffic, as only a small PDCP header is added to each IP packet. Thus, the fronthaul traffic is also comparable to the backhaul traffic in LTE, which enables reutilization of backhaul networks. That is, operators could reuse their former backhaul infrastructure as fronthaul network. Finally, the standardization effort required to implement this split is low, given that it has already been considered in the past for LTE Dual Connectivity [3GP17].

Owing to its advantages, PDCP-RLC is the split currently considered by the 3GPP Releases 15 and 16 for the NR specifications [3GP21j]. Nonetheless, the PDCP-RLC lacks enough centralized functions to perform any kind of advanced coordination technique with other gNBs, and the DU is still required to implement the computationally intensive MAC and PHY layers. Therefore, a more centralized split would be a better option if the fronthaul capacity allows it.

5.3.2. MAC-PHY

In this split, the PDCP, RLC, and MAC functions are centralized in the CU, whereas the PHY, and RF functions are located in the DU. More centralized functions mean less operating costs with respect to PDCP-RLC. Furthermore, coordination techniques such as coordinated scheduling or coordinated link adaptation are possible with a centralized MAC [3GP17], while the fronthaul traffic is still considerably lower than those of Intra-PHY or C-RAN splits [MAGVK19b].

Due to its characteristics, the MAC-PHY split has been selected by the Small Cell Forum for their envisioned network, and standardized in the nFAPI initiative [Sma17]. Nevertheless, the MAC-PHY split has also disadvantages. On the one hand, the presence of additional headers and control signals between MAC and PHY layers increases the fronthaul capacity and latency requirements with respect to PDCP-RLC [MAGVK19b]. On the other hand, its coordination capabilities are limited with respect to C-RAN or Intra-PHY splits.

5.4. Adaptive functional split

In this section, we provide an overview of the objectives and challenges that an adaptive 5G RAN faces. Furthermore, we introduce the details of our solution, which implements an adaptive functional split between PDCP-RLC and MAC-PHY.

5.4.1. Objectives and challenges

Given that a centralized architecture outperforms a distributed one in terms of cost and coordination capabilities, the objective of an adaptive 5G RAN is to make sure that each gNB in the network operates at the most centralized functional split that is supported by the fronthaul network. Since the load of the fronthaul network depends on the user traffic, which may change over time, the functional split should also be able to change at runtime. The main difference between functional splits is the location of the functions of the 5G processing chain, as explained in Sec. 2.2.4. As a result, switching between functional splits at runtime is equivalent to live migrate functions from the CU to the DU, or vice versa.

There are at least two main obstacles when live migrating RAN functions: increased fronthaul traffic and function downtime. The former refers to the additional information that needs to be exchanged between CU and DU during the migration, which leads to an increase in the fronthaul traffic. The latter is the time during which the functions being migrated are not available, owing to a possible need of halting these functions in order to complete the migration. These two obstacles lead to two secondary objectives. On the one hand, the overhead traffic during the migration should be minimized, as the sheer motivation of the migration may be the reduction of the traffic on the fronthaul. That is, if the user traffic increases and the centralization level has to be reduced in order to decrease the fronthaul load, a migration that produces high overhead would be counterproductive. On the other hand, the downtime of the migration should be minimized as well, since it may negatively impact the experience of the user. This is specially concerning for the communication between low layers, as any interruption may result in missing a whole subframe. For instance, in our case, a downtime between PDCP and RLC layers translates into a delayed transmission of PDCP packet data units (PDUs), which may be a problem for low-latency users. Moreover, an interruption of the communication between MAC and PHY layers cannot be higher than the scheduling interval, or else entire transmission and reception slots would be wasted. In 5G, the slot duration ranges from 1 ms to $62.5, \mu$s, so very small downtimes are tolerated.

5.4.2. Migration strategy

As mentioned above, any change in the functional split implies moving functions from the CU to the DU, or vice versa. In our case, the difference between MAC-PHY and PDCP-RLC is the location of the MAC and RLC functions. In the MAC-PHY split, these functions are located in the CU, whereas in PDCP-RLC split they are located in the DU. Hence, when switching from PDCP-RLC to MAC-PHY, we need to move the MAC and RLC functions from the DU to the CU, and vice versa. In order to do this, we need an underlying migration strategy that fits the characteristics of the functions and the requirements of the network.

The live migration of functions is a well-studied topic in the field of network function virtualization (NFV). Indeed, a common strategy to live migrate a function is to deploy it on a virtual machine and then migrate the virtual machine [CC14]. A virtualization platform is hence needed to manage the migration, since the hardware of CU and DU may be different. For instance, platforms like OpenStack, based on hypervisors such as KVM [KVM] or Xen [Theb], allow for such live migrations [Thec]. However, this *virtualization-based* approach conflicts directly with the two limitations presented in the previous section. To the best of our knowledge, no existing virtualization platform can offer a downtime comparable to the scheduling interval of a 5G network (1 ms or less) [BNK16; Bas+19; Tor+21]. Moreover, live migrating a virtual machine often entails copying its disk and memory to the destination, thus producing a high traffic overhead until the migration is completed [Tor+21].

A faster, lighter type of live migration is therefore needed to change functional splits in a 5G RAN. We propose a *replication-based* approach, in which the MAC and RLC functions are simultaneously deployed in both DU and CU. That is, at every instant there is one active set of MAC and RLC functions at either the DU or the CU, and an inactive set at the other unit. When the migration is performed, the roles are exchanged: the active functions are disabled, and the inactive functions are enabled. This approach can be considerably faster and produces much less overhead during the migration than the virtualization-based approach. Its main drawback is that it requires to have MAC and RLC processes running simultaneously on both units, even when they are not used. This creates additional memory and CPU consumption on the inactive unit, which should be taken into account as operating expenses. This drawback is discussed in more detail in Chapter 6. Furthermore, in order to provide uninterrupted service to the users, the MAC and RLC functions cannot be just turned on or off. In fact, there are multiple state variables and data structures that are created, modified, and used by both functions during runtime. These have to be carefully transferred to the destination before completing the migration. In the next section, we address the mater of transferring the state of a base station to switch functional splits.

5.4.3. State transfer

In a replication-based migration, there are three basic tasks to accomplish: (i) transfer the *state* from the old set of functions to the new one, (ii) deactivate the old set of functions, and (iii) activate the new set of functions. That is, before being able to toggle the two sets of functions on and off, we must guarantee that they are in the same state. In this work, we define the state of the MAC and RLC functions as the set of parameters that are required for the correct operation of these functions and are susceptible to change over time. Note that this state is defined within the scope of the internal operation of a base station and should not be confused with the concept of network state presented in Chapter 3.

We can classify the parameters defining the state of the MAC and RLC functions into three types, according to their origin and updating frequency. First, we have the *external parameters*, which are used by the MAC and RLC layers but not created or modified by them. These are mostly details of the connected UEs and defined radio bearers provided by either the UEs or by higher layers of the gNB. Some examples of external parameters are logical channel IDs (LCIDs), Radio Network Temporary Identifiers (RNTIs), or timer values. These parameters do not change frequently, as they are only created or modified when a UE connects or disconnects, or when the higher layers decide so. Second, the *internal parameters* are those created or modified by the MAC and RLC layers themselves, such as the RLC sequence number, frame and subframe numbers, and the list of active HARQ processes. In general, these parameters change every slot or subframe. Finally, we consider the *content of the buffers* as the last type of state parameters, including the RLC transmission and retransmission buffers, and the HARQ retransmission buffer. The content of these buffers varies every time slot, after transmission and reception, or when new data is received.

Given their different characteristics, the three types of parameters should be handled differently in order to minimize the overhead and downtime of the migration. For instance, the external parameters can be forwarded to both active and inactive RLC and MAC functions every time they are updated, e.g., when a new UE connects or disconnects. This results in a very small overhead traffic following these events, but since it is performed ahead of time, it reduces the amount of data exchanged during the migration. Conversely, the internal parameters, owing to their fast updating frequency, have to be transmitted during the migration, as only the active MAC and RLC layers are aware of the most updated values. The amount of overhead traffic caused by transferring these internal parameters during a migration is, however, almost negligible, as only a few bytes per UE are needed to store them. Namely, up to two bytes for the RLC sequence number, two bytes for frame and subframe numbers, and one byte for the HARQ processes [3GP21b; 3GP21f].

In contrast, transferring the content of the buffers is a more challenging task. According to the 3GPP specifications for 5G [3GP21k], the RLC retransmission buffer should store from 50 to 160 RLC PDUs in order to account for the maximum expected acknowledgment delay. This implies that, if the content of this buffer is transferred in one scheduling interval (in order to avoid downtime), the overhead traffic produced during the migration would be between 50 and 160 times higher than the downlink data rate of the air interface, which is actually comparable to the capacity needed for full centralization. Indeed, in [3GP17] the capacity required for the C-RAN split of in one gNB is estimated to be around 40 times larger than the aggregated user traffic. Hence, an adaptive functional split would be pointless with such a high overhead, since, if it could be supported, full centralization would probably also be supported, thus removing the need for performing a functional split.

In order to reduce the overhead, a solution would be to put both active and inactive buffers into a common pre-defined state just before the migration, instead of transferring the content. As the packets stored in the buffer cannot be modified, this can only mean to empty the buffers before completing the migration. There are two options to achieve this: either to drop the content of the old buffers, or to redirect new packet arrivals to the new buffers while the old buffers drain normally. The former, or *hard migration*, provides the fastest migration with no overhead traffic, although it implies packet losses. The latter, or *soft migration*, prevents packet losses, but it may introduce delay while the old buffers are emptied, and produces an overhead traffic equal to the arrival rate of new user packets. In addition, a combination of both migration options, or *custom migration*, can be also defined. For this option, the old buffers are given a fixed amount of time to empty, after which all remaining content is dropped. Therefore, a custom migration provides an adjustable trade-off between latency and packet losses.

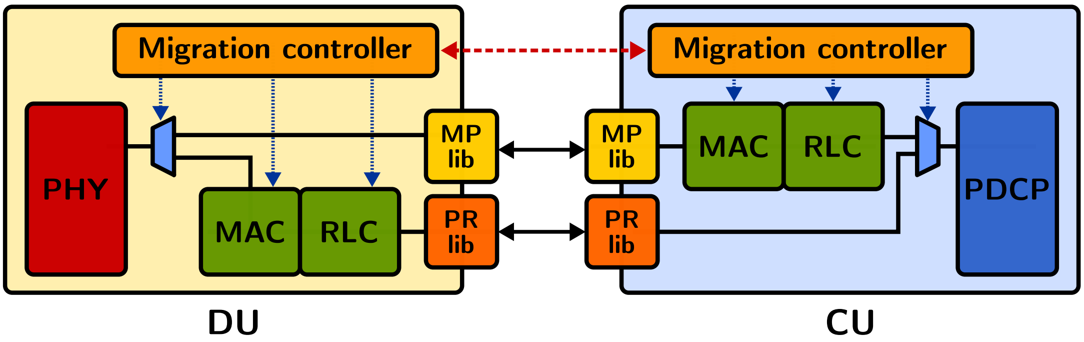


Figure 5.1.: Functional architecture of the two binaries supporting an adaptive functional split. The blue trapezoid symbolizes the ability of the migration controller to steer the flow of data.

5.4.4. Migration platform

With the objective of testing the viability of the aforementioned migration scheme, we implemented it using a modified version of srsRAN as software platform [GM+16]. As commented in Sec. 5.2.2, both OpenAirInterface and srsRAN would be good options to implement our proof-of-concept. OpenAirInterface, for example, already provides partial implementation of the MAC-PHY and PDCP-RLC splits in its original code. Nonetheless, since it features multiple advanced characteristics, realizing modifications on its code can occasionally be more challenging than on that of srsRAN, which is relatively simpler. Although srsRAN does not provide functional splits out of the box, its code is clearly organized into layers, thus facilitating the implementation of functional splits. As a result, even though OpenAirInterface would also be a valid option to realize a similar proof-of-concept, in this work we choose srsRAN as the software platform. We split the original, monolithic srsRAN binary that contains all the RAN layers into two different 5G binaries: one containing the PHY, MAC, and RLC functions for the DU, and another hosting the MAC, RLC, PDCP, and RRC functions for the CU (see Fig. 5.1). The code implementing the MAC and RLC functions can be disabled during runtime in both binaries, thus providing the software basis for switching functional splits.

In order to make these two binaries work with each other, interfaces for the PDCP-RLC and MAC-PHY splits need to be defined. For the sake of simplicity, these interfaces are implemented as libraries whose purpose is to convert the original function calls between layers into Ethernet packets. This is done by means of Google Protocol Buffers [Goo], which is a tool that helps transforming C++ objects into serialized data. The two libraries are known as the *PR library* for the PDCP-RLC split, and the *MP library* for the MAC-PHY split. The PR library performs a pure serialization of the objects exchanged by the srsRAN function calls, which already resembles the application protocol of the F1 interface in 5G [3GP21j], whereas the MP library is explicitly customized to follow the general structure of nFAPI [Sma17]. Both libraries are logically connected in pairs through the fronthaul network. This is also shown in Fig. 5.1.

Apart from the RAN layers and their interfaces, two *migration controllers* are needed to orchestrate the migration in both units. These functions can activate or deactivate the MAC and RLC functions of their unit, and steer the user traffic accordingly. Furthermore, they have East-West interfaces to communicate with each other, in order to coordinate the migration, and a northbound interface to receive the migration command from upper layers. In this chapter, we neglect the origin of this migration command, which can either result from an automatic decision or manual intervention. In Chapter 6, however, we discuss this topic further by proposing multiple strategies to automatically trigger the migration at the CU.

5.4.5. Proposed migration procedure

In this section, we describe our migration procedure between PDCP-RLC and MAC-PHY splits. Based on what is explained in the previous section, we treat the synchronization of external and internal parameters differently. We synchronize the external parameters before the migration by duplicating configuration messages at two points. First, when a UE performs a successful random access procedure, the data structures defining the established radio bearers are copied to the inactive MAC and RLC functions. Second, after the core has successfully registered or updated a UE, the RRC function forwards its configuration to both active and inactive MAC and RLC functions before generating the messages *RRC Connection Setup* or *Reconfiguration*.

The internal parameters and the content of the buffers are synchronized by means of a five-step migration procedure. For the sake of brevity, we focus on a migration from MAC-PHY to PDCP-RLC with only downlink flows. However, extending this procedure to the other direction or to the uplink is straightforward, as this does not affect the essential operation of MAC and RLC layers. The procedure is as follows:

1. **CU request handling:** The CU receives the command for a soft, hard, or custom migration through its northbound interface. In the case of custom migration, it also receives the maximum time allocated for the draining of RLC and HARQ buffers.
2. **CU traffic steering:** The migration controller function redirects the flow of arriving PDCP PDUs to the DU. As a consequence, the RLC buffer at the CU stops receiving new arrivals and the RLC buffer at the DU starts receiving them.
3. **Buffer synchronization:** This step may consist of up to three stages, depending on the type of migration.
   3.1) **RLC draining:** Only for soft and custom migrations. The MAC scheduler continues its normal operation until the RLC buffers are empty, or until the allocated time runs out (in a custom migration).
   3.2) **MAC draining:** Only for soft and custom migrations. The MAC layer at the CU waits until the acknowledgment of the last active HARQ process is received, or until the allocated time runs out (in a custom migration). In a

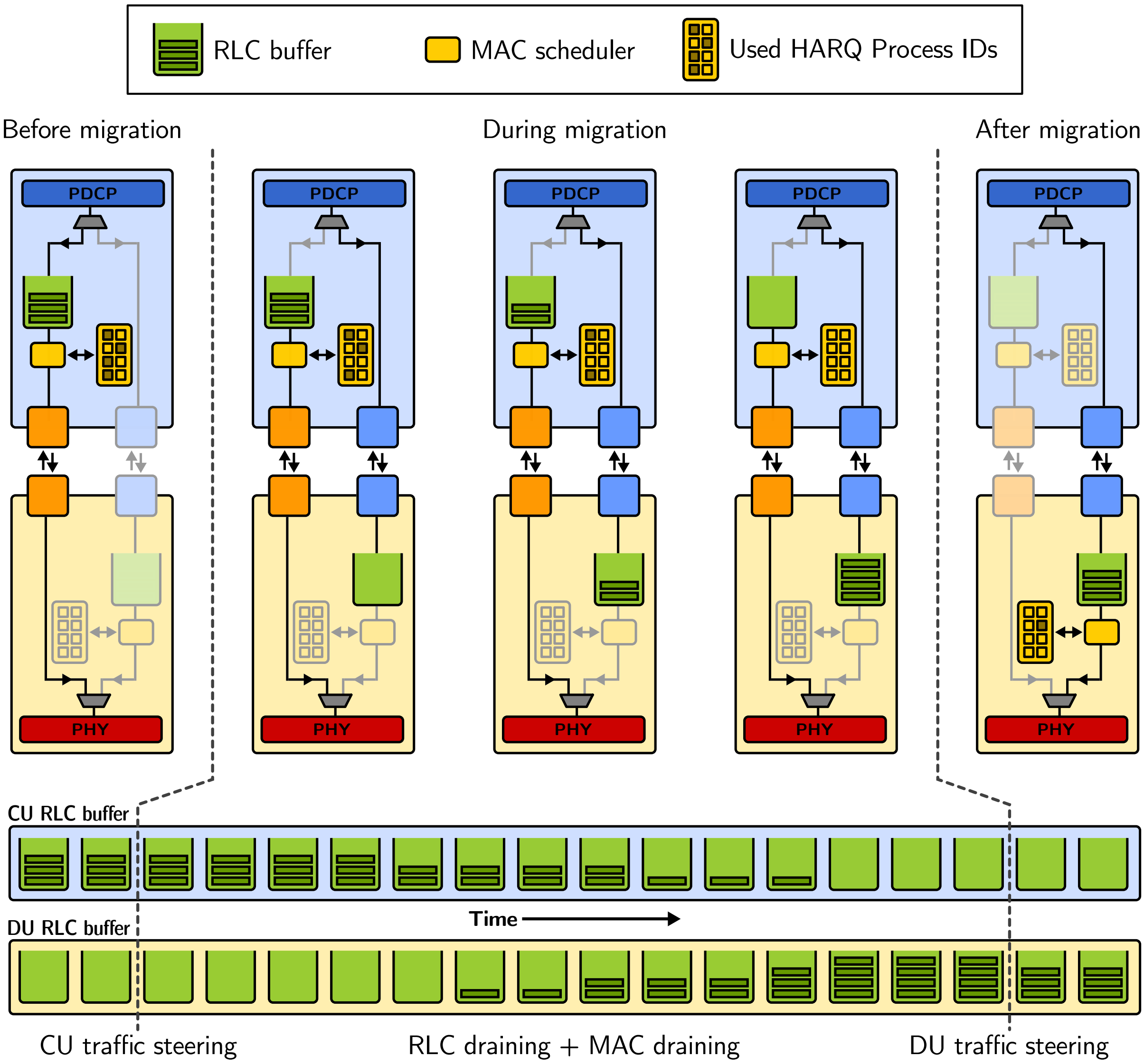


Figure 5.2.: Evolution of the internal operation of RLC and MAC layers during a soft migration from the MAC-PHY split to the PDCP-RLC split.

soft migration, this step guarantees that no packets are lost.

3.3) **MAC and RLC reset:** Only for hard migrations, and custom migrations after the allocated time. The content of the RLC and HARQ buffers is dropped and the list of active HARQ processes is reset.

4. **MAC and RLC synchronization**: The current frame, subframe, and RLC sequence numbers are sent from the CU to the DU, and the RLC and MAC functions at the DU are updated accordingly.

5. **DU traffic steering:** The migration controller activates the MAC and RLC functions at the DU, so they can start processing the PDUs stored in the RLC buffers.

In Fig. 5.2, we show a representation of the state of the RLC buffers, HARQ PIDs, and user packet routes before, during, and after a soft migration from the MAC-PHY split to the PDCP-RLC split. Before the migration, in the leftmost figure, the RLC buffer at the CU contains four user packets and four HARQ PIDs are in use. The MAC scheduler at the CU decides, based on information provided by the UEs, when to let packets at the RLC buffer proceed to the MAC layer, and which HARQ PIDs to use and release. Right after receiving the migration command (step 1), the CU disables the route connecting outgoing PDCP packets to its RLC buffer, and instead forward them to the PR library to be directly sent to the RLC buffer at the DU (step 2). After this is performed, the RLC buffer at the CU keeps draining normally as decided by the MAC scheduler (step 3) until this buffer is empty (step 3.1) and no HARQ PIDs are in use (step 3.2). At this point, the remaining internal parameters are sent from the MAC and RLC functions at the CU to those at the DU (step 4). Finally, the CU disables its MAC and RLC functions and those at the DU are enabled (step 5), thus allowing packet flow from the MAC function at the DU to the PHY layer.

This procedure produces no downtime, as it guarantees that there are always active MAC and RLC functions during the migration. This is achieved by letting the MAC and RLC functions at the DU store the new arriving PDUs (step 2), while the functions at the CU are still processing those in the buffer when the migration starts. However, the absence of downtime does not imply that the migration does not cause additional latency to the users. Indeed, step 3.2 (MAC draining) is basically a waiting step, which may delay the processing of PDUs arriving at that time. This delay is prevented by hard migrations or limited by custom migrations, at the cost of potential packet losses (step 3.3).

5.5. Implementation results

In this section, we present the details of our flexible platform implementing an adaptive 5G RAN, as well as measurement results. This platform is able to switch from PDCP-RLC to MAC-PHY, or vice versa, at runtime without interrupting ongoing user traffic. In order to do this, it follows the aforementioned procedure for replication-based migrations of the three types: hard, soft, and custom.

5.5.1. Hardware platform description

The hardware equipment consists of four off-the-shelf Intel i7 PCs for the UE, DU, CU, and core network. The DU and CU are connected by means of a fronthaul link, which is realized with a 1 Gb/s Ethernet link. The same type of link is used for the backhaul, which connects the CU to the core network. For the radio interface between the UE and the DU, two programmable USRP B200 are used to transmit a single-carrier 10 MHz LTE signal, which implies a maximum user data rate of 31.7 Mb/s. Although only one UE is used in this setup, the results are applicable to any multi-UE which produces the same joint downlink rate.

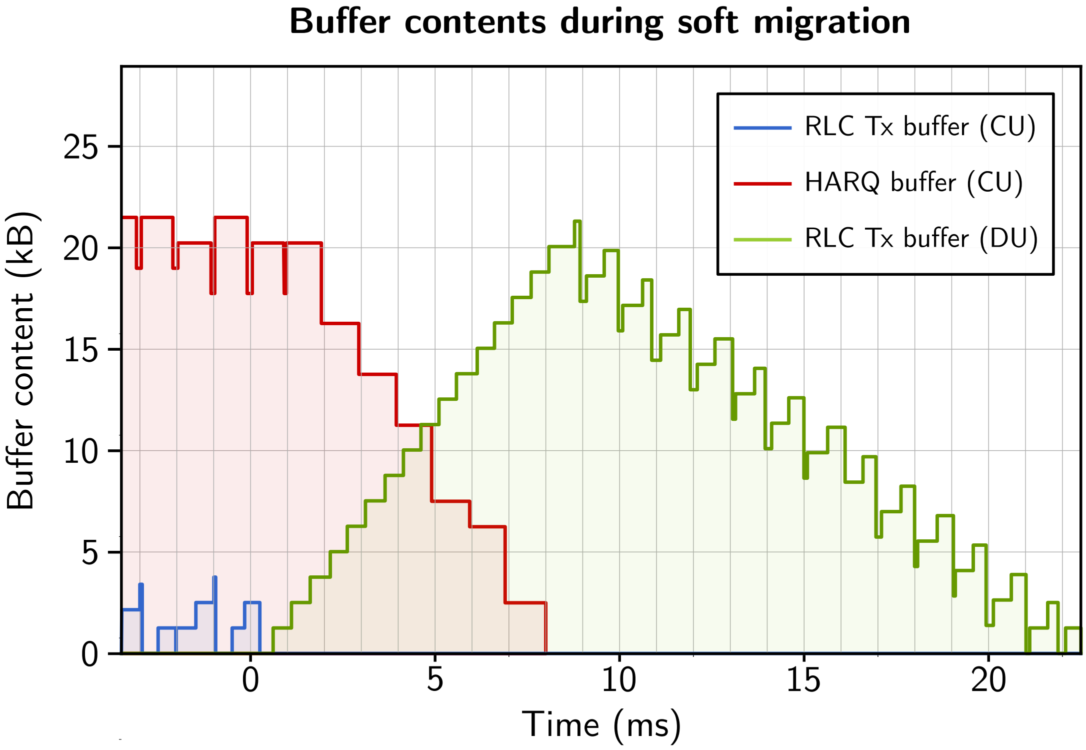


Figure 5.3.: Content of the RLC and HARQ buffers during a soft migration. The input traffic is a constant-rate 20 Mb/s stream, and the network downlink capacity is 31.7 Mb/s. The migration starts at $t = 0$ ms.

5.5.2. Measurement results

In order to show the performance of our implemented 5G RAN, in this section we present the most relevant experimental results. In Fig. 5.3, we can observe an exemplary measurement of the evolution of the content of the RLC and HARQ buffers when a soft migration from MAC-PHY to PDCP-RLC is performed, that is, when the MAC and RLC functions are migrated from the CU to the DU. For this experiment, a constant-rate 20 Mb/s downlink stream is used to represent the user traffic. The service rate of the base station is 31.7 Mb/s, corresponding to the downlink data rate supported by the air interface of a UE experiencing the highest channel quality in a 10 MHz carrier. Thus, the RAN operates at 63% capacity, on average. The data shown in the figure is extracted directly from execution logs, which report the status of the buffers every time they are modified, thus ensuring an update periodicity of one millisecond or less. The migration command is received by the migration controller at time $t = 0$, which performs the CU traffic steering step immediately and starts draining the RLC buffer. We can see that the RLC draining step lasts a small fraction of a millisecond, after which the RLC buffer at the CU is already empty. Then, the HARQ buffer, which is full when the migration starts, takes around 8 ms to drain. During that time, the RLC buffer at the DU receives PDCP PDUs from the CU, but it is not yet ready to pass them on to the MAC layer. Shortly after the HARQ buffer at the CU is empty, implying that all HARQ PIDs are unused, the migration is finished and the

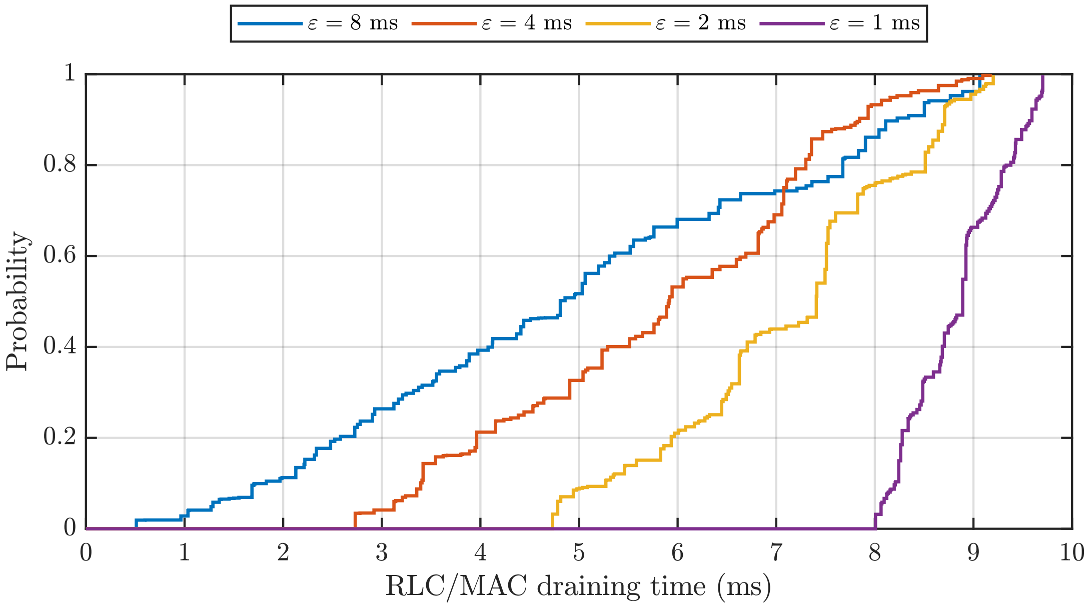


Figure 5.4.: Cumulative distribution functions of the RLC/MAC draining times for different inter-arrival times ε of incoming downlink PDCP PDUs. Each curve corresponds to 100 soft migrations.

