UNIVERSIDADE FEDERAL DE GOIÁS (UFG)

INSTITUTO DE INFORMÁTICA (INF)

PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DANIEL CAMPOS DA SILVA

Channel-Aware Inter-Slice Scheduling for SLA Assurance: Theoretical and Simulation-Based Approaches

GOIÂNIA
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

[X] Dissertação [ ] Tese [ ] Outro*:_____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

Exemplos: Estudo de caso ou Revisão sistemática ou outros formatos.

2. Nome completo do autor

Daniel Campos da Silva

3. Título do trabalho

Channel-Aware Inter-Slice Scheduling for SLA Assurance: Theoretical and Simulation-Based Approaches

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
a) consulta ao(à) autor(a) e ao(à) orientador(a);
b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Kleber Vieira Cardoso**, **Professor do Magistério Superior**, em 30/06/2025, às 17:56, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Daniel Campos Da Silva**, **Discente**, em 01/07/2025, às 00:15, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5470692** e o código CRC **2C95E11D**.

Referência: Processo nº 23070.024274/2025-22 SEI nº 5470692

DANIEL CAMPOS DA SILVA

Channel-Aware Inter-Slice Scheduling for SLA Assurance: Theoretical and Simulation-Based Approaches

Master's thesis in Scandinavian format submitted to the Programa de Pós-Graduação em Ciência da Computação (PPGCC) of the Instituto de Informática (INF) of Universidade Federal de Goiás (UFG), in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Area: Computer Science.

Mentor: Prof. Dr. Kleber Vieira Cardoso

GOIÂNIA
2025

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Silva, Daniel Campos da
Channel-Aware Inter-Slice Scheduling for SLA Assurance: Theoretical and Simulation-Based Approaches [manuscrito] / Daniel Campos da Silva. - 2025.
lxxxi, 81 f.: il.

Orientador: Prof. Dr. Kleber Vieira Cardoso.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2025.
Bibliografia. Apêndice.
Inclui algoritmos, lista de figuras, lista de tabelas.

1. Service level agreement. 2. Network slicing. 3. Resource block. 4. Radio resource scheduling. 5. Energy efficiency. I. Cardoso, Kleber Vieira, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº 17 da sessão de Defesa de Dissertação de **Daniel Campos da Silva**, que confere o título de Mestre/Doutor em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos vinte dias do mês de junho de dois mil e vinte e cinco, a partir das dez horas, via sistema de webconferência, realizou-se a sessão pública de Defesa de Dissertação intitulada **"Channel-Aware Inter-Slice Scheduling for SLA Assurance: Theoretical and Simulation-Based Approaches"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Kleber Vieira Cardoso (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Flávio Geraldo Coelho Rocha (EMC/UFG), membro titular externo; Professor Doutor Jacek Kibilda (Virginia Tech), membro titular externo. A realização da banca ocorreu por meio de videoconferência. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Kleber Vieira Cardoso, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos vinte dias do mês de junho de dois mil e vinte e cinco.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Kleber Vieira Cardoso**, **Professora do Magistério Superior**, em 22/06/2025, às 21:11, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Daniel Campos Da Silva**, **Discente**, em 22/06/2025, às 22:52, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Jacek Kibilda**, **Usuário Externo**, em 23/06/2025, às 11:05, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Flavio Geraldo Coelho Rocha**, **Professor do Magistério Superior**, em 24/06/2025, às 15:13, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5452595** e o código CRC **5C4D09D1**.

All rights reserved. Reproduction of this work in whole or in part without prior authorization from the university, the author, and the advisor is prohibited.

Daniel Campos da Silva

Graduated in Computer Science from Universidade Federal de Goiás (UFG). Was a peer tutor on the courses of Fundamentals of Mathematics for Computing (2018-2019) and Analysis and Design of Algorithms 2 (2020). Did undergraduate research in Vectorial Optimization (2021) and Graph Theory (2021-2022). Worked as a researcher on Rede Nacional de Ensino e Pesquisa (RNP)'s projects CloudNEXT (2022) and OpenRAN@Brasil FASE 2 (2023-2025). Was a teaching assistant in the course of Computer Networks (2024). Has experience and develops research in the following themes: software-defined networks, mobile and wireless networks, Open-RAN, radio resource management, operational research, and simulators.

I dedicate this work to my family, friends, colleagues, and professors for all their support and belief during the beginning of my journey to become a professor.

Special Thanks

First, I thank the Universidade Federal de Goiás (UFG), the Rede Nacional de Ensino e Pesquisa (RNP), and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for providing the faculty support, material resources, and financial assistance necessary for me to dedicate myself exclusively to my master's research over the past two years. Had I needed to work while pursuing my degree, the final result would certainly not have achieved the same quality.

I would also like to thank my advisor, Professor Kleber Vieira Cardoso, who taught me so much about the life of a researcher. He has been an inspiration and a role model, showing me how to navigate an academic career. He consistently encouraged me to tackle more complex tasks, publish in international journals and conferences, collaborate with researchers abroad, and share our work with the global community. Thanks to his connections, I had the opportunity to pursue a Ph.D. at Virginia Tech – an idea that had never even crossed my mind before. Kleber exemplifies how to create opportunities for those from challenging backgrounds, empowering them to channel their determination into meaningful contributions to science and become exceptional researchers. That is the kind of professor I aspire to be.

Though I always envisioned an academic career, there were moments of doubt when difficulties arose, and I relied on my friends and labmates to persevere. I especially thank Gabriel Almeida, William Teixeira, and João Paulo Esper for making our laboratory feel like a second home, where we supported each other, shared our struggles, and chased our goals – all over freshly ground coffee and warm pão de queijo straight from the oven. I am also grateful to my long-time friend Thiago Monteles, who has been by my side since the beginning of our bachelor's degree seven years ago, joining me for parties and movie nights to unwind our hard-working minds.

Lastly, I thank my family for the unwavering love they have given – and still give – me throughout my life. Despite financial struggles, my parents, Sandra Campos and Marcelo Brito, worked tirelessly – often juggling multiple jobs – to ensure I never knew what hunger was like while growing up. They understood that education was the key to opportunities they never had, and I am deeply grateful for their efforts in nurturing my critical thinking and fostering the mindset of a scientist. I also thank my aunt Aracele

Campos, whom I see as a second mother, for helping my parents in taking care of me, especially when I needed to move away from them to pursue a better education in another city. Finally, I thank my little brother and best friend, Guilherme Campos, who shared his childhood with me and remains a constant presence, whether to discuss life's challenges, show new songs on the guitar, play video games, or watch movies together. I love him with all my heart and will always support him in achieving goals even greater than my own.

Teaching is not transferring knowledge, but creating the possibilities for its own production or construction.

Paulo Freire,
Brazilian patron of education, Brazil.

Resumo

da Silva, Daniel Campos. **Channel-Aware Inter-Slice Scheduling for SLA Assurance: Theoretical and Simulation-Based Approaches**. GOIÂNIA, 2025. 80p. Dissertação de Mestrado. Programa de pós graduação em ciência da computação, Instituto de Informática (INF), Universidade Federal de Goiás (UFG).

Conforme a diversidade de aplicações com requisitos de Qualidade de Serviço (QoS) heterogêneos cresce em redes móveis, o fatiamento de rede emerge como uma tecnologia essencial para garantir Acordos de Nível de Serviço (SLAs) através do isolamento de recursos entre diferentes tipos de serviço agrupados em fatias independentes. Um escalonamento eficiente de recursos de rádio entre fatia (RRS inter-fatias) é crucial nesse contexto, pois controla diretamente a vazão atingível - e, consequentemente, a garantia de SLAs - além de promover ganhos em eficiência energética em cenários de baixa demanda, mediante a redução do uso de recursos e do consumo de energia em estações base. Esta dissertação investiga características de RRS inter-fatias de alto desempenho, incluindo: ciência de canal, predição de RRS intra-fatia, alocação orientada a desvio de SLA, proporções dinâmicas dos recursos das fatias e justiça entre usuários de uma mesma fatia. O problema de RRS é formulado matematicamente, viabilizando o projeto de heurísticas aproximando soluções ótimas. Por meio de simulações, demonstramos que os algoritmos propostos superam os escalonadores do estado da arte tanto na garantia de SLAs, quanto em eficiência no uso de recursos.

Palavras–chave

Acordo de nível de serviço, fatiamento de rede, bloco de recurso, escalonamento de recurso de rádio, eficiência energética, otimização.

Abstract

da Silva, Daniel Campos. **Channel-Aware Inter-Slice Scheduling for SLA Assurance: Theoretical and Simulation-Based Approaches**. GOIÂNIA, 2025. 80p. MSc. Dissertation. Programa de pós graduação em ciência da computação, Instituto de Informática (INF), Universidade Federal de Goiás (UFG).

As the diversity of applications with heterogeneous Quality of Service (QoS) requirements grows in mobile networks, network slicing emerges as a key technology to meet Service Level Agreements (SLAs) by isolating resources among different types of services grouped into independent slices. Efficient inter-slice radio resource scheduling (RRS) is crucial in this context, directly governing the achievable throughput - and thus SLA assurance - while also enabling energy efficiency gains in low-demand scenarios through reduced resource usage and power consumption in the base station. This thesis investigates high-performance inter-slice RRS characteristics, including channel-awareness, intra-slice RRS prediction, SLA-drift-oriented allocation, dynamic slice resource proportions, and fairness among users within the same slice. We formulate the RRS problem mathematically, facilitating the design of RRS heuristics approximating optimal solutions. Through simulations, we demonstrate how our proposed algorithms outperform state-of-the-art schedulers in both SLA assurance and resource efficiency.

Keywords

Service level agreement, network slicing, resource block, radio resource scheduling, energy efficiency, optimization.

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

1.1 Contextualization and theoretical basis

The introduction of 5G and 6G standards by the 3rd Generation Partnership Project (3GPP) set a broader range of service diversity as one of the main aspects of future mobile networks, expanding 5G's use cases of Massive Machine-Type Communication (mMTC), Enhanced Mobile Broadband (eMBB), and Ultra Reliable Low-Latency Communication (URLLC) to 6G's new six cases: (*i*) immersive human-centric communications, (*ii*) sensing, localization, and imaging, (*iii*) full-capability industry 4.0 and beyond, (*iv*) smart city and life, (*v*) global coverage for mobile services, and (*vi*) connected machine learning and networked artificial intelligence (AI) [Tong e Zhu 2021]. Consequently, the radio access network (RAN) is expected to support heterogeneous demands with different Service Level Agreements (SLAs) that describe how to achieve their Quality of Service (QoS) requirements. In this context, network slicing is a key 5G technology that enables efficient RAN management among slices, which are logical networks with isolated resources [5G Slicing Association 2020]. Such resources can range from dedicated vCPUs to instances of RAN functions, but allocated radio resources are the ones most directly impacting QoS, since they directly define the data throughput of User Equipments (UEs). In this context, it is crucial to develop Radio Resource Scheduling (RRS) algorithms to allocate radio resources among slices intelligently. Figure 1.1 illustrates a network slicing scenario for a single Base Station (BS) with a static RRS decision among three slices, each with two associated UEs.

However, a static RRS may be insufficient to effectively respond to the dynamism in demand and channel quality for the different slices. Therefore, we define RRS problems not only in the domain of frequency, but also of time. To reduce complexity, both domains are discretized into Physical Resource Blocks (PRBs), also called Resource Blocks (RBs), which are allocated during 1 Transfer Time Interval (TTI) and span a fixed bandwidth. The 3GPP specifications define a numerology $\mu \in \{0, 1, 2, 3, 4\}$ as a parameter to calculate the TTI length as $2^{-\mu}$ milliseconds and the PRB bandwidth as $12 \cdot 2^{\mu} \cdot 15$ kHz [Chen et al. 2023]. Moreover, PRBs can be grouped into Resource Block Groups (RBGs)

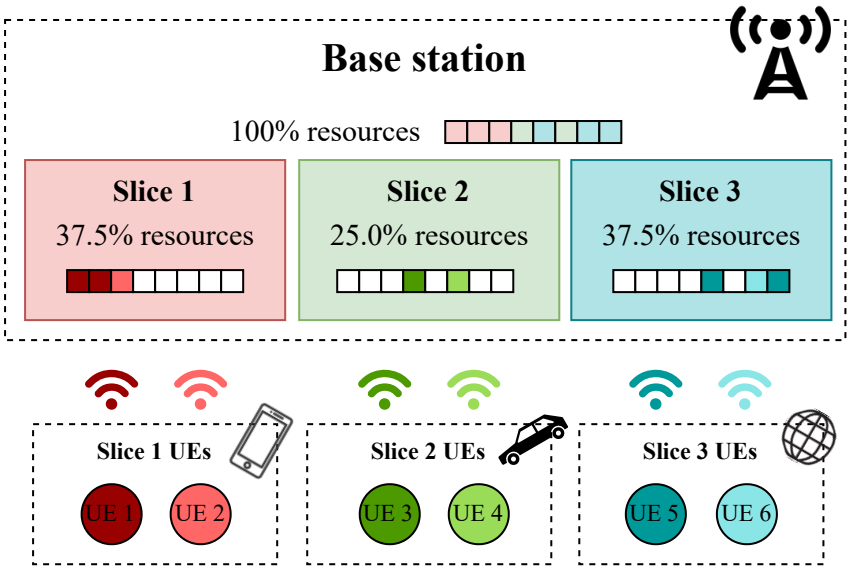


Figure 1.1: Network slicing scenario with static RRS decision.

of size up to 4 RBs to accelerate scheduling by allocating multiple RBs at a time.