RLC buffer at the DU starts to be emptied by the MAC layer. In this measurement, we can clearly see how the handover between old and new MAC and RLC functions is performed, and how this affects the normal operation of the buffers.

As mentioned in Sec. 5.4.5, the time it takes for the RLC and HARQ buffers to drain is important, as it impacts the end-user latency. Therefore, a specific experiment is performed in order to find out the distribution of this RLC/MAC draining time as a function of the inter-arrival time ε of the incoming PDCP packets. The results, shown in Fig. 5.4, allow us to conclude that the lower the inter-arrival time, the higher the average draining time. This is to be expected, as the lower the inter-arrival time, the more HARQ processes will be active, leading to a slower buffer draining time. In addition, we see that the maximum RLC/MAC draining time is around 10 ms, which corresponds to 1 ms to empty the RLC buffer and 8–9 ms to empty the HARQ buffer (containing up to 8 MAC PDUs).

In Fig. 5.5 we show boxplots representing the distributions of the additional latencies experienced by users packets that are affected by migration events. We consider eleven types of migrations: soft, hard, and nine custom migrations whose deadlines range from 1 to 9 ms. In these experiments, the RAN is dealing with a downlink traffic of periodic packets transmitted every 0.3 ms. Besides, the interference conditions produce a 10% packet error rate, that is, 1 out of 10 MAC PDUs has to be retransmitted. We observe that hard migrations cause a negligible impact on end-user latency, whereas soft migrations may cause an additional delay of 14 ms, with a median of around 7 ms. Custom migrations can be used to finely tune the experienced latency down to the acceptable values between these two extremes. Nonetheless, we conclude

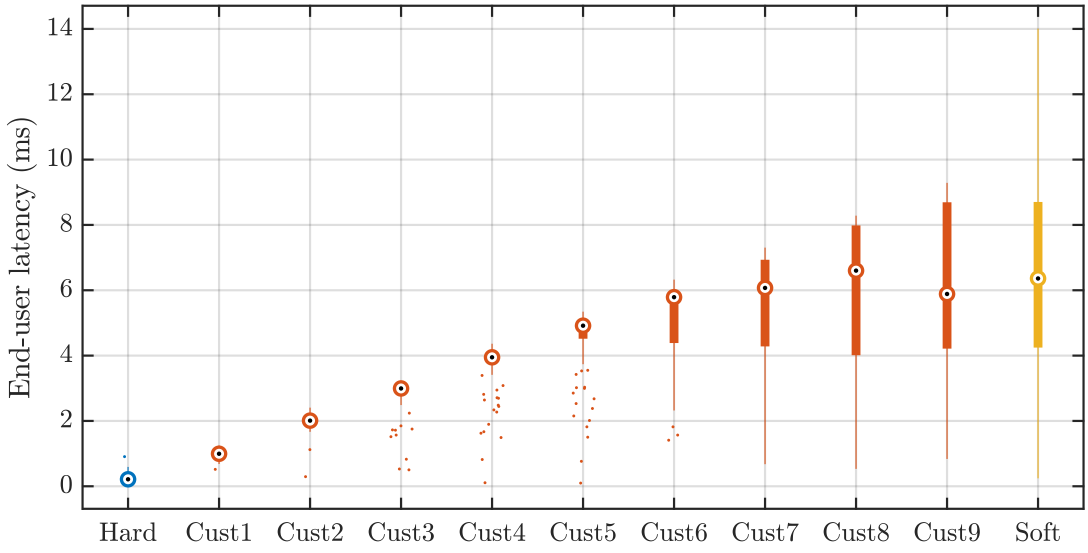


Figure 5.5.: Additional end-user latency for soft, hard, and nine custom migrations with allocated times ranging from 1 to 9 ms. Each point represents 50 migrations from MAC-PHY to PDCP-RLC.

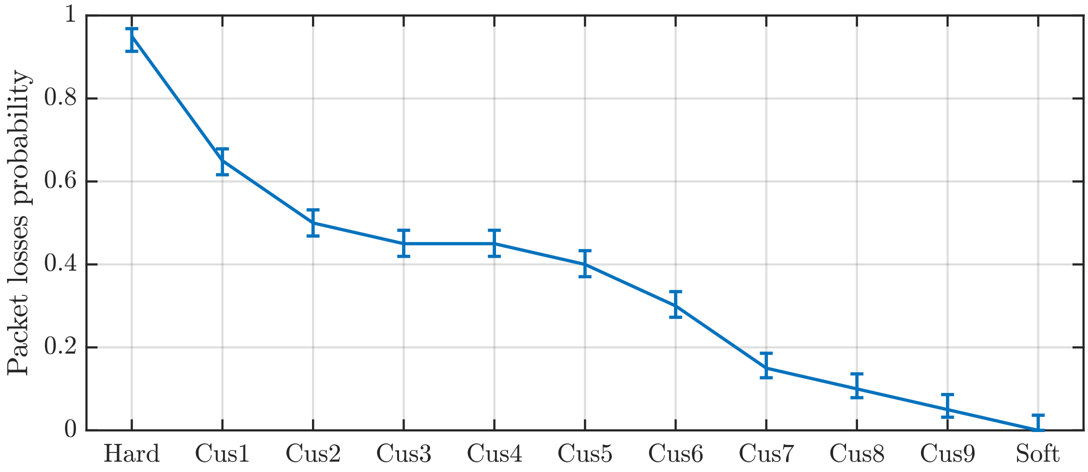


Figure 5.6.: Probability of experiencing a migration with packet losses for soft, hard, and nine custom migrations with allocated times ranging from 1 to 9 ms. Each point represents 50 migrations from MAC-PHY to PDCP-RLC, and vertical bars represent 95%-confidence intervals.

that even soft migrations lead to very small additional latencies, which may only be a problem for ultra-low latency use cases.

Finally, in Fig. 5.6 we show the probability of experiencing packet losses during a migration. We conclude that hard migrations are prone to suffer from packet losses, since 95% of the observed migrations resulted in packet losses. Conversely, soft migrations can be used to guarantee no packet losses, at the cost of introducing additional end-user latency as shown in Fig. 5.6. The probability of experiencing packet losses when using custom migrations lay in the middle between these two extremes, showing a roughly linear decay as the deadline grows.

5.6. Summary

In previous chapters, we explore the advantages of a 5G RAN featuring the ability of changing the functional splits of its base stations during runtime. However, this ability is presently neither included in any deployed network, nor described in current specifications, and only recently considered in previous research work. Owing to this, it is unclear whether changing the functional split is actually feasible and, if it is, how fast and costly it can be. In this chapter, we address the challenge of providing an actual implementation of a 5G RAN that is able to change its functional split without interrupting its normal operation. Although it is developed using experimental research equipment in a simplified scenario, this proof-of-concept implementation shows that live adaptation of the functional split is indeed possible.

After reviewing the state of the art and selecting the most adequate subset of functional splits, we enumerate the challenges that changing the functional split entails. Based on these challenges, we conclude that relying on off-the-shelf virtualization platforms is not the best option, and instead we propose our own replication-based migration strategy. This includes a detailed description of the state of a base station, consisting of those parameters that need to be transferred during a migration. We describe the hardware and software platforms that we use to implement this proof-of-concept and list the steps that are required to perform a change in the functional split with minimal packet losses and impact on end-user latency.

Finally, we provide actual measurements of the performance of our flexible 5G RAN during a migration. We observe that the buffer draining time, which is directly related to the total migration time, does not exceed 10 ms in a base station providing a 10 MHz channel and working at high load. This translates into a maximum additional end-user latency of 14 ms when using soft migrations, although this additional latency can be completely prevented by using hard migrations. The main disadvantage of a hard migration is, however, increased packet losses, which occur in 95% of the observed experiments. If this is undesired, custom migrations can be used to select the appropriate operating point within the trade-off between packet losses and end-user latency, or we can resort to soft migrations to completely remove packet losses.

6. Dynamic Functional Split Adaptation in Real Time

6.1. Introduction

6.1.1. Motivation, scope, and challenges

In Chapter 2 we introduce the motivation for deploying a 5G RAN that can adjust its functional split during runtime. We base this on the different characteristics that each option features in terms of required fronthaul capacity and interference-coordination capabilities. Nonetheless, implementing such an adaptive network requires us to carefully monitor the cost associated to both operating the network in a stable state and changing from an obsolete state to a more appropriate one. That is the reason why we introduce a novel, dedicated cost model in Chapter 3 that captures the most relevant components of the total cost in a flexible network. In addition, deciding on the optimal functional split for a given instant is not a trivial task, but it entails solving non-linear mixed-integer optimization problem as quickly as possible. In Chapter 4, we lay out multiple strategies to tackle this problem, evaluate them, and finally select the most promising one. However, even if we trust that adapting the functional split leads to enhanced performance and/or reduced cost, it is still unclear whether realizing this adaptation in actual networks is possible. In order to demonstrate that this adaptation is indeed feasible, in Chapter 5 we describe and evaluate a proof-of-concept implementation of a 5G RAN that is able to dynamically change its functional split with minimal impact on normal operation. Therefore, at this point we know that a 5G RAN may benefit from adapting its functional split and that the migration to the optimal functional split is possible and reasonably fast. Nevertheless, thus far we have focused on modeling and selecting the functional split for fixed time instants, neglecting the potential degradation of its optimality over time and the influence of the cost of dynamically reconfiguring the network.

Mobile networks face large variability in their demands, owing mainly to the movement of their users, changes in the channel quality, and user traffic variations (which can be part of well-known patterns or result from unexpected events). Thus, the network state that is optimal for a given instant may perform very suboptimally after some time. As a result, the network operator must be able to not only select the instantaneously optimal network configuration, but also keep track of the network evolution and decide when moving to a new configuration is desirable. Ideally, the operator should try to always operate in the optimal network state. Nonetheless, there are two reasons why this is not possible in actual deployments. On the one hand, finding the optimal state requires solving an NP-Hard problem, as shown in Chapter 4. Therefore, the only way of checking the optimality of a state is solving the problem anew [Aro09], which may be too computationally intensive to perform at every time instant. On the other hand, changing the network state may imply a non-negligible action cost, as anticipated in Chapter 3 and observed in Chapter 5. Thus, it may not be worth migrating to the optimal network state if the current state is just minimally suboptimal, since the additional revenue of operating in the optimal state may not compensate the action cost.

In this chapter, we propose and evaluate multiple *adaptation strategies* that a 5G RAN operator may follow in order to decide in real time when to move from its current state to the optimal state, based on the cost and benefit of doing so. Investigating these strategies entails facing a series of challenges. First, we have to decide an appropriate manner in which the network operator keeps track of the evolution of the optimal state. Second, we must identify the time-varying parameters that may affect the optimality of our current state. Third, we need to take into account all cost components that contribute to the total cost of operating and changing the network state in real time. Fourth, we have to use these information so as to find good-behaving adaptation strategies that are implementable in a real 5G RAN and lead to minimal total cost. Finally, we must ensure that these strategies perform correctly in realistic network conditions.

6.1.2. Key contributions

With the intention of addressing the aforementioned challenges, this chapter bases on our previous work [MAK21] to feature the following contributions.

1. We propose a simple but effective rule for the network operator to monitor the optimal state of a 5G RAN from the available UE and channel information. We base on the cost-of-flexibility model to properly configure the parameters of this monitoring rule.
2. We model the demand of a 5G RAN in order to represent the influence of the most relevant time-varying parameters. This demand can be used in the subsequent analysis to investigate and evaluate the performance of the adaptation strategies.
3. Based on the cost-of-flexibility model and the implementation results shown in Chapter 5, we present a detailed cost model of an adaptive 5G RAN, which includes the cost of selecting new states and migrating to them.
4. We derive multiple adaptation strategies with the intention of exploring a wide range of adaptation features. Some of these strategies are very simple to implement but may result in relatively poor performance, other strategies exhibit a good behavior but they may be difficult to implement, whereas the remaining strategies feature balanced characteristics.

5. In order to find the most appropriate adaptation strategies, we simulate their performance against realistic 5G RAN conditions and evaluate them in terms of achieved total cost and simplicity.

The remaining of this chapter is organized as follows. In Sec. 6.2, we summarize the previous work related to the problem of selecting an optimal adaptation strategy in a 5G RAN that can change its functional split. Sec. 6.3 discusses the monitoring rule that the network operator may use in order to keep track of the optimal network state with sufficient accuracy. In Sec. 6.4, we present the time-dependent cost modeling that we use to derive and analyze adaptation strategies, including a model for the time-varying demand. Sec. 6.5 lays out and discusses seven different adaptation strategies, which cover a wide range of trade-off features that may be desirable for the network operator. In Sec. 6.6 we apply the cost-of-flexibility model to the adaptation strategies in order to find critical frontiers points for the action cost that the operator may use to simplify the selection of the adaptation strategy. Sec. 6.7 shows a detailed evaluation of the performance of the adaptation strategies, as well as the accuracy of the estimated action cost frontiers. Finally, Sec. 6.8 summarizes and concludes this chapter.

6.2. Related work

We can divide the work related to this chapter into two categories. First, we have those works tackling, to some extent, the selection of a functional split dynamically, which have been also presented in previous chapters. For example, in [Mae+14; Sab+13] the concept of flexible functional split is introduced, in [GS+18a; GS+18b] the authors tackle the problem of optimally selecting the functional split at the design phase, [DGA19; DHA20; Die+21] tackle the delay characterization of a network featuring a changing functional split, in [Cha+17b] the authors explain a platform featuring reconfigurable functional splits, and in [HR18b] a dynamic functional split is also discussed with the purpose of allocating slices in a virtual RAN framework. Nevertheless, none of these works deal with the issue of optimally deciding the moment of changing the functional split.

Since, to the best of our knowledge, there are no previous works dealing directly with this decision, the second category of related work comprises those works addressing similar problems or generic approaches to dynamic problems. For example, similar problems can be found in the field of optimal decision theory, which deals with the selection of the decisions taken by a generic agent so as to optimize an expected outcome [DeG69; Cen00]. One such problem is optimal stopping, which addresses the problem of selecting when to stop a certain action in order to maximize earnings or minimize losses [AS07; BCJ19]. In our case, we can interpret that the network operator has to decide when to stop using the current state and move to the optimal state, and thus we can use the concepts and techniques employed in this field.

In addition, our problem can be considered a dynamic optimization problem, since our intention is to find the sequence of solutions that optimizes a dynamically-varying problem [CGP11]. There are several approaches to tackle a dynamic optimization problem, such as dynamic programming [Bel03], in which the dynamic problem is decomposed into increasingly simpler subproblems in order to find the optimal sequence of actions. Dynamic programming is a rather popular strategy that has been also applied to 5G networks problems, such as for dynamic content placement [Aye+18] or resource allocation in network slicing [Zha+18]. Alternative approaches to solving dynamic problems include evolutionary optimization [NYB12], which uses evolutionary algorithms to tackle hard dynamic problems, such as non-linear problems and those whose future parameters are difficult to estimate. One example of this is our previous work [MAJK20], which uses a genetic algorithm to find a near-optimal functional split of a simplified network. Another popular approach to tackle dynamic problems is reinforcement learning, in which an agent automatically learns the action rules that maximize a reward function [Sut18]. Reinforcement learning has been successfully applied to several dynamic configuration problems in 5G networks [Xio+19], such as resource scheduling [Com+18] or caching [SSG17].

In this chapter, we base on the concepts and techniques presented in the aforementioned works to tackle the problem of optimally adapting the functional splits in real time for 5G RANs. To the best of our knowledge, our contribution is the first work that addresses the dynamic selection of the optimal functional split, taking into account the continuous degradation of implemented functional splits and the cost of reconfiguring them.

6.3. Optimal state monitoring

In Chapter 2, Sec. 2.4.2, we introduce the decision-making entity, which is in charge of monitoring the correspondence between the current network state and the instantaneous demand. If the network state optimally satisfies the demand, that is, if $\mathbf{s}(\tau) = \mathbf{s}^*(\tau) = \chi(\mathbf{d}(\tau))$ at time instant τ, where $\chi(\,\cdot\,)$ is the adaptation function as defined in Sec. 3.3.2, then no special action is required. However, if $\mathbf{s}(\tau) \neq \mathbf{s}^*(\tau) = \chi(\mathbf{d}(\tau))$, the decision-making entity has to decide whether the network should move to the optimal state $\mathbf{s}^*(\tau)$ or the current state $\mathbf{s}(\tau)$ is still near-optimal enough not to motivate a state migration. As a result, this decision-making entity has two important tasks [Kel+19]: (i) keep track of the instantaneous optimal state and (ii) decide on state adaptations.

Monitoring the evolution of the demand is a relatively simple task, since the CU, which contains the decision-making entity, is periodically updated about the location, traffic load, and channel quality of its served UEs [DPS18, Sec. 8.2]. Nonetheless, attempting to figure out the corresponding optimal state for every demand may be rather challenging. Indeed, in Chapter 4 we show that in order to obtain the optimal state of a 5G RAN for fixed conditions we need to solve the FSSP, which is an NP-Hard problem. This hinders the achievement of a continuously-adapted sequence of optimal states, since tackling an NP-Hard problem is time consuming and computationally intensive. In addition, NP-Hardness implies that we cannot tell whether a

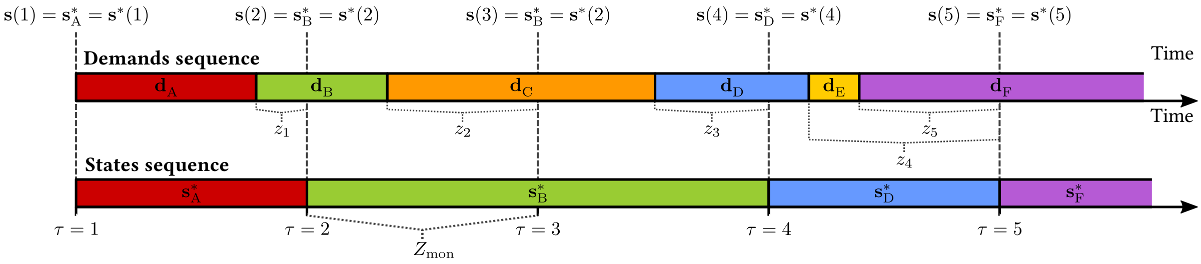

$\mathbf{s}(1) = \mathbf{s}^*_{\mathrm{A}} = \mathbf{s}^*(1)$
$\mathbf{s}(2) = \mathbf{s}^*_{\mathrm{B}} = \mathbf{s}^*(2)$
$\mathbf{s}(3) = \mathbf{s}^*_{\mathrm{B}} = \mathbf{s}^*(2)$
$\mathbf{s}(4) = \mathbf{s}^*_{\mathrm{D}} = \mathbf{s}^*(4)$
$\mathbf{s}(5) = \mathbf{s}^*_{\mathrm{F}} = \mathbf{s}^*(5)$
Demands sequence
Time
$\mathbf{d}_{\mathrm{A}}$
$\mathbf{d}_{\mathrm{B}}$
$\mathbf{d}_{\mathrm{C}}$
$\mathbf{d}_{\mathrm{D}}$
$\mathbf{d}_{\mathrm{E}}$
$\mathbf{d}_{\mathrm{F}}$
Time
z_1
z_2
z_3
z_5
z_4
States sequence
$\mathbf{s}^*_{\mathrm{A}}$
$\mathbf{s}^*_{\mathrm{B}}$
$\mathbf{s}^*_{\mathrm{D}}$
$\mathbf{s}^*_{\mathrm{F}}$
$\tau = 1$
$\tau = 2$
$\tau = 3$
$\tau = 4$
$\tau = 5$
Z_{mon}

Figure 6.1.: Example diagram of the sequences of demands and states in the considered network, along with demand-sampling instants. The optimal states are related to their corresponding demand as $\mathbf{s}^*_i = \chi(\mathbf{d}_i)$, where $i \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{F}\}$.

previously optimal state is still optimal after a change in the optimization problem unless we solve it again.

In this thesis, we suggest a simple, yet effective approach for monitoring and controlling the state of the network. First, the decision-making entity periodically samples the network demand and solves the FSSP for each sample, so as to obtain an updated, instantaneous network configuration that optimizes the revenue. We refer to the *monitoring period* between demand samples as Z_{mon}. The length of Z_{mon} plays an important role in the performance of the functional split adaptation. If the period between samples is too long, the network may spend a large portion of the time in severely obsolete states. If it is too short, we may waste computational resources. In Sec. 6.7.3, we apply the cost-of-flexibility model to obtain an appropriate value for Z_{mon}, taking into account both factors. Once the new optimal state is known, the decision-making entity chooses whether to stay in the current state or adapt to the optimal state. In Sec. 6.5 we discuss several adaptation strategies that the decision-making entity may use to select the right option.

In Fig. 6.1 we show a diagram of possible sequences of demands and states over time for an abstract RAN. Demands change at unforeseeable instants, and the network is first aware of the change at the next monitoring instant, which are homogeneously spaced with period Z_{mon}. These instants are indexed via $\tau \in \{1, ..., 5\}$. At these times, the operator can decide to either set $\mathbf{s}(\tau) = \mathbf{s}^*(\tau)$ or stay at $\mathbf{s}(\tau) = \mathbf{s}(\tau - 1)$, depending on the adaptation strategy. The duration of action phases between demand and state changes are denoted as $\{z_1, ..., z_5\}$. Note that, in some cases, the demand may change multiple times before the next monitoring instant. In fact, in an actual deployment, we may argue that the demand changes in a continuous manner, since it is affected by the location and activity of the UEs, as we explain in Sec. 6.4.1. This is, however, taken into account by the cost-of-flexibility framework used in previous and subsequent analysis.

In Chapter 3 we assume that the network can instantaneously perceive changes in the demands, and it responds to them by immediately starting the action phase so as to find and implement the new optimal state. The duration of this action phase is modeled by the random variable $\mathcal{Z}$. In our case, however, we do not detect demand changes immediately, but at a periodic rate. This is not a problem from the modeling point-of-view, since we can simply merge the actual action phase with the difference between demand changes (occurring at unknown instants) and the next monitoring instants. In fact, since the time required to solve the FSSP is in the order of $1 - 2$ s (see Sec. 4.7.2), and the time to migrate the network state is in the order of $1 - 10$ ms (see Sec. 5.5.2), this demand-to-monitoring delay may dominate $\mathcal{Z}$ for a sufficiently large Z_{mon}. In that case, it is clear that the duration of the action phase $\mathcal{Z}$ follows a uniform distribution ranging from 0 to Z_{mon}. The reason is that, since monitoring and demand changes are independent processes (due to our ignorance about when demand changes occur), the difference between a demand change and the next monitoring instant can be modeled as a uniformly distributed random variable ranging from 0 to Z_{mon}. Consequently:

$$\mathcal{Z} \sim U(0, Z_{\text{mon}}) \quad \Leftrightarrow \quad f_{\mathcal{Z}}(z) = \begin{cases} \frac{1}{Z_{\text{mon}}} & \text{if } 0 \leq z \leq Z_{\text{mon}}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

6.4. Time-dependent cost modeling

With the intention of providing a consistent basis for the derivation and analysis of adaptation strategies, in this section we introduce our model of time-dependent cost components. We first define the time-dependent demand of a 5G RAN featuring an adaptive functional split, then we present the mathematical notation to refer to time-dependent variables, and finally we discuss the definition of time-dependent readiness and action costs.

6.4.1. Demand modeling

In Chapter 3, we define the network demand as the set of all parameters that affect the profitability of the network but cannot be directly modified by the operator. A precise definition of demand is not specially relevant for the instantaneous problem formulations shown in Chapter 4, since in that case the demand can be interpreted as fixed problem parameters. Nonetheless, when addressing the evolution of cost and performance of an adaptive 5G RAN over time, it is important to clearly state which problem parameters are part of the demand, so as to track their variation and properly adapt to them.

In principle, we could associate to the demand every parameter playing a role in the performance-maximizing, operating-cost-minimizing, or readiness-cost-minimizing FSSPs, as formulated in (P4), (P15), and (P17), respectively (see Chapter 4, Sec. 4.4, 4.5, and 4.6). However, some of these parameters describe fixed network features, such as the sets of fronthaul edges $\mathbb{E}$ and nodes $\mathbb{N}$, the function modeling capacity required by each functional split $\nu(\,\cdot\,)$, etc. These parameters are unlikely to change during normal network operation, and thus for simplicity we neglect them as part of the demand. Nevertheless, it is straightforward to include these parameters in the definition of demand if this is required, since this does not affect the validity of the subsequent modeling and analysis.

There are four different sets of variables in our FSSP formulations, as presented in (P4), (P15), and (P17), whose values are susceptible to change over time. First, the received signal power p_u received by UE $u \in \mathbb{U}$ from its serving gNB can change, due either to UE mobility or to changes in the channel quality. Second, the association h_u between UE $u \in \mathbb{U}$ and its serving gNBs may also vary because of handover procedures. Third, the interference power $i_{u,g}$ received by UE $u \in \mathbb{U}$ from all gNBs is likely to change frequently, owing again to mobility and channel fluctuations. Finally, the aggregated downlink traffic ρ_g for all gNBs could also change, due to variations in received signal quality of their served UEs. As a result, we formally define the demand $\mathbf{d}$ of a dynamically adapting 5G RAN as:

$$\mathbf{d} \triangleq \langle \mathbf{p}, \mathbf{h}, \mathbf{i}, \boldsymbol{\rho} \rangle \,, \tag{2}$$

where $\mathbf{p} \triangleq [p_u]\ \forall u \in \mathbb{U}$ is the vector of all signal powers, $\mathbf{h} \triangleq [h_u]\ \forall u \in \mathbb{U}$ is the vector of all gNB-to-UE associations, $\mathbf{i} \triangleq [i_{u,g}]\ \forall \langle u, g \rangle \in \mathbb{U} \times \mathbb{G}$ is the vector of all interference powers, and $\boldsymbol{\rho} \triangleq [\rho_g]\ \forall g \in \mathbb{G}$ is the vector of all downlink traffics. We thus use $\mathbf{d}$ in our subsequent formulations to emphasize the dependency of performance or cost on these variable parameters.

6.4.2. Time dependency modeling

Since the values of the parameters which make up the demand may change over time, we use the following notation to refer to the value of demand $\mathbf{d}$ at time instant τ:

$$\mathbf{d}(\tau) \triangleq \langle \mathbf{p}(\tau), \mathbf{h}(\tau), \mathbf{i}(\tau), \boldsymbol{\rho}(\tau) \rangle \,. \tag{3}$$

This time dependency can be transmitted to every other function or variable that depends on $\mathbf{d}$. For example, vectors $\mathbf{p}, \mathbf{h}, \mathbf{i}$ affect the spectral efficiency η_u that each UE u can achieve as defined in Sec. 4.4.1. Hence, we can include explicit dependency on $\mathbf{d}(\tau)$ as follows:

$$\eta_u(\mathbf{a}, \mathbf{d}(\tau)) = \log_2 \left(1 + \frac{p_u(\tau)}{\varsigma + \sum_{g=1}^{G} i_{u,g}(\tau) \cdot \text{И}(\min(a_{h_u}, a_g))} \right) , \tag{4}$$

From $\eta_u(\mathbf{a}, \mathbf{d}(\tau))$, we can calculate the time-dependent geometric mean of spectral efficiencies as follows:

$$\widetilde{\eta}(\mathbf{a}, \mathbf{d}(\tau)) \triangleq \left(\prod_{u=1}^{U} \eta_u(\mathbf{a}, \mathbf{d}(\tau)) \right)^{\frac{1}{U}} = \exp\left(\frac{1}{U} \sum_{u=1}^{U} \log\left(\eta_u(\mathbf{a}, \mathbf{d}(\tau)) \right) \right) . \tag{5}$$

Similarly, the variation of $\boldsymbol{\rho}$ influences the computational cost of running centralized and decentralized functions, as discussed in Sec. 4.5. Therefore we use the notation $K_{\text{comp}}(\mathbf{a}, \mathbf{d}(\tau))$ to refer to the instantaneous computational cost at time τ and, by extension, $K_{\text{oper}}(\mathbf{s}, \mathbf{d}(\tau))$, $K_{\text{rev}}(\mathbf{s}, \mathbf{d}(\tau))$, and $K(\mathbf{s}, \mathbf{d}(\tau))$ to denote the operating cost, the revenue, and readiness costs resulting from applying state s at time τ, respectively. The same notation applies for any other cost-related function.

In principle, the time index τ could take any continuous value. In Sec. 6.3, however, we state that our preferred approach for a dynamically adapting RAN implies sampling the network at periodic intervals, instead of continuous monitoring. Consequently, we henceforth assume that the time index τ is discrete and takes values within the set $\tau \in \{1,\, 2,\, ...,\, \tau^{\max}\}$, where $\tau^{\max}$ is the last considered instant. Since the state s of the network may also change as a result of conscious adaptations at those points, we use the notation $\mathbf{s}(\tau)$ to reflect the time dependency of the network state, as it is already done in Chapter 3. In addition, we define

$$\mathbf{S} \triangleq [\mathbf{s}(1) \, \cdots \, \mathbf{s}(\tau^{\max})] \tag{6}$$

as the sequence vector containing all network states in the interval $\tau \in \{1, ..., \tau^{\max}\}$. Therefore, S is a finite realization of the stochastic sequence of states $\{\mathcal{S}_j\}_{j\in\mathbb{Z}}$ that is defined and discussed in Sec. 3.3.1. Given the optimal network state $\mathbf{s}^*(\tau)$ at time τ, we also define

$$\mathbf{S}^* \triangleq [\mathbf{s}^*(1) \, \cdots \, \mathbf{s}^*(\tau^{\max})] \tag{7}$$

as the sequence vector containing all optimal network states. Note that the elements of S and $\mathbf{S}^*$ may differ if the network decides not to operate in the optimal state, but both their values are always known up to the present value of τ. Similarly, we can define the sequence of demands

$$\mathbf{D} \triangleq [\mathbf{d}(1) \, \cdots \, \mathbf{d}(\tau^{\max})] \tag{8}$$

as a finite realization of the stochastic sequence of demands $\{\mathcal{D}_j\}_{j\in\mathbb{Z}}$ that is defined in Sec. 3.3.1.

Finally, we use the following shorthand notation to refer to the readiness cost of the optimal state $\mathbf{s}^*(\tau)$ with respect to current demand $\mathbf{d}(\tau)$ at time τ:

$$K^*(\tau) \triangleq K(\mathbf{s}^*(\tau), \mathbf{d}(\tau)) = K(\chi(\mathbf{d}(\tau)), \mathbf{d}(\tau)), \tag{9}$$

where $\chi(\,\cdot\,)$ is the adaptation function as defined in Sec. 3.3.2.

6.4.3. Mean readiness and action costs

In this section, we provide time-dependent estimations of the mean readiness, proaction, and reaction costs over time, following the notation described above. In the end, we combine these three cost components into a single total cost, which can be used as the minimization objective for our adaptation strategies.

Readiness cost

The readiness cost $K(\mathbf{s}, \mathbf{d})$ depends on time in two different manners. On the one hand, the state $\mathbf{s}(\tau)$ of the RAN can be dynamically selected by the adaptation strategy in order to minimize readiness cost. On the other hand, even if the state remains constant, the demand $\mathbf{d}(\tau)$ of the network does change over time, which also impacts the resulting readiness cost. Thus, the empirical mean readiness cost $\overline{K}(\mathbf{S}, \mathbf{D})$ for a finite sequence of states $\mathbf{S}$ and a finite sequence of demands $\mathbf{D}$ is calculated as:

$$\overline{K}(\mathbf{S}, \mathbf{D}) \triangleq \frac{1}{\tau^{\max}} \sum_{\tau=1}^{\tau^{\max}} K(\mathbf{s}(\tau), \mathbf{d}(\tau)) \tag{10}$$

We can compute the instantaneous readiness cost associated to arbitrary state $\mathbf{s}$ and demand $\mathbf{d}$ as described in Sec. 4.5 and 4.6, which can be summarized as:

$$\begin{aligned} K(\mathbf{s}, \mathbf{d}) &= K_{\text{oper}}(\mathbf{s}, \mathbf{d}) + K_{\text{rev}}(\mathbf{s}, \mathbf{d}) && (11) \\ &= K_{\text{inst}} + K_{\text{comp}}(\mathbf{a}, \boldsymbol{\rho}) + K_{\text{rout}}(\mathbf{f}) + \widetilde{K}_{\text{rev}}(\widetilde{\eta}(\mathbf{a}, \mathbf{d})), && (12) \end{aligned}$$

where

$$K_{\text{comp}}(\mathbf{a}, \boldsymbol{\rho}) = \sum_{g=1}^{G} \left(K_{\text{comp,CU}}(a_g)\gamma_{\text{CU}} + K_{\text{comp,DU}}(a_g)\gamma_{\text{DU}} \right) \rho_g \tag{13}$$

now includes explicit dependency on $\boldsymbol{\rho}$. Assuming that $\{\mathcal{S}_j\}_{j\in\mathbb{Z}}$ and $\{\mathcal{D}_j\}_{j\in\mathbb{Z}}$ are stationary processes, we can define the theoretical mean readiness cost as:

$$\overline{K} \triangleq \lim_{\tau^{\max}\to\infty} \frac{1}{\tau^{\max}} \sum_{\tau=1}^{\tau^{\max}} K(\mathbf{s}(\tau), \mathbf{d}(\tau)). \tag{14}$$

We can also employ the cost-of-flexibility framework presented in Sec. 3.4.2 to estimate the mean readiness cost $\overline{K}$ of a dynamically adapting network as:

$$\overline{K} = \alpha\varphi K_{\Delta}(0) + (1 - \alpha\varphi)\beta\varphi\widehat{K}_{\beta}(\varphi), \tag{15}$$

where

$$\widehat{K}_{\beta}(\varphi) \triangleq \sum_{\delta=1}^{\infty} K_{\Delta}(\delta)(1 - \beta\varphi)^{\delta-1}, \tag{16}$$

$$\alpha \triangleq \frac{1}{T} \int_{0}^{\infty} F_{\mathcal{Z}}(t) \left(1 - F_{\mathcal{T}}(t)\right) dt, \tag{17}$$

$$\beta \triangleq \int_0^\infty F_\mathcal{Z}(t) f_\mathcal{T}(t) dt, \tag{18}$$

φ is the maximum flexibility parameter, which reflects the ratio of demands that could be eventually satisfied assuming infinite action time, $K_\Delta(\cdot)$ is the readiness degradation function (RDF), a characteristic function of the network, $F_\mathcal{Z}(\cdot)$ is the cumulative distribution function of the action phase duration $\mathcal{Z}$, $F_\mathcal{T}(\cdot)$ is the cumulative distribution function of the demand duration $\mathcal{T}$, and $f_\mathcal{T}(\cdot)$ is the probability distribution function of the demand duration $\mathcal{T}$.

The utility of using (15) to compute $\overline{K}$ is twofold. On the one hand, since we can relate the distribution of $\mathcal{Z}$ to the monitoring period Z_mon, as stated in (1), we can use this expression to observe the relationship between $\overline{K}$ and Z_mon and thus estimate an adequate value of Z_mon. The monitoring period Z_mon should be short enough to closely keep track of the changes in the optimality of the state vector, but long enough to be feasible (and profitable) in a real RAN. This selection is addressed in Sec. 6.7.3. On the other hand, the dependency of (15) on φ allows us to model the total cost of certain adaptation strategies, as it is discussed in Sec. 6.5. In addition, we can use this modeling to draw conclusions for all adaptation strategies.

In order to use (15), we need to find the RDF and the distribution of $\mathcal{T}$ in a RAN featuring a dynamically adapted functional split. We can obtain this information from either previous experimental data or new simulations. Although it is thus required to perform preliminary measurements or simulations to enable the use the cost-of-flexibility framework, once the RDF and the distribution of $\mathcal{T}$ are known the theoretical model can be used to predict the readiness cost for any distribution of $\mathcal{Z}$ and any value of φ. This removes the necessity for dedicated measurements or simulations to test the validity of multiple adaptation strategies or Z_mon values.

For estimating the distribution of $\mathcal{T}$, we can simply simulate our network with a fine time granularity and observe how frequently an optimal state vector s* changes over time. This has been already performed in our previous work [MAJK20], where we present an approximation to the actual duration of the demands with a time granularity of 1 s. This can be used directly as estimates of $F_\mathcal{T}(t)$ and $f_\mathcal{T}(t)$, although more advanced techniques can be also applied to yield a better approximation to the original distribution of $\mathcal{T}$. We show one such technique in Appendix A.3, which can also be used to estimate $F_\mathcal{T}(t)$ and $f_\mathcal{T}(t)$ from simulations featuring coarser time granularity.

Regarding the estimation of the RDF $K_\Delta(\delta)$, we can also measure it directly or infer it from related functions, such as the evolution of the cost of an optimal solution over time since the moment it is first implemented (also provided in [MAJK20]). If we denote this time-based degradation function as $K_\Delta^\tau(\tau)$, the following expression holds:

$$K_\Delta^\tau(\tau) = \sum_{\delta=0}^\infty K_\Delta(\delta) \Pr\{\Delta = \delta|\tau\}, \tag{19}$$

where $\Pr\{\Delta = \delta|\tau\}$ is the probability that the state delay is δ after τ time units since the optimal state was implemented. In other words, it is the probability of experiencing $\Delta = \delta$ demand changes in a time interval of length τ. We present a procedure to estimate $K_\Delta(\delta)$ from $K^\tau_\Delta(\tau)$ in Appendix A.4.

For notational convenience, we also define $\overline{K}^*$ as the mean readiness cost for the sequence of optimal states:

$$\overline{K}^* \triangleq \lim_{\tau^{\max}\to\infty} \overline{K}(\mathbf{S}^*, \mathbf{D}). \tag{20}$$

Since the optimal state is the one that minimizes the readiness cost, it is clear the mean readiness cost of the optimal sequence is less or equal than the mean readiness cost of any other sequence, that is:

$$\overline{K}^* \leq \overline{K}. \tag{21}$$

Proaction cost

The mean proaction cost $\overline{C}^P$ is the additional computational cost of finding a new optimal state vector s$^*(\tau)$ every monitoring interval Z_{mon}. We define ζ as the average computational effort of solving the FSSP, in RC · s (as defined in Sec. 4.5). We can convert this computational effort into proaction cost by using the following expression:

$$\overline{C}^P = \frac{\zeta \cdot \gamma_{\text{CU}}}{Z_{\text{mon}}}, \tag{22}$$

where γ_{CU} is the computational cost per reference core at the CU, since we can safely assume that solving the FSSP occurs at the CU, and Z_{mon} is the monitoring period.

In Sec. 6.7.4, we provide an estimation of the value of $\overline{C}^P$ for our simulated parameters and those provided by previous work. We observe that the value of this cost component is almost negligible when compared to the readiness or the reaction cost, thus playing little role in the profitability of the network or the selection of the adaptation strategy.

Reaction cost

The instantaneous reaction cost $C^R(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}}, \mathbf{d})$ reflects the effort or inconvenience of changing the state vector from an initial state s$^{\text{ini}}$ to a final state s$^{\text{fin}}$ during runtime. This cost component is the most challenging to estimate, since the technology for changing the functional split of a RAN during runtime is not yet widespread, thus little details about the actual reaction cost are known. Therefore, instead of considering a single estimation of the reaction cost, we cover a wide range of values with the intention of exploring the profitability limits of dynamic adaptation as a function of the reaction cost.

As with the readiness cost, we define the empirical mean reaction cost $\overline{C}^R(\mathbf{S}, \mathbf{D})$ of operating with finite state sequence $\mathbf{S}$ and demand sequence $\mathbf{D}$ as:

$$\overline{C}^R(\mathbf{S}, \mathbf{D}) \triangleq \frac{1}{\tau^{\max}} \sum_{\tau=2}^{\tau^{\max}} C^R(\mathbf{s}(\tau-1), \mathbf{s}(\tau), \mathbf{d}(\tau)). \tag{23}$$

Assuming that $\{\mathcal{S}_j\}_{j\in\mathbb{Z}}$ and $\{\mathcal{D}_j\}_{j\in\mathbb{Z}}$ are stationary processes, we can define the theoretical mean reaction cost as:

$$\overline{C}^R \triangleq \lim_{\tau^{\max}\to\infty} \frac{1}{\tau^{\max}} \sum_{\tau=2}^{\tau^{\max}} C^R(\mathbf{s}(\tau-1), \mathbf{s}(\tau), \mathbf{d}(\tau)). \tag{24}$$

In addition, we define $\overline{C}^{R*}$ as the mean reaction cost of the sequence of optimal states:

$$\overline{C}^{R*} \triangleq \lim_{\tau^{\max}\to\infty} \overline{C}^R(\mathbf{S}^*, \mathbf{D}). \tag{25}$$

The sequence of optimal states $\mathbf{S}^*$ is the one featuring the highest number of state changes out of all sensible strategies, since it makes no sense to adapt more often than the frequency of the changes in the optimal states. Therefore it follows that the reaction cost of always operating in the optimal state, that is, the impatient strategy, is an upper bound to any other reasonable strategy:

$$\overline{C}^R \leq \overline{C}^{R*}. \tag{26}$$

It is worth noting that $\overline{C}^{R*}$ can also be interpreted as the *average cost per migration*, since $\mathbf{S}^*$ includes all potential migrations that any strategy may feature.

We identify two main sources of reaction cost: the migration cost $C^R_{\text{migr}}(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}}, \mathbf{d})$ directly related to a state migration (including computational efforts and flow rerouting) and the penalization cost $C^R_{\text{pen}}(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}}, \mathbf{d})$ coming from penalizations due to potential service interruptions. Hence:

$$C^R(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}}, \mathbf{d}) = C^R_{\text{migr}}(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}}, \mathbf{d}) + C^R_{\text{pen}}(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}}, \mathbf{d}) \tag{27}$$

We can estimate the former subcomponent based on our previous experimental results. In Chapter 5, we present an implementation of a RAN featuring a dynamically configurable functional split. The migration between two functional splits is accomplished by means of function redundancy, that is, the function to be migrated runs at both the DU and CU simultaneously while the transmission and reception buffers are being emptied. As a result, we model the additional instantaneous cost of migrating from an initial state vector $\mathbf{s}^{\text{ini}}$ to a final state vector $\mathbf{s}^{\text{fin}}$ by means of the following function:

$$C^R_{\text{migr}}(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}}, \mathbf{d}) = \frac{Z_{\text{migr}}}{Z_{\text{mon}}} \Bigg(\sum_{g=1}^{G} \Big[K_{\text{comp,CU}}(\max(a^{\text{ini}}_g, a^{\text{fin}}_g))\gamma_{\text{CU}} \\ + K_{\text{comp,DU}}(\min(a^{\text{ini}}_g, a^{\text{fin}}_g))\gamma_{\text{DU}} \Big] \rho_g - K_{\text{comp}}(\mathbf{a}^{\text{fin}}) \Bigg), \quad (28)$$

where Z_{migr} is the migration length, assumed constant for simplicity. The factor $\frac{Z_{\text{migr}}}{Z_{\text{mon}}}$ scales the migration cost according to its duration, since it may only apply for a very short interval. Indeed, all experimental migrations shown in Chapter 5 conclude in less than 20 ms. The term $-K_{\text{comp}}(\mathbf{a}^{\text{fin}})$ prevents counting the readiness cost of the final state vector twice, which implies that $C^R_{\text{migr}}(\mathbf{s}, \mathbf{s}, \mathbf{d}) = 0$, that is, there is no migration-related reaction cost if there is no change in the state vector. Regarding additional routing cost, the migration strategy described in Chapter 5 does not incur in an increased network usage, apart from negligible additional signaling. As a result, we do not include an explicit dependence on $\mathbf{f}^{\text{ini}}$ or $\mathbf{f}^{\text{fin}}$ in $C^R_{\text{migr}}(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}})$.

An estimation for the value of the penalization cost $C^R_{\text{pen}}(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}}, \mathbf{d})$ is rather difficult to obtain, since operators rarely disclose these operational details. In order to compensate for the lack of information about $C^R_{\text{pen}}(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}})$ and also potential inaccuracies in our estimation of $C^R_{\text{migr}}(\mathbf{s}^{\text{ini}}, \mathbf{s}^{\text{fin}})$, we use $\overline{C}^{R*}$, the average cost per migration, as a variable in our experiments. In order to decide on an appropriate range for $\overline{C}^{R*}$, we define $\overline{C}^{R*}_{\text{I}}$ as the known contribution of the migration costs to the reaction cost:

$$\overline{C}^{R*}_{\text{I}} \triangleq \frac{1}{\tau^{\max}} \sum_{\tau=2}^{\tau^{\max}} C^R_{\text{migr}}(\mathbf{s}^*(\tau-1), \mathbf{s}^*(\tau)), \quad (29)$$

Since $\overline{C}^{R*}_{\text{I}}$ only includes migration costs, it is clearly a lower bound of $\overline{C}^{R*}$. As a reference upper bound, we can simply multiply $\overline{C}^{R*}_{\text{I}}$ by a large value. In our experiments we consider the following range of values for $\overline{C}^{R*}$:

$$\overline{C}^{R*}_{\text{I}} \leq \overline{C}^{R*} \leq 2 \cdot 10^6 \cdot \overline{C}^{R*}_{\text{I}}. \quad (30)$$

As we show in Sec. 6.7, this wide range is chosen since it allows us to clearly see the influence of the reaction cost on the overall profitability and identify important frontiers that impact the selection of the adaptation strategy.

Total cost

The mean total cost $\overline{Q}$ of a RAN featuring a dynamically-adapting functional split is the sum of the mean readiness, proaction, and reaction costs.

$$\overline{Q} \triangleq \overline{K} + \overline{C}^P + \overline{C}^R. \tag{31}$$

This theoretical definition can be approximated for long sequences of states $\mathbf{S}$ and demands states $\mathbf{D}$ as:

$$Q(\mathbf{S}, \mathbf{D}) \triangleq K(\mathbf{S}, \mathbf{D}) + \overline{C}^P + C^R(\mathbf{S}, \mathbf{D}) \approx \overline{Q}. \tag{32}$$

Since $\overline{Q}$ combines all cost components in the network (operating costs, revenue, and cost of adapting), we use it as the main indicator of the profitability of the network when comparing scenarios and adaptation strategies.

6.5. Adaptation strategies

In Chapter 4, we present a method to find the state vector that minimizes the instantaneous readiness cost, given a fixed demand. This method is applied periodically, namely every Z_{mon} seconds, in order for the network to be updated of potential changes in the environment that lead to a change in the optimal state. Thus, the network knows at each monitoring interval whether the current state is optimal or not, along with the readiness cost associated with the current, the optimal, and past states.

Since both operating in a suboptimal state and moving to the optimal state are costly, we need a set of rules to decide whether staying at the current state is more beneficial than moving to the optimal state, or vice-versa. We refer to this set of rules as the *adaptation strategy*. We make a distinction between *static* strategies, which feature a state vector that does not change over time, and *dynamic* strategies, whose state vector may be adapted to changes in the environment. Although it may seem counter-intuitive to consider static strategies as adaptation strategies, in some cases the optimal rule may be not adapting at all.

If we could accurately predict the future, we could find the optimal adaptation strategy by using dynamic programming, and thus there would be no need to consider other strategies, unless it could not be implemented efficiently. Nonetheless, in this thesis we assume that no sophisticated method for predicting the evolution of the network is available, apart from having access to basic statistics. This is done with the intention of *not having to rely on the accuracy of our predictions* to justify the feasibility and profitability of a dynamic functional split adaptation.

In this section, we present seven adaptation strategies to compare its performance against each other. While some of them are described for reference purposes, such as the dynamic programming strategy, others are implementable strategies that could be used in a real RAN.

6.5.1. Uniform-static strategy

The *uniform-static strategy* is a static strategy in which the state vector

$$\mathbf{s}(\tau) = \mathbf{s}^{\text{ufst}} \qquad \forall \tau \in \{1, ..., \tau^{\max}\} \tag{33}$$

is constant and minimizes the readiness cost for a uniform distribution of UEs within the covered area. The state vector $\mathbf{s}^{\text{ufst}}$ can be obtained by firstly calculating the average ratios between interference and signal powers $\frac{i_{u,g}}{p_u}$ $\forall u \in \mathbb{U},\ \forall g \in \mathbb{G}$ for a uniform UE distribution (e. g. via a straightforward simulation), then computing coefficients $\ell^{m}_{g,k}$ $\forall s \in \mathbb{S},\ \forall g, k \in \mathbb{G}$ from these ratios, and finally use them to obtain the optimal state vector in (P20).

The total cost of obtained when using this approach serves as a upper bound for that of any *profitable* adaptation strategy, since it features no action cost and its readiness cost is only optimal for a uniform distribution of UEs average situation. Although probably suboptimal, this strategy can be always implemented in real RANs and requires almost no information from the environment.

Since the state vector never changes, there is no action cost component for this strategy. Hence, the mean total cost of the uniform-static strategy is:

$$\overline{Q}^{\text{ufst}} \approx K(\mathbf{S}^{\text{ufst}}, \mathbf{D}), \tag{34}$$

for long sequences of states $\mathbf{S}^{\text{ufst}} = [\mathbf{s}^{\text{ufst}} \ \cdots \ \mathbf{s}^{\text{ufst}}]$ and demands $\mathbf{D}$.

6.5.2. Mean-static strategy

Even if the network operator prefers to implement a static strategy, it may happen that the average UE concentration in the RAN is not uniform, but instead UEs tend to cluster around known places. In that case, it would be reasonable to take this information into account when calculating the optimal static solution. That is, instead of assuming a uniform distribution of UEs, the operator may use a different distribution of UEs that better represents their average location. Thus, we define the *mean-static strategy* as the static strategy in which the state vector is:

$$\mathbf{s}(\tau) = \mathbf{s}^{\text{mnst}} \qquad \forall \tau \in \{1, ..., \tau^{\max}\}, \tag{35}$$

where $\mathbf{s}^{\text{mnst}}$ is the state vector that minimizes the readiness cost for the actual, average distribution of UEs.

The readiness cost $\overline{Q}^{\text{mnst}}$ of the mean-static strategy is lower or equal than $\overline{Q}^{\text{ufst}}$ if the average distribution of users is estimated correctly, as can be trivially shown. Indeed, the mean-static strategy is the static strategy that yields the lowest possible readiness cost in a well-characterized network. As a result, it is a tighter upper bound for profitable dynamic strategies than the uniform-static strategy, and a lower bound for static strategies.

Similar to that of the uniform-static strategy, the mean total cost of the mean-static strategy is:

$$\overline{Q}^{\text{mnst}} \approx K(\mathbf{S}^{\text{mnst}}, \mathbf{D}), \tag{36}$$

for long sequences of states $\mathbf{S}^{\text{mnst}} = [\mathbf{s}^{\text{mnst}} \ \cdots \ \mathbf{s}^{\text{mnst}}]$ and demands $\mathbf{D}$.

6.5.3. Impatient strategy

The *impatient strategy* is the simplest of the dynamic strategies, as it consists in changing the state vector every time a new optimal one is detected, that is:

$$\mathbf{s}(\tau) = \mathbf{s}^*(\tau) \qquad \forall \tau \in \{1, ..., \tau^{\max}\}. \tag{37}$$

As extensively discussed in Chapter 4, and especially in Sec. 4.6, we define the optimal state $\mathrm{s}^*(\tau)$ as that state minimizing the instantaneous readiness cost at time τ. The optimization problem that has to be solved to achieve this is:

$$\langle \mathbf{b}^*, \mathbf{r}^*, \mathbf{f}^* \rangle = \arg\min_{\mathbf{b},\mathbf{r},\mathbf{f}} \ \widehat{K}_{\text{comp}}(\mathbf{b}) + K_{\text{rout}}(\mathbf{f}) + \xi K^{\text{ref}}_{\text{oper}}(\upsilon - 1)\frac{\beth_{\text{lin}}\beth_{\text{sim}}}{2\widetilde{\eta}_{\text{ref}}} \sum_{m=1}^{M-1}\sum_{g=1}^{G} r_g^m \tag{P20a}$$

subject to

$$\sum_{g=1}^{G} f_e^g \leq \vartheta_e \qquad \forall e \in \mathbb{E}, \tag{P1c}$$

$$f_e^g \geq 0 \qquad \forall e \in \mathbb{E},\ \forall g \in \mathbb{G}, \tag{P1d}$$

$$\sum_{e \in \mathbb{E}^+(n)} f_e^g - \sum_{e \in \mathbb{E}^-(n)} f_e^g = \begin{cases} 0 & \forall n \in \mathbb{N} \backslash \{n_0, n_g\} \\ \widehat{\nu}(\mathbf{b}_g) & \text{for } n = n_0 \\ -\widehat{\nu}(\mathbf{b}_g) & \text{for } n = n_g \end{cases} \qquad \forall g \in \mathbb{G}, \tag{P5b}$$

$$0 \leq r_g^m \leq R_g^m b_g^m \qquad \forall m \in \widehat{\mathbb{M}}, \forall g \in \mathbb{G}, \tag{P12b}$$

$$r_g^m \geq \sum_{k=1}^{G} \ell_{g,k}^m b_g^m - (1 - b_g^m) R_g^m \qquad \forall m \in \widehat{\mathbb{M}}, \forall g \in \mathbb{G}, \tag{P12c}$$

$$r_g^m \leq \sum_{k} \ell_{g,k}^m b_g^m \qquad \forall m \in \widehat{\mathbb{M}}, \forall g \in \mathbb{G}, \tag{P12d}$$

$$b_g^1 \geq b_g^2 \geq ... \geq b_g^{M-1} \qquad \forall g \in \mathbb{G}, \tag{P5c}$$

$$\mathbf{b} \in \{0, 1\}^G. \tag{P5d}$$

From optimal variables $\mathbf{b}^*$ and $\mathbf{f}^*$ we can obtain the optimal instantaneous state $\mathbf{s}^*$ as:

$$\mathbf{s}^* \triangleq \langle \mathbf{a}^*, \mathbf{f}^* \rangle = \langle \mathbf{b}^* \cdot (\mathbf{I}_G \otimes \mathbf{1}_{M-1}), \mathbf{f}^* \rangle \,, \tag{38}$$

where $\mathbf{I}_g$ is the identity matrix of size g, $\mathbf{1}_m$ is an all-ones column vector of size m, and $\otimes$ is the Kronecker product.

From solving (P20) every monitoring interval, we obtain the sequence of optimal states $\mathbf{S}^*$, from which we can estimate $\overline{K}^*$ and $\overline{C}^{R*}$ as discussed in Sec. 6.4.3. Therefore, the mean total cost $\overline{Q}^{\text{impa}}$ of the impatient strategy can be approximated from a long optimal state sequence $\mathbf{S}^*$ as:

$$\overline{Q}^{\text{impa}} = \overline{K}^* + \overline{C}^P + \overline{C}^{R*} \approx \overline{Q}(\mathbf{S}^*, \mathbf{D}). \tag{39}$$

We can see that the readiness cost achieved by this impatient strategy is always minimal, while the action cost is always maximal, as we can deduce from (21) and (26).