Since each slice has independence on how to distribute the received RBG, the RRS is divided in the context of network slicing into two stages: inter-slice RRS and intra-slice RRS. The former is executed once per TTI and defines which of the available RBGs in the BS will be allocated to the slices, while the latter runs for each slice, in parallel and isolated, distributing the slice's RBGs to its associated UEs. The two-stage RRS in the network slicing scenario is represented in Figure 1.2.

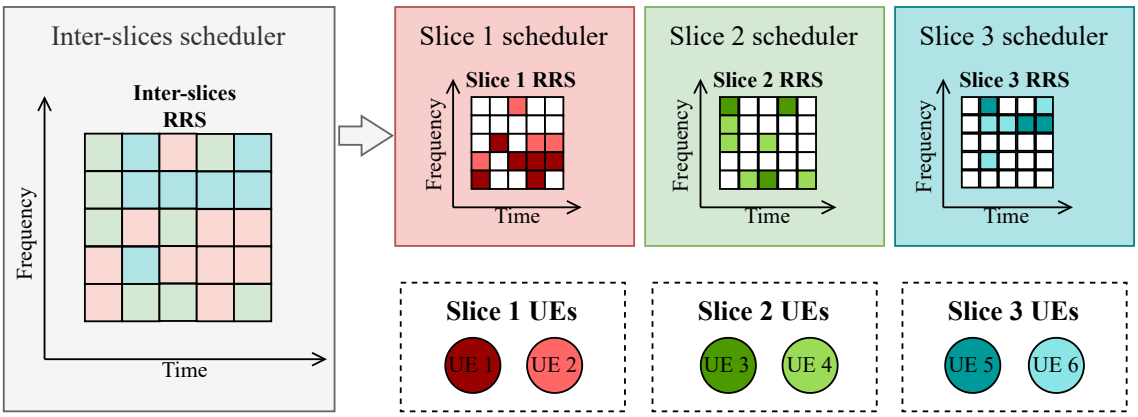


Figure 1.2: Inter-slice and Intra-slice scheduling.

The intra-slice RRS algorithm may vary between slices, since the objective of the scheduler (e.g., maximizing throughput, or maximizing fairness) is application-specific, depending on what type of service the slice provides. The inter-slice scheduler, on the other hand, has to deal with the slices' heterogeneity and ensure different SLAs while prioritizing distinct slices. Besides guaranteeing QoS, another task the RRS is crucial

for is improving energy efficiency, since it can allocate fewer resources to reduce the BS power consumption. Thus, we can define inter-slice RRS primary and secondary objectives as maximizing SLA assurance and minimizing resource usage, respectively.

Another important aspect for RRS algorithms, aside from their objectives, is considering differences in channel quality depending on the RBG frequency for a UE, which we call channel-awareness. Such differences occur in the real world due to the physical phenomenon of frequency-selective fading, and the scheduler may perceive them by receiving Channel Quality Indicators (CQIs) for each RBG of each UE. Channel-awareness is implemented in many intra-slice RRS algorithms, selecting RBGs for UEs with higher CQIs to improve allocation efficiency. However, inter-slice schedulers allocate RBGs to slices, not UEs, so their performance is limited by how the intra-slice schedulers will distribute the allocated resources. Fortunately, the inter-slice scheduler can predict the intra-slice scheduler algorithm, if the latter is greedy, to know which UE will receive each allocated RBG, thus enabling channel-aware inter-slice RRS [Chen et al. 2023].

1.2 Research problem

The research problem considered in this thesis is that of designing channel-aware solutions for inter-slice RRS that efficiently maximize SLA assurance while also minimizing resource usage to enable improvements in energy efficiency.

1.3 Related work

The present thesis is based on two main works. First, [Nahum et al. 2024] develops a Deep Reinforcement Learning (DRL) agent for inter-slice RRS trained to minimize intent-drift, a metric that indicates SLA-non-assurance. In this thesis, we refer to intent-drift as SLA-drift (SLAd) to broaden the term. Although SLA assurance can be a binary objective, quantifying it with SLAd allows us to differentiate schedules in scenarios where SLAs cannot be fully ensured. The paper also describes important metrics for SLA requirements, such as long-term throughput and buffer latency. Nevertheless, we identify some points of improvement in the work: (*i*) the agent design assumes inputs with a fixed number of slices, so it needs to be retrained if new slices are created or removed, (*ii*) the agent's output is a ratio of RBGs for each slice, thus its allocation cannot be channel-aware, and (*iii*) the evaluated scenario considers only Round-Robin as the intra-slice RRS algorithm, which is not commonly used in industry implementation due to its simplicity. All the previous points are addressed in our second main reference [Chen et al. 2023], which develops a polynomial channel-aware inter-slice RRS algorithm for selecting RBGs to maximize the overall BS throughput. Notwithstanding, the work considers a

fixed ratio of RBGs for each slice and is unaware of SLA requirements, limiting the scheduler's adaptability in scenarios with highly dynamic demands. Moreover, since the scheduler predicts intra-slice allocations to choose RRS decisions that maximize throughput, it tends to select unfair allocations prioritizing UEs with better channel quality, while UEs in worse conditions within the same slice have their SLAs not assured, regardless of the intra-slice RRS algorithm being fairness-oriented.

[Chen et al. 2023], as well as [Balasingam, Kotaru e Bahl 2024] and [Dai et al. 2024], compare their performance with a well-known scheduler [Kokku et al. 2012] from the early 2010s. However, its assumptions are outdated and lead to performance issues. For instance, it does not consider orthogonal frequency division multiple access (OFDMA) and schedules all available RBGs to a single slice at every TTI. A further limitation, that also applies to [Chen et al. 2023], is that resource distribution follows a weighted round-robin approach, with slice weights based on historical throughput. Such weights might not reflect the number of resources needed to fulfill different SLA requirements, since some may not directly relate to throughput.

Most of the inter-slice schedulers in the state-of-the-art explicitly allocate all available resources or aim to maximize throughput, thus leading to a high use of bandwidth. We cite two works that go in the opposite direction, improving energy efficiency in scenarios where resources are plentiful. [Yang et al. 2024] formulates a Reinforcement Learning (RL) agent with a reward function that maximizes the weighted sum of throughput minus delay minus resource usage. However, this approach requires fine-tuning the weights to avoid allocation decisions where saving resources has a higher reward than ensuring SLAs, which leads to user starvation. Additionally, static weights may not adjust to the variation of demand in the scenario. On the other hand, [Balasingam, Kotaru e Bahl 2024] provides a framework for inter-slice RRS that minimizes allocated bandwidth as its main objective, subject to ensuring throughput and latency SLA requirements, which are formulated exactly as the SLAd metric.

Motivated by recent developments in artificial intelligence/machine learning algorithms, the majority of recent works approaching inter-slice RRS develop RL agents to define the proportion of radio resources each slice should receive. [Sherif, Ahmed e Kotb 2025] proposes different DRL agents leveraging the Proximal Policy Optimization (PPO) and Transfer Learning (TF) algorithms and evaluates how different reward functions impact the carbon footprint of their training process. The authors from [Nahum et al. 2024] improve the Soft Actor-Critic (SAC) RL solution, evaluated in scenarios where all slices use Round-Robin as intra-slice RRS algorithm, to [Nahum et al. 2025], a PPO Multi-Agent RL (MARL) solution where an agent defines the ratio of resources for each slice and another chooses each slice's intra-slice RRS algorithm, among Round-Robin, Proportional Fair, and Maximum Throughput. After

[Boutiba et al. 2022] structured a solution integrated with the MAC layer to support multi-slice association (i.e., one UE can be associated with more than one slice) and mixed numerology (i.e., multiple numerologies in the same base station), the authors expand their work in [Boutiba et al. 2023] by formulating a Mixed Integer Linear Problem (MILP) and approximating the optimal solution with a Deep Q-Learning (DQN) RL algorithm. All of the mentioned RL agents, however, have the same two downsides: (*i*) they need to be trained with a large dataset or interactive environment, which can take hours to finish and (*ii*) they need to be retrained for new scenarios with different slice configurations since their state spaces have a fixed number and order of slices. Moreover, almost all of the defined action spaces are a list of real numbers representing the ratio of resources reserved to each slice, which cannot represent channel-aware allocations, since RBGs are not selected individually.

There are also studies evaluating inter-slice RRS on Open-RAN environments, where the scheduler usually runs as an xApp, which is a RAN-optimization microservice application executed in the Near Real-Time RAN Intelligent Controller (Near-RT RIC) platform. [Polese et al. 2022] evaluates three PPO DRL xApps, one of them trained online, that maximize or minimize network metrics (e.g., buffer size, physical transport blocks, or PRB ratio) for specific slices in the Colosseum wireless testbed [Bonati et al. 2021]. [Dai et al. 2024] maximizes the transmitted bits minus the head of buffer delay as a DDPG RL agent implemented in an xApp optimizing the srsRAN radio stack. [Cheng et al. 2024] modifies Open Air Interface's radio stack to support slicing and follow xApp-defined Radio Resource Management (RRM) policies, which specify the ratio of dedicated, prioritized, and shared resources in the BS for each slice. [Navidan et al. 2024] also addresses the definition of RRM policies, which are generated by a DQN agent maximizing slice performance index metrics, similar to SLAd, and evaluated on the Colosseum testbed. However, exploring inter-slice RRS solutions in the Open-RAN architecture is out of the scope of this thesis.

1.4 Justificative, significance, and motivation

The growing variety of services supported by 3GPP networks demands more efficient RRS algorithms that can consider the heterogeneity of applications with different QoS requirements. Such aspects of RRS become possible in 5G and Beyond scenarios thanks to technologies and concepts recently introduced in mobile networks to enable intelligent RAN management and optimization, such as network slicing, Open-RAN, and Artificial Intelligence (AI)-RAN. Leveraging those concepts, we propose and evaluate dynamic solutions for the RRS problem that ensure QoS and, following recent discussions on RAN energy efficiency, minimize power consumption by reducing resource allocation.

In this thesis, we identify desirable characteristics for inter-slice RRS algorithms and discuss metrics for their evaluation in scenarios with high service heterogeneity. We compare our proposed solutions with state-of-the-art algorithms from the literature in simulated scenarios and analyse their performance to show points of improvement. Moreover, all code implementing the simulation, optimization models, heuristics, baselines, and graphic generation is made publicly available so that the scientific community can replicate, study, and improve the obtained results.

The selection of the theme researched in this thesis was made by the present master's student to leverage: (*i*) his knowledge in optimization and heuristics, and (*ii*) his mentor being the co-author of [Nahum et al. 2024], which had a good development of inter-slice RRS concepts, but lacked the formulation of an optimization problem, giving a clear room for improvement. Additionally, the RRS problem is well-known and has classical solutions (e.g., Round-Robin and Proportional Fair), facilitating the definition of the research problem, which could grow in complexity as new results are obtained (e.g., considering resource minimization and channel-awareness).

1.5 Objectives

The general objective of this thesis is to propose and evaluate solutions for the channel-aware inter-slice RRS oriented to maximizing SLA assurance and minimizing resource allocation. The specific objectives of this thesis include:

- Selecting relevant metrics for evaluating inter-slice RRS algorithms;
- Reproducing and comparing state-of-the-art inter-slice RRS algorithms from the literature with our proposed solutions;
- Formulating the research problem as an optimization problem;
- Evaluating the proposed and literature solutions compared with optimal or approximated solutions obtained from optimization models.

1.6 Methodology

The methodology followed during the thesis and each of its papers consists of iteratively revisiting the steps described below, in the following order:

1. Reviewing the literature to outline key aspects of the RRS problem in the context of network slicing;
2. Formulating the research problem as a mathematical optimization problem;
3. Designing and implementing optimization models to find optimal or approximate solutions for the formulated problem;

4. Designing and implementing heuristics to solve the formulated problem in a reasonable time with performance similar to the optimal solution;
5. Studying and implementing inter-slice RRS schedulers from the literature as baselines for comparing with the proposed solution;
6. Building simulated environments to evaluate the heuristic, baselines, and optimal scheduling decisions;
7. Generating graphs, figures, and metrics describing performances obtained in the simulated environment;
8. Analyzing and discussing the obtained results;
9. Writing a paper presenting the work, obtained results, and insights.