Alternatively, we can also use the cost-of-flexibility framework of Chapter 3 to predict $\overline{Q}^{\text{impa}}$, since its readiness cost is that of a dynamic network with maximum flexibility $\varphi = 1$. Therefore, we can use (15) to formulate the mean total cost as:

$$\overline{Q}^{\text{impa}} = \alpha K_\Delta(0) + (1-\alpha)\beta \widehat{K}_\beta(1) + \overline{C}^P + \overline{C}^{R*}. \tag{40}$$

6.5.4. Random deferral strategy

Whereas with the impatient strategy the state vector $\mathbf{s}(\tau)$ changes every time there is a change in the optimal state vector, with the *random deferral strategy* the decision of moving to the optimal state is taken randomly with probability φ. That is, the resulting state sequence of one random realization can be formulated as:

$$\mathbf{S}^{\text{rds}} = [\mathbf{s}^{\text{rds}}(1) \;\cdots\; \mathbf{s}^{\text{rds}}(\tau^{\max})], \tag{41}$$

where

$$\mathbf{s}^{\text{rds}}(\tau) = \begin{cases} \mathbf{s}^*(\tau) & \text{with probability } \varphi, \\ \mathbf{s}^{\text{rds}}(\tau-1) & \text{otherwise.} \end{cases} \tag{42}$$

The advantage of this strategy, apart from its simplicity, is that its mean total cost $\overline{Q}^{\text{rds}}$ accepts a direct modeling from the cost-of-flexibility framework, as a function of the adaptation probability φ (which is equivalent to the concept of maximum flexibility) and the mean reaction cost of the optimal state sequence:

$$\overline{Q}^{\text{rds}}\Big(\varphi, \overline{C}^{R*}\Big) = \alpha\varphi K_\Delta(0) + (1-\alpha\varphi)\beta\varphi \widehat{K}_\beta(\varphi) + \varphi \overline{C}^{R*}. \tag{43}$$

In addition, for a given $\overline{C}^{R*}$, the value of φ may not be arbitrary, but we can select

$\varphi = \varphi^{\text{ords}}$ so that the total cost is minimized. We refer to this strategy as the *optimal random deferral strategy*. The value of φ^{ords} can be obtained after solving the following equation:

$$\frac{\partial}{\partial\varphi}\overline{Q}^{\text{rds}}\Big(\varphi, \overline{C}^{R*}\Big)\bigg|_{\varphi=\varphi^{\text{ords}}} = 0. \tag{44}$$

Equation (44) does not accept a closed-form solution, but it can be easily solved by numerical methods.

6.5.5. Dynamic programming strategy

The *dynamic programming strategy* uses dynamic programming to find the optimal sequence of states given full information about the optimal states in the future, along with their readiness and action costs. Since we assume that a real network may not have access to sophisticated predictions, we only use the results of this strategy to provide a lower bound to the total cost that any feasible strategy can achieve.

In order to use dynamic programming, we model the system in the following way. We assume that we already know the full vector of optimal states $\mathbf{S}^*$ for the whole time interval $\{1, ..., \tau^{\max}\}$. This optimal state vector contains $\tau^{\max}$ states, which are all the states that our dynamic programming algorithm can choose from. In theory, we could not only provide those states in $\mathbf{S}^*$ but also the optimal states for each possible combination of monitoring intervals, or even all possible states in the state space $\mathbb{S}^G$. However, either option may lead to a combinatorial explosion even for moderate values of G or $\tau^{\max}$, which is known as the *curse of dimensionality* in dynamic programming [Pow07]. Fortunately, since our goal when using this technique is only to provide a lower bound to the cost of non-prediction-based strategies, using just the optimal states for every time instant is enough, since these are also the only information that is available to the other techniques.

We define the total cost-to-go function $\Lambda(\tau, \mathbf{s}^*(\tau'))$ as the minimum total cost that the network may produce when starting from time instant τ and state $\mathbf{s}^*(\tau')$ till $\tau^{\max}$. From this definition, we can define Bellman's equation as [Bel66]:

$$\Lambda(\tau, \mathbf{s}^*(\tau')) = \min_{\epsilon\in\{1,...,\tau^{\max}\}} \Big\{\Lambda(\tau+1, \mathbf{s}^*(\epsilon)) + C^R(\mathbf{s}^*(\tau'), \mathbf{s}^*(\epsilon), \mathbf{d}(\tau)) + K(\mathbf{s}^*(\epsilon), \mathbf{d}(\tau+1))\Big\} \tag{45}$$

The term $C^R(\mathbf{s}^*(\tau'), \mathbf{s}^*(\epsilon), \mathbf{d}(\tau))$ represents the reaction cost of moving from state $\mathbf{s}^*(\tau')$, which is an arbitrary state, to the next selected state $\mathbf{s}^*(\epsilon)$. Conversely, the term $K(\mathbf{s}^*(\epsilon), \mathbf{d}(\tau+1))$ reflects the readiness cost of operating in the next selected state. Equation (45) can be efficiently solved using backwards induction [Bel66]. The result is a vector of states that minimizes the total cost $\overline{Q}^{\text{DP}}$, which can be calculated as:

$$\overline{Q}^{\text{DP}} = \min_{\epsilon\in\{1,...,\tau^{\max}\}} \Big\{\Lambda(1, \mathbf{s}^*(\epsilon))\Big\}. \tag{46}$$

This total cost is a lower bound to any non-prediction-based strategies, but it cannot be implemented in actual RANs unless advanced prediction techniques are used, since it requires knowledge about future optimal states.

6.5.6. Greedy strategy

Although a real 5G RAN cannot predict future optimal states, it can indeed be aware of current past optimal states, which can be used to decide when it is a good time to move to an optimal state. The *greedy strategy* follows this principle by triggering the adaptation at local minima in the accumulated cost gap between current and the optimal states.

Let us define function $Д(\tau, \epsilon)$ for $\epsilon \in \{0, 1, 2, ...\}$ as:

$$Д(\tau, \epsilon) \triangleq \sum_{\sigma=0}^{\epsilon} \left[K(\mathbf{s}^*(\tau), \mathbf{d}(\tau + \sigma)) - K^*(\tau + \sigma) \right], \tag{47}$$

This function is the accumulated readiness cost gap between operating at an obsolete optimal state and the optimal readiness cost. Let us consider a simple case in which $Д(\tau, \epsilon)$ is deterministic, fully known, and wide-sense stationary such that it only depends on the gap ϵ but not on the initial instant τ. Then, with slight abuse of notation, we can drop the dependency on τ:

$$Д(\tau, \epsilon) = Д(\epsilon) \quad \forall \tau, \epsilon. \tag{48}$$

Since $Д(\epsilon)$ is the accumulated readiness cost gap with respect to the optimal, minimal readiness cost, it clear that $Д(\epsilon)$ is an increasing function, and thus it is easy to show that the optimal strategy would be to change the state with a fixed periodicity ϵ_{min} such that:

$$\frac{Д(\epsilon_{\text{min}}) + \widehat{C}^R}{\epsilon_{\text{min}}} \leq \frac{Д(\epsilon) + \widehat{C}^R}{\epsilon}, \quad \forall \epsilon \in \{0, 1, ...\}, \tag{49}$$

where

$$\widehat{C}^R \triangleq \frac{\tau^{\max} \overline{C}^{R*}}{\sum_{\tau=2}^{\tau^{\max}} \left[\mathbf{s}^*(\tau) \neq \mathbf{s}^*(\tau - 1) \right]} \tag{50}$$

is the average reaction cost of non-zero migrations, and $[\cdot]$ is the Iverson bracket, which returns 1 if the expression inside is true. This follows from the fact that $\frac{Д(\epsilon)+\widehat{C}^R}{\epsilon}$ is the mean total cost for a fixed ϵ, which includes the accumulated cost of not changing the state for ϵ intervals and the cost of eventually adapting. As a result, the network only needs to keep track of the evolution of $\frac{Д(\epsilon)+\widehat{C}^R}{\epsilon}$ every monitoring interval and perform a state change whenever an increase is detected from the previous value, thus following a greedy behavior. A detailed description of the procedure to set the

Algorithm 2: Greedy strategy to dynamically select the optimal functional split.

```
Input: $\widehat{C}^R$ (estimated)
$\tau \leftarrow 1$                                          // Time index
$\mathbf{s}^{\text{grdy}}(\tau) \leftarrow \mathbf{s}^*(\tau)$                       // Initial adaptation
$\tau_{\text{last}} \leftarrow \tau$                                // Time of last adaptation
repeat
    $\tau \leftarrow \tau + 1$
    $\mathbf{s}^*(\tau) \leftarrow$ from solving (P20)         // Calculate new optimal state
    $Д'_{\text{last}} \leftarrow K(\mathbf{s}^*(\tau_{\text{last}}), \mathbf{d}(\tau)) - K^*(\tau)$  // Cost gap from previous state
    if $\frac{Д_{\text{last}}+\widehat{C}^R}{\tau-\tau_{\text{last}}-1} \leq \frac{Д_{\text{last}}+Д'_{\text{last}}+\widehat{C}^R}{\tau-\tau_{\text{last}}}$ then
        $\mathbf{s}^{\text{grdy}}(\tau) \leftarrow \mathbf{s}^*(\tau)$  // If relative cost gap increases, adapt
        $\tau_{\text{last}} \leftarrow \tau$
        $Д_{\text{last}} \leftarrow 0$
    else
        $Д_{\text{last}} \leftarrow Д_{\text{last}} + Д'_{\text{last}}$ // If not, stay and accumulate cost gap
    end
end
```

sequence of the state vectors

$$\mathbf{S}^{\text{grdy}} \triangleq [\mathbf{s}^{\text{grdy}}(\tau) \ \cdots \ \mathbf{s}^{\text{grdy}}(\tau^{\max})] \tag{51}$$

is shown in Algorithm 2.

Since the operation of the greedy strategy depends on the specific events that the network is facing, we can only model its total cost by applying definition (32):

$$\overline{Q}^{\text{grdy}} = Q(\mathbf{S}^{\text{grdy}}, \mathbf{D}). \tag{52}$$

6.5.7. Adaptive threshold strategy

The greedy strategy assumes that there is a single, deterministic accumulated readiness cost gap function $Д(\tau, \epsilon)$. This is a rather unrealistic assumption, since the cost gap between current and optimal solutions can evolve in multiple ways depending on the initial state vector and the evolution of the network. The *adaptive threshold strategy* strategy relaxes these assumptions and makes them closer to the actual system dynamics, while providing a method for iterative adjustment. While being simple to implement, if offers very promising results with respect to the other approaches.

Let us suppose that instead of a single $Д(\epsilon)$ function, we have an arbitrary number $Л$ of them. That is, the readiness cost of a state can degrade in $Л$ different ways after it stops being optimal. We denote each of these functions as $Д_i(\epsilon_i)$, $i \in \{1, ..., Л\}$. Without loss of generality, we assume that all $Д_i(\epsilon_i)$ are equally likely, i. e., they appear at similar rates. We denote by ϵ_i the time during which these functions apply, which we can modify by performing or delaying adaptations. Then, after time $\sum_{i=1}^{Л} \epsilon_i$, the mean total cost can be computed as:

$$\overline{Q} = \frac{\sum_{i=1}^{Л} Д_i(\epsilon_i) + Л\widehat{C}^R}{\sum_{i=1}^{'} \epsilon_i} \tag{53}$$

If we relax the ϵ_i variables and assume they are continuous, we can find the minimum[1] by setting their partial derivatives to 0:

$$\frac{\partial}{\partial \epsilon_i} \frac{\sum_{i=1}^{Л} Д_i(\epsilon_i) + Л\widehat{C}^R}{\sum_{i=1}^{Л} \epsilon_i} = 0, \qquad \forall i \in \{1, ..., Л\}, \tag{54}$$

which leads to the following identities:

$$Д_1'(\epsilon_1) = Д_2'(\epsilon_2) = ... = Д_Л'(\epsilon_Л) \tag{55}$$

$$Д_1'(\epsilon_1) \cdot \left(\sum_{i=1}^{Л} (\epsilon_i \right) - \sum_{i=1}^{Л} Д_i(\epsilon_i) = Л\widehat{C}^R \tag{56}$$

where $Д_i'(\epsilon_i) \triangleq \frac{\partial}{\partial \epsilon_i} Д_i(\epsilon_i)$. The first set of equations leads to an interesting conclusion: all derivatives of $Д_i(\epsilon_i)$ reach the same value for an optimal selection of continuous ϵ_i. In our model, variables ϵ_i are discrete, thus we cannot actually compute derivatives. Nonetheless, we can approximate the continuous first derivative by the discrete first difference:

$$\begin{aligned} Д_i'(\epsilon_i) &\approx Д_i(\epsilon_i) - Д_i(\epsilon_i - 1) \qquad (57) \\ &= K(\mathbf{s}^*(\tau), \mathbf{d}(\tau + \epsilon_i)) - K^*(\tau + \epsilon_i) \qquad (58) \end{aligned}$$

If we combine (55) and (56), we can define the threshold value $Д_{\text{thres}}'$ at which we reach the optimal selection of ϵ_i $\forall i \in \{1, ..., Л\}$ as:

$$Д_{\text{thres}}' \triangleq Д_1'(\epsilon_1) = \frac{\widehat{C}^R + \overline{Д}}{\overline{\epsilon}}, \tag{59}$$

[1] Since we model the first derivatives $Д_i'(\epsilon_i)$ as positive increasing functions, it can be easily shown that $Д_i(\epsilon_i)$ and (53) are convex functions.

where

$$\overline{\text{Д}} \triangleq \frac{1}{D}\sum_{i=1}^{D} \text{Д}_i(\epsilon_i) \tag{60}$$

and

$$\overline{\epsilon} \triangleq \frac{1}{D}\sum_{i=1}^{D} \epsilon_i. \tag{61}$$

That is, $\overline{\text{Д}}$ is simply the average accumulated readiness cost gap right before an adaptation, and $\overline{\epsilon}$ is the average duration of holding an obsolete state before an adaptation. Both components can be easily observed by the network, and they can be used to iteratively find the optimal threshold $\text{Д}'_{\text{thres}}$. A complete description of the procedure to set the values the sequence of the state vectors

$$\mathbf{S}^{\text{adth}} \triangleq [\mathbf{s}^{\text{adth}}(\tau) \ \cdots \ \mathbf{s}^{\text{adth}}(\tau^{\max})] \tag{62}$$

is shown in Algorithm 3.

In order to kick-start this algorithm, we can initially use the greedy algorithm to obtain the first estimations of $\widehat{C}^R$ and $\text{Д}'_{\text{thres}}$. Possible extensions to this algorithm may be to use an exponential moving average to compute $\overline{\text{Д}}$ and $\overline{\epsilon}$, so that the threshold responds faster to changes in the UE distribution or channel conditions. In addition, the average non-zero reaction cost $\widehat{C}^R$ can also be iteratively calculated from the observed state changes if a good estimation is not available.

As with the greedy strategy, we can only model the total cost of the adaptive threshold strategy by applying definition (32):

$$\overline{Q}^{\text{adth}} = Q(\mathbf{S}^{\text{adth}}, \mathbf{D}). \tag{63}$$

In Table 6.1, we present a comparison of all proposed adaptation strategies, where we show their main features and the optimality of their readiness and action costs.

6.6. Reference action cost frontiers

In a 5G RAN featuring a dynamically-adapted functional split, the relationship between the readiness cost of operating at a possibly suboptimal state and the action cost of performing and state change is crucial. If the action cost is negligible, then there is no objection to adapt whenever possible. Nonetheless, as the action cost grows with respect to the readiness cost, adaptation should be more careful. Indeed, if the action cost is high, dynamic adaptation may not be worthwhile at all, thus being static states desirable instead. As a result, it is useful for an operator to have a set of *reference action cost frontiers* in order to decide whether dynamic adaptation is worthwhile or not.

In addition to the lower bound of the reaction cost $\overline{C}_{\text{I}}^{R*}$ presented in Sec. 3.4.2, we

Algorithm 3: Adaptive threshold strategy to dynamically select the optimal functional split.

```
Input: $\widehat{C}^R$, $Д'_{\text{thres}}$ (estimated)
$\tau \leftarrow 1$                                          // Time index
$\mathbf{s}^{\text{adth}}(\tau) \leftarrow \mathbf{s}^*(\tau)$              // Initial adaptation
$\tau_{\text{last}} \leftarrow \tau$                                 // Time of last adaptation
$m \leftarrow 0$                                             // Adaptations counter
repeat
    $\tau \leftarrow \tau + 1$
    $\mathbf{s}^*(\tau) \leftarrow$ from solving (P20)              // Calculate new optimal state
    $Д'_{\text{last}} \leftarrow K(\mathbf{s}^*(\tau_{\text{last}}), \tau) - K^*(\tau)$    // Cost gap from previous state
    $Д_{\text{last}} \leftarrow Д_{\text{last}} + Д'_{\text{last}}$            // Accumulate total cost gap
    if $Д'_{\text{last}} \geq Д'_{\text{thres}}$ then                    // Threshold check
        $\mathbf{s}^{\text{adth}}(\tau) \leftarrow \mathbf{s}^*(\tau)$              // Adaptation
        $\tau_{\text{last}} \leftarrow \tau$
        $m \leftarrow m + 1$
        $\bar{\epsilon} \leftarrow \bar{\epsilon}\frac{m}{m+1} + \frac{\tau - \tau_{\text{last}}}{m+1}$
        $\overline{Д} \leftarrow \overline{Д}\frac{m}{m+1} + Д_{\text{last}}\frac{1}{m+1}$
        $Д'_{\text{thres}} \leftarrow \frac{\widehat{C}^R + \overline{Д}}{\bar{\epsilon}}$             // Update threshold
        $Д_{\text{last}} \leftarrow 0$
    end
end
```

Strategy	Type	Implementable	Readiness cost	Action cost
Uniform-static	Static	Yes	Suboptimal	None
Mean-static	Static	Yes	Minimal (of static strategies)	None
Impatient	Dynamic	Yes	Minimal	Maximal
Random deferral	Dynamic	Yes	Suboptimal	Suboptimal
Dynamic programming	Dynamic	No	Optimal	Optimal
Greedy	Dynamic	Yes	Suboptimal	Suboptimal
Adaptive threshold	Dynamic	Yes	Suboptimal	Suboptimal

Table 6.1.: Comparison of proposed adaptation strategies. The term "optimal" implies that the readiness or action costs are such that the total cost is optimal.

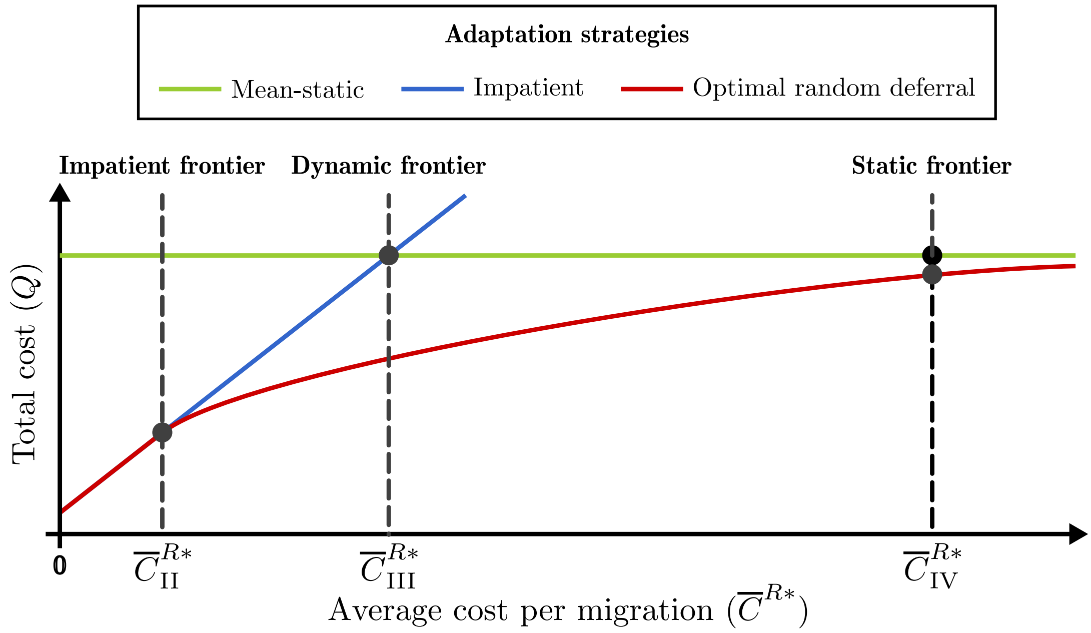


Figure 6.2.: Location of the reference action cost frontiers.

propose three theoretically-derived reference points for the reaction cost: the *impatient frontier* $\overline{C}_{\text{II}}^{R*}$, the *dynamic frontier* $\overline{C}_{\text{III}}^{R*}$, and the *static frontier* $\overline{C}_{\text{IV}}^{R*}$. We only focus on the reaction cost as the main component of the action cost, since the proaction cost is probably negligible, as we show in Sec. 6.7.4.

The theoretical values of $\overline{C}_{\text{II}}^{R*}$, $\overline{C}_{\text{III}}^{R*}$, and $\overline{C}_{\text{IV}}^{R*}$ can be calculated from $F_{\mathcal{T}}(t)$, the RDF $K_{\Delta}(\delta)$, and a reference upper limit to the total cost, such as that of the mean-static strategy $\overline{Q}^{\text{mnst}}$. Although we still need to simulate the behavior of our 5G RAN to obtain accurate characterizations of these components, its use removes the need for simulating the adaptation strategies individually. Owing to its good modeling characteristics, we use the *optimal random deferral strategy* as the reference dynamic strategy to calculate these frontiers.

6.6.1. Impatient frontier

The impatient frontier $\overline{C}_{\text{II}}^{R*}$ is defined as the last value of $\overline{C}^{R*}$ where the impatient and the optimal random deferral strategies are equivalent. In other words, up to $\overline{C}_{\text{II}}^{R*}$, the mean reaction cost $\overline{C}^{R*}$ of always adapting is so low that the total cost of impatient adaptation is near-optimal, and thus adapting the state every time the optimal state changes is the best strategy.

The cost-of-flexibility framework presented in Chapter 3 allows us to directly compute the point where the random deferral strategy first departs from the impatient strategy. We know from Properties 3.4.9 and 3.4.10 that (15) is a decreasing and convex function on φ (see Sec. 3.4.2 and Appendices A.2 and A.1). As a result, (43) is also convex, which implies that there may be at most one value of φ that produces a global minimum. Nonetheless, since $\varphi \in [0,1]$ for realistic strategies, if the φ that minimizes (43) is greater or equal than 1, the φ that actually minimizes (43) must be $\varphi = 1$, in other words, the impatient strategy. Therefore, $\overline{C}_{\mathrm{II}}^{R*}$ must satisfy this identity:

$$\left.\frac{\partial \overline{Q}^{\mathrm{rds}}(\varphi, \overline{C}_{\mathrm{II}}^{R*})}{\partial \varphi}\right|_{\varphi=1} = 0. \tag{64}$$

After solving (64) for $\overline{C}_{\mathrm{II}}^{R*}$, we obtain the following expression:

$$\overline{C}_{\mathrm{II}}^{R*} = -\alpha K_{\Delta}(0) + \beta \sum_{\delta=1}^{\infty} K_{\Delta}(\delta) \left[1 - 2\alpha - \beta \frac{1-\alpha}{1-\beta}(\delta - 1)\right] (1-\beta)^{\delta-1}. \tag{65}$$

In Fig. 6.2, we show the location of the impatient frontier in an abstract representation of the possible relationships between the total cost and the average cost per migration for four adaptation strategies. We see that the total cost of the mean-static strategy does not depend on $\overline{C}^{R*}$, whereas the impatient and optimal random deferral strategies exhibit linear and concave dependencies on $\overline{C}^{R*}$, respectively. It is clear that if $0 \leq \overline{C}^{R*} \leq \overline{C}_{\mathrm{II}}^{R*}$ the impatient strategy is optimal, hence the operator may select this simple strategy in this region.

6.6.2. Dynamic frontier

The dynamic frontier $\overline{C}_{\mathrm{III}}^{R*}$ is defined as the value of $\overline{C}^{R*}$ where the total cost of the impatient strategy intersects a reference upper bound, such as the total cost of the mean-static strategy $\overline{Q}^{\mathrm{mnst}}$. This implies that the mean total cost resulting from using dynamic strategies with a mean action cost between $\overline{C}_{\mathrm{II}}^{R*}$ and $\overline{C}_{\mathrm{III}}^{R*}$ is still lower than that of static strategies, but the impatient strategy is not optimal in this region anymore. Therefore, we define $\overline{C}_{\mathrm{III}}^{R*}$ as the value of $\overline{C}^{R*}$ that fulfills the following identity:

$$\left.\overline{Q}^{\mathrm{impa}}\right|_{\overline{C}^{R*}=\overline{C}_{\mathrm{III}}^{R*}} = \overline{Q}^{\mathrm{mnst}}. \tag{66}$$

After solving (40) for $\overline{C}_{\mathrm{III}}^{R*}$, we obtain:

$$\overline{C}_{\mathrm{III}}^{R*} = \overline{Q}^{\mathrm{mnst}} - \alpha K_{\Delta}(0) - (1-\alpha)\beta \widehat{K}_{\beta}(1). \tag{67}$$

In Fig. 6.2, the dynamic frontier is represented as the point where the mean-static and impatient strategies intersect. If $\overline{C}^{R*} < \overline{C}_{\mathrm{III}}^{R*}$, the impatient strategy may be suboptimal but is still less costly than the mean-static strategy. However, if $\overline{C}^{R*} \geq \overline{C}_{\mathrm{III}}^{R*}$ only advanced dynamic strategies may result in less total cost than static strategies.

6.6.3. Static frontier

The static frontier $\overline{C}^{R*}_{\text{IV}}$ is defined as the value of $\overline{C}^{R*}$ beyond which the total cost of using dynamic adaptation strategies is comparable to the cost of implementing the mean-static strategy. Thus, for $\overline{C}^{R*} > \overline{C}^{R*}_{\text{IV}}$, using a static strategy may be the best option, since dynamic strategies may perform just marginally better or even worse.

One possible approach to estimate $\overline{C}^{R*}_{\text{IV}}$ would be to find the mean action cost such that:

$$\left.\frac{\partial \overline{Q}^{\text{rds}}(\varphi, \overline{C}^{R*}_{\text{IV}})}{\partial \varphi}\right|_{\varphi=0} = 0. \tag{68}$$

Similarly to the derivation of $\overline{C}^{R*}_{\text{II}}$, solving (68) for $\overline{C}^{R*}_{\text{IV}}$ would give us the mean reaction cost such that the random deferral strategy is as costly as no adaptation at all ($\varphi = 0$). However, this approach has two important drawbacks. On the one hand, this would yield the maximum reaction cost for which the random deferral approach is, on average, less costly *than an obsolete static state*. We want, however, to compare dynamic strategies with the mean-static strategy, which is a simple but reasonable strategy. On the other hand, the accuracy of solving (68) for $\overline{C}^{R*}_{\text{IV}}$ may also be low, since a measurement- or simulation-based RDF $K_\Delta(\delta)$ may include few points for large δ, as they correspond to large intervals without adaptations, which are relatively rare events.

With the intention of both improving the accuracy and making sure that $\overline{C}^{R*}_{\text{IV}}$ is a lower bound to the maximum admissible reaction cost, we define $\overline{C}^{R*}_{\text{IV}}$ as the point where the cost of random deferral strategy equals the cost of the mean-static strategy for a reference flexibility $\varphi = \varphi_{\text{ref}}$:

$$\overline{Q}^{\text{rds}}(\varphi_{\text{ref}}, \overline{C}^{R*}_{\text{IV}}) = Q^{\text{mnst}}. \tag{69}$$

After solving for $\overline{C}^{R*}_{\text{IV}}$:

$$\overline{C}^{R*}_{\text{IV}} = \frac{Q^{\text{mnst}}}{\varphi_{\text{ref}}} - \alpha K_\Delta(0) - (1 - \alpha\varphi_{\text{ref}})\beta \widehat{K}_\beta(\varphi_{\text{ref}}) \tag{70}$$

The value of φ_{ref} should be low enough to obtain a tight lower bound, but cannot be lower than the accuracy that we have for the RDF. Based on our simulation results, we suggest using $\varphi_{\text{ref}} = 0.1$, as it is shown in the next section.