1.7 Contribution

Below, we list the contributions and subproducts of this research, including our main findings and publicly available code:

- A listing of key points for improving inter-slice RRS performance: (*i*) channel-awareness, (*ii*) intra-slice RRS prediction, (*iii*) SLAd-oriented, (*iv*) resource minimization, and (*v*) dynamic slice resource proportion;
- The implementation of a modular simulation for testing inter-slice and intra-slice RRS algorithms;
- The problem formulation and implementation of an optimization model for solving the stepwise inter-slice RRS in plentiful scenarios;
- The design and implementation of the Stepwise Optimal Algorithm (SOA), optimally solving the problem above;
- An analysis of how SOA can assure SLAs in plentiful resources scenarios, even with fewer resources, overperforming the DRL agent from [Nahum et al. 2024];
- The problem formulation and implementation of an optimization model for solving the channel-aware inter-slice RRS oriented to minimizing SLAd and resource allocation in both plentiful and scarce resource scenarios;
- The design and implementation of the Drift and Resource Allocation MINimization (DREAMIN) scheduler, approximating the optimal solution for the problem above;
- An analysis of the SLAd metric and how inter-slice RRS algorithms based on it, as DREAMIN, can improve SLA-assurance, energy efficiency, and intra-slice fairness, compared to algorithms that maximize throughput, as [Chen et al. 2023].

1.8 Thesis structure description

This thesis follows the Scandinavian model, containing an introduction, produced papers, and a conclusion. The documents proving that the two produced papers were published or accepted are included in the appendix section at the end of the thesis. We organize the next chapters as follows. Chapter 2 contains the first paper produced in this research, entitled "Stepwise Optimal Inter-Slices Radio Resource Scheduling for Service-Level Agreement Assurance". It provides a first look at the inter-slice RRS problem, considering only plentiful scenarios and evaluating the SOA algorithm. It was written in English and published on the Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, which occurred in May 2024, was organized by the Sociedade Brasileira de Computação (SBC), and had an acceptance rate of 42%. Chapter 3 contains the second paper produced in this research, entitled "DREAMIN: Channel-Aware Inter-Slices Radio Resource Scheduling for Efficient SLA Assurance". It expands the problem from Chapter 2 by formulating a channel-aware problem for multiple steps, considering both plentiful and scarce scenarios, and solving it with DREAMIN. It was written in English and accepted for publication at the International Conference on Communications, which occurred in June 2025, was organized by the Institute of Electrical and Electronics Engineers (IEEE), and had an acceptance rate of 40% in its previous edition. Finally, Chapter 4 concludes the thesis with some final considerations and discussions about the findings, limitations, and recommended future work for continuing this research.

CHAPTER 2

Stepwise Optimal Inter-Slices Radio Resource Scheduling for Service-Level Agreement Assurance

2.1 Abstract

In 5G networks and beyond, radio access networks (RANs) must be able to support multiple services with different service level agreements (SLAs). Network slicing is a critical concept in this context and it depends on an efficient approach for radio resource scheduling (RRS). Inter-slices RRS is responsible for allocating resource block groups (RBGs) to the slices to ensure their SLAs. Mainly motivated by the O-RAN initiative, several works in the literature have presented proposals based on machine learning (ML) to solve this problem. However, there is still a lack of problem formalization and an optimal strategy, which are both introduced in this work. Through simulations, we compare our approach with a state-of-the-art deep reinforcement learning (DRL) agent. The results show the excess resources employed by the agent when they are plentiful, suggesting an unnecessary increase in energy consumption. Additionally, we show the relevant gap between solutions when the resources are scarce. Finally, we discuss guidelines on how to improve ML-based approaches to the inter-slice RRS problem.

2.2 Introduction

While the adoption of Open RAN is still in the very beginning and surrounded by discussions, the impact of the O-RAN Alliance and its standards is already huge in the telecommunications ecosystem. ML-based approaches in the RAN are not new since 3GPP Release 8 (from 2006) already had self-organizing networks (SON) related exactly to this type of approach. The several advances in the ML field and the design adopted for the O-RAN architecture [Polese et al. 2023] have kept most of the academic interest strongly guided in this direction when investigating RAN-related issues. Two key

components of the O-RAN architecture are the near real-time RAN intelligent controller (Near-RT RIC) and the non-real-time RAN intelligent controller (Non-RT RIC). O-RAN compatible solutions must be developed as xApps (running in the Near-RT RIC) and rApps (running in the Non-RT RIC), which are generally ML-based applications. As a consequence, commonly, the ML-based proposals found in the literature are compared only against other ML-based counterparts. In the context of resource allocation, this may raise a basic question: how far from the optimal are the solutions?

In the RAN, network slicing is critical and involves non-trivial resource allocation. Network slicing is the main enabler for supporting multiple services with different SLAs over the same infrastructure. An SLA takes into account a set of requirements that the network operator must ensure to provide the necessary quality of service (QoS) to the users' applications. To ensure the QoS, the network operator creates slices to serve the services with SLAs that may be very different. For example, video streaming from a smartphone requires a high throughput but tolerates packet loss and some latency, thus it may belong to an enhanced mobile broadband (eMBB) slice. On the other hand, self-driving vehicles need a very low packet loss and latency, best supported by ultra-reliable low-latency communication (URLLC) slices. This means that each slice must receive a specific amount of resources to satisfy the correspondent SLA and this resource allocation is highly dynamic in the RAN due to two main reasons. First, the number of user equipment (UEs) associated with each base station (BS) changes as the users move, thus, the consumption of resources also varies. Second, wireless channel conditions of each UE also vary, not only due to user mobility but also due to other environmental characteristics. This context is appealing to model-free approaches such as ML-based ones, but we argue that this does not preclude the pursuit of problem formalization and optimal solutions. Thus, we can build consistent performance references and find important insights that can contribute to designing better approaches, including the ML-based ones as we will discuss later.

Related work – There are several papers in the literature investigating issues related to network slicing in the RAN. In the following, we do not try to be comprehensive, but cite some critical works on the topic, mainly considering inter-slices allocation and including some recent state-of-the-art papers. [Kokku et al. 2012] is a well-known solution for inter-slice scheduling referenced by other inter-slice schedulers. Because it was developed in the early 2010s, its assumptions are outdated. The main problem of its scheduling is not considering orthogonal frequency division multiple access (OFDMA), thus scheduling radio resources by allocating all RBGs to a single slice at every transmission time interval (TTI). Another issue is that the resources are distributed in a weighted round-robin way, where each slice weight is defined by its historical throughput. This weight may not represent the needed resources to ensure each slice's SLA, as its restrictions may not

directly relate to throughput.

[Chen et al. 2023] shows that the spectral efficiency can differ between RBGs for a single UE. It then develops a heuristic that maximizes the total throughput by selecting the best pair of RBG and UE. The intra-slice scheduling must be greedy to enable predicting its allocation to select the best RBGs for each slice. Nonetheless, this solves only the problem of selecting the resources, but the number of RBGs for each slice is determined similarly to [Kokku et al. 2012], using a pre-determined slice weight. [Nahum et al. 2024] develops a DRL agent to solve the inter-slices scheduling among eMBB, URLLC, and BE slices. We call this solution the DRL agent. It uses *intents*, which are equivalent to SLAs from the scheduler's perspective, to determine *intent-drifts*: the normalized difference between a required metric and its required value. Therefore, the agent's objective is to minimize the intent-drift for all slices.

[Khodapanah et al. 2020] provides a framework for inter-slice RRS using artificial neural networks to maximize the number of fulfilled SLA requirements assuming the allocation of fractional resources. In [Lotfi, Afghah e Ashdown 2023], an attention-based DRL agent is presented for scheduling resource blocks (RBs), not RBGs, between eMBB, URLLC, and massive machine-type communication (mMTC) slices. [Polese et al. 2022] uses proximal policy optimization to train a DRL agent that executes as an xApp in O-RAN architecture, scheduling resources to maximize or minimize metrics for each slice. Additionally, [Mei et al. 2021] proposes a DRL framework combining a deep deterministic poly gradient and a deep-Q-network algorithm to address the inter-slices scheduling problem. A common aspect of the related works is that every solution always allocates 100% of the BS RBGs, which may be inefficient in several scenarios. Allocating only the minimal resources necessary to satisfy the SLAs brings benefits, such as the possibility to serve more users or minimize energy consumption.

Our contributions and paper organization – In this work, we optimally solve the stepwise optimal inter-slices RRS for SLA assurance problem, i.e., employing the minimum number of RBGs to assure the SLA of every UE. In summary, our main contributions are:

- The formalization of the problem of stepwise inter-slices RRS for SLA assurance.
- The design of an algorithm that solves the problem in polynomial time.
- A comparison of our approach with a state-of-the-art DRL agent, using simulation, which illustrates the room for improvement and provides insights on how to improve ML-based approaches.

Section 2.3 presents the system model and the problem formulation. Section 2.4 describes how to solve the formulated problem with a polynomial algorithm. Section 2.5 evaluates our solution in comparison with the DRL agent and a weighted round-robin in a simulated environment. Lastly, Section 2.6 contains our conclusions and future works.

2.3 System model and problem formulation

In this section, we first introduce the system model employed to define the problem, delineating parameters pertinent to our study. Subsequently, we present the problem formulation, specifying the metrics associated with the SLA of each slice type and outlining the objective of our formulation.

2.3.1 System model

We assume that each UE u possesses a buffer represented as an array $L_u = [B_u(0), B_u(1), \ldots, B_u(L-1)]$, where L is the maximum buffer latency, in TTIs. Each element $B_u(i) \in L_u$ indicates the number of packets awaiting transmission for i timesteps. For example, if $L_u = [1,3,5]$, it means that 1 packet has just arrived (waiting for 0 timesteps), 3 packets have been waiting for 1 timestep, and 5 packets have been waiting for 2 timesteps. Furthermore, we denote the cumulative count of transmitted packets with an array $B_u^{sent} = [B_u^{sent}(0), B_u^{sent}(1), \ldots, B_u^{sent}(\eta-1)]$. Each $B_u^{sent}(i)$ represents how many packets waited for i timesteps until being transmitted since the beginning of the environment. For instance, if $B_u^{sent} = [3,5,4]$, it indicates that, until now, 3 packets were immediately sent (no waiting), 5 packets waited for 1 timestep before transmission, and 4 packets waited for 2 timesteps before being transmitted.

The BS bandwidth is discretized into RBs with $2^\mu \cdot 180$ kHz each, where $\mu \in [0,1,2,3,4]$ is the BS option, as in the 5G standards [ETSI 2020]. The TTI time length is directly related to the RB bandwidth, expressed as $l = 2^{-\mu}$ ms. Moreover, ρ RBs are aggregated into one RBG with a total bandwidth of $R = \rho \cdot 2^\mu \cdot 180$ kHz [ETSI 2020]. This way, we discretize the time in our model as timesteps lasting 1 TTI each. We define a set $\mathcal{T} = \{0,1,\ldots,\eta\}$ comprising the timesteps, where η is the current one. Additionally, we introduce a time window of $W \in \mathbb{N}_+$ timesteps for calculating historical metrics. As W may be greater than the number of past steps, $\omega = \min(W, \eta+1)$ is used as the current window. The stages of a timestep are as follows: (*i*) packets arrive in each UE's buffer, then (*ii*) the inter-slice schedulers allocate the available RBGs of the BS among the slices, which (*iii*) distribute the received resources among the UEs to (*iv*) transmit packets from the buffer to the BS.

2.3.2 Problem formulation

We address the stepwise optimal inter-slices RRS for SLA assurance problem by defining the minimal number of RBGs necessary for ensuring the SLA requirements of every UE in the actual timestep. We denote by $\mathcal{S}$ the set of slices in the environment, while $\mathcal{U}_s$ is the set of users assigned to the slice $s \in \mathcal{S}$. In this way, we express $\alpha_s \in \mathbb{N}$

as the amount of RBG allocated to slice $s \in \mathcal{S}$ in timestep η. To calculate the predicted metrics for the UEs, we define β_u as the number of RBGs that will be allocated for $u \in \mathcal{U}_s$ if $s \in \mathcal{S}$ receives α_s RBGs. In this work, we address three types of slices: best effort (BE), eMBB, and URLLC. The BE slice has services with tolerant throughput requirements [Khodapanah et al. 2020]. The SLA for BE is defined as minimum values for the fifth-percentile throughput and long-term throughput. Moreover, the SLA for eMBB and URLLC comprises a minimum value for the served throughput and maximum values for the packet loss rate and the average buffer latency [Nahum et al. 2024]. To achieve a generic formulation that may be expanded to include new slices and SLAs, we categorize the set of slices into two subsets $\mathcal{S} = \{\mathcal{S}^{buf} \cup \mathcal{S}^{thr}\}$. The first, $\mathcal{S}^{buf}$, comprises slices whose SLAs relate to buffering, while $\mathcal{S}^{thr}$ consists of slices with SLAs depending solely on throughput metrics. In this work, we assume $\mathcal{S}^{thr} = \{BE\}$ and $\mathcal{S}^{buf} = \{eMBB, URLLC\}$. In the following, we describe the five SLA constraints and the objective function of our problem, assuming that the network operator defines the required value for each constraint based on the application type.