In Fig. 6.2, we depict the static frontier as the point where the difference between the optimal random deferral and the mean-static strategies surpasses a small predefined threshold. As a result, if $\overline{C}^{R*} > \overline{C}^{R*}_{\text{IV}}$, the benefits of operating a dynamically adaptive RAN are low, thus the operator may consider implementing a static strategy instead.

6.7. Experimental results

In this section, we simulate a realistic 5G RAN to evaluate the adaptation strategies proposed in Sec. 6.5. In addition, we apply the cost-of-flexibility framework to derive an adequate monitoring period for the network, as discussed in Sec. 6.4.3, and to predict the reference action cost frontiers, as presented in Sec. 6.6.

6.7.1. Simulator description

We use a MATLAB simulator to generate the gNBs, the fronthaul network, and the UE positions from which we obtain all required parameters to formulate the FSSP as in (P20). This problem is tackled using a commercial optimization solver on operator-grade hardware, and then another MATLAB simulator is used to assess the cost of the adaptation strategies.

Simulated mobile coverage

We follow the recommendations for simulating dense urban scenarios provided in 3GPP TS38.193 [3GP20b] so as to produce simulation results as realistic as possible. Consequently, gNBs are divided into a macro and a micro layers. The macro gNBs are located at the nodes of a hexagonal grid with an inter-site distance of 200 m. The micro gNBs are randomly (and uniformly) distributed over the covered area. This results in a density of around 29 macro gNBs per square kilometer. There are three times more micro gNBs than macro gNBs, which leads to an the average cell density of approximately 115 gNBs/km^2. In our experiments, we set the total number of gNBs to be $G = 300$, thus the covered area is 2.6 km^2, which approximately corresponds to the center of a medium-sized or large city. Regarding the number of UEs, it is recommended to consider 10 active UEs per gNB on average, that is, $U = 3000$ and a UE density of 1150 UEs/km^2.

Time-dependency and UE mobility

Each simulation run consists of $\tau^{\max} = 960$ monitoring intervals, which for a monitoring period of $Z_{\text{mon}} = 30$ s (as we estimate in Sec. 6.7.3) corresponds to a simulated time of 8 hours per simulation run. For each monitoring interval τ, a new UE distribution is generated as an evolution from that of $\tau - 1$, and then (P20) is solved to obtain a new $\mathrm{s}^*(\tau)$. The distribution and mobility of UEs during this simulated time correspond to that described in [Xu+16] for transport, entertainment, and comprehensive areas, since those are the area types that we can most frequently find in city centers. Namely, UEs are divided into two layers of simulated mobility. A subset of UEs roam across the covered area without any preference for the traversed places, while the remaining UEs move around randomly-positioned UE clusters.

The *number of UE clusters* for a given simulation run is denoted by parameter $\varpi \in \{0, ..., 8\}$. This allows us to evaluate slow-varying scenarios with no UE clusters (except for those occasionally formed by the random roaming of UEs) when $\varpi = 0$, and fast-varying scenarios with one UE cluster per hour, on average, when $\varpi = 8$. The rising and falling times of each cluster are randomly selected between 25 and 45 minutes, while their peak durations range from 15 minutes to 4 hours, in accordance with the mobility data presented in [Xu+16]. Since the number of UE clusters ϖ reflects how variable the UE distribution is, it is used as a variable in the experiments.

Fronthaul network

Based on the results of our previous work presented in Chapter 4, we set the number of possible centralization levels to $M = 4$, which in our case corresponds to the PDCP-RLC, MAC-PHY, Intra-PHY, and C-RAN functional split options. In accordance with these options, we use the interference-cancellation vector $\mathbf{c} = \langle 1, 0.6, 0.2, 0.01 \rangle$ as suggested in [PM17]. This implies that there is no interference cancellation with PDCP-RLC split, whereas interference power is reduced by 20 dB when using C-RAN.

Regarding the fronthaul network, we use the capacities vector $\mathbf{r} = \langle 4, 8, 80, 160 \rangle$ Gb/s, as provided in [3GP17]. The link capacity is set to $\vartheta_e = 1$ Tb/s according to [GS+18a], where the authors describe actual fronthaul networks on Italy, Romania, and Switzerland. Moreover, we base on the results shown in Chapter 4 [MAJK21] to generate the fronthaul network with an *average fronthaul network degree*, defined as the ratio of links to fronthaul switches, of 3.5 by means of a Waxman topology model [Wax88].

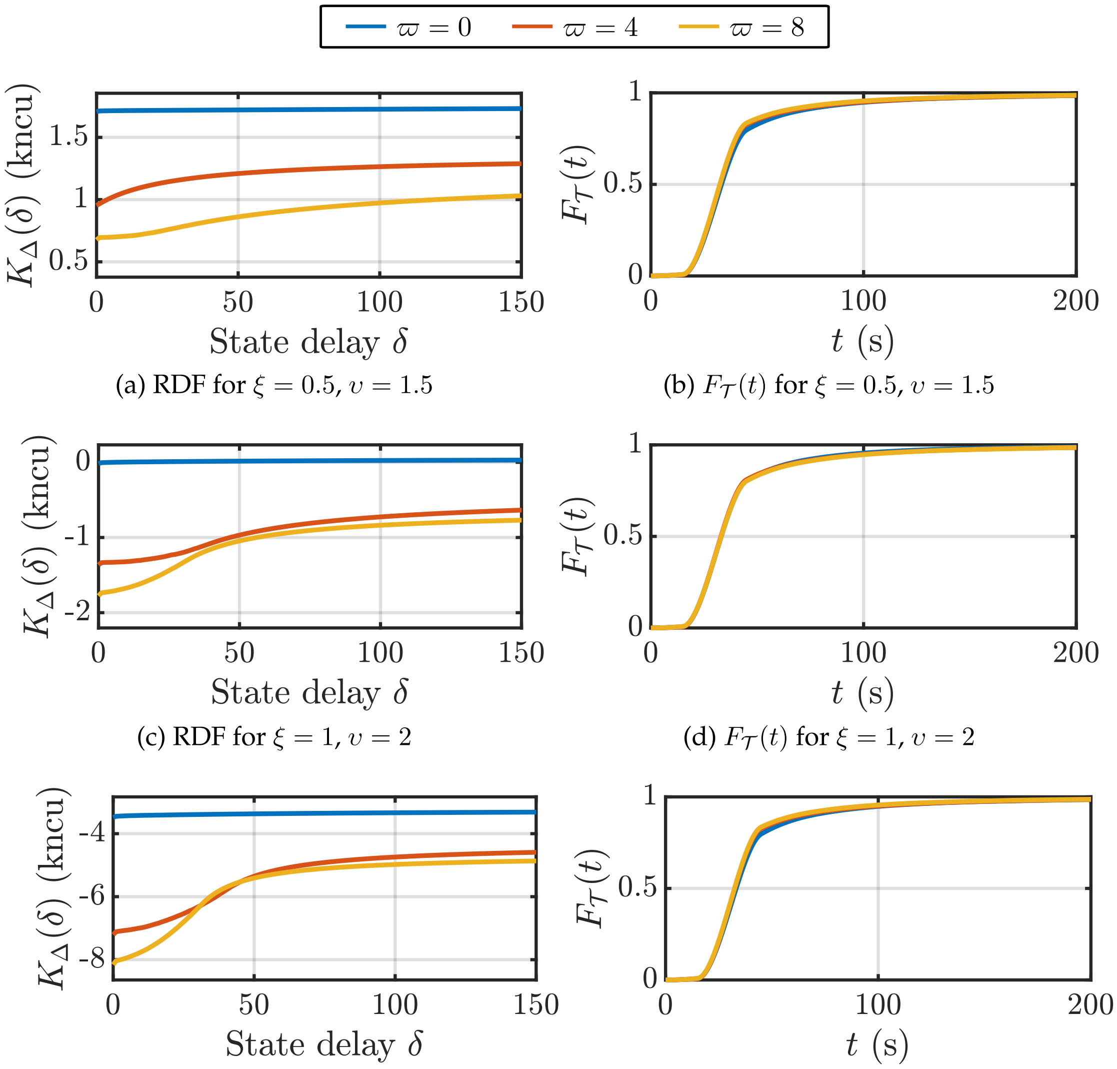
Computing platform and simulation time

With the intention of providing realistic simulation results, we use an operator-grade hardware platform featuring 44 Intel Xeon E5 cores [Bas+17]. For solving the FSSP, we use the commercial Gurobi MILP solver [Gur].

We repeat each simulation run (consisting of $\tau^{\max} = 960$ data points) 100 times to ensure statistically tight results. Consequently, for each simulation experiment we have a total of 96,000 data points, corresponding to approximately 33 days of simulated time.

6.7.2. Derivation of $K_\Delta(\delta)$ and $F_\mathcal{T}(t)$

After performing simulations for $\varpi \in \{0, ..., 8\}$ UE clusters, $\xi = \{0.5, 1, 2\}$ reference revenue-operating cost ratios, and $\upsilon = \{1.5, 2, 3\}$ revenue growth rates (81 simulation instances in total, one per combination of parameters), we apply the techniques described in Sec. 6.4.3, Appendix A.3, and Appendix A.4 to estimate the readiness degradation function $K_\Delta(\delta)$ and the distribution of demand durations $F_\mathcal{T}(t)$.



(a) RDF for $\xi = 0.5$, $\upsilon = 1.5$

(b) $F_\mathcal{T}(t)$ for $\xi = 0.5$, $\upsilon = 1.5$

(c) RDF for $\xi = 1$, $\upsilon = 2$

(d) $F_\mathcal{T}(t)$ for $\xi = 1$, $\upsilon = 2$

(e) RDF for $\xi = 2$, $\upsilon = 3$

(f) $F_\mathcal{T}(t)$ for $\xi = 2$, $\upsilon = 3$

Figure 6.3.: Estimated RDF $K_\Delta(\delta)$ and $F_\mathcal{T}(t)$ for revenue growth rates and revenue-operating cost ratios $(\xi, \upsilon) = \{(0.5, 1.5), (1, 2), (2, 3)\}$, and cluster sizes $\varpi \in \{0, 4, 8\}$ from simulation data. The values of the RDFs are depicted in thousands of normalized cost units (kncu).

In Fig. 6.3, we show the resulting $K_\Delta(\delta)$ and $F_\mathcal{T}(t)$ functions for a selection of parameters. We observe noticeable differences among the RDFs depending on the parameters ϖ, ξ, and υ. As the number of UE clusters ϖ increases, the mean readiness cost decreases for almost all state delays, which is consistent with the fact that a higher UE concentration leads to higher achieved spectral efficiencies as observed in Sec. 4.7. Conversely, there is an inverse relationship between ϖ and the degradation rate. When there are no UE clusters ($\varpi = 0$), the degradation of the readiness cost is much less noticeable than when $\varpi = 4$ or $\varpi = 8$. This makes intuitive sense, since old states are less likely to operate correctly when the UE distribution changes in a clustered scenario.

The distribution of the duration of the demands $F_{\mathcal{T}}(t)$, that is, the time between changes in the optimal state, seems remarkably consistent for all values of ϖ, ξ, and υ. The explanation for this is twofold. On the one hand, we observe that the FSSP is rather sensitive to small changes in the distribution of UEs, leading to frequent changes in the optimal solutions. The average time between these changes, in the order of one minute, is small when compared to the evolution of the UE clusters, whose shortest raising or falling times are 25 minutes. Since short-term variations are similar for all scenarios, $F_{\mathcal{T}}(t)$ is little affected by the value of ϖ. On the other hand, since the performance-to-revenue function is a linear function and the values of (ξ, υ) are within a limited range, their specific values do not substantially change the shape of these short-term variations either, leading to a negligible influence of these parameters.

6.7.3. Selection of monitoring period Z_{mon}

Once we have $K_{\Delta}(\delta)$ and $F_{\mathcal{T}}(t)$, we can use them to select an adequate value of Z_{mon}. As we discuss in Sec. 6.4.3, the time between a demand change, i. e., a change in the optimal state vector, and the next opportunity for a state change is modeled by means of the random variable $\mathcal{Z}$, which follows a uniform distribution for periodic monitoring. That is, $\mathcal{Z} \sim U(0, Z_{\text{mon}})$. From this and (15), we can formulate the mean readiness cost as a function of Z_{mon} as:

$$\overline{K}(Z_{\text{mon}}) = \alpha(Z_{\text{mon}})K_{\Delta}(0) + (1 - \alpha(Z_{\text{mon}}))\beta(Z_{\text{mon}})\widehat{K}_{\beta(Z_{\text{mon}})}(1), \tag{71}$$

where

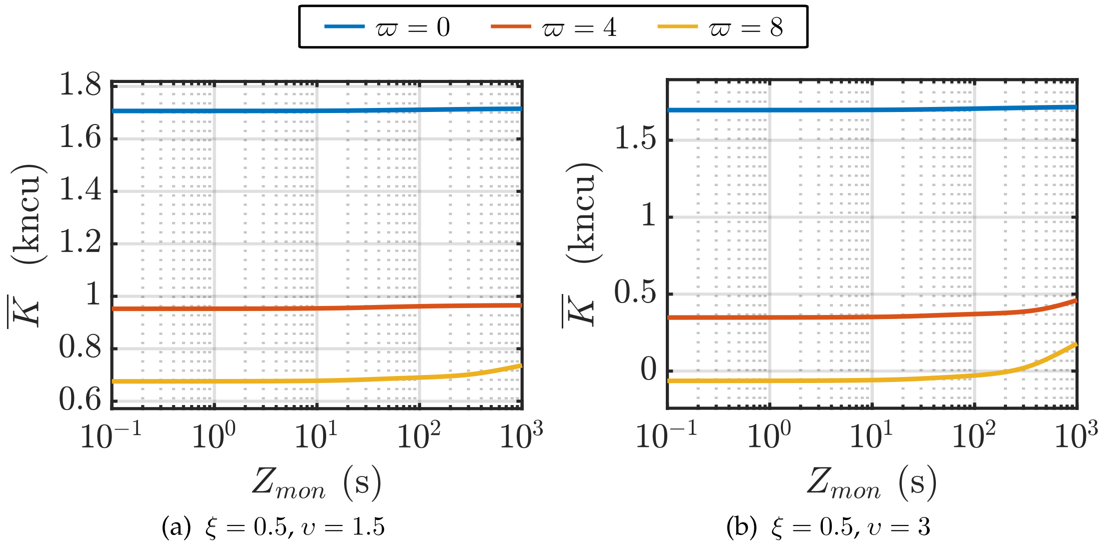
$$\alpha(Z_{\text{mon}}) = \frac{1}{T}\left[\int_{0}^{Z_{\text{mon}}} \frac{t\,(1 - F_{\mathcal{T}}(t))}{Z_{\text{mon}}} dt + \int_{Z_{\text{mon}}}^{\infty} (1 - F_{\mathcal{T}}(t))\,dt\right], \tag{72}$$

and

$$\beta(Z_{\text{mon}}) = 1 - F_{\mathcal{T}}(Z_{\text{mon}}) + \int_{0}^{Z_{\text{mon}}} \frac{t f_{\mathcal{T}}(t)}{Z_{\text{mon}}} dt. \tag{73}$$

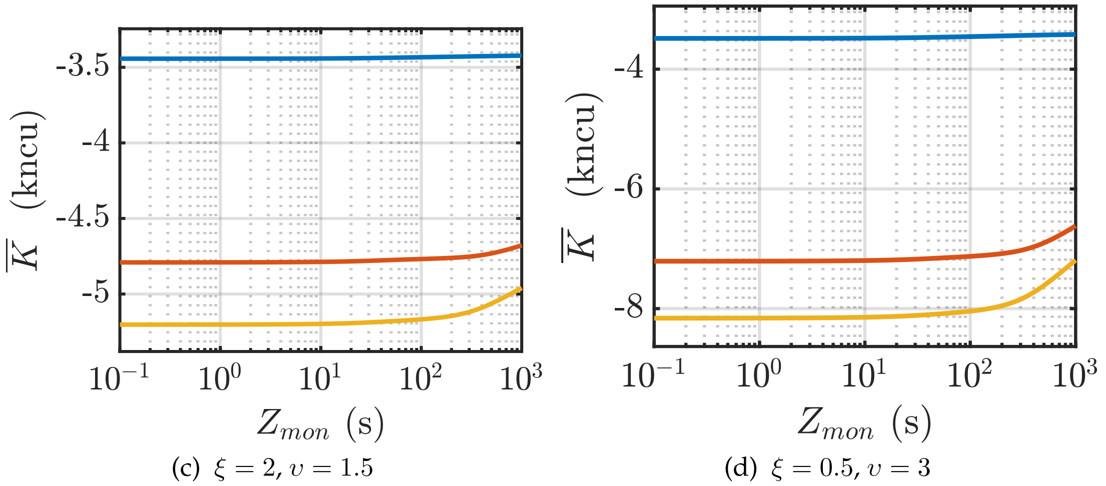
Note that we have set $\varphi = 1$ in (71) since this is the value that minimizes the readiness cost.

In Fig. 6.4 we show the evolution of $\overline{K}(Z_{\text{mon}})$ as Z_{mon} ranges from 0.1 s to 1000 s for multiple values of ϖ, ξ, and υ. We observe that the mean readiness cost stays relatively constant for all cases until $Z_{\text{mon}} \approx 100$ s, where the long monitoring period leads to a noticeable cost increase, specially for high values of ϖ, ξ, and υ. Thus, we conclude that monitoring periods in the range $0 \leq Z_{\text{mon}} \lesssim 100$ s do not to affect substantially the mean readiness cost that can be achieved in the simulated network. In our case, we henceforth choose $Z_{\text{mon}} = 30$ s as the monitoring interval for all simulations.



(a) $\xi = 0.5, \upsilon = 1.5$

(b) $\xi = 0.5, \upsilon = 3$



(c) $\xi = 2, \upsilon = 1.5$

(d) $\xi = 0.5, \upsilon = 3$

Figure 6.4.: Evolution of the mean readiness cost achievable by the network as a function of the monitoring interval Z_{mon}.

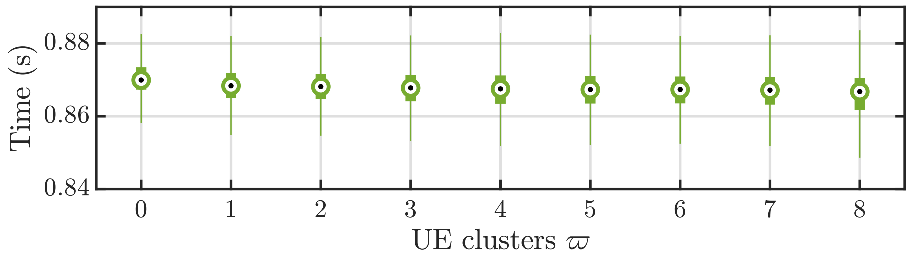


Figure 6.5.: Distribution of solving times of the FSSP as the number of UE clusters ϖ varies.

Parameter	Value	Units	Source
$K_{\text{inst,CU}}$	1	ncu	[GS+18b, Table I]
$K_{\text{inst,DU}}$	0.5	ncu	[GS+18b, Table I]
$K_{\text{comp,CU}}(a)$	$[1\ 1.8\ 3.4\ 5]$	RC · s/Gb/s	[GS+18b, Table I]
$K_{\text{comp,DU}}(a)$	$[4\ 3.2\ 1.6\ 0]$	RC · s/Gb/s	[GS+18b, Table I]
γ_{CU}	0.017	ncu/RC	[GS+18b, Table I]
γ_{DU}	1	ncu/RC	[GS+18b, Table I]
ω_{DU}	0	ncu/Gb/s	[GS+18b, Table I]
Z_{mon}	30	s	Sec. 6.7.3
Z_{migr}	20	ms	Sec. 5.5

Table 6.2.: Summary of cost and simulation parameters.

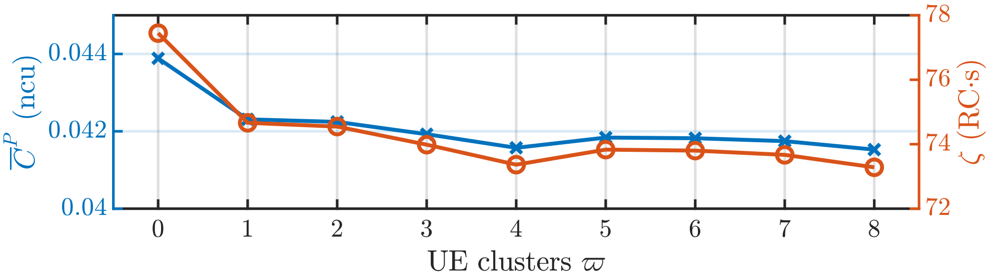
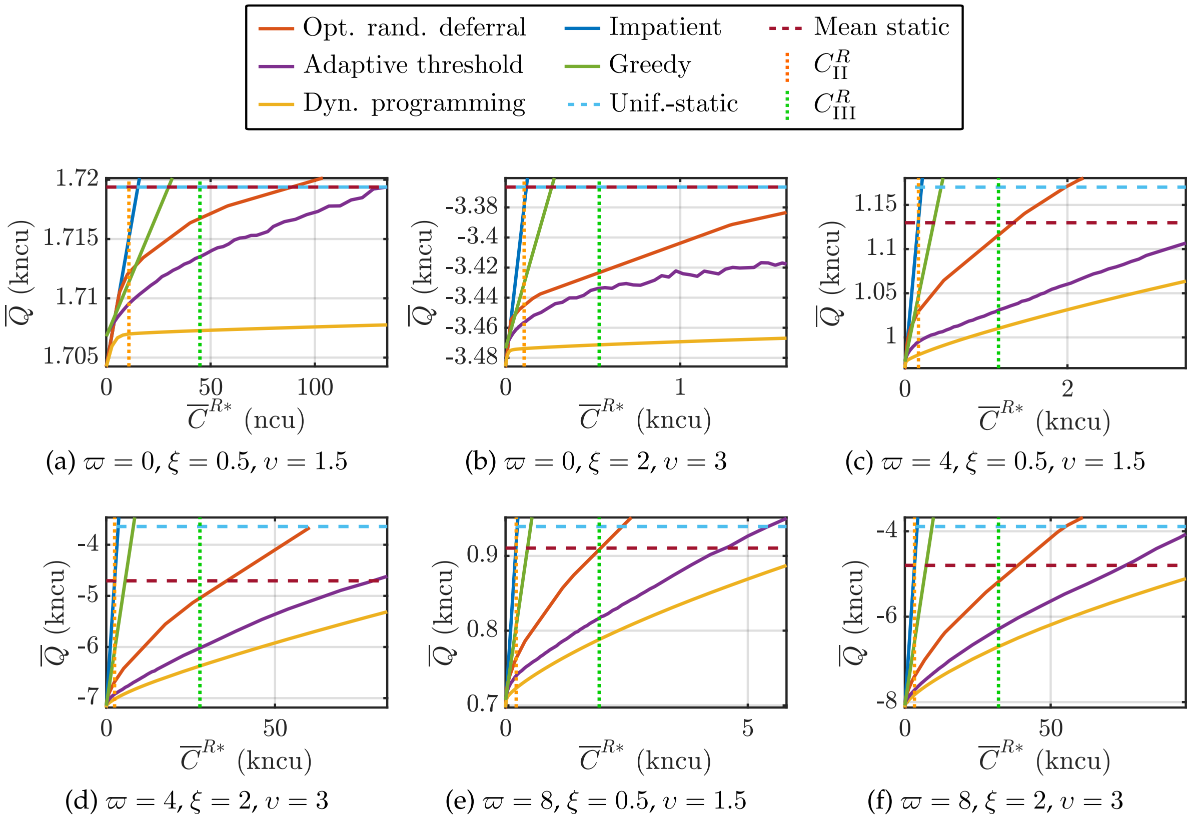


Figure 6.6.: Average computational effort of solving the FSSP and its associated proaction cost.

6.7.4. Proaction cost $\overline{C}^P$ estimation

We can compute the proaction cost from (22) as discussed in Sec. 6.4.3. Equation (22) depends on Z_{mon}, γ_{CU} whose values can be found in Table 6.2, and the mean computational effort ζ, which we need to measure from our simulations. In Fig. 6.5 we show the distribution of the solving times of (P20), which is directly related to the mean computational effort. Namely, in order to convert solving times into ζ, we just need to multiply the mean solving time by the number of used cores (44 cores) and scale by the consumed power, since our Intel Xeon E5 require 1.7 W per core whereas the reference Intel i7-4770 cores in [GS+18b] require 3.5 W per core [Int].

The resulting computational effort ζ and its associated proaction cost $\overline{C}^P$ are shown in Fig. 6.6. If we compare its value with the ranges that we deal with for the readiness cost (see for example Fig. 6.4), we conclude that the proaction cost has a negligible impact on the overall cost for the considered scenarios.



(a) $\varpi = 0, \xi = 0.5, \upsilon = 1.5$
(b) $\varpi = 0, \xi = 2, \upsilon = 3$
(c) $\varpi = 4, \xi = 0.5, \upsilon = 1.5$
(d) $\varpi = 4, \xi = 2, \upsilon = 3$
(e) $\varpi = 8, \xi = 0.5, \upsilon = 1.5$
(f) $\varpi = 8, \xi = 2, \upsilon = 3$

Figure 6.7.: Comparison of the mean total cost $\overline{Q}$ of static and dynamic strategies as the average action cost $\overline{C}^{R*}$ grows for $\varpi \in \{0, 4, 8\}$ UE clusters and readiness cost parameters $(\xi, \upsilon) \in \{(1, 1), (3, 3)\}$.

6.7.5. Comparison of adaptation strategies

In Fig. 6.7 we show the mean total cost achieved by all adaptation strategies considered in Sec. 6.5. In order to observe the influence of the reaction cost, we vary the mean reaction cost $\overline{C}^{R*}$ of the optimal state sequence, which represents the mean cost of potentially changing the state after each monitoring interval. We explore values of $\overline{C}^{R*}$ in the following range:

$$\overline{C}_{\mathrm{I}}^{R*} \leq \overline{C}^{R*} \leq 2 \cdot 10^{6} \overline{C}_{\mathrm{I}}^{R*}, \tag{74}$$

where $\overline{C}_{\mathrm{I}}^{R*}$ is a lower bound of $\overline{C}^{R*}$ defined in (29).

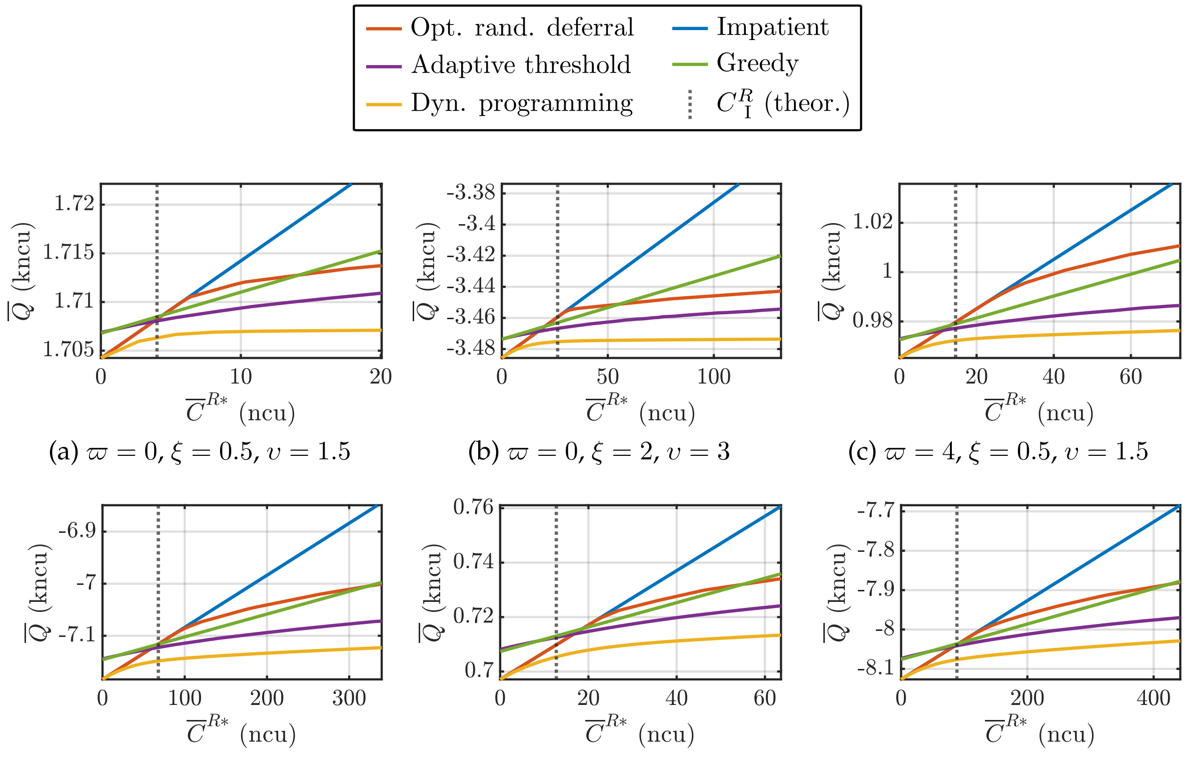
At first glance, we observe that the relative performance of all adaptation strategies is rather consistent for all scenarios. That is, the dynamic programming strategy always achieves the lowest total cost, followed by the adaptive threshold and the optimal random deferral strategies. Moreover, the total cost achieved by these three strategies is a concave function of $\overline{C}^{R*}$ in the explored range, which is clearly a desirable feature, and the adaptive threshold strategy is noticeably closer to the lower bound provided by dynamic programming than any other strategy. The cost of the greedy strategy seems to grow linearly with $\overline{C}^{R*}$ and is almost always worse than that of the adaptive threshold and optimal random deferral strategy, with few exceptions. As we expected, the cost of the impatient strategy also grows linearly with $\overline{C}^{R*}$ and features the steepest slope out of all strategies.

If we look into the values of $\overline{C}^{R*}$ at which the optimal random deferral and adaptive threshold strategies meet the mean-static strategy, we can draw interesting conclusions regarding the impact of the number of UE clusters ϖ and the shape of the performance-to-revenue. For instance, in Fig. 6.7a, which represents an scenario with no UE clusters and a rather flat performance-to-revenue function ($\upsilon = 1.5$), the mean reaction cost at which the random deferral and the adaptive threshold strategies stop being less costly than the mean static approach are $\overline{C}^{R*} \approx 91$ ncu and $\overline{C}^{R*} \approx 130$ ncu, respectively. These are 5% to 8% of the readiness cost of using the mean-static strategy ($\overline{Q}^{\text{mnst}} \approx 1720$ ncu), which implies that the network is only able to benefit from a dynamic adaptation if the cost of changing the functional split is $5-8\%$ of the cost of running the network during a single monitoring interval. Nevertheless, when $\varpi = 8$ UE clusters with the same performance-to-revenue function (Fig. 6.7e), the points where the random deferral and adaptive threshold strategies meet the mean-static readiness cost are $\overline{C}^{R*} \approx 2.1$ kncu and $\overline{C}^{R*} \approx 4.6$ kncu, respectively, which are 2.6 and 5.1 larger than the total cost of running the mean-static strategy ($\overline{Q}^{\text{mnst}} \approx 890$ ncu). For $\varpi = 8$ and a steep performance-to-revenue function ($\upsilon = 3$) as shown in Fig. 6.7f, the maximum mean reaction costs that we can tolerate for the random deferral and adaptive threshold strategies are $\overline{C}^{R*} \approx 39$ kncu and $\overline{C}^{R*} \approx 77$ kncu, respectively, which is almost 20 times larger compared to the previous case. We thus conclude that only in scenarios where the UEs are uniformly distributed and the performance-to-revenue function is almost flat, static strategies may be less costly than dynamic strategies, whereas in the remaining cases the dynamic strategies clearly outperform the static ones.

In Fig. 6.8 we show the same results as in Fig. 6.7 for a lower range of $\overline{C}^{R*}$, in order to better appreciate the behavior of the dynamic strategies when the mean reaction cost is low. We observe that the impatient, and random deferral strategies behave in the same way when $\overline{C}^{R*}$ is very small, such as $\overline{C}^{R*} \lesssim 20$ ncu in all but the case where $(\varpi, \xi, \upsilon) = (0, 0.5, 1.5)$. This implies that the impatient strategy is preferable over the greedy or adaptive threshold strategy for those cases. Conversely, it becomes clear that the greedy strategy is never less costly than all other implementable strategies, which allows us to discard it as a desirable option for any $\overline{C}^{R*}$.

6.7.6. Reference action-cost frontiers

Finally, we test the accuracy of the action-cost frontiers that are presented in Sec. 6.6. In Fig. 6.9 we show the theoretical values of $\overline{C}_{\text{II}}^{R*}$, $\overline{C}_{\text{III}}^{R*}$, and $\overline{C}_{\text{I}}^{R*}$ (calculated from the



(a) $\varpi = 0, \xi = 0.5, \upsilon = 1.5$

(b) $\varpi = 0, \xi = 2, \upsilon = 3$

(c) $\varpi = 4, \xi = 0.5, \upsilon = 1.5$

(d) $\varpi = 4, \xi = 2, \upsilon = 3$

(e) $\varpi = 8, \xi = 0.5, \upsilon = 1.5$

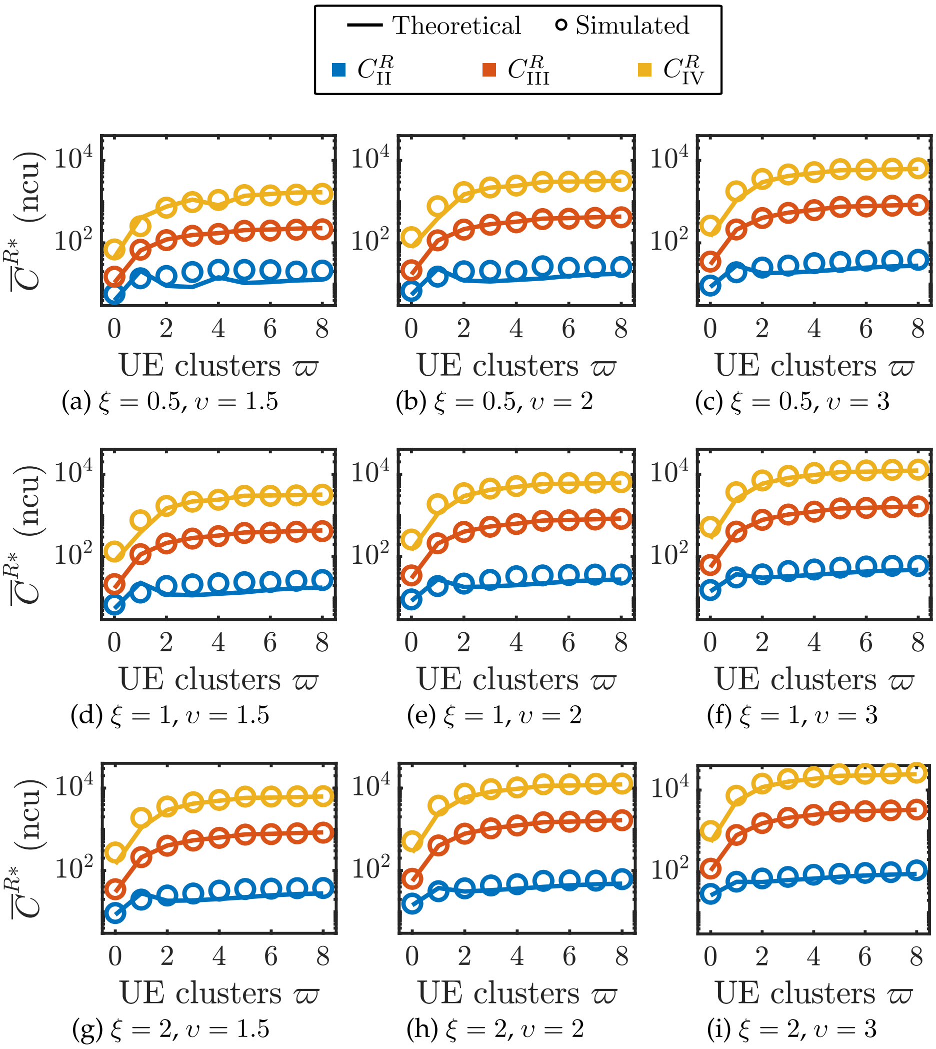
(f) $\varpi = 0, \xi = 2, \upsilon = 3$

Figure 6.8.: Comparison of the mean total cost $\overline{Q}$ of dynamic strategies as the average action cost $\overline{C}^{R*}$ grows for $\varpi \in \{0, 4, 8\}$ UE clusters and readiness cost parameters $(\xi, \upsilon) \in \{(1, 1),\ (3, 3)\}$.

estimated $F_{\mathcal{T}}(t)$, RDF $K_\Delta(\delta)$, and $\overline{Q}^{\mathrm{mnst}}$), alongside the corresponding values obtained by individually simulating the random deferral strategy for the $\overline{C}^{R*}$ range shown in (74), $\varpi = \{0, ..., 8\}$ UE clusters, $\xi = \{0.5, 1, 2\}$, and $\upsilon = \{1.5, 2, 3\}$. We observe a close correspondence between the theoretical and simulated values, thus validating the accuracy of the proposed model and the derivation of the RDF.

The value of $\overline{C}^{R*}_{\mathrm{II}}$ is also depicted in Fig. 6.8, indicating the predicted points where the random deferral strategy start producing less total cost than the impatient strategy. We can see that this frontier is a good predictor of the point beyond which the impatient strategy is clearly not optimal for all simulation instances. Thus, its value can be used to decide whether always adapting is an adequate option or a more advanced strategy, such as the adaptive threshold strategy, should be used.

Frontiers $\overline{C}^{R*}_{\mathrm{III}}$ and $\overline{C}^{R*}_{\mathrm{IV}}$ are also shown in Fig. 6.7, marking the predicted values of $\overline{C}^{R*}$ where the impatient strategy meets the mean-static strategy, and providing a lower bound to the intersection between the random deferral strategy and the mean-static strategy, respectively. We can see that these predictions are indeed fairly accurate, specially for $\varpi > 0$. $\overline{C}^{R*}_{\mathrm{III}}$ can be used to estimate the maximum mean action cost beyond which an advanced adaptation strategy is mandatory. Nonetheless, $\overline{C}^{R*}_{\mathrm{IV}}$ is,



(a) $\xi = 0.5, \upsilon = 1.5$

(b) $\xi = 0.5, \upsilon = 2$

(c) $\xi = 0.5, \upsilon = 3$

(d) $\xi = 1, \upsilon = 1.5$

(e) $\xi = 1, \upsilon = 2$

(f) $\xi = 1, \upsilon = 3$

(g) $\xi = 2, \upsilon = 1.5$

(h) $\xi = 2, \upsilon = 2$

(i) $\xi = 2, \upsilon = 3$

Figure 6.9.: Comparison between theoretical and simulated values of the reference action-cost frontiers $\overline{C}^{R*}_{\text{II}}$, $\overline{C}^{R*}_{\text{III}}$, and $\overline{C}^{R*}_{\text{IV}}$ for $\varpi = \{0, ..., 8\}$, $\xi = \{0.5, 1, 2\}$, and $\upsilon = \{1.5, 2, 3\}$.

as expected, a rather conservative frontier for dynamic strategies. This is due to two main reasons. On the one hand, $\overline{C}^{R*}_{\text{IV}}$ is a lower bound to the actual intersection between the optimal random deferral strategy and the mean-static strategy. On the other hand, advanced adaptation strategies such as the adaptive threshold strategy are able to provide a total cost lower than that of the mean-static strategies even when the optimal random deferral strategy yields a higher cost, as we can observe in Fig. 6.7. As a result, if $\overline{C}^{R*} \leq \overline{C}^{R*}_{\text{IV}}$ we can be confident that using the adaptive threshold strategy results in lower total cost than impatient or static strategies, whereas if $\overline{C}^{R*} > \overline{C}^{R*}_{\text{IV}}$ the adaptive threshold strategy may still be a valid option, although further analysis is required.

6.8. Summary

In previous chapters, we motivate the implementation of a 5G RAN whose functional split can be dynamically adapted arguing that it may increase the revenue of the operator. We present a comprehensive cost model to capture all the adaptation dynamics, an approach to select the instantaneously optimal functional split, and a proof-of-concept implementation to show that a dynamic functional split is technically feasible. In this chapter, we finally cope with the issue of estimating the revenue resulting from a dynamic functional split adaptation.

We propose a simple strategy to monitor changes in the network demand and, after introducing the corresponding notation, we provide a detailed description of all cost components that need to be considered in a dynamically adapting network. We show how the readiness degradation function can be obtained from a limited number of measurements or simulations in order to provide a simplified estimation of the readiness cost. This reduces number of simulations needed to characterize a dynamic 5G RAN and enables using the cost-of-flexibility model to predict its behavior in other conditions. We model the proaction and reaction costs using our proof-of-concept implementation as a reference, although a wide range of possible values is included so as to cover unforeseen cost components.

We present, describe, and compare seven adaptation strategies that a network operator may consider for running a dynamic 5G RAN. Two of these are static strategies, which represent networks which do not change the functional split dynamically but still intend to maximize revenue. The impatient strategy reflects a naive approach in which the network always attempts to operate in the optimal state, regardless of the cost of doing so. The dynamic programming strategy serves as a lower bound for all other strategies, since it relies on unrealistic future knowledge to provide the optimal sequence of state changes. The optimal random deferral strategy bases on the cost-of-flexibility model to select the optimal frequency of random state adaptation. The greedy strategy provides a small enhancement over the impatient strategy, since the state is not always changed, but only when immediate reward is detected. Finally, the adaptive threshold strategy improves upon the greedy strategy to provide a simple, yet promising approach to decide when the functional split should be adapted.

Taking these strategies as a reference, we derive three action cost frontiers. The first two, the impatient and dynamic frontiers, predict the points at which the impatient strategy stops being optimal and better than static strategies (owing to its high action cost), respectively. The third, static frontier, estimates the mean value of the action cost beyond which static strategies may be desirable over dynamic strategies.

Based on the previous analysis, we finally simulate a dynamic 5G radio access network in order to test the accuracy of our models and the viability of a dynamic functional split adaptation. We observe that we can afford a monitoring period of up to 100 s without noticeably degrading network performance, thus allowing more than enough time for solving the FSSP dynamically. We conclude that the proaction cost is negligible when compared to the other cost components, owing to the short convergence time of the algorithms solving the FSSP and the low cost of computational resources at the CU. We evaluate our proposed adaptation strategies and realize that dynamic strategies clearly outperform static strategies in all but extreme cases. When the action cost is low, that is, below frontier $\overline{C}_{\text{II}}^{R*}$, the impatient strategy is enough to achieve between 1% to 200% higher revenue (or lower cost) than static strategies. For higher reaction costs, namely $\overline{C}_{\text{II}}^{R*} \leq \overline{C}^{R*} \leq \overline{C}_{\text{IV}}^{R*}$, the adaptive threshold approach still provides near-optimal performance in all considered scenarios, specially in those cases where the UEs are not uniformly distributed over the covered area. Indeed, there is a large range of the action cost values that ensure that dynamic operation yields higher revenue than static operation. Finally, we successfully validate the accuracy of our estimated action cost frontiers by comparing it with the results of dedicated simulations.

7. Conclusion and Outlook

Every new generation of mobile networks features a common improvement over its predecessor: increasing user data rates. In 5G, multiple technology enhancements have been and are being developed to achieve this goal, such as the introduction of new modulation and coding schemes, the allocation of additional spectrum, and cell densification. Although cell densification is often considered among the most promising options to provide better service to the users, it also entails increasing inter-cell interference. This may hinder the achievement of the ambitious data rates that 5G networks promise to their customers.

In order to counter the effects of inter-cell interference, mobile networks may use interference-mitigation techniques, such as coordinated scheduling or joint transmission and reception. These techniques, however, often require high-throughput and low-latency communication among base stations. Centralized RAN architectures may enable this, but in reality they are difficult to implement, owing to the high requirements they that pose on the fronthaul networks connecting centralized and remote units. As a result, partial function centralization is the most promising option for implementing a 5G RAN.

Designing and deploying a partially centralized architecture implies deciding which RAN functions should be centralized and which functions may remain distributed. The optimality of this decision, however, depends on the instantaneous location of the UEs, their activity, and the channel quality. Since these aspects are constantly changing in actual networks, implementing a static functional split may result into highly suboptimal performance. Conversely, if the RAN is able to adapt its functional split to the instantaneous situation, it can not only improve user data rates, but also reduce operating cost.

In this thesis, we investigate the motivation, feasibility, cost, and performance of implementing a functional split that can be dynamically adapted to the instantaneous RAN conditions. We model the implications that a functional split has over the required fronthaul capacity, interference-mitigation capabilities, operating cost, and total revenue of a 5G RAN. We formulate the selection of the best functional split as an optimization problem and propose efficient algorithms to tackle it. In addition, we present of a proof-of-concept implementation in order to demonstrate the feasibility of dynamic adaptation. Finally, we derive multiple adaptation strategies so as to show that dynamic operation is indeed effective and desirable in a wide range of realistic conditions.

7.1. Summary

This thesis presents four main contributions to the problem of optimally selecting and adapting the functional split in 5G radio access networks. The first contribution, discussed in Chapter 3, comprises the definition of a cost model of flexible communication networks. The second contribution, explained in Chapter 4, deals with the formulation and optimal resolution of the functional split selection problem. The third contribution, presented in Chapter 5, demonstrates the feasibility of a dynamically adapted 5G RAN by means of a practical implementation. The fourth and final contribution, described in Chapter 6, tackles the selection of the most adequate functional splits over time. In this section, we briefly discuss the details and main conclusions of each contribution.

Derivation of a cost model for flexible networks. Network sofwarization technologies, such as network function virtualization and software-defined networking, offer enhanced flexibility and scalability to all types of communication networks, including radio access networks. They allow the opportunity for networks to dynamically adapt its operational state to changes in the demand, thus promising superior performance. As a result, this feature can be exploited to improve the profitability of the network. However, finding the appropriate state that satisfies a demand, realizing a state change, and spending transitional time in obsolete states may lead to increased operating costs. In this thesis, we present a novel cost model for flexible networks that takes into account all these cost components and combines them into a single cost metric. We use a probability-theory framework to derive the relationships among three main components: readiness, proaction, and reaction cost. In addition to providing generic examples of possible applications of this model, we apply it to the problem of dynamically selecting the optimal functional split in a 5G radio access network. Namely, we use it to calculate an adequate sampling rate for monitoring the evolution of the demand and to estimate the behavior of our proposed adaptation strategies.

Proposal and evaluation of optimization approaches to select the best functional split. The increased cell density required by 5G RAN deployments unavoidably leads to aggravated inter-cell interference. A fully centralized RAN architecture, in which all the processing of the base station is performed at a single data center, may be able to mitigate this additional interference, but the current limitations of fronthaul networks render it infeasible. A partially centralized architecture, in which only a subset of the processing functions is centralized, is thus the only viable option. Nonetheless, selecting the most adequate division between centralized and distributed functions is not a trivial task. In this thesis, we analyze three different approaches to select the optimal functional split that maximizes proportionally-fair user data rates, minimizes operating cost, and minimizes readiness cost (which combines operating cost and performance-related revenue). After deriving and assessing multiple problem formulations, we present a readiness-cost-minimizing approach that yields near optimal results with a very low convergence time.

Implementation of a proof-of-concept adaptive 5G RAN. By comparing the performance of optimally selected functional splits with that of static deployments, we are able to show that a dynamically adaptive functional split is beneficial for the profitability of a 5G RAN. Nevertheless, performing live changes in the location of the processing functions is unprecedented for mobile networks. In order to demonstrate that this feature is indeed feasible, in this thesis we show a detailed description of a proof-of-concept 5G RAN implementation that is able to switch between two functional split options during runtime. We build upon existing software that implements the 4G/5G protocol stack on conventional equipment and realize the required modifications to allow dynamic operation. We describe a migration strategy that allows changing the functional split of a base station with almost no disturbance to its normal operation. Indeed, packet losses can be fully prevented if a minimal end-to-end additional delay (in the order of 10 ms) is permitted. Conversely, migrations can be instantaneous if packet losses can be afforded. This implementation not only shows that dynamically adapting the functional split is indeed feasible, but it can be used to provide actual operational data for selecting an adequate adaptation strategy.

Proposal and evaluation of dynamic adaptation strategies. From the previous contributions, we have an efficient approach to select the instantaneously-optimal functional split and we are confident that the functional split can be adapted in a timely and affordable manner. However, it is still unclear when the network operator should trigger an adaptation of the functional split. On the one hand, the network demand changes continuously, but this does not guarantee that these changes are substantial enough to motivate an adaptation. On the other hand, adaptations of the functional split may still be costly, owing to additional resource consumption and potential penalizations related to packet losses or increased end-user delay. In this thesis, we apply the cost-of-flexibility model and dynamic optimization techniques to derive an efficient adaptation strategy. After comparing several options, we observe that there are indeed dynamic adaptation strategies that lead to remarkable readiness cost reductions with respect to static approaches. We thus conclude that a dynamic functional split adaptation is a feasible and very promising option to increase the efficiency and profitability of 5G radio access networks.

7.2. Future work

In this section, we present some promising research directions for future continuation of this work.

Extension of the cost-of-flexibility model. The current cost-of-flexibility model, as presented in Chapter 3, only considers *action-interrupting* networks. These networks interrupt their ongoing action phases if there is a new demand change, so that when a new state is implemented it is guaranteed to be optimal. We argue that this is probably the best strategy that a communication network facing varying demands should follow, since it ensures dealing with the most updated information about the demands. Nevertheless, in some cases it may be possible to prefer an *action-persistent* network, for which action phases are not interrupted. For example, if the demand changes frequently and operating in obsolete states is not particularly harmful, an action-persistent network may perform better than an action-interrupting network. However, this behavioral difference fundamentally changes the probabilistic analysis on which the cost-of-flexibility model bases. Therefore, extending the model to include action-persistent networks may be an interesting research direction.

Realization of a more complete dynamic 5G proof-of-concept. The proof-of-concept of a 5G RAN implementing a dynamically-adaptive functional split that we present in Chapter 5 has several limitations. On the one hand, it only features two functional split options, which hinders the experimentation about the features of other centralization levels. On the other hand, our proof-of-concept consists in two base stations providing service to up to three UEs. This results in a very simple fronthaul network, whose management can be tackled by uncomplicated algorithms. Therefore, a more realistic deployment featuring multiple base stations and centralization levels may be used to assess the effectiveness of our proposed adaptation strategies.

Use of reinforcement learning to improve adaptation strategies. In Chapter 6, we present a collection of adaptation strategies, ranging from static and naive strategies to more complicated approaches. Although none of these adaptation strategies requires an accurate estimation of the future evolution of the RAN, they all rely on information about simple network statistics, such as the average UE distribution or the expected degradation of the readiness cost. Whereas some strategies require this information as an external input, the adaptive threshold strategy is able to autonomously update these statistics from its own experience. This is similar to what reinforcement learning is able to do, and indeed there are many examples of using reinforcement learning to improve performance of communication networks. Hence, future work could explore the application of reinforcement learning as an alternative adaptation strategy, and compare its advantages and disadvantages with those of our current strategies.

A. Proofs and derivations

A.1. Proof of Property 3.4.9

Proof. If we differentiate (20), we obtain:

$$\frac{d\overline{K}(\varphi)}{d\varphi} = \alpha K_\Delta(0) + \beta(1-2\alpha\varphi)\sum_{\delta=1}^{\infty} K_\Delta(\delta)(1-\beta\varphi)^{\delta-1} \\ - \beta^2\varphi(1-\alpha\varphi)\sum_{\delta=1}^{\infty} K_\Delta(\delta)(\delta-1)(1-\beta\varphi)^{\delta-2} \quad (1)$$

Now we define $\mathfrak{D}_\Delta(\delta) \triangleq K_\Delta(\delta) - K_\Delta(0)$. Since $K_\Delta(\delta)$ is an increasing function, $\mathfrak{D}_\Delta(\delta)$ is positive and also increasing. After replacing $\mathfrak{D}_\Delta(\delta)$ into (1):

$$\frac{d\overline{K}(\varphi)}{d\varphi} = \alpha K_\Delta(0) + \beta(1-2\alpha\varphi)\left[\sum_{\delta=1}^{\infty} \mathfrak{D}_\Delta(\delta)(1-\beta\varphi)^{\delta-1} + \frac{K_\Delta(0)}{\beta\varphi}\right] \\ - \beta^2\varphi(1-\alpha\varphi)\left[\sum_{\delta=1}^{\infty} \mathfrak{D}_\Delta(\delta+1)\delta(1-\beta\varphi)^{\delta-1} + \frac{K_\Delta(0)}{(\beta\varphi)^2}\right], \quad (2)$$

This comes as a consequence of the following identities, whose proof is straightforward:

$$\sum_{\delta=1}^{\infty} K_\Delta(0)(1-\beta\varphi)^{\delta-1} = \frac{K_\Delta(0)}{\beta\varphi}, \quad (3)$$

$$\sum_{\delta=1}^{\infty} K_\Delta(0)x(1-\beta\varphi)^{\delta-1} = \frac{K_\Delta(0)}{(\beta\varphi)^2}. \quad (4)$$

Based on the following trivial identity:

$$\alpha K_\Delta(0) + \beta(1-2\alpha\varphi)\frac{K_\Delta(0)}{\beta\varphi} - \beta^2\varphi(1-\alpha\varphi)\frac{K_\Delta(0)}{(\beta\varphi)^2} = 0, \quad (5)$$

we can simplify (2) into:

$$\begin{aligned}\frac{d\overline{K}(\varphi)}{d\varphi} &= \beta(1-2\alpha\varphi)\left[\sum_{\delta=1}^{\infty}\mathfrak{D}_{\Delta}(\delta)(1-\beta\varphi)^{\delta-1}\right] \\ &\qquad - \beta^2\varphi(1-\alpha\varphi)\left[\sum_{\delta=1}^{\infty}\mathfrak{D}_{\Delta}(\delta+1)\delta(1-\beta\varphi)^{\delta-1}\right] \qquad (6) \\ &= \beta\sum_{\delta=1}^{\infty}\mathfrak{D}_{\Delta}(\delta)\left[(1-2\alpha\varphi)-\frac{\beta\varphi(1-\alpha\varphi)}{1-\beta\varphi}(\delta-1)\right](1-\beta\varphi)^{\delta-1} \qquad (7)\end{aligned}$$

Each addend in the summation of (7) has three factors: $\mathfrak{D}_{\Delta}(\delta)$, which is a positive and increasing function of δ, $(1-\beta\varphi)^{\delta-1}$, which is a positive and decreasing function of δ, and

$$\mathfrak{V}(\delta,\alpha,\beta,\varphi) \triangleq (1-2\alpha\varphi)-\frac{\beta\varphi(1-\alpha\varphi)}{1-\beta\varphi}(\delta-1), \qquad (8)$$

which is a negative function, if $1-2\alpha\varphi<0$, or, otherwise, positive from $1\leq\delta\leq\tilde{\delta}^{+}$, and then negative, for some value of $\tilde{\delta}^{+}$. In the former case, it is clear that $\frac{d\overline{K}(\varphi)}{d\varphi}\leq 0$. In the latter case, we divide the summation into two parts, one positive and one negative:

$$\begin{aligned}\frac{d\overline{K}(\varphi)}{d\varphi} &= \beta\sum_{\delta=1}^{\tilde{\delta}^{+}}\mathfrak{D}_{\Delta}(\delta)\mathfrak{V}(\delta,\alpha,\beta,\varphi)(1-\beta\varphi)^{\delta-1} \\ &\qquad + \beta\sum_{\delta=\tilde{\delta}^{+}+1}^{\infty}\mathfrak{D}_{\Delta}(\delta)\mathfrak{V}(\delta,\alpha,\beta,\varphi)(1-\beta\varphi)^{\delta-1} \qquad (9)\end{aligned}$$