Maximum tolerated average buffer latency – This constraint prohibits the solution from surpassing the average buffer latency threshold defined for each slice $s \in \mathcal{S}^{buf}$, denoted as l_s^{req}. We use $\psi_u(i)$ as the predicted number of packets in $B_u(i)$ that will be transmitted if u receives β_u RBGs. The constraint is defined as follows:

$$\frac{\sum_{i=0}^{L-1}(\psi_u(i) + B_u^{sent}(i)) \cdot i}{\sum_{i=0}^{L-1}(\psi_u(i) + B_u^{sent}(i))} \cdot l \leq l_s^{req}, \qquad \forall u \in \mathcal{U}_s, s \in \mathcal{S}^{buf}. \tag{2-1}$$

Packet loss rate – This constraint prevents the violation of the packet loss rate requirement for a slice $s \in \mathcal{S}^{buf}$, denoted by p_s^{req}. We consider a scenario where packets can be dropped due to two reasons: (i) the buffer reaches its maximum capacity at the moment a packet arrives, and (ii) a packet achieves its maximum latency L. We denote $\overline{D_u^{arr}}(t)$ as the number of packets dropped at timestep $t \in \mathcal{T}$ due to maximum buffer capacity and $\overline{D_u^{lat}}(t)$ as the number of packets dropped at timestep $t \in \mathcal{T}$ due to the maximum latency constraint. As $\overline{D_u^{lat}(\eta)}$ can only be known after scheduling, we call ϕ_u^{lat} the number of packets that will be dropped in this timestep if u receives β_u RBGs. Similarly, we write ϕ_u^{arr} as the packets that will be dropped upon arrival in the $\eta+1$ timestep, given β_u. We assume that λ_u packets will arrive at the UE u in the timestep $\eta+1$ when calculating ϕ_u^{arr}. Considering $\overline{B_u^{start}(t)}$ as the number of packets in the buffer at the beginning of timestep $t \in \eta$, before packet arrival, and $\overline{A_u}(t)$ as the number of packets that arrived the buffer at timestep $t \in \mathcal{T}$, we define the packet loss rate constraint as:

$$\frac{\sum_{t=\eta-\omega+1}^{\eta} \overline{D_u^{arr}}(t) + \sum_{t=\eta-\omega+1}^{\eta} \overline{D_u^{lat}}(t) + \phi_u^{lat} + \phi_u^{arr}}{\overline{B_u^{start}}(\eta-\omega+1) + \sum_{t=\eta-\omega+1}^{\eta} \overline{A_u}(t) + \lambda_u} \leq p_u^{req}, \ \forall u \in \mathcal{U}_s, s \in \mathcal{S}^{buf}. \tag{2-2}$$

Served throughput – This constraint dictates that the required served throughput t_s^{req} must be upheld for $u \in \mathcal{U}_s$. We denote E_u as the spectral efficiency of the user $u \in \mathcal{U}_s$ in the current timestep. The constraint is defined as:

$$\beta_u \cdot R \cdot E_u \geq t_s^{req} \qquad \forall u \in \mathcal{U}_s, s \in \mathcal{S}^{buf}. \tag{2-3}$$

Long-term throughput – This constraint prevents the solution from violating the long-term throughput requirement for all slices $s \in \mathcal{S}^{thr}$, denoted by g_s^{req}. We represent the historically served throughput of user $u \in \mathcal{U}_s$ at timestep $t \in \mathcal{T}$ as $\overline{T_u}(t)$. The long-term throughput is defined as the average throughput of the UE over the time window of the last ω timesteps. Therefore, the long-term throughput constraint is defined as follows:

$$\frac{1}{\omega} \cdot \left(\beta_u \cdot R \cdot E_u + \sum_{t=\eta-\omega+1}^{\eta-1} \overline{T_u}(t) \right) \geq g_s^{req}, \qquad \forall u \in \mathcal{U}_s, s \in \mathcal{S}^{thr}. \tag{2-4}$$

Fifth-percentile throughput – This constraint assures the fifth-percentile throughput requirement f_s^{req} for $s \in \mathcal{S}^{thr}$. This constraint prevents the solution from letting the fifth-percentile throughput of a user $u \in \mathcal{U}_s$ be below f_s^{req}. The fifth-percentile throughput is calculated by obtaining the h-th element of the sorted list $[\overline{T_u}(\eta-\omega+1), \ldots, \overline{T_u}(\eta-1), \beta_u \cdot R \cdot E_u]$ of the throughput in the last ω timesteps, where $h = \lfloor \frac{5}{100}\omega \rfloor$. Considering $W < 20$, we can simplify the metric to represent the constraint as:

$$\min\left(\overline{T_u}(\eta-\omega+1), \ldots, \overline{T_u}(\eta-1), \beta_u \cdot R \cdot E_u \right) \geq f_s^{req}, \qquad \forall u \in \mathcal{U}_s, s \in \mathcal{S}^{thr}. \tag{2-5}$$

We aim to minimize radio resource usage while ensuring QoS for every user. Thus, the stepwise optimal inter-slices RRS for SLA assurance problem is formalized as:

$$\begin{array}{l} \text{minimize} \sum_{s \in \mathcal{S}} \alpha_s \\ \text{Subject to Equation}(2-1) \text{ to } (2-5) \end{array} . \tag{2-6}$$

However, linearizing the formulated problem is a complex task, which includes restricting β_u to how the intra-slice scheduler would work and expressing ψ_u, ϕ_u^{lat} and ϕ_u^{arr} as variables that depend on β_u for each UE u. Fortunately, this problem can be solved by a polynomial algorithm, as explained in Section 2.4. It is important to note that our stepwise problem is not equivalent to solving the scheduling for all the timesteps all at once, which is an NP-hard problem.

2.4 Stepwise Optimal Algorithm

The problem in Equation 2-6 can be solved with a polynomial greedy algorithm. We call our strategy the stepwise optimal algorithm (SOA), which considers a scenario where it is possible to meet the QoS requirements for all UEs at each timestep. In the following, we describe how SOA assures SLA and predicts intra-slice scheduling.

2.4.1 Minimum throughput necessary

The output of the scheduling must be an allocation of RBGs, which is directly related to the UE's throughput. Based on this fact, we convert every SLA requirement into a Minimum Throughput Necessary (MTN) to respect the restriction of $u \in \mathcal{U}_s$ at the actual timestep. This strategy enables SOA to be expanded to new scenarios by calculating the MTN for new SLA restrictions, which may describe different slices.

The MTN to respect the served throughput constraint of Equation 2-3 is expressed as $MTN_s^t(u) = t_s^{req}$. The same simple expression happens to the fifth-percentile throughput MTN, noted as $MTN_s^f(u)$. Because SOA respects all restrictions at every timestep, then $\overline{T_u}(t) \geq f_s^{req}$, $\forall t \in \mathcal{T} \setminus \{\eta\}$. Thus, $MTN_s^f(u) = f_s^{req}$ ensures the constraint in Equation 2-5. We denote by $MTN_s^g(u)$ the MTN to assure the long-term throughput requirement of Equation 2-4, obtained by isolating the served throughput of the actual timestep:

$$MTN_s^g(u) = g_s^{req} \cdot \omega - \sum_{t=\eta-\omega+1}^{\eta-1} \overline{T_u}(t). \tag{2-7}$$

The average buffer latency MTN, denoted as $MTN_s^l(u)$, ensures the constraint of Equation 2-1. We noted that respecting this constraint while minimizing resources, in the long term, tends to a scenario where packets are sent at the last moment before l_s^{req}. Hence, we approximate $MTN_s^l(u)$ as the MTN to send the packets that will have waited for more than l_s^{req} in the next timestep:

$$MTN_s^l(u) = B_u\left(\left\lfloor \frac{l_s^{req}}{l} \right\rfloor\right) \cdot Z_s \cdot \frac{1}{l}, \tag{2-8}$$

where Z_s is the packet size for all packets from UEs associated with the slice $s \in \mathcal{S}$. The packet loss rate MTN $MTN_s^p(u)$ must ensure that ϕ_u^{lat} and ϕ_u^{arr} are small enough to respect the constraint of Equation 2-2. We approximate the number of packets that will arrive in the next timestep as $\lambda_u = \overline{A_u}(\eta)$. Considering that no resource is allocated to the UE u, we calculate $\phi_u^{lat} = B_u(L-1)$ and $\phi_u^{arr} = \left\lceil \frac{1}{Z_s} \cdot \max(0, (\overline{A_u}(\eta) + \sum_{i=0}^{L_u-1} B_u(i)) \cdot Z_s - B^{max}) \right\rceil$, with B^{max} as the buffer capacity in bits. Then, the number of packets that will drop between this scheduling and the next is $\max(\phi_u^{lat}, \phi_u^{arr})$, as $\phi_u^{lat} + \phi_u^{arr}$ may count the same packet twice. We express the denominator of Equation 2-2, the total of packets,

as $\theta = \overline{A_u}(\eta) + \overline{B_u^{start}}(\eta - \omega + 1) + \sum_{n=\eta-\omega+1}^{\eta} \overline{A_u}(n)$. Thus, the MTN to send packets under drop risk and respect p_s^{req} is:

$$MTN_s^p(u) = \left\lceil Z_s \cdot \max(0, \sum_{n=\eta-\omega+1}^{\eta} \overline{D_u^{arr}}(n) + \sum_{n=\eta-\omega+1}^{\eta-1} \overline{D_u^{lat}}(n) + \max(\phi_u^{lat}, \phi_u^{arr}) - p_s^{req} \cdot \theta) \right\rceil \frac{1}{l}. \quad (2\text{-}9)$$

Lastly, we express as $MTN_s(u)$ the MTN to ensure the SLA of a UE $u \in \mathcal{U}_s$ in the actual step. It is calculated as the maximal MTN among the restrictions of u:

$$MTN_s(u) = \begin{cases} \max(MTN_s^t(u), MTN_s^l(u), MTN_s^p(u)), & \text{if } s \in S^{buf} \\ \max(MTN_s^f(u), MTN_s^g(u)), & \text{if } s \in S^{thr} \end{cases}. \quad (2\text{-}10)$$

2.4.2 RBG allocation

The SOA achieves $MTN_s(u)$ at every timestep by ensuring the allocation of at least $\beta_u^{min} = \left\lceil \frac{MTN_s(u)}{E_u \cdot R} \right\rceil$ RBGs for each $u \in \mathcal{U}_s$, $s \in \mathcal{S}$. Although the intra-slice scheduler allocates RBGs for the UEs, we can predict its scheduling similarly to [Chen et al. 2023]. As in [Nahum et al. 2024], we consider that the intra-slice scheduler for each slice $s \in \mathcal{S}$ is a Round-Robin, which cycles through $\mathcal{U}_s$ uniformly distributing the slices' RBGs among the users. The Round-Robin state is saved as an `offset`, the index of the next UE in the cycle. Knowing the `offset`, we act as a Round-Robin, allocating 1 RBG at a time until the number of RBGs for $u \in \mathcal{U}_s$ is $\beta_u \geq \beta_u^{min}$. Hence, SOA allocates $\alpha_s = \sum_{u \in \mathcal{U}_s} \beta_u$ for each slice $s \in \mathcal{S}$ as the minimal number of RBGs needed to assure the SLA of every $u \in \mathcal{U}_s$. The SOA allocation is described in Algorithm 2.1. The $MTN_s(u)$ function can be implemented in time $\mathcal{O}(1)$ using cumulative variables for the summations. Considering a scenario with limited resources, the loop at line 7 could stop the algorithm if the total allocated RBGs reaches G, which is the number of available RBGs in the BS. Hence, as every RBG is allocated only once, the time complexity of SOA is $\mathcal{O}(G)$.

2.5 Evaluation

In this section, we present the evaluation results of the proposed SOA. First, we describe the simulation environment used to implement the wireless network and its UEs. Then, we compare the SOA solutions with two state-of-the-art models: a weighted

Algorithm 2.1: SOA allocation process.

Data: Set of slices $\mathcal{S} = \{s_1, \ldots, s_{|\mathcal{S}|}\}$ and set of users for each slice $\mathcal{U}_{s_1}, \ldots, \mathcal{U}_{|\mathcal{S}|}$
Result: Number of RBGs allocated for each slice $s \in \mathcal{S}$: $\alpha_{s_1}, \ldots, \alpha_{s_{|\mathcal{S}|}}$

for $s \in \mathcal{S}$ **do**
  `offset` $\leftarrow$ getIntraSliceRoundRobinOffset(s)
  for $u \in \mathcal{U}_s$ **do**
    $\beta_u \leftarrow 0$
    $\beta_u^{min} \leftarrow \left\lceil \frac{MTN_s(u)}{E_u \cdot R} \right\rceil$
  end
  while $\exists u \in \mathcal{U}_s$ *such that* $\beta_u < \beta_u^{min}$ **do**
    $u \leftarrow \mathcal{U}_s$[`offset`] `// Mimicking Round-Robin`
    `offset` $\leftarrow$ `offset` $+ 1 \pmod{|\mathcal{U}_s|}$ `// intra-slice allocation`
    $\beta_u \leftarrow \beta_u + 1$
  end
  $\alpha_s \leftarrow \sum_{u \in \mathcal{U}_s} \beta_u$
end
return $\alpha_{s_1}, \ldots, \alpha_{s_{|\mathcal{S}|}}$

round-robin algorithm and the DRL agent proposed in [Nahum et al. 2024]. All evaluation results are publicly available at GitHub[1].