In order for $\frac{d\overline{K}(\varphi)}{d\varphi}$ to be positive, the former (positive) summation in (9) has to be larger in absolute value than the latter (negative) one. For fixed α, β, φ, the largest value that $\mathfrak{D}_{\Delta}(\delta)$ could take in the range $0\leq\delta\leq\tilde{\delta}^{+}$ is $\mathfrak{D}_{\Delta}(\tilde{\delta}^{+})$, since $\mathfrak{D}_{\Delta}(\delta)$ is an increasing function. Similarly, a lower bound to the minimum value that $\mathfrak{D}_{\Delta}(\delta)$ can take when $\delta>\tilde{\delta}^{+}$ is also $\mathfrak{D}_{\Delta}(\tilde{\delta}^{+})$. As a result:

$$\begin{aligned}\frac{d\overline{K}(\varphi)}{d\varphi} &\leq \beta\mathfrak{D}_{\Delta}(\tilde{\delta}^{+})\sum_{\delta=1}^{\infty}\mathfrak{V}(\delta,\alpha,\beta,\varphi)(1-\beta\varphi)^{\delta-1} \qquad (10) \\ &= \beta\mathfrak{D}_{\Delta}(\tilde{\delta}^{+})\left[\frac{1-2\alpha\varphi}{\beta\varphi}-\frac{\beta\varphi(1-\alpha\varphi)}{(\beta\varphi)^2}\right] \qquad (11) \\ &= -\alpha\mathfrak{D}_{\Delta}(\tilde{\delta}^{+}) \qquad (12)\end{aligned}$$

Since $\mathfrak{D}_{\Delta}(\tilde{\delta}^{+})\geq 0$, it follows that $\frac{d\overline{K}(\varphi)}{d\varphi}\leq 0$. □

A.2. Proof of Property 3.4.10

Proof. If we derive (7), we obtain:

$$\frac{d^2\overline{K}(\varphi)}{d\varphi^2} = \beta\sum_{\delta=1}^{\infty}\mathfrak{D}_\Delta(\delta)\left[-\frac{2\alpha}{1-\beta\varphi} - \frac{2\beta(1-2\alpha\varphi)}{(1-\beta\varphi)^2}(\delta-1) \right. \\ \left. + \frac{\beta^2(1-\alpha\varphi)\varphi}{(1-\beta\varphi)^3}(\delta-1)(\delta-2)\right](1-\beta\varphi)^\delta \quad (13)$$

There are three factors in each addend of the summation in (13). $\mathfrak{D}_\Delta(\delta)$ and $(1-\beta\varphi)^\delta$ are always positive, whereas

$$\mathfrak{W}(\delta,\alpha,\beta,\varphi) \triangleq -\frac{2\alpha}{1-\beta\varphi} - \frac{2\beta(1-2\alpha\varphi)}{(1-\beta\varphi)^2}(\delta-1) + \frac{\beta^2(1-\alpha\varphi)\varphi}{(1-\beta\varphi)^3}(\delta-1)(\delta-2) \quad (14)$$

may be positive or negative. Namely, the first addend in (14) is negative, the third addend is positive, and the second addend may be positive or negative depending on the value of δ. Nonetheless, the value of the third addend grows with δ^2, whereas the second and first addends are linear and constant terms, respectively. Thus, it is clear that there exists a $\tilde{\delta}^+$ such that $\mathfrak{W}(\delta,\alpha,\beta,\varphi) > 0\ \forall\delta > \tilde{\delta}^+$, although $\mathfrak{W}(0,\alpha,\beta,\varphi) < 0$. This allows us to split (13) into the sum of a negative-valued and a positive-valued summations:

$$\frac{d^2\overline{K}(\varphi)}{d\varphi^2} = \beta\sum_{\delta=1}^{\tilde{\delta}^+}\mathfrak{D}_\Delta(\delta)\mathfrak{W}(\delta,\alpha,\beta,\varphi)(1-\beta\varphi)^\delta \\ + \beta\sum_{\delta=\tilde{\delta}^+ +1}^{\infty}\mathfrak{D}_\Delta(\delta)\mathfrak{W}(\delta,\alpha,\beta,\varphi)(1-\beta\varphi)^\delta \quad (15)$$

In order for $\frac{d^2\overline{K}(\varphi)}{d\varphi^2}$ to be negative, the former (negative) summation in (15) has to be larger in absolute value than the latter (positive) one. For fixed α, β, φ, the largest value that $\mathfrak{D}_\Delta(\delta)$ could take in the range $0 \le \delta \le \tilde{\delta}^+$ is $\mathfrak{D}_\Delta(\tilde{\delta}^+)$, since $\mathfrak{D}_\Delta(\delta)$ is an increasing function. Similarly, a lower bound to the minimum value that $\mathfrak{D}_\Delta(\delta)$ can take when $\delta > \tilde{\delta}^+$ is also $\mathfrak{D}_\Delta(\tilde{\delta}^+)$. As a result:

$$\frac{d^2\overline{K}(\varphi)}{d\varphi^2} \ge \beta\mathfrak{D}_\Delta(\tilde{\delta}^+)\sum_{\delta=1}^{\infty}\mathfrak{W}(\delta,\alpha,\beta,\varphi)(1-\beta\varphi)^\delta \quad (16)$$

$$= \beta\mathfrak{D}_\Delta(\tilde{\delta}^+)\left[-\frac{2\alpha}{\beta\varphi} - \frac{2\beta(1-2\alpha\varphi)}{(\beta\varphi)^2} + \frac{2\beta^2\varphi(1-\alpha\varphi)}{(\beta\varphi)^3}\right] = 0, \quad (17)$$

which proves that $\overline{K}(\varphi)$ is a convex function of φ. □

A.3. Estimation of $F_{\mathcal{T}}(t)$ from sampled durations

The distribution of the duration of demands in the network is modeled by $F_{\mathcal{T}}(t)$. We assume that this distribution is unknown, but we sample the demand every Z_{mon} seconds and denote the duration of the sampled demands by the random variable $\widehat{\mathcal{T}}$, whose units are sampling intervals of length Z_{mon}. For a certain value $\mathcal{T} = t$, the corresponding value of $\widehat{\mathcal{T}}$ can be either $\left\lceil \frac{t}{Z_{\text{mon}}} \right\rceil$ or $\left\lfloor \frac{t}{Z_{\text{mon}}} \right\rfloor$, depending on whether the first sampling instant lies on the first $t - \lfloor t \rfloor$ seconds or not, respectively. As a result, we can define the conditional probability mass function of $\widehat{\mathcal{T}}$ when $\mathcal{T} = t$ as:

$$f_{\widehat{\mathcal{T}}|\mathcal{T}}(\hat{t} \mid \mathcal{T} = t) = \begin{cases} t - \lfloor t \rfloor & \text{if } \hat{t} = \left\lceil \frac{t}{Z_{\text{mon}}} \right\rceil, \\ 1 - t + \lfloor t \rfloor & \text{if } \hat{t} = \left\lfloor \frac{t}{Z_{\text{mon}}} \right\rfloor. \end{cases} \tag{18}$$

Applying the law of total probability, the probability mass function of $\widehat{\mathcal{T}}$ can be formulated as:

$$f_{\widehat{\mathcal{T}}}(\hat{t}) = \int\limits_{-\infty}^{\infty} f_{\widehat{\mathcal{T}}|\mathcal{T}}(\hat{t} \mid \mathcal{T} = t) f_{\mathcal{T}}(t) dt. \tag{19}$$

After combining (18) and (19) and some straightforward algebra, we obtain the following expression for $f_{\widehat{\mathcal{T}}}(\hat{t})$ as a function of $F_{\mathcal{T}}(t)$:

$$f_{\widehat{\mathcal{T}}}(\hat{t}) = \frac{\int_{Z_{\text{mon}}\hat{t}}^{Z_{\text{mon}}(\hat{t}+1)} F_{\mathcal{T}}(t)dt - \int_{Z_{\text{mon}}(\hat{t}-1)}^{Z_{\text{mon}}\hat{t}} F_{\mathcal{T}}(t)dt}{Z_{\text{mon}} - \int_0^{Z_{\text{mon}}} F_{\mathcal{T}}(t)dt}. \tag{20}$$

Now, for some arbitrary $\hat{t}^{\max}$, we define the following vector:

$$\mathfrak{F} = \left[\int_0^{Z_{\text{mon}}} F_{\mathcal{T}}(t)dt \quad \cdots \quad \int_0^{Z_{\text{mon}}\hat{t}^{\max}} F_{\mathcal{T}}(t)dt \right], \tag{21}$$

the following coefficient matrix:

$$\mathfrak{H} = \begin{bmatrix} -2+f_{\widehat{\mathcal{T}}}(1) & 1+f_{\widehat{\mathcal{T}}}(2) & f_{\widehat{\mathcal{T}}}(3) & \cdots & f_{\widehat{\mathcal{T}}}(\hat{t}^{\max}) \\ 1 & -2 & 1 & 0 & \cdots \\ 0 & 1 & -2 & 1 & \cdots \\ 0 & 0 & 1 & -2 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{22}$$

and the following coefficient vector:

$$\mathfrak{f} = [f_{\widehat{\mathcal{T}}}(1) \quad \cdots \quad f_{\widehat{\mathcal{T}}}(\hat{t}^{\max})]. \tag{23}$$

We can combine $\mathfrak{F}$, $\mathfrak{H}$, and $\mathfrak{f}$ in the following way according to (20):

$$\mathfrak{F}^{\mathsf{T}} \cdot \mathfrak{H} = Z_{\text{mon}} \mathfrak{f}^{\mathsf{T}} \tag{24}$$

It is clear that if we had $\mathfrak{F}$, we could directly calculate $F_{\mathcal{T}}(t)$ by taking derivatives. From (24), we can find an estimate $\mathfrak{F}$ by either computing:

$$\mathfrak{F} = Z_{\text{mon}} \mathfrak{f}^{\mathsf{T}} \mathfrak{H}^{+}, \tag{25}$$

where $\mathfrak{H}^{+}$ denotes the Moore–Penrose inverse of $\mathfrak{H}$, or by minimizing $\|\mathfrak{F}^{\mathsf{T}} \cdot \mathfrak{H} - Z_{\text{mon}} \mathfrak{f}^{\mathsf{T}}\|^2$ with additional constraints if more information is known about $F_{\widehat{\mathcal{T}}}(t)$.

A.4. Estimation of RDF from time-based cost degradation

We can establish the following relationship between the time-based degradation function $K_{\Delta}^{\tau}(\tau)$ and the RDF $K_{\Delta}(\delta)$, as mentioned in Sec. 6.4.3:

$$K_{\Delta}^{\tau}(\tau) = \sum_{\delta=0}^{\infty} K_{\Delta}(\delta) \Pr\{\Delta = \delta | \tau\}. \tag{26}$$

The probability $\Pr\{\Delta = \delta | \tau\}$ of the state delay being δ after τ time units can be written as:

$$\Pr\{\Delta = \delta \mid \tau\} = \Pr\left\{ \sum_{i=1}^{\delta+1} \mathcal{T}_i \geq \tau \;\middle|\; \sum_{i=1}^{\delta} \mathcal{T}_i < \tau \right\}, \tag{27}$$

since a state delay $\Delta = \delta$ at time τ since the optimal state was implemented implies that there has been exactly δ changes in the demand since then. Let $\mathcal{T}_{\Sigma}^{\delta} \triangleq \sum_{i=1}^{\delta} \mathcal{T}_i$, we can calculate (27) as:

$$\Pr\{\Delta = \delta \mid \tau\} = \int_0^{\tau} \left(1 - F_{\mathcal{T}}(\tau - t)\right) f_{\mathcal{T}_{\Sigma}^{\delta}}(t) dt, \tag{28}$$

where the density function $f_{\mathcal{T}_{\Sigma}^{\delta}}(t)$ can be obtained by convoluting $f_{\mathcal{T}}(t)$ δ times with itself. Now, for a long, arbitrary $\tau^{\max}$, we define the following vectors:

$$\mathfrak{K}_{\tau} = [K_{\Delta}^{\tau}(1) \ \cdots \ K_{\Delta}^{\tau}(\tau^{\max})] \tag{29}$$

$$\mathfrak{K} = [K_{\Delta}(1) \ \cdots \ K_{\Delta}(\tau^{\max})] \tag{30}$$

$$\mathfrak{D} = [\Pr\{\Delta = \delta \mid 1\} \ \cdots \ \Pr\{\Delta = \delta \mid \tau^{\max}\}] \tag{31}$$

Combining (29)–(31) with (27), we derive the following relationship:

$$\mathfrak{K}_{\tau} = \mathfrak{K} \cdot \mathfrak{D}^{\mathsf{T}} \tag{32}$$

We can solve for an estimation of $\mathfrak{K}$ by either computing:

$$\mathfrak{K} = \mathfrak{K}_\tau \cdot \mathfrak{D}^+ \tag{33}$$

where $\mathfrak{D}^+$ denotes the Moore–Penrose inverse of $\mathfrak{D}$, or by minimizing $\|\mathfrak{K}_\tau - \mathfrak{K} \cdot \mathfrak{D}^\intercal\|^2$ with additional constraints if more information is known about $K_\Delta(\tau)$.

Bibliography

Publications by the author

Journal publications

[Bab+20] P. Babarczi, M. Klügel, A. Martínez Alba, M. He, J. Zerwas, P. Kalmbach, A. Blenk, and W. Kellerer. “A mathematical framework for measuring network flexibility”. In: *Computer Communications* 164 (2020), pp. 13–24.

[Die+21] L. Diez, A. Martínez Alba, W. Kellerer, and R. Agüero. “Flexible Functional Split and Fronthaul delay: A queuing-based model”. In: *IEEE Access* (2021). [Under review].

[He+19] M. He, A. Martínez Alba, A. Basta, A. Blenk, and W. Kellerer. “Flexibility in softwarized networks: Classifications and research challenges”. In: *IEEE Communications Surveys & Tutorials* 21.3 (2019), pp. 2600–2636.

[Kel+18] W. Kellerer, A. Basta, P. Babarczi, A. Blenk, M. He, M. Klugel, and A. Martínez Alba. “How to Measure Network Flexibility? A Proposal for Evaluating Softwarized Networks”. In: *IEEE Communications Magazine* (2018).

[MAJK21] A. Martínez Alba, S. Janardhanan, and W. Kellerer. “Enabling dynamically centralized RAN architectures in 5G and beyond”. In: *IEEE Transactions on Network and Service Management* (2021). Early access.

[MAK21] A. Martínez Alba and W. Kellerer. “Dynamic Functional Split Adaptation in Next-Generation Radio Access Networks”. In: *IEEE Transactions on Network and Service Management* (2021). [Under review].

Conference publications

[MA+18] A. Martínez Alba, A. Basta, J. H. Gómez Velásquez, and W. Kellerer. “A realistic coordinated scheduling scheme for the next-generation RAN”. In: *IEEE Global Communications Conference (GLOBECOM)*. 2018.

[MA+21] A. Martínez Alba, P. Babarczi, A. Blenk, M. He, P. Krämer, J. Zerwas, and W. Kellerer. “Modeling the Cost of Flexibility in Communication Networks”. In: *IEEE Conference on Computer Communications (INFOCOM)*. IEEE. 2021.

[MAGVK19a] A. Martínez Alba, J. H. Gómez Velásquez, and W. Kellerer. “An adaptive functional split in 5G networks”. In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2019, pp. 410–416.

[MAGVK19b] A. Martínez Alba, J. H. Gómez Velásquez, and W. Kellerer. “Traffic characterization of the MAC-PHY split in 5G networks”. In: *IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2019, pp. 1–6.

[MAJK20] A. Martínez Alba, S. Janardhanan, and W. Kellerer. “Dynamics of the flexible functional split selection in 5G networks”. In: *IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2020, pp. 1–6.

[MAK19a] A. Martínez Alba and W. Kellerer. “A Dynamic Functional Split in 5G Radio Access Networks”. In: *IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2019, pp. 1–6.

[MAK19b] A. Martínez Alba and W. Kellerer. “Large-and Small-Scale Modeling of User Traffic in 5G Networks”. In: *International Conference on Network and Service Management (CNSM)*. IEEE. 2019, pp. 1–5.

[MAPK21] A. Martínez Alba, S. Pundt, and W. Kellerer. “Comparison of performance- and cost-optimal functional splits in 5G and beyond”. In: *IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2021.

General publications

[3GP17] 3GPP. *Study on new radio access technology: Radio access architecture and interfaces*. Technical Report (TR) 38.801. Version 14.0.0. 3rd Generation Partnership Project (3GPP), Mar. 2017.

[3GP20a] 3GPP. *Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 general aspects and principles*. Technical Specification (TS) 36.420. Version 16.0.0. 3rd Generation Partnership Project (3GPP), July 2020.

[3GP20b] 3GPP. *Study on scenarios and requirements for next generation access technologies*. Technical Report (TR) 38.913. Version 16.0.0. 3rd Generation Partnership Project (3GPP), July 2020.

[3GP21a] 3GPP. *5G; NG-RAN; Architecture description*. Technical Specification (TS) 38.401. Version 16.6.0. 3rd Generation Partnership Project (3GPP), July 2021. URL: `http://www.3gpp.org/\-DynaReport/\-38401.htm`.

[3GP21b] 3GPP. *5G; NR; Medium Access Control (MAC) protocol specification*. Technical Specification (TS) 38.321. Version 16.5.0. 3rd Generation Partnership Project (3GPP), July 2021.

[3GP21c] 3GPP. *5G; NR; Multiplexing and channel coding*. Technical Specification (TS) 38.212. Version 16.6.0. 3rd Generation Partnership Project (3GPP), June 2021.

[3GP21d] 3GPP. *5G; NR; NR and NG-RAN Overall description; Stage-2*. Technical Specification (TS) 38.300. Version 16.6.0. 3rd Generation Partnership Project (3GPP), July 2021.

[3GP21e] 3GPP. *5G; NR; Physical channels and modulation*. Technical Specification (TS) 38.211. Version 16.6.0. 3rd Generation Partnership Project (3GPP), June 2021.

[3GP21f] 3GPP. *5G; NR; Radio Link Control (RLC) protocol specification*. Technical Specification (TS) 38.322. Version 16.2.0. 3rd Generation Partnership Project (3GPP), Jan. 2021.

[3GP21g] 3GPP. *5G; NR; User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone*. Technical Specification (TS) 38.101-1. Version 16.6.0. 3rd Generation Partnership Project (3GPP), June 2021.

[3GP21h] 3GPP. *5G; NR; User Equipment (UE) radio transmission and reception; Part 2: Range 2 Standalone*. Technical Specification (TS) 38.101-2. Version 16.6.0. 3rd Generation Partnership Project (3GPP), June 2021.

[3GP21i] 3GPP. *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2*. Technical Specification (TS) 36.300. Version 16.6.0. 3rd Generation Partnership Project (3GPP), July 2021.

[3GP21j] 3GPP. *NG-RAN; F1 Application Protocol (F1AP)*. Technical Specification (TS) 38.473. Version 16.6.0. 3rd Generation Partnership Project (3GPP), Aug. 2021.

[3GP21k] 3GPP. *NR; User Equipment (UE) radio access capabilities*. Technical Specification (TS) 38.306. Version 16.5.0. 3rd Generation Partnership Project (3GPP), July 2021.

[Afo+18] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck. "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions". In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 2429–2453.

[Alf+18] Y. Alfadhli, M. Xu, S. Liu, F. Lu, P. Peng, and G. Chang. "Real-Time Demonstration of Adaptive Functional Split in 5G Flexible Mobile Fronthaul Networks". In: *Optical Fiber Communications Conference and Exposition (OFC)*. 2018, pp. 1–3.

[Ali+17] E. Ali, M. Ismail, R. Nordin, and N. F. Abdulah. "Beamforming techniques for massive MIMO systems in 5G: overview, classification, and trends for future research". In: *Frontiers of Information Technology & Electronic Engineering* 18.6 (2017), pp. 753–772.

[AMO13] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Always learning. Pearson, 2013. ISBN: 9781292042701.

[Aro09] S. Arora. *Computational complexity: a modern approach*. Cambridge New York: Cambridge University Press, 2009. ISBN: 9780521424264.

[AS07] A. Aries and A. Shiryaev. *Optimal Stopping Rules*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg, 2007. ISBN: 9783540740100.

[Aye+18] T. M. Ayenew, D. Xenakis, N. Passas, and L. Merakos. "Dynamic programming based content placement strategy for 5G and beyond cellular networks". In: *IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE. 2018, pp. 1–6.

[Bae+19] J. H. Bae, A. Abotabl, H.-P. Lin, K.-B. Song, and J. Lee. "An overview of channel coding for 5G NR cellular communications". In: *APSIPA Transactions on Signal and Information Processing* 8 (2019), e17.

[Bar+15] J. Bartelt, P. Rost, D. Wubben, J. Lessmann, B. Melis, and G. Fettweis. "Fronthaul and backhaul requirements of flexibly centralized radio access networks". In: *IEEE Wireless Communications* 22.5 (2015), pp. 105–111.

[Bas+17] A. Basta, A. Blenk, K. Hoffmann, H. J. Morper, M. Hoffmann, and W. Kellerer. "Towards a cost optimal design for a 5G mobile core network based on SDN and NFV". In: *IEEE Transactions on Network and Service Management* 14.4 (2017), pp. 1061–1075.

[Bas+19] D. Basu, X. Wang, Y. Hong, H. Chen, and S. Bressan. "Learn-as-you-go with megh: Efficient live migration of virtual machines". In: *IEEE Transactions on Parallel and Distributed Systems* 30.8 (2019), pp. 1786–1801.

[BCJ19] S. Becker, P. Cheridito, and A. Jentzen. "Deep optimal stopping". In: *Journal of Machine Learning Research* 20 (2019), p. 74.

[Bel03] R. Bellman. *Dynamic Programming*. Dover Books on Computer Science Series. Dover Publications, 2003. ISBN: 9780486428093.

[Bel66] R. Bellman. "Dynamic programming". In: *Science*. Dover Books on Computer Science Series 153.3731 (1966), pp. 34–37.

[Ber+14] D. Bergman, A. A. Cire, A. Sabharwal, H. Samulowitz, V. Saraswat, and W.-J. van Hoeve. "Parallel combinatorial optimization with decision diagrams". In: *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*. Springer. 2014, pp. 351–367.

[Bet+17] M. Bettner, S. Haka, J. Williams, and J. Carcello. *Financial & Managerial Accounting*. McGraw-Hill Education, 2017. ISBN: 9781259692406.

[BGH92] D. P. Bertsekas, R. G. Gallager, and P. Humblet. *Data networks*. Vol. 2. Prentice-Hall International New Jersey, 1992.

[BGP16] J. S. Borrero, C. Gillen, and O. A. Prokopyev. "A simple technique to improve linearized reformulations of fractional (hyperbolic) 0–1 programming problems". In: *Operations Research Letters* 44.4 (2016), pp. 479–486.

[Bil95] P. Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics. Wiley, 1995. ISBN: 9780471007104.

[BK16] N. Bizanis and F. A. Kuipers. "SDN and virtualization solutions for the Internet of Things: A survey". In: *IEEE Access* 4 (2016), pp. 5591–5606.

[BKM20] C. Bouras, A. Kollia, and E. Maligianni. "Techno-economic Comparison of Cognitive Radio and Software Defined Network (SDN) Cost Models in 5G Networks". In: *Wireless Personal Communications* (2020).

[BNK16] C. H. Benet, K. A. Noghani, and A. J. Kassler. "Minimizing Live VM Migration Downtime Using OpenFlow Based Resiliency Mechanisms". In: *IEEE International Conference on Cloud Networking (Cloudnet)*. 2016, pp. 27–32.

[BNP16] C. Bouras, P. Ntarzanos, and A. Papazois. "Cost modeling for SDN/NFV based mobile 5G networks". In: *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE. 2016, pp. 56–61. ISBN: 9781467388184.

[Bou+09] G. Boudreau, J. Panicker, N. Guo, R. Chang, N. Wang, and S. Vrzic. "Interference coordination and cancellation for 4G networks". In: *IEEE Communications Magazine* 47.4 (2009), pp. 74–81.

[Bou+15] M. Bouet, J. Leguay, T. Combe, and V. Conan. "Cost-based placement of vDPI functions in NFV infrastructures". In: *International Journal of Network Management* (2015).

[Bou+18] C. Bouras, S. Kokkalis, A. Kollia, and A. Papazois. "Techno-economic comparison of MIMO and DAS cost models in 5G networks". In: *Wireless Networks* (2018).

[Cas18] B. Casemore. *How Network Disaggregation Facilitates Datacenter and IT Modernization*. White Paper. IDC, 2018.

[CC14] W. Cerroni and F. Callegati. "Live migration of virtual network functions in cloud-based edge networks". In: *IEEE International Conference on Communications (ICC)*. 2014, pp. 2963–2968.

[Cen00] N. Cencov. *Statistical Decision Rules and Optimal Inference*. Translations of mathematical monographs. American Mathematical Society, 2000. ISBN: 9780821813478.

[CGP11] C. Cruz, J. R. González, and D. A. Pelta. "Optimization in dynamic environments: a survey on problems, methods and measures". In: *Soft Computing* 15.7 (2011), pp. 1427–1448.

[Cha+17a] O. Chabbouh, S. B. Rejeb, N. Agoulmine, and Z. Choukair. "Cloud RAN Architecture Model Based upon Flexible RAN Functionalities Split for 5G Networks". In: *International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. 2017, pp. 184–188.

[Cha+17b] C. Chang, N. Nikaein, R. Knopp, T. Spyropoulos, and S. S. Kumar. "FlexCRAN: A flexible functional split framework over ethernet fronthaul in Cloud-RAN". In: *IEEE International Conference on Communications (ICC)*. 2017, pp. 1–7.

[Che+14a] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann. "Cloud RAN for mobile networks–A technology overview". In: *IEEE Communications surveys & tutorials* 17.1 (2014), pp. 405–426.

[Che+14b] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang. "A vision of IoT: Applications, challenges, and opportunities with China perspective". In: *IEEE Internet of Things journal* 1.4 (2014), pp. 349–359.

[Che+16a] A. Checko, A. P. Avramova, M. S. Berger, and H. L. Christiansen. "Evaluating C-RAN fronthaul functional splits in terms of network level energy and cost savings". In: *Journal of Communications and Networks* 18.2 (2016), pp. 162–172.

[Che+16b] S. Chen, S. Sun, Q. Gao, and X. Su. "Adaptive beamforming in TDD-based mobile communication systems: State of the art and 5G research directions". In: *IEEE Wireless Communications* 23.6 (2016), pp. 81–87.

[Cis17] Cisco. *Prepare to succeed with the Internet of Things*. White Paper. 2017.

[Cis19] Cisco public. *Cisco visual networking index: global mobile data traffic forecast update, 2017–2022*. White Paper. 2019.

[Cis20] Cisco public. *Cisco annual internet report (2018–2023)*. White Paper. 2020.

[Com+18] I.-S. Comşa, S. Zhang, M. E. Aydin, P. Kuonen, Y. Lu, R. Trestian, and G. Ghinea. "Towards 5G: A reinforcement learning-based scheduling solution for data traffic management". In: *IEEE Transactions on Network and Service Management* 15.4 (2018), pp. 1661–1675.

[Dav+13] A. Davydov, G. Morozov, I. Bolotin, and A. Papathanassiou. "Evaluation of joint transmission CoMP in C-RAN based LTE-A HetNets with large coordination areas". In: *IEEE Globecom Workshops (GC Wkshps)*. IEEE. 2013, pp. 801–806.

[DeG69] M. DeGroot. *Optimal statistical decisions*. New York: McGraw-Hill, 1969. ISBN: 0070162425.

[DGA19] L. Diez, V. Gonzalez, and R. Agüero. “Minimizing Delay in NFV 5G Networks by Means of Flexible Split Selection and Scheduling”. In: *IEEE Vehicular Technology Conference (VTC)*. IEEE. 2019, pp. 1–6.

[DHA20] L. Diez, C. Hervella, and R. Agüero. “Understanding the performance of flexible functional split in 5G vRAN Controllers: A Markov Chain-based model”. In: *IEEE Transactions on Network and Service Management* 18.1 (2020), pp. 456–468.

[Döt+13] U. Dötsch, M. Doll, H.-P. Mayer, F. Schaich, J. Segel, and P. Sehier. “Quantitative analysis of split base station processing and determination of advantageous architectures for LTE”. In: *Bell Labs Technical Journal* 18.1 (2013), pp. 105–128.

[Dow05] A. B. Downey. “Lognormal and Pareto distributions in the Internet”. In: *Computer Communications* 28.7 (2005), pp. 790–801.

[DPS18] E. Dahlman, S. Parkvall, and J. Skold. *5G NR: The Next Generation Wireless Access Technology*. Elsevier Science, 2018. ISBN: 9780128143247.

[Dur10] R. Durrett. *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010. ISBN: 9781139491136.

[EAN19] N. C. Ericsson AB Huawei Technologies Co. Ltd and Nokia. *Common Public Radio Interface: eCPRI Interface Specification*. Interface Specification. Version 2.0. May 2019.

[EIS75] S. Even, A. Itai, and A. Shamir. “On the complexity of time table and multi-commodity flow problems”. In: *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*. IEEE. 1975, pp. 184–193.

[Eng+98] S. Engstrom, T. Johansson, F. Kronestedt, M. Larsson, S. Lidbrink, and H. Olofsson. “Multiple reuse patterns for frequency planning in GSM networks”. In: *VTC’98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No. 98CH36151)*. Vol. 3. IEEE. 1998, pp. 2004–2008.

[EPP19] A. Eira, J. Pedro, and J. J. Pires. “Modeling Cost Versus Flexibility in Optical Transport Networks”. In: *Journal of Lightwave Technology* (2019).

[Eur17] European Union. *Flash Eurobarometer 456. SMEs, resource efficiency and green markets*. Report. 2017.

[Eur21] European Union. *Special Eurobarometer 513. Climate Change*. Report. 2021.

[FBB19] N. Farrugia, J. A. Briffa, and V. Buttigieg. “Solving the Multi-Commodity Flow Problem using a Multi-Objective Genetic Algorithm”. In: *IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2019, pp. 2816–2823.

[FLF19] C.-H. Fang, P.-R. Li, and K.-T. Feng. "Joint interference cancellation and resource allocation for full-duplex cloud radio access networks". In: *IEEE Transactions on Wireless Communications* 18.6 (2019), pp. 3019–3033.

[Fou+16] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis. "FlexRAN: A flexible and programmable platform for software-defined radio access networks". In: *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. 2016, pp. 427–441.

[FYK91] K. Furuta, M. Yamakita, and S. Kobayashi. "Swing up control of inverted pendulum". In: *IECON*. Vol. 91. 1991, pp. 2193–2198.

[GA18] K. Govindaraj and A. Artemenko. "Container live migration for latency critical industrial applications on edge computing". In: *IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vol. 1. IEEE. 2018, pp. 83–90.

[Gal91] M. Gallagher. "Proportionality, disproportionality and electoral system". In: *Electoral studies* 10.1 (1991), pp. 33–51.

[GAO17] R. A. Gustav A. Oertzen. *On the Technical Future of the Telecommunications Industry*. White Paper. Oliver Wyman, 2017.

[Gha+15] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, R. Ahmed, and R. Boutaba. "Elastic virtual network function placement". In: *IEEE 4th International Conference on Cloud Networking, CloudNet 2015*. 2015. ISBN: 9781467395014.

[Glo15] G. Glockner. *Parallel and distributed optimization with gurobi optimizer*. 2015.

[Glo75] F. Glover. "Improved linear integer programming formulations of nonlinear integer problems". In: *Management Science* 22.4 (1975), pp. 455–460.

[GM+16] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. 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*. ACM. 2016, pp. 25–32.

[GS+18a] A. Garcia-Saavedra, X. Costa-Perez, D. J. Leith, and G. Iosifidis. "FluidRAN: Optimized vRAN/MEC orchestration". In: *IEEE Conference on Computer Communications (INFOCOM)*. IEEE. 2018, pp. 2366–2374.

[GS+18b] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. J. Leith. "Joint optimization of edge computing architectures and radio access networks". In: *IEEE Journal on Selected Areas in Communications* 36.11 (2018), pp. 2433–2443.

[GW74] F. Glover and E. Woolsey. “Converting the 0–1 polynomial programming problem to a 0–1 linear program”. In: *Operations research* 22.1 (1974), pp. 180–182.

[Ham+13] A. S. Hamza, S. S. Khalifa, H. S. Hamza, and K. Elsayed. “A survey on inter-cell interference coordination techniques in OFDMA-based cellular networks”. In: *IEEE Communications Surveys & Tutorials* 15.4 (2013), pp. 1642–1670.

[Hor86] R. Horst. “A general class of branch-and-bound methods in global optimization with some new approaches for concave minimization”. In: *Journal of Optimization Theory and Applications* 51.2 (1986), pp. 271–291.

[HP17] A. Hajisami and D. Pompili. “Dynamic joint processing: Achieving high spectral efficiency in uplink 5G cellular networks”. In: *Computer Networks* 126 (2017), pp. 44–56.

[HR18a] D. Harutyunyan and R. Riggio. “Flex5G: Flexible Functional Split in 5G Networks”. In: *IEEE Transactions on Network and Service Management* 15.3 (2018), pp. 961–975.

[HR18b] D. Harutyunyan and R. Riggio. “Flex5G: Flexible functional split in 5G networks”. In: *IEEE Transactions on Network and Service Management* 15.3 (2018), pp. 961–975.

[Hue+08] R. Huelsermann, M. Gunkel, C. Meusburger, and D. A. Schupke. “Cost modeling and evaluation of capital expenditures in optical multilayer networks”. In: *Journal of Optical Networking* 7.9 (2008), pp. 814–833.

[HW10] H. Huang and L. Wang. “P&P: A combined push-pull model for resource monitoring in cloud computing environment”. In: *IEEE 3rd International Conference on Cloud Computing (CLOUD)*. 2010. ISBN: 9780769541303.

[Ips20] Ipsos. *5G Awareness and Needs*. European Study. 2020.

[IT12] ITU-T. *Framework of network virtualization for future networks*. Recommendation Y.3011. Telecommunication Standardization Sector of ITU, Jan. 2012.

[IT18] ITU-T. *Transport network support of IMT-2020/5G*. Technical Report GSTR-TN5G. Telecommunication Standardization Sector of ITU, Feb. 2018.

[Jun+10] V. Jungnickel, A. Forck, S. Jaeckel, F. Bauermeister, S. Schiffermueller, S. Schubert, S. Wahls, L. Thiele, T. Haustein, W. Kreher, et al. “Field trials using coordinated multi-point transmission in the downlink”. In: *IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops*. IEEE. 2010, pp. 440–445.

[Jun+14] V. Jungnickel, K. Manolakis, W. Zirwas, B. Panzner, V. Braun, M. Lossow, M. Sternad, R. Apelfröjd, and T. Svensson. “The role of small cells, coordinated multipoint, and massive MIMO in 5G”. In: *IEEE communications magazine* 52.5 (2014), pp. 44–51.

[KDV08] G. Kesidis, A. Das, and G. de Veciana. "On flat-rate and usage-based pricing for tiered commodity internet services". In: *Annual Conference on Information Sciences and Systems*. IEEE. 2008, pp. 304–308.

[Kel+19] W. Kellerer, P. Kalmbach, A. Blenk, A. Basta, M. Reisslein, and S. Schmid. "Adaptable and data-driven softwarized networks: Review, opportunities, and challenges". In: *Proceedings of the IEEE* 107.4 (2019), pp. 711–731.

[KH05] H. Kim and Y. Han. "A proportional fair scheduling for multicarrier transmission systems". In: *IEEE Communications letters* 9.3 (2005), pp. 210–212.

[Klü+19] M. Klügel, M. He, W. Kellerer, and P. Babarczi. "A Mathematical Measure for Flexibility in Communication Networks". In: *IFIP Networking Conference (IFIP Networking)*. IEEE. 2019, pp. 1–9.

[KMT98] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. "Rate control for communication networks: shadow prices, proportional fairness and stability". In: *Journal of the Operational Research society* 49.3 (1998), pp. 237–252.

[Kos+12] C. Kosta, B. Hunt, A. U. Quddus, and R. Tafazolli. "On interference avoidance through inter-cell interference coordination (ICIC) based on OFDMA mobile systems". In: *IEEE Communications Surveys & Tutorials* 15.3 (2012), pp. 973–995.

[KS17] G. P. Koudouridis and P. Soldati. "Spectrum and network density management in 5G ultra-dense networks". In: *IEEE Wireless Communications* 24.5 (2017), pp. 30–37.

[LCC18] L. M. Larsen, A. Checko, and H. L. Christiansen. "A survey of the functional splits proposed for 5G mobile crosshaul networks". In: *IEEE Communications Surveys & Tutorials* (2018).

[Li11] X. Li. *Radio Access Network Dimensioning for 3G UMTS*. Springer, 2011.

[Li94] H.-L. Li. "A global approach for general 0–1 fractional programming". In: *European Journal of Operational Research* 73.3 (1994), pp. 590–596.

[Lin+13] L. Lingitz, C. Morawetz, D. T. Gigloo, S. Minner, and W. Sihn. "Modelling of flexibility costs in a decision support system for midterm capacity planning". In: *Procedia CIRP*. 2013.

[Lin16] J. C. Lin. "Human exposure to RF, microwave, and millimeter-wave electromagnetic radiation [Health Effects]". In: *IEEE Microwave Magazine* 17.6 (2016), pp. 32–36.

[Lom54] K. S. Lomax. "Business failures: Another example of the analysis of failure data". In: *Journal of the American Statistical Association* 49.268 (1954), pp. 847–852.

[Mac+13] G. R. MacCartney, J. Zhang, S. Nie, and T. S. Rappaport. “Path loss models for 5G millimeter wave propagation channels in urban microcells”. In: *IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2013, pp. 3948–3953.

[Mae+14] A. Maeder, M. Lalam, A. De Domenico, E. Pateromichelakis, D. Wübben, J. Bartelt, R. Fritzsche, and P. Rost. “Towards a flexible functional split for cloud-RAN networks”. In: *European Conference on Networks and Communications (EuCNC)*. IEEE. 2014, pp. 1–5.

[Mae+16] A. Maeder, A. Ali, A. Bedekar, A. F. Cattoni, D. Chandramouli, S. Chandrashekar, L. Du, M. Hesse, C. Sartori, and S. Turtinen. “A scalable and flexible radio access network architecture for fifth generation mobile networks”. In: *IEEE Communications Magazine* 54.11 (2016), pp. 16–23.

[Mak+17] N. Makris, P. Basaras, T. Korakis, N. Nikaein, and L. Tassiulas. “Experimental evaluation of functional splits for 5G cloud-RANs”. In: *2017 IEEE International Conference on Communications (ICC)*. IEEE. 2017, pp. 1–6.

[Mar+18] A. Marotta, D. Cassioli, K. Kondepu, C. Antonelli, and L. Valcarenghi. “Efficient Management of Flexible Functional Split through Software Defined 5G Converged Access”. In: *IEEE International Conference on Communications (ICC)*. 2018, pp. 1–6.

[MMD91] R. Mazumdar, L. G. Mason, and C. Douligeris. “Fairness in network optimal flow control: Optimality of product forms”. In: *IEEE Transactions on Communications* 39.5 (1991), pp. 775–782.

[MT90] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 1990. ISBN: 9780471924203.

[MTK01] K. Man, K. Tang, and S. Kwong. *Genetic Algorithms: Concepts and Designs*. Advanced Textbooks in Control and Signal Processing. Springer London, 2001. ISBN: 9781852330729.

[Nar+18] G. Nardini, G. Stea, A. Virdis, A. Frangioni, L. Galli, D. Sabella, and G. M. Dell’Aera. “Practical feasibility, scalability and effectiveness of coordinated scheduling algorithms in cellular networks towards 5G”. In: *Journal of Network and Computer Applications* 106 (2018), pp. 1–16.

[NGM15] NGMN Alliance. “5G white paper”. In: *Next generation mobile networks, white paper* 1 (2015), pp. 1–125.

[Nik+14] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet. “OpenAirInterface: A flexible platform for 5G research”. In: *ACM SIGCOMM Computer Communication Review* 44.5 (2014), pp. 33–38.

[Nik15] N. Nikaein. “Processing radio access network functions in the cloud: Critical issues and modeling”. In: *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*. 2015, pp. 36–43.

[NMH14] G. Nigam, P. Minero, and M. Haenggi. “Coordinated multipoint joint transmission in heterogeneous networks”. In: *IEEE Transactions on Communications* 62.11 (2014), pp. 4134–4146.

[NTT20] NTT DOCOMO. *5G Evolution and 6G*. White Paper. Jan. 2020.

[NYB12] T. T. Nguyen, S. Yang, and J. Branke. “Evolutionary dynamic optimization: A survey of the state of the art”. In: *Swarm and Evolutionary Computation* 6 (2012), pp. 1–24.

[Ope14] Open Data Center Alliance. *Open Data Center Alliance: Software-Defined Networking Rev. 2.0*. Technical Report. 2014.

[Opp+97] A. Oppenheim, A. Willsky, S. Nawab, w. Hamid, and I. Young. *Signals & Systems*. Prentice-Hall signal processing series. Prentice Hall, 1997. ISBN: 9780138147570.

[PL15] R. K. Polaganga and Q. Liang. “Self-similarity and modeling of LTE/LTE-A data traffic”. In: *Measurement* 75 (2015), pp. 218–229.

[PM17] H. Paixão Martins. “Analysis of CoMP for the Management of Interference in LTE”. In: *Master Thesis* (2017).

[Pow07] W. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley Series in Probability and Statistics. Wiley, 2007. ISBN: 9780470182956.

[Rup+18] A. Ruprecht, D. Jones, D. Shiraev, G. Harmon, M. Spivak, M. Krebs, M. Baker-Harvey, and T. Sanderson. “Vm live migration at scale”. In: *ACM SIGPLAN Notices* 53.3 (2018), pp. 45–56.

[Sab+13] D. Sabella, P. Rost, Y. Sheng, E. Pateromichelakis, U. Salim, P. Guitton-Ouhamou, M. Di Girolamo, and G. Giuliani. “RAN as a service: Challenges of designing a flexible RAN architecture in a cloud-based heterogeneous mobile network”. In: *Future Network & Mobile Summit*. IEEE. 2013, pp. 1–8.

[SCF21] SCF. *5G nFAPI specifications*. Document 225.2.0. Small Cell Forum (SCF), May 2021.

[Sch+03] K. Schuster, F. Pukelsheim, M. Drton, and N. R. Draper. “Seat biases of apportionment methods for proportional representation”. In: *Electoral Studies* 22.4 (2003), pp. 651–676.

[Sey+16] T. Seyama, D. Jitsukawa, T. Kobayashi, T. Oyama, T. Dateki, H. Seki, M. Minowa, T. Okuyama, S. Suyama, and Y. Okumura. “Study of Coordinated Radio Resource Scheduling Algorithm for 5G Ultra High-Density Distributed Antenna Systems–Performance Evaluation of Joint Transmission Multi-User MIMO”. In: *IEICE Technical Report; IEICE Tech. Rep.* 115.472 (2016), pp. 181–186.

[Sha49] C. E. Shannon. “Communication in the presence of noise”. In: *Proceedings of the IRE* 37.1 (1949), pp. 10–21.

[SL05] G. Song and Y. Li. “Cross-layer optimization for OFDM wireless networks-part I: theoretical framework”. In: *IEEE transactions on wireless communications* 4.2 (2005), pp. 614–624.

[Sma17] Small Cell Forum. *FAPI and nFAPI specifications*. Document 082.09.05. Release 9.0. May 2017.

[Sor+17] B. Soret, A. De Domenico, S. Bazzi, N. H. Mahmood, and K. I. Pedersen. “Interference coordination for 5G new radio”. In: *IEEE Wireless Communications* 25.3 (2017), pp. 131–137.

[SRF15] V. Suryaprakash, P. Rost, and G. Fettweis. “Are heterogeneous cloud-based radio access networks cost effective?” In: *IEEE Journal on Selected Areas in Communications* 33.10 (2015), pp. 2239–2251. arXiv: `1503.03366`.

[SS10] P. K. Sharma and R. Singh. “Comparative analysis of propagation path loss models with field measured data”. In: *International Journal of Engineering Science and Technology* 2.6 (2010), pp. 2008–2013.

[SSG17] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis. “Optimal and scalable caching for 5G using reinforcement learning of space-time popularities”. In: *IEEE Journal of Selected Topics in Signal Processing* 12.1 (2017), pp. 180–190.

[Sta15] W. Stallings. *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. Always Learning. Networking. Pearson, 2015. ISBN: 9780134175393.

[Sut18] R. Sutton. *Reinforcement learning: an introduction*. Cambridge, Massachusetts London, England: The MIT Press, 2018. ISBN: 9780262039246.

[SY14] R. Srikant and L. Ying. *Communication Networks: An Optimization, Control and Stochastic Networks Perspective*. Cambridge University Press, 2014. ISBN: 9781107036055.

[SYCP18] V. Sciancalepore, F. Z. Yousaf, and X. Costa-Perez. “Z-TORCH: An Automated NFV Orchestration and Monitoring Solution”. In: *IEEE Transactions on Network and Service Management* (2018). arXiv: `1807.02307`.

[Tal+20] T. Taleb, R. L. Aguiar, I. Grida Ben Yahia, B. Chatras, G. Christensen, U. Chunduri, A. Clemm, X. Costa, L. Dong, J. Elmirghani, et al. *White paper on 6G networking*. White paper. 6G Research Visions no. 6, University of Oulu, 2020.

[TAS02] M. Tawarmalani, S. Ahmed, and N. V. Sahinidis. “Global optimization of 0–1 hyperbolic programs”. In: *Journal of Global Optimization* 24.4 (2002), pp. 385–416.

[TBK15] T. Taleb, M. Bagaa, and A. Ksentini. "User mobility-aware Virtual Network Function placement for Virtual 5G Network Infrastructure". In: *IEEE International Conference on Communications*. 2015. ISBN: 9781467364324.

[TGD19] H. D. Trinh, L. Giupponi, and P. Dini. "Urban anomaly detection by processing mobile traffic traces with LSTM neural networks". In: *IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2019, pp. 1–8.

[Tom66] J. Tomlin. "Minimum-cost multicommodity network flows". In: *Operations Research* 14.1 (1966), pp. 45–51.

[Tor+21] R. Torre, R.-S. Schmoll, F. Kemser, H. Salah, I. Tsokalo, and F. H. Fitzek. "Benchmarking Live Migration Performance Under Stressed Conditions". In: *IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2021, pp. 1–2.

[Uni21] I. I. T. Union). *Measuring Digital Development: ICT Price Trends 2020*. 2021.

[Upd21] I. Update. *Ericsson Mobility Report*. Tech. rep. Ericsson, June 2021.

[Val+16] L. Valcarenghi, K. Kondepu, F. Giannone, and P. Castoldi. "Requirements for 5G fronthaul". In: *International Conference on Transparent Optical Networks (ICTON)*. 2016, pp. 1–5.

[Wax88] B. M. Waxman. "Routing of multipoint connections". In: *IEEE journal on selected areas in communications* 6.9 (1988), pp. 1617–1622.

[Wu97] T.-H. Wu. "A note on a global approach for general 0–1 fractional programming". In: *European Journal of Operational Research* 101.1 (1997), pp. 220–223.

[Xio+19] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L.-C. Wang. "Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges". In: *IEEE Vehicular Technology Magazine* 14.2 (2019), pp. 44–52.

[Xu+16] F. Xu, Y. Li, H. Wang, P. Zhang, and D. Jin. "Understanding mobile traffic patterns of large scale cellular towers in urban environment". In: *IEEE/ACM transactions on networking* 25.2 (2016), pp. 1147–1161.

[You+16] F. Z. Yousaf, C. Goncalves, L. Moreira-Matias, and X. C. Perez. "RAVA - Resource aware VNF agnostic NFV orchestration method for virtualized networks". In: *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*. 2016. ISBN: 9781509032549.

[Zan97] J. Zander. "On the cost structure of future wideband wireless access". In: *IEEE 47th Vehicular Technology Conference. Technology in Motion*. Vol. 3. IEEE. 1997, pp. 1773–1776.

[Zha+17] Y. Zhang, J. Ding, M.-W. Kwan, J. Ni, E. K. Tsang, Y.-N. R. Li, and J. Li. "Measurement and Evaluations of Coherent Joint Transmission for 5G Networks". In: *IEEE 85th Vehicular Technology Conference (VTC Spring)*. IEEE. 2017, pp. 1–5.

[Zha+18] P. Zhao, H. Tian, S. Fan, and A. Paulraj. "Information prediction and dynamic programming-based RAN slicing for mobile edge computing". In: *IEEE Wireless Communications Letters* 7.4 (2018), pp. 614–617.

[ZWW14] L. Zhang, W. Wu, and D. Wang. "Time dependent pricing in wireless data networks: Flat-rate vs. usage-based schemes". In: *IEEE Conference on Computer Communications (INFOCOM)*. IEEE. 2014, pp. 700–708.

Cited websites

[ATT] ATT North America. *Next-Generation IP MPLS Backbone*. URL: `https://www.att.com/Common/merger/files/pdf/wired-network/Domestic\_OC-768\_Network.pdf` (visited on 01/14/2021).

[Ben] Ben Wojtowicz. *OpenLTE*. URL: `https://sourceforge.net/projects/openlte/` (visited on 09/06/2021).

[Goo] Google Developers. *Google Protocol Buffers*. URL: `https://developers.google.com/protocol-buffers` (visited on 09/06/2021).

[Gur] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. URL: `https://www.gurobi.com` (visited on 07/20/2021).

[Int] Intel Corporation. *Intel Processors for PC, Laptops, Servers, and AI*. URL: `https://intel.com/content/www/us/en/products/details/processors.html`.

[KVM] KVM. *Kernel Virtual Machine*. URL: `https://www.linux-kvm.org/` (visited on 09/06/2021).

[Opea] OpenAirInterface Software Alliance. *Open Air Interface RAN Feature Set*. URL: `https://gitlab.eurecom.fr/oai/openairinterface5g/blob/master/doc/feAture_sEt.md` (visited on 09/06/2021).

[Opeb] OpenAirInterface Software Alliance. *OpenAirInterface 5G Radio Access Network Project*. URL: `https://openairinterface.org/oai-5g-ran-project/` (visited on 09/06/2021).

[Opec] OpenAirInterface Software Alliance. *OpenAirInterface AMF Feature Set*. URL: `https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-amf/-/blob/master/docs/fEature_Set.md` (visited on 09/06/2021).

[Oped] OpenAirInterface Software Alliance. *OpenAirInterface Main Page*. URL: `https://openairinterface.org` (visited on 09/06/2021).

[Opee] OpenAirInterface Software Alliance. *OpenAirInterface SMF Feature Set*. URL: https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-smf/-/blob/master/docs/fEature_Set.md (visited on 09/06/2021).

[Opef] OpenAirInterface Software Alliance. *OpenAirInterface Software Alliance Members*. URL: https://openairinterface.org/osa-members/ (visited on 09/06/2021).

[Ran] Range Networks. *OpenBTS*. URL: http://openbts.org/ (visited on 09/06/2021).

[SRSa] SRS. *srsRAN - Your own mobile network*. URL: https://srsran.com/ (visited on 09/06/2021).

[SRSb] SRS. *srsRAN 21.04 Documentation*. URL: https://docs.srsran.com/ (visited on 09/06/2021).

[SRSc] SRS. *The srsLTE project is evolving*. URL: https://www.srslte.com/srslte-srsran (visited on 09/06/2021).

[Thea] The Internet Topology Zoo. URL: http://www.topology-zoo.org/ (visited on 09/06/2021).

[Theb] The Linux Foundation Projects. *Xen Project*. URL: https://xenproject.org/ (visited on 09/06/2021).

[Thec] The OpenStack Fundation. *Open Source Cloud Computing Infrastructure – OpenStack*. URL: https://www.openstack.org/ (visited on 09/06/2021).

Acronyms and abbreviations

3G	Third generation of mobile networks
3GPP	Third Generation Partnership Project
4G	Fourth generation of mobile networks
5G	Fifth generation of mobile networks
AES	Advanced Encryption Standard
AMF	Access and Mobility Management Function
ARQ	Automatic Repeat Request
AUSF	Authentication Server Function
BGP	Borrero-Gillen-Prokopyev
BBU	Baseband unit
BPSK	Binary Phase-Shift Keying
CAPEX	Capital Expenditure
CDF	Cumulative Distribution Function
CMS	Cloud Management System
CPU	Central Processing Unit
CQI	Channel Quality Indicator
C-RAN	Cloud-RAN
CU	Centralized Unit
DAS	Distributed Antenna System
DPI	Deep Packet Inspection
DRAN	Distributed RAN
DU	Distributed Unit
eCPRI	Enhanced Common Public Radio Interface
eMBB	Enhanced Mobile Broadband
eMBMS	Evolved Multimedia Broadcast Multicast Service
eNodeB	Evolved NodeB
EPC	Evolved Packet Core
FDD	Frequency Domain Duplex
FFR	Fractional Frequency Reuse
FFT	Fast Fourier Transform
FR1	Frequency Range 1
FR2	Frequency Range 2
FSSP	Functional Split Selection Problem
gNB	Next-generation NodeB
GPRS	General Packet Radio Service
GTP	GPRS Tunneling Protocol
HARQ	Hybrid ARQ

HSS	Home Subscriber Server
ICIC	Inter-cell Interference Coordination
IFFT	Inverse Fast Fourier Transform
IoT	Internet of Things
IP	Internet Propotcol
KVM	Kernel-based Virtual Machine
LCID	Logical Channel ID
LDPC	Low-density Parity-check Code
LP	Linear Problem
LTE	Long Term Evolution
LWT	Li-Wu-Tawarmalani
MAC	Medium Access Control
MILP	Mixed-Integer Linear Problem
MIMO	Multiple Input Multiple Output
MINLP	Mixed-Integer Non-Linear Problem
MME	Mobility Management Entity
mMTC	Massive Machine-Type Communication
NAS	Non-Access Stratum
nFAPI	Network Functional Application Platform Interface
NFV	Network Function Virtualization
NG-RAN	Next-Generation RAN
NR	New Radio
NRF	Network Repository Function
NV	Network Virtualization
OPEX	Operating Expenses
PC	Personal Computer
PDCP	Packet Data Convergence Protocol
PDN	Public Data Network
PDU	Packet Data Unit
P-GW	PDN-Gateway
PHY	Physical layer
PID	Process ID
QAM	Quadrature Amplitude Modulation
QIP	Quadratic Integer Problem
QoD	Quality of Decisions
QoS	Quality of Service
RAN	Radio Access Network
RDF	Readiness Degradation Function
RF	Radiofrequency
RLC	Radio Link Control
RNTI	Radio Network Temporary Identifier
RRC	Radio Resource Control
RRH	Remote Radio Head
RU	Remote Unit
SDAP	Service Data Adaptation Protocol

SDN	Software Defined Networking
SDR	Software Defined Radio
SFFR	Soft Fractional Frequency Reuse
S-GW	Serving Gateway
SINR	Signal to Interference and Noise Ratio
SMF	Session Management Function
TCO	Total Cost of Ownership
TDD	Time Domain Duplex
UDM	Unified Data Management Function
UDR	Unified Data Repository
UE	User Equipment
UPF	User Plane Function
URLLC	Ultra-Reliable Low-Latency Communication
VNF	Virtual Network Function
WSL	Webster/Sainte-Laguë

List of Figures

List of Tables