2.5.1 Simulation

To evaluate the SOA in different scenarios, we implemented a simulator that leverages [Nahum et al. 2024] dataset of realistic spectral efficiency values generated with the channel impulse responses for a wireless network simulated with QUAsi Deterministic RadIo channel GenerAtor (QuaDRiGa)[2] [Jaeckel et al. 2014]. The dataset contains 50 different trials of uplink communication that consider a massive Multiple Input Multiple Output (MIMO) system, the dual-slope path loss statistical models of 3GPP 38.901 UMi [Mondal et al. 2015, Zhu et al. 2021], the interference from the six more interfering nearby cells, shadow fading, and the MIMO spectral efficiency estimate equation from [Jr e Lozano 2018]. Each trial collects the data of 10 UEs throughout 2000 timesteps lasting 1 ms each. A timestep in our simulation has a time length of 1 TTI and is structured as represented in Figure 2.1. The main parameters of the simulation are listed in Table 2.1.

We consider a base station with 100 MHz of bandwidth, resulting in 138 RBGs. The packet arrival for a UE in each timestep is dictated by a Poisson distribution with

[1] https://github.com/LABORA-INF-UFG/paper-DGWCAMK-2024
[2] https://quadriga-channel-model.de/

Parameter	W	I	L	R	B^{max}	G	ρ	μ
Value	10 timesteps	1 ms	100 TTIs	720 KHz	32 kbytes	138 RBGs	4 RBs	0

Table 2.1: Simulation parameters.

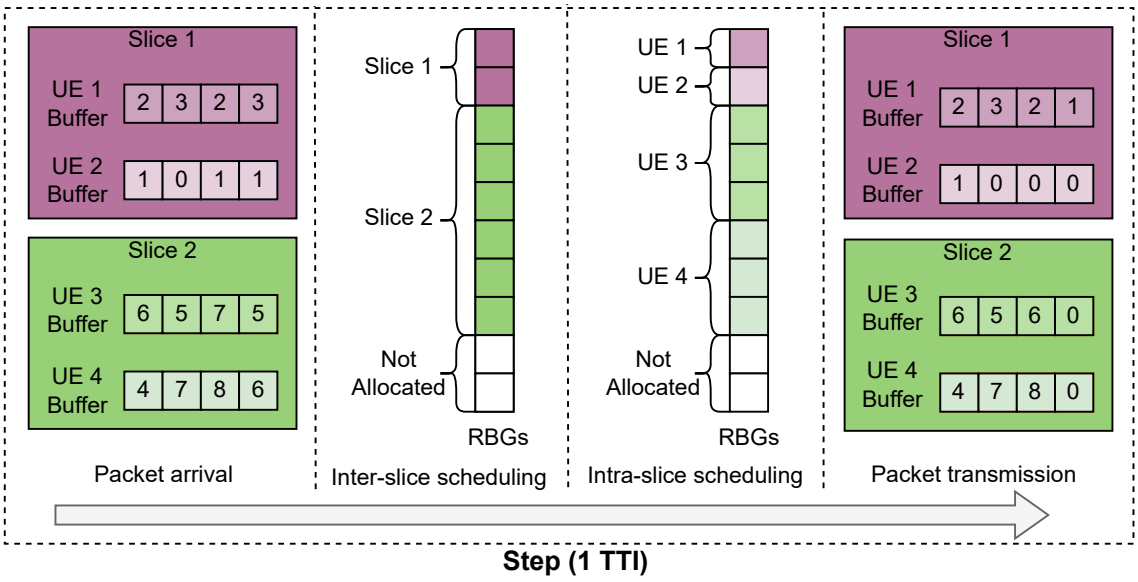


Figure 2.1: Example of one timestep in a scenario with 10 RBGs and 2 slices (2 UEs in each). One RBG is enough for a UE to send 2 packets during 1 TTI. Note that UE 1 will drop a packet when advancing the step due to reaching the maximum buffer latency (3 TTIs).

mean μ_s, where $s \in \mathcal{S}$ is the assigned slice to the UE. Our evaluation scenario considers three different slices: an eMBB slice, a URLLC slice, and a BE slice. The slices are instantiated following the parameters of Table 2.2. Each slice has a round-robin algorithm as its intra-slice scheduler, as in [Nahum et al. 2024].

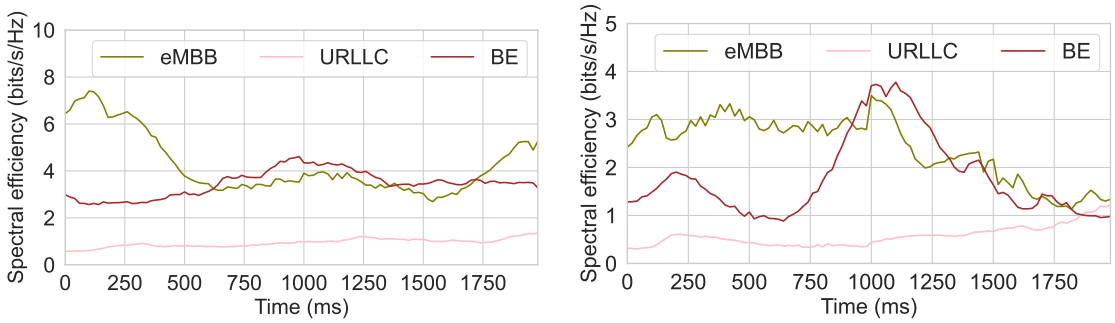
Figure 2.2 illustrates the spectral efficiency of the trial used in our evaluation scenario. Since SOA assures the SLAs of all UEs, the ones with the worst channel conditions will be allocated more RBGs. To do this, the SOA must schedule more resources to every UE in the slice, since we consider a round-robin intra-slice scheduler. Therefore, the dynamic of SOA's scheduling follows Figure 2.2(b) instead of Figure 2.2(a).

2.5.2 Baselines

To assess the quality of the SOA for the inter-slice scheduling problem, we compare its performance to two baselines: a heuristic and a state-of-the-art DRL scheduler. The first is the weighted round-robin (RR) scheduler, which distributes resources uniformly among slices based on their assigned weights. The second is the DRL agent introduced by [Nahum et al. 2024], trained using the same Soft Actor-Critic algorithm of the

Slice type	eMBB	URLLC	BE
Requirements	t^{req}_{urllc} 10 Mbps l^{req}_{embb} 20 ms p^{req}_{embb} 20%	t^{req}_{embb} 1 Mbps l^{req}_{urllc} 1 ms p^{req}_{urllc} 0.001%	f^{req}_{be} 2 Mbps g^{req}_{be} 5 Mbps
$\|\mathcal{U}_s\|$	3	3	4
Z_s	1500 bytes	500 bytes	1500 bytes
μ_s	15 Mbps	1 Mbps	15 Mbps

Table 2.2: Slice parameters.



(a) *Average* (b) *Worst*

Figure 2.2: Spectral efficiency in the evaluated scenario.

original work. Both baselines rely on weights that define the priority of each slice type, such as eMBB, URLLC, and BE. In the RR algorithm, the weights indicate the priority of slices in receiving RBGs and the proportion of the scheduling. In our scenario, the proportion in RR scheduling is 30% for URLLC, 30% for eMBB and 40% for BE, as the first two have 3 UEs and the latter has 4 UEs.

The weights used by the DRL agent, however, determine not the RBG allocation itself, but the impact of respecting the SLA requirements. In the original work [Nahum et al. 2024], for each SLA requirement, an intent-drift is calculated normalizing the distance to the required value if the requirement is not respected, otherwise, it is zero. Each intent-drift is then multiplied by a normalized weight associated with the SLA requirement. The authors consider the same five distinct SLA requirements we described in Section 2.3. We denote their weights as $\{w^t_s, w^l_s, w^p_s, w^f_s, w^g_s\}$, corresponding to the weights for served throughput, average buffer latency, packet loss rate, fifth-percentile throughput, and long-term throughput requirements of a slice s, respectively. Therefore, we use the same values from [Nahum et al. 2024]: $w^t_{eMBB} = 0.2$, $w^l_{eMBB} = 0.05$, $w^p_{eMBB} = 0.05$, $w^t_{URLLC} = 0.1$, $w^l_{URLLC} = 0.25$, $w^p_{URLLC} = 0.25$, $w^f_{BE} = 0.05$, and $w^g_{BE} = 0.05$. The agent's reward function is then defined as the summation of the product between each

intent-drift and the weight, a sum that must be minimized.

Besides the reward, the DRL agent is defined by two other components: (*i*) the action space, which represents the possible schedulings for the agent, determined in the original work as a number for each slice, and (*ii*) the observation space, which describes the state of the environment and is used as input for the agent to select the action that minimizes the intent-drift. In [Nahum et al. 2024], two observation space strategies are defined: (*i*) the limited observation space, which includes the SLA requirement values and 9 metrics for each slice, calculated as the average for its UEs, and (*ii*) the full observation space, which includes also the 9 metrics for each UE. Between the metrics are the spectral efficiency and the 5 metrics used in the intent-drift calculation. However, as highlighted by [Nahum et al. 2024], both observation spaces demonstrate similar performance. Consequently, we adopt the limited observation space to streamline the training complexity. The training process for the DRL agent follows the approach described by [Nahum et al. 2024], where we create a training dataset with 45 trials from the spectral efficiency dataset and the agent undergoes training 10 times over the 2000 timesteps in each trial. This results in a total of 900,000 training steps. One trial, not included in the training dataset, is reserved for evaluating the three schedulers.

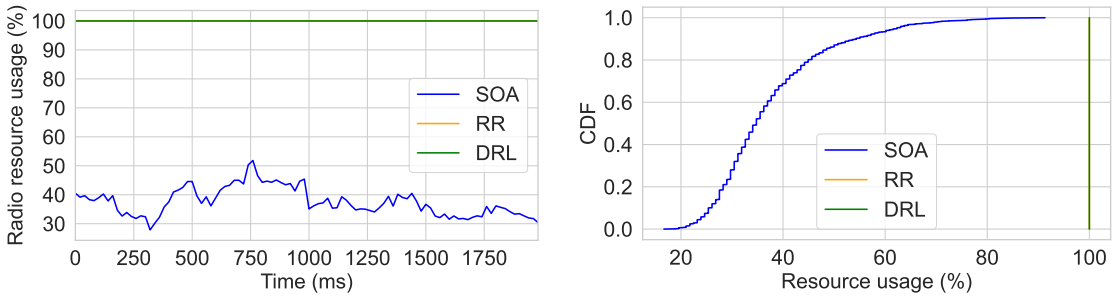
It is important to notice that the DRL agent was originally evaluated in a different scenario, where ensuring the SLA for every UE is not possible. We use a different number of UEs per slice, higher spectral efficiencies, and lower buffer sizes, and we do not consider changes in the traffic or requirements during the experiment. The number of RBGs is also higher, since we divide the resources into RBs with a lower bandwidth to follow the specifications of [ETSI 2020]. For instance, while [Nahum et al. 2024] considers only 17 RBGs, we have 138, despite summing up the same 100 MHz bandwidth. Hence, the comparisons we perform consider the scenario of our evaluation and differ from the results in [Nahum et al. 2024].

2.5.3 Results

To evaluate SOA solutions, we consider two scenarios: a standard and a limited scenario. In the standard scenario, the SOA allocates the minimum RBGs necessary, while RR and DRL schedule all available RBGs in the BS at each timestep. This is a consequence of how the two baselines are formulated to not consider minimizing resource usage. While this strategy offers lower complexity for scenarios with lower demands, it does not prioritize radio resource optimization, resulting in the usage of more resources than is strictly necessary, complicating the comparison of SOA with the baselines. Therefore, we consider a limited scenario where the number of available RBGs in the BS dynamically changes to the minimum needed to assure the SLA of all UEs

at each timestep, defined by the SOA resource usage. In this scenario, we emphasize the significance of considering RBG minimization and how it can be difficult for the baselines to ensure SLA requirements for each UE with fewer resources.

Standard scenario – In this scenario, we compare the RBG usage for all models. Figure 2.3 depicts the radio resource usage for the RR algorithm, the DRL agent, and SOA across all timesteps. It is important to observe that, while the RR algorithm and the DRL agent utilize all radio resources in every timestep (lines are overlapped at 100%), our SOA dynamically adjusts the radio resource usage based on the current demand. Consequently, we achieved an average reduction of 62% in radio resource usage over the 2000 timesteps, while the worst-case demand, around 750 ms, has a peak of 91% allocated resources.



(a) *Radio resource usage as a function of time*

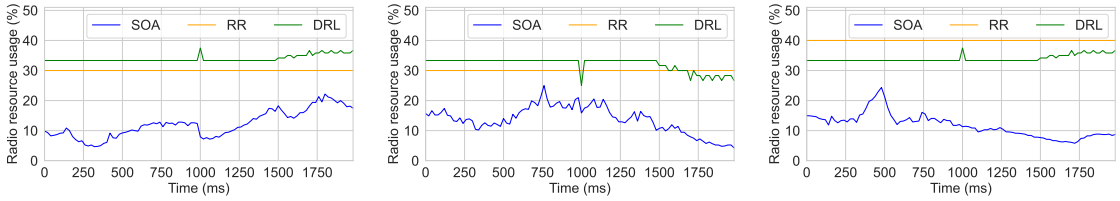
(b) *CDF of radio resource usage*

Figure 2.3: Radio resource usage of the schedulers in the standard experiment.

Although both baselines utilize all available radio resources for each timestep, it is important to note that neither method explicitly prevents the violation of slice requirements. Despite the DRL agent incorporating an intent-drift in its formulation, penalizing the agent when slice requirements are broken, this approach does not entirely eliminate the possibility of such faults occurring. To quantify SLA violations, we count the instances where UEs fail to meet the specified requirements for each slice at every timestep. For instance, if the URLLC slice receives no resource allocation throughout all 2000 timesteps, the served throughput requirement is violated for each of its 3 UEs, resulting in a total of $3 \times 2000 = 6000$ instances.

As expected, the SOA respected all SLAs throughout the 2000 timesteps. The same happened to the RR algorithm but with a less efficient resource utilization. Despite also allocating 100% of the resources, the DRL agent performs worse than RR and exhibits SLA violations during the experiment. It especially violates the served throughput requirement for the URLLC slice a total of 210 times. This is due to the set of actions chosen by the agent throughout the simulation: (*i*) equal distribution for all slices, chosen in 1930 timesteps, and (*ii*) 50% for eMBB, 0% for URLLC, and 50% for BE, chosen in 70 timesteps and thus violating $70 \times 3 = 210$ times this SLA requirement.

We also analyze the radio resource allocation for each slice during the experiment. Figure 2.4 illustrates the resource allocation for eMBB, URLLC, and BE slices, comparing the RR algorithm, the DRL agent, and our SOA. As expected, the RR algorithm exhibits a static radio resource allocation among all slices during all timesteps due to its formulation. A similar behavior is observed in the DRL agent allocation. As it chooses an equal distribution across 1930 timesteps, the overall allocated resources are mostly constant. Its only fluctuations in allocation happen around 1000 ms and after 1400 ms, when the channel conditions for eMBB and BE are worse, therefore they are prioritized. This is the reason why URLLC gets no resources from the DRL agent in some timesteps.



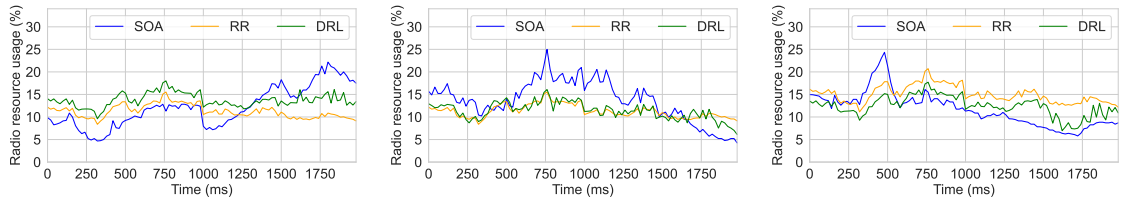
(a) *eMBB* (b) *URLLC* (c) *BE*

Figure 2.4: Allocated radio resources for each slice in the standard experiment.

Limited scenario – In this scenario, we use the SOA's minimal resource allocation represented in Figure 2.4 as the total number of RBGs available at every timestep to the RR algorithm and the DRL agent. It is noteworthy that the DRL agent was trained to use 100% resources, so it may not have learned how to solve scarce scenarios.

Figure 2.5 illustrates the radio resource allocation for each slice during this experiment. Comparing the slices, we can see that both RR and DRL allocate more RBGs to eMBB and BE than the SOA's optimal solution most of the time, while the opposite happens to URLLC. The DRL agent consistently distributes resources equally among the slices for the majority of the experiment. This allocation strategy is similar to the RR algorithm, where the allocation is determined as a constant fraction of the available RBGs at each timestep. Additionally, we observe that the SOA strategy has a highly different dynamic from the baselines. This occurs as the SOA decision is based on channel and buffer conditions, thus leading to an adaptable solution responding to the current demand.

As expected, the RR algorithm and the DRL agent solutions in scarce scenarios are worse, as shown by the SLA violations in Figure 2.6. We can see that as the DRL agent allocation for eMBB is higher than RR's, it has fewer SLA violations for the requirements of this slice. The same does not happen to BE requirements, which are more respected by the RR algorithm since the slice has a higher weight. We recall the BE intent-drifts having the lowest weights in the DRL's reward as a reason for not prioritizing it. Lastly, although the URLLC allocation is similar between the baselines, the DRL agent performs



(a) *eMBB* (b) *URLLC* (c) *BE*

Figure 2.5: Allocated radio resources for each slice in the limited experiment.

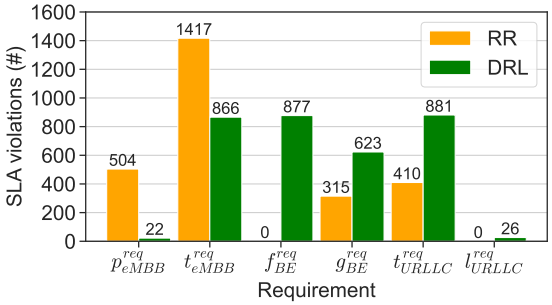


Figure 2.6: SLA violations in the limited experiment. SOA has no SLA violation.

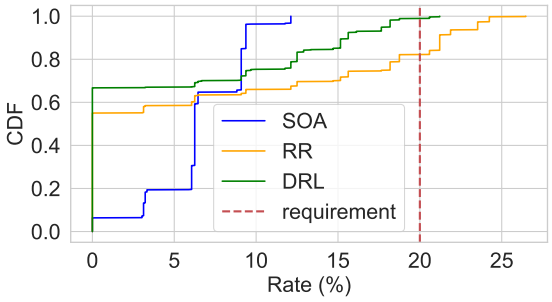


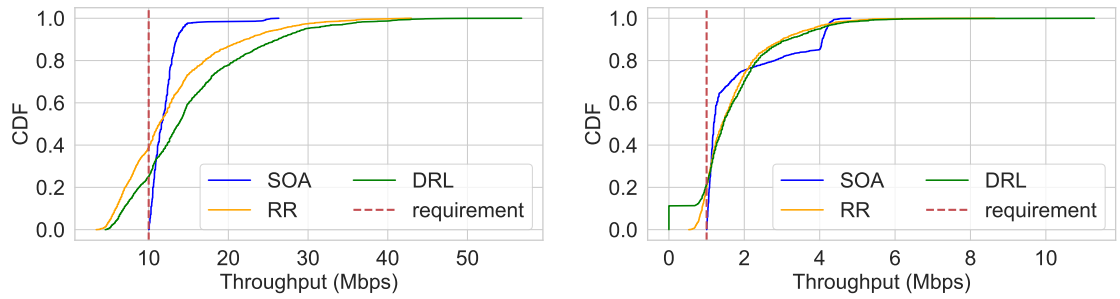
Figure 2.7: CDF of worst packet loss rate for eMBB in the limited experiment.

worse. This happens due to the DRL agent scheduling zero resources to the slice in a few timesteps.

As eMBB has larger packets and a higher demand than URLLC, if high throughput is not achieved, it leads to high packet losses. Figure 2.7 illustrates the CDF for the worst eMBB packet loss, calculated as the maximum packet loss for an eMBB UE in each timestep. We note that the DRL agent maintains zero packet loss most of the time due to over-provisioning, but violates the requirement when the channel quality is low.

Figure 2.8 depicts the worst served throughput for eMBB and URLLC. Again, the DRL agent shows better performance than the RR algorithm regarding the eMBB slice. However, we can see that URLLC has a throughput of 0 Mbps for the DRL scheduling during a considerable portion of the simulation. This is explained by its set of selected actions: (*i*) equal distribution for the three slices, (*ii*) 100% for eMBB, (*iii*) half for eMBB and half for BE, and (*iv*) half for eMBB and half for URLLC. An option where no resource is scheduled to URLLC and BE is selected in 11% and 4%, where eMBB always receives at least 33% of the available RBGs.

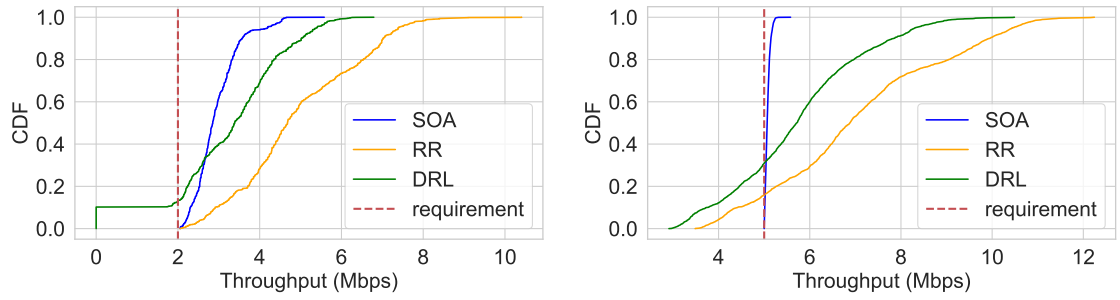
The BE performance is represented by Figure 2.9. Since the fifth-percentile throughput is calculated as the minimum throughput for our time window of 10 timesteps, scheduling no RBGs to BE impacts the actual timestep and the next 9 ones. This explains why the DRL agent has a value of 0 Mbps for the metric almost 10% of the time. The long-



(a) *eMBB* (b) *URLLC*

Figure 2.8: CDF of worst served throughput in the limited experiment.

term throughput does not achieve such low values as it is more tolerant to fluctuations in allocation. This metric also exemplifies how SOA respects the SLA of each UE while reducing resource usage, as it allocates exactly the required value at every timestep.



(a) *Fifth-percentile throughput* (b) *Long-term throughput*

Figure 2.9: CDF of worst metrics for BE in the limited experiment.

Guidelines for RL solutions – A critical aspect of RL methods is designing an efficient reward function. Defining a constant weight for summing different intent-drift may be a solution for prioritizing slices, but it also may lead to SLA violations. For example, when the DRL agent allocates half resources for eMBB and half for BE, it chooses to disrespect a requirement of weight $w^t_{URLLC} = 0.2$ while assuring the requirements of weight $w^g_{BE} + w^t_{eMBB} = 0.25$. Another crucial aspect of RL solutions is defining the observation space. Using the average spectral efficiency for a slice may invisibilize the heterogeneous channel conditions of UEs in a slice. A scenario with two UEs u_1 and u_2 where $E_{u_1} = 0.5$ and $E_{u_2} = 5.5$ bits/s/Hz will be equivalent to another where $E_{u_1} = E_{u_2} = 3$ bits/s/Hz, as both scenarios have the same average spectral efficiency. Lastly, much of the resource usage can be reduced if the agent is designed not to allocate every RBG, as seen in SOA's allocation. An improvement can be made in the DRL agent by adding an action value for the unused resources and then adding it as part of the reward function.

2.6 Conclusions and future works

In this work, we presented the stepwise optimal inter-slice RRS for SLA assurance problem. We proposed SOA, a polynomial algorithm that solves this problem by ensuring the SLA requirements of each UE and minimizing resource allocation. Our solution is used to evaluate a state-of-the-art DRL scheduler in a scenario where every SLA can be ensured. We showed that the ML approach may fail in ensuring the SLA of every UE even with a less efficient resource utilization. Moreover, the SOA achieves zero SLA violations while reducing resource usage by an average of 62%. We expect that new ML solutions could leverage our strategy as a baseline to improve their approaches. Lastly, our future works include expanding the scenarios where SOA can be used to address competition and prioritization when resources are scarce.

2.7 Acknowledgements

This work was supported by CAPES, MCTIC/CGI.br/São Paulo Research Foundation (FAPESP) through the Project Smart 5G Core And MUltiRAn Integration (SAMURAI) under Grant 2020/05127-2, by CNPq through the Project Universal under Grant 405111/2021-5, by RNP/MCTIC, Grant No. 01245.010604/2020-14, under the 6G Mobile Communications Systems project, and Program OpenRAN@Brasil.

CHAPTER 3

DREAMIN: Channel-Aware Inter-Slices Radio Resource Scheduling for Efficient SLA Assurance

3.1 Abstract

Network slicing addresses Quality of Service (QoS) needs through efficient inter-slice Radio Resource Scheduling (RRS) to meet Service Level Agreements (SLAs) of each slice. Given the independence of each slice's scheduler and the discrepancy in channel quality among users and resources in different frequencies, an effective inter-slice RRS must predict intra-slice RRS and be channel-aware. In plentiful scenarios, where all SLAs are assured, RRS can also improve energy efficiency by reducing allocated resources, while in scarce scenarios, where disrespecting SLA is inevitable, we can reduce the SLA drift (SLAd), a metric quantifying SLA non-compliance. In this context, we formulate a constraint programming problem for channel-aware RRS oriented to SLAd and resource usage minimization and propose the Drift and Resource Allocation Minimization (DREAMIN) scheduler as a scalable approximation of the solution. Our simulated results show DREAMIN outperforming the state-of-the-art RadioSaber (RS) reducing the resource usage by 62% in the plentiful scenario and lowering by 62% the SLAd in the scarce scenario. DREAMIN also has 14 times more occurrences of high intra-slice fairness indices than RS in the scarce scenario, demonstrating how schedulers oriented to minimizing SLAd tend to be fairer than others maximizing total capacity.

3.2 Introduction and Related Work

5G and Beyond brought an increasing diversity of services supported by the radio access network (RAN). In this broad scenario, network slicing is critical to meet quality of service (QoS) requirements by effectively scheduling resources among slices, i.e., isolated logical networks on top of a shared RAN infrastructure serving user equipments (UEs)

with similar applications. These resources could be CPU or memory for running RAN functions, but the radio resources are the ones directly affecting UEs as they dictate the achievable capacity. Radio resources are discretized into resource blocks (RBs) consisting of a specific frequency range allocated to a UE during one transfer time interval (TTI) and are grouped into resource block groups (RBGs) to reduce scheduling complexity. Therefore, solving the inter-slice scheduling problem of allocating RBGs among slices is key to achieving QoS in scenarios of high service diversity. Moreover, the problem depends on the intra-slice scheduler since it distributes the RBGs received by the slice to its assigned UEs.

Two aspects must be considered on the inter-slice scheduling to improve efficiency [Chen et al. 2023]. First, the scheduling must be channel-aware, accounting for varying channel qualities among RBGs to the same UE caused by frequency-selective fading, so it not only defines the number of RBGs each slice receives but also selects them. Second, the inter-slice scheduling must predict the intra-slice scheduling to appropriately select RBGs, knowing which UEs will receive them. Most works of the literature map QoS to a service level agreement (SLA), which can be expressed as a list of network metrics and their required values. Therefore, instead of always allocating all available resources to maximize or minimize metrics, leading to overprovisioning, a scheduler can satisfy SLAs and save the remaining RBGs, which may improve energy efficiency.

Related work – Table 3.1 compares this paper with recent works on solving radio resource scheduling in network slicing scenarios. The approach in [Nahum et al. 2024] leverages the translation of natural language network configuration descriptions, called intents, into SLAs to define intent drift: the normalized difference between the SLA metric and its desired value. In this work, we refer to intent drift as SLA drift (SLAd) to broaden the term. Although SLA assurance can be a binary objective, quantifying it with SLAd allows us to differentiate schedules in scenarios where SLAs cannot be fully ensured. While [Nahum et al. 2024] develops a deep reinforcement learning (DRL) agent to minimize SLAd using all available resources, our previous work [Campos et al. 2024] introduces a linear algorithm to achieve zero SLAd and improve energy efficiency by minimizing resource usage in plentiful scenarios.

The work in [Chen et al. 2023] explores the advantages of a channel-aware inter-slice scheduler, which not only decides the number of RBGs allocated for each slice but also selects the RBGs. This can only be done if the intra-slice scheduler is greedy, so its allocation is predictable. The proposed solution is RadioSaber, a state-of-the-art algorithm for maximizing capacity by selecting the RBG-slice allocation with the highest capacity at each iteration until all RBGs are allocated. However, the RBGs quota for each slice follows a static proportion defined as the slice weight in the base station (BS), not adapting to the dynamic channel conditions.

Table 3.1: Related work.

Works	Problem formulation	Optimized model	Dynamic slice resource proportion	Slice priorities	Channel-aware[1]	Minimize resources	SLAd-oriented	Intra-slice fairness
[Chen et al. 2023]	○	○	○	●	●	○	○	○
[Nahum et al. 2024]	●	○	●	●	○	○	●	○
[Campos et al. 2024]	●	●	●	○	○	●	○	○
[Rana et al. 2024]	●	●	●	●	●	○	○	○
[Li et al. 2024]	●	●	●	●	○	◐	◐	○
[Boutiba et al. 2022]	○	○	●	○	◐	○	○	○
[Boutiba et al. 2023]	●	●	●	○	◐	○	○	○
[Dai et al. 2024]	●	○	●	○	○	○	○	○
[Cheng et al. 2024]	○	○	●	○	○	○	○	○
[Navidan et al. 2024]	●	●	●	●	○	○	○	○
This work	●	●	●	●	●	●	●	●

[1]Works considering differences in achievable capacity only between RBs of different numerologies are half-filled.

A linear complexity knapsack algorithm approximates a chance-constrained optimization problem formulated in [Rana et al. 2024] to maximize the priorities of delivered packets by scheduling RBGs for each packet individually. A hierarchical framework is proposed in [Li et al. 2024] to maximize the SLA satisfaction rate by a multi-arm bandit algorithm selecting one from a set of scheduling actions generated with a deep neural network. The works in [Boutiba et al. 2022] and [Boutiba et al. 2023] approach multi-slice mixed numerology scenarios where the same UE is assigned to multiple slices and the available RBGs in the BS have different numerologies. Open-RAN compliant testbeds are used in the evaluations of [Dai et al. 2024], [Cheng et al. 2024], and [Navidan et al. 2024], which implement inter-slice schedulers as xApps, microservice applications optimizing the RAN that run on near real-time RAN intelligent controllers.
Our contributions – We solve the problem of channel-aware inter-slice scheduling oriented to minimizing SLAd and resource allocation. Our main contributions are:

- An optimization model for scheduling RBGs between slices throughout multiple TTIs by dynamically adapting to changes in channel conditions through time.
- DREAMIN: a greedy scalable approximation to the optimal solution selecting the allocations that most reduce the overall SLAd.
- An evaluation of DREAMIN compared to the state-of-the-art RadioSaber scheduler [Chen et al. 2023], allocating 62% less resources in plentiful scenarios and, in scarce scenarios, 62% lower SLAd and more than 14 times the TTIs with high intra-slice fairness index.
- An analysis of how channel-aware inter-slice schedulers impact intra-slice fairness and how minimizing SLAd tends to higher fairness indexes.

Paper organization – Sections 3.3 and 3.4 present the system model and the problem formulation. Section 3.5 explains how DREAMIN works and its complexity. We evaluate DREAMIN compared to RadioSaber in multiple simulated scenarios and discuss their results in Section 3.6. Finally, Section 3.7 contains our conclusions and future works.

3.3 System model

We model a downlink scenario where the UEs' buffers at the BS receive packets from the user plane function (UPF) and transmit them to the users. A TTI begins with packets arriving at the buffers, followed by radio resource scheduling and packet transmission based on the achieved capacity. We assume the only channel information input into the schedulers is Channel Quality Indicators (CQIs), which can be mapped to achievable capacity values (in bits/second). This abstracts UE mobility and leads to channel-model-independent solutions.

We define $\mathcal{S} = \{s_1, s_2, \ldots\}$ as the set of instantiated slices and $\mathcal{U}_s = \{u_j, u_{j+1}, \ldots\}$ as the set of UEs assigned to a slice $s \in \mathcal{S}$. The set $\mathcal{U} = \bigcup_{s\in\mathcal{S}} \mathcal{U}_s$ is used when referring to all UEs in the BS. Note we consider each user as assigned to only one slice, thus $\bigcap_{s\in\mathcal{S}} \mathcal{U}_s = \emptyset$. Our system model discretizes time as the sequence of TTIs $\mathcal{T} = \{0, \ldots, |\mathcal{T}| - 1\}$. The set $\mathcal{R} = \{r_1, \ldots, r_{|\mathcal{R}|}\}$ contains all RBGs in the BS. The output of the problem we define in Section 3.4 is the binary decision variable $\rho^u_{r,t} \in \{0, 1\}$, which indicates if RBG $r \in \mathcal{R}$ is allocated to UE $u \in \mathcal{U}$ at TTI $t \in \mathcal{T}$. We denote by $C^u_{r,t}$ the achievable capacity for RBG r if allocated to UE u at TTI t, calculated based on the respective informed CQI. The total capacity achieved by u at t is then defined as:

$$c^u_t = \sum_{r\in\mathcal{R}} \rho^u_{r,t} \cdot C^u_{r,t}. \tag{3-1}$$

Buffer modeling – We assume that each user u in each slice s has a buffer receiving A^u_t packets at the beginning of TTI t. Depending on the allocated capacity for u at t, we calculate k^u_t as the integer number of packets from u's buffer delivered at TTI t. Two types of packet dropping are also considered in the buffer model. First, df^u_t counts how many packets arrived at u's buffer at TTI t but were instantly dropped for surpassing its maximum capacity Z_u (in packets). Second, dl^u_t accounts for the packets dropped at TTI t after achieving a maximum tolerated latency L_u (in TTIs) and not being transmitted. Considering arrived, transmitted, and dropped packets, we define the total amount of packets in u's buffer at the beginning of TTI t, before any packet arrives, as:

$$b^u_t = \begin{cases} 0, & \text{if } t = 0 \\ b^u_{t-1} + A^u_{t-1} - k^u_{t-1} - df^u_{t-1} - dl^u_{t-1}, & \text{if } t > 0 \end{cases}. \tag{3-2}$$

The number of delivered packets in each TTI is limited by the allocated capacity for the UE and how many packets are in their buffer. As we discretize time in TTIs, we must consider that a packet can start being delivered in one TTI and finish in another. Discarding this possibility would result in a high reduction in capacity, e.g.: if the capacity allocated in every TTI can only transmit 0.9 packets, the k^u_t calculation would floor it to zero in all TTIs, so packets are never delivered. Therefore, we express partially delivered packets by $\bar{k}^u_t$. To avoid counting packets no longer in the buffer, we set $\bar{k}^u_t$ as zero if all packets are delivered or if a packet is lost due to latency. Considering $\bar{k}^u_{-1} = 0$, l as the TTI length and P_u the packet size for UE u, we define k^u_t and $\bar{k}^u_t$ as:

$$k^u_t = \min\left(\left\lfloor \frac{c^u_t \cdot l}{P_u} + \bar{k}^u_{t-1} \right\rfloor, b^u_t + A^u_t - df^u_t \right), \tag{3-3}$$

$$\bar{k}^u_t = \begin{cases} 0, & \text{if } b^u_{t+1} = 0 \text{ or } dl^u_t > 0 \\ \frac{c^u_t \cdot l}{P_u} - k^u_t, & \text{otherwise} \end{cases}. \tag{3-4}$$

We express as $p^u_{t,l}$ the number of packets waiting for l TTIs at the moment t, i.e., arrived at $t-l$, and will still be in the buffer at the end of t. Assuming the buffer always delivers older packets first, we subtract how many packets arrived or were already in the buffer at $t-l$ by how many of those were delivered or dropped:

$$p^u_{t,l} = \max\Big(0, b^u_{t-l} + A^u_{t-l} - df^u_{t-l} - \sum_{t'=t-l}^{t} k^u_{t'} - \sum_{t'=t-l}^{t-1} dl^u_{t'}\Big), \tag{3-5}$$

which uses a max function with a zero because the expression is negative when newer packets, i.e., those that arrived after $t-l$, are also delivered. Since we drop packets that were not delivered after waiting for L_u TTIs due to latency, we can define $dl^u_t = p^u_{t,L_u}$. Lastly, the number of packets dropped due to buffer full can be defined similarly using a max function:

$$df^u_t = \max(0, b^u_t + A^u_t - Z_u). \tag{3-6}$$

SLA metrics – We consider three metrics when defining SLAs. The first is the instantaneous capacity, denoted for a user u at TTI t as $CAP^u_t = c^u_t$. Second, the long-term capacity (LTC), denoted by LTC^u_t, is the average capacity in a window of TTIs. The window size AW_t adjusts the fixed window TW for each TTI t so TTIs before $t = 0$ are not considered.

$$LTC^u_t = \frac{\sum_{t'=t-AW_t+1}^{t} c^u_{t'}}{AW_t}, \tag{3-7}$$

$$AW_t = \begin{cases} t+1, & \text{if } t < TW \\ TW, & \text{if } t \geq TW \end{cases}. \tag{3-8}$$

Lastly, the latency LAT^u_t is calculated at each TTI t for a user u associated with slice s as the number of TTIs spent by the oldest remaining packet in the buffer. We define $\beta^u_{t,l}$ as a variable indicating $p^u_{t,l} > 0$ and express LAT^u_t as:

$$LAT^u_t = \max_{l \in 0,\ldots,L_u} (l \cdot \beta^u_{t,l}). \tag{3-9}$$

3.4 Problem formulation

This section defines the optimization model for the inter-slice radio resource scheduling problem. As one resource block is allocated to at most one user in a TTI, we restrict:

$$\sum_{u \in \mathcal{U}} \rho^u_{r,t} \leq 1, \qquad \forall r \in \mathcal{R}, t \in \mathcal{T}. \tag{3-10}$$

Intra-slice scheduling – We follow [Chen et al. 2023] in predicting the intra-slice scheduling by adding restrictions to mimic an independent proportional fair (PF) scheduler for each slice. The PF is a score-based scheduler allocating each RBG to the UE with the highest score, defined as the ratio between capacity and historical capacity to maximize throughput while ensuring fairness. The historical capacity h_t^u is an exponential weighted moving average calculated for the window TW assuming a given starting historical capacity y_u for the user u at the first TTI. Therefore, we define the PF score for allocating the RBG r to the user u at TTI t as $\frac{C_{r,t}^u}{h_t^u}$ and restrict the selected allocation as the one maximizing the score in the slice:

$$h_t^u = \begin{cases} y_u, & \text{if } t = 0 \\ (1 - \frac{1}{TW})h_{t-1}^u + \frac{1}{TW}c_{t-1}^u, & \text{if } t > 0 \end{cases}. \tag{3-11}$$

$$\sum_{u' \in \mathcal{U}_s} \frac{C_{r,t}^{u'}}{h_t^{u'}} \cdot \rho_{t,r}^{u'} \geq \frac{C_{r,t}^u}{h_t^u} \cdot \sum_{u' \in \mathcal{U}_s} \rho_{t,r}^{u'}, \qquad \forall s \in \mathcal{S}, u \in \mathcal{U}_s, r \in \mathcal{R}, t \in \mathcal{T}. \tag{3-12}$$

Note that $\sum_{u' \in \mathcal{U}_s} \rho_{t,r}^{u'} \in \{0,1\}$ indicates if the RBG r is assigned to the slice s since each RBG is allocated to at most one UE. Hence, the constraint becomes $0 \geq 0$ when r is not allocated to the slice s.

SLAd – We represent the SLA for each slice s as a required value Q_s^m for each metric $m \in \mathcal{M}_s \subseteq \{CAP, LTC, LAT\}$, representing the capacity, LTC, and latency requirements, respectively. For each slice s, TTI t, UE $u \in \mathcal{U}_s$, and metric $m \in \mathcal{M}_s$, we calculate the SLAd $f_{t,s}^{m,u}$ as zero if the requirement is achieved, otherwise, it is a number between zero and one proportional to how distant the metric is from the required:

$$f_{t,s}^{CAP,u} = \begin{cases} \frac{Q_s^{CAP} - CAP_t^u}{Q_s^{CAP}}, & \text{if } CAP_t^u < Q_s^{CAP} \\ 0, & \text{if } CAP_t^u \geq Q_s^{CAP} \end{cases}, \tag{3-13}$$

$$f_{t,s}^{LTC,u} = \begin{cases} \frac{Q_s^{LTC} - LTC_t^u}{Q_s^{LTC}}, & \text{if } LTC_t^u < Q_s^{LTC} \\ 0, & \text{if } LTC_t^u \geq Q_s^{LTC} \end{cases}, \tag{3-14}$$

$$f_{t,s}^{LAT,u} = \begin{cases} \frac{LAT_t^u - Q_s^{LAT}}{L_u - Q_s^{LAT}}, & \text{if } LAT_t^u > Q_s^{LAT} \\ 0, & \text{if } LAT_t^u \leq Q_s^{LAT} \end{cases}. \tag{3-15}$$

The SLAd is aggregated assuming normalized weights: W_s^m represents the importance of the metric m for slice s, while W_s can be seen as the price of slice s. We consider every UE within the same slice as equally important. Therefore, for each TTI t, we define $f_{t,s}^u$ as the UE SLAd for $u \in \mathcal{U}_s$, $f_{t,s}$ as the slice SLAd for $s \in \mathcal{S}$, and f_t as the total SLAd in the BS:

$$f_{t,s}^{u} = \sum_{m \in \mathcal{M}_s} f_{t,s}^{m,u} \cdot W_s^m, \tag{3-16}$$

$$f_{t,s} = \frac{1}{|\mathcal{U}_s|} \sum_{u \in \mathcal{U}_s} f_{t,s}^{u}, \tag{3-17}$$

$$f_t = \sum_{s \in \mathcal{S}} f_{t,s} \cdot W_s. \tag{3-18}$$

Objective function – The primary objective of the scheduler is maximizing QoS by minimizing SLAd, while the secondary objective is minimizing resource usage if, and only if, it achieves zero SLAd. To distinguish a scarce scenario, where SLAd is inevitable, from a plentiful scenario, where we can minimize resource allocation, we define a binary indicator variable a_t for each TTI t.

$$a_t = \begin{cases} 0, & \text{if } f_t = 0 \\ 1, & \text{if } f_t > 0 \end{cases}. \tag{3-19}$$

We formulate the problem of channel-aware inter-slice radio resource scheduling for jointly minimizing SLAd and resource usage as:

$$\underset{\rho_{r,t}^{u}}{\text{minimize}} \quad \sum_{t \in \mathcal{T}} \left(a_t(1 + f_t) + (1 - a_t) \frac{1}{|\mathcal{R}|} \sum_{u \in \mathcal{U}} \sum_{r \in \mathcal{R}} \rho_{r,t}^{u} \right) \tag{3-20}$$

subjected to Equations (3-10) and (3-12).

Note how $(1 + f_t)$ is always higher than the normalized resource allocation, thus there is no TTI where the solution neglects SLAs to reduce resource allocation.

3.5 Proposed solution

In order to approximate the optimal solution in a scalable manner, we introduce the Drift and REsource Allocation MINimization (DREAMIN) scheduler. DREAMIN behaves as RadioSaber [Chen et al. 2023]: at each TTI, iterates through all possible allocations, predicting the intra-slice scheduling, to select the one that most reduces the overall SLAd, until there are no more available RBGs. The SLAd is minimized by choosing the allocation with the highest UE SLAd reduction multiplied by the UE and slice weights, and the resource usage is minimized by stopping the allocation once all SLAs are satisfied, i.e., the overall BS SLAd is zero. As each allocation impacts a single UE u and the BS SLAd is $f_t = \sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s} f_{t,s}^{u} \cdot W_s \cdot \frac{1}{|\mathcal{U}_s|}$, we can solely evaluate the reduction on the u's SLAd $f_{t,s}^{u} \cdot W_s \cdot \frac{1}{|\mathcal{U}_s|}$.

The Algorithm 3.1 describes our approach relying on two functions: intra_sched(s, r) returns the user u that will receive the RBG r if it is allocated to the slice s and; slad_for_cap(u, c) calculates the SLAd at the current TTI for the user u if it has a capacity c. Both can be executed in constant time if we assume that *(i)* the intra-slice scheduler scores are already calculated, and *(ii)* the time window for the LTC metric in Equation (3-7) and the maximum latency for the LAT metric in Equation (3-9) are also constant. As the worst-case scenario happens when every RBG has to be allocated, the number of evaluated allocations is upper-bounded by $\sum_{i=1}^{|\mathcal{R}|} i \cdot |\mathcal{S}| = \frac{|\mathcal{S}| \cdot |\mathcal{R}| \cdot (|\mathcal{R}|+1)}{2}$. Therefore, DREAMIN's time complexity is $O(|\mathcal{R}|^2 \cdot |\mathcal{S}|)$. Since [Chen et al. 2023] has the same complexity, we expect DREAMIN to run in real-time if implemented in an efficient language with parallelization.

Algorithm 3.1: DREAMIN allocation process.

Data: $\mathcal{R}, \mathcal{S}, \mathcal{U}, C^u_{r,t}, t$
Result: RBGs allocated for each slice $s \in \mathcal{S}$

for $s \in \mathcal{S}$, **do**
  allocation[s] $\leftarrow \emptyset$
end
for $u \in \mathcal{U}$ **do**
  cap[u] $\leftarrow 0$
  slad[u] $\leftarrow$ slad_for_cap(u, 0)
end
available_rbgs $\leftarrow R$
while |available_rbgs| > 0 **do**
  if $\sum_{u \in \mathcal{U}}$ slad[u] = 0 **then**
    return allocation
  end
  best_reduction $\leftarrow 0$
  for $r \in$ available_rbgs **do**
    for $s \in \mathcal{S}$ **do**
      $u \leftarrow$ intra_sched(s, r)
      reduction $\leftarrow$ slad[u] - slad_for_cap(u, cap[u] + $C^u_{r,t}$)
      if best_reduction $<$ reduction / $|\mathcal{U}_s|$ * W_s **then**
        best_reduction $\leftarrow$ reduction / $|\mathcal{U}_s|$ * W_s
        $u^* \leftarrow u$
        $r^* \leftarrow r$
        $s^* \leftarrow s$
      end
    end
  end
  cap[u^*] $\leftarrow$ cap[u^*] + $C^{u^*}_{r^*,t}$
  slad[u^*] $\leftarrow$ slad_for_cap(u^*, cap[u^*])
  available_rbgs $\leftarrow$ available_rbgs $\setminus \{r^*\}$
  allocation[s^*] $\leftarrow$ allocation[s^*] $\bigcup \{r^*\}$
end
return allocation

3.6 Evaluation

We compare DREAMIN's performance in a trace-driven simulation with the approximated (APPR) solution, the state-of-the-art channel-aware RadioSaber (RS) scheduler [Chen et al. 2023], and a Weighted Round-Robin (WRR). All code is available on GitHub[2]. At each TTI, packets arrive from the UPF into the UEs' buffers following a Poisson distribution with a mean equal to their demanded throughput (in bits/second). The schedulers distribute RBGs among the slices, which are allocated to the UEs by PF schedulers. Finally, the packets in each UE's buffer are transmitted using the allocated RBGs.

The capacity per RBG for each UE throughout the TTIs is based on the CQI traces dataset from [Chen et al. 2023], which extends 20MHz LTE traces from LTScope [Xie, Yi e Jamieson 2020] to 100 MHz 5G traces. There are CQI values for 158 UEs collected throughout 475 TTIs for 512 RBs, reported with the granularity of 4 RBs. Numerology 0 is used, hence the TTI length is 1 ms, and an RB spans 180 KHz with a subcarrier interval of 15 KHz. Each CQI is mapped to a reachable capacity following the spectral efficiency and code rate values from the 3GPP's CQI table[3].

We follow 3GPP's standardized 5G Quality Indicator (5QI) to QoS definitions[4] to specify the SLA of three slices in our scenario, one for each 5QI resource type: Guaranteed Bit Rate (GBR), Non-GBR, and Delay-Critical GBR (DC-GBR). Since GBR and DC-GBR require guaranteed bitrates, their SLA capacity requirement is instantaneous. On the other hand, the Non-GBR has a more tolerant requirement of LTC. The values for capacity and LTC requirements are defined as the slice demands, while the latency requirement is the Packet Delay Budget (PDB) for the radio interface[4]. Since 3GPP assumes DC-GBR packets surpassing the PDB as dropped, we do not define a latency requirement as every packet in the buffer would respect it. We use 256 Kbits as the buffer size for every UE in the BS and $TW = 10$ TTIs as the time window to calculate the LTC and the historical capacity used by the intra-slice schedulers. Table 3.2 summarizes the configurations of each slice.

We evaluate four scenarios: small-scale plentiful, small-scale scarce, large-scale plentiful, and large-scale scarce. In plentiful scenarios, zero SLAd can be achieved during most of the TTIs by respecting every SLA requirement, so DREAMIN's goal is to reduce resource allocation. In scarce scenarios, SLAd is inevitable for almost all TTIs, thus DREAMIN's focus is to reduce SLAd per TTI. Due to the limitations of optimally solving the scheduling problem, we define simplified small-scale scenarios with fewer

[2] github.com/LABORA-INF-UFG/paper-DGMK-2024
[3] Table 5.2.2.1-2, 3GPP TS 38.214 v18.4.0 (2024-09-23)
[4] Table 5.7.4-1, 3GPP TS 23.501 v19.1.0 (2024-09-24)

users, TTIs, and RBGs, to effectively approximate the optimal solution. To maintain total bandwidth when limiting the number of RBGs, we increase the RBG size, i.e. the number of RBs in one RBG, in the same proportion. In the small-scale scenario, we compare how close the schedulers are to the APPR solution, while the large-scale scenario evaluates how DREAMIN can improve performance in a more realistic setting. For each scenario, we run 20 simulations with different sets of randomly selected users and ensemble the results of all their TTIs to plot graphs.

Table 3.2: Slice configurations.

Slice	**GBR**	**Non-GBR**	**DC-GBR**
5QI	2	80	86
Service	Conversational video	Augmented reality	V2X messages
Demand/UE	12 Mbps[5]	50 Mbps[6]	10 Mbps[7]
Packet size P_u	256 bytes	1024 bytes	128 bytes
Max. latency L_u	200 ms	100 ms	3 ms
Weight W_s	0.333	0.333	0.333
Q_s^{CAP} (W_s^{CAP})	12 Mbps (0.5)	-	10 Mbps (1.0)
Q_s^{LTC} (W_s^{LTC})	-	50 Mbps (0.5)	-
Q_s^{LAT} (W_s^{LAT})	130 ms (0.5)	8 ms (0.5)	-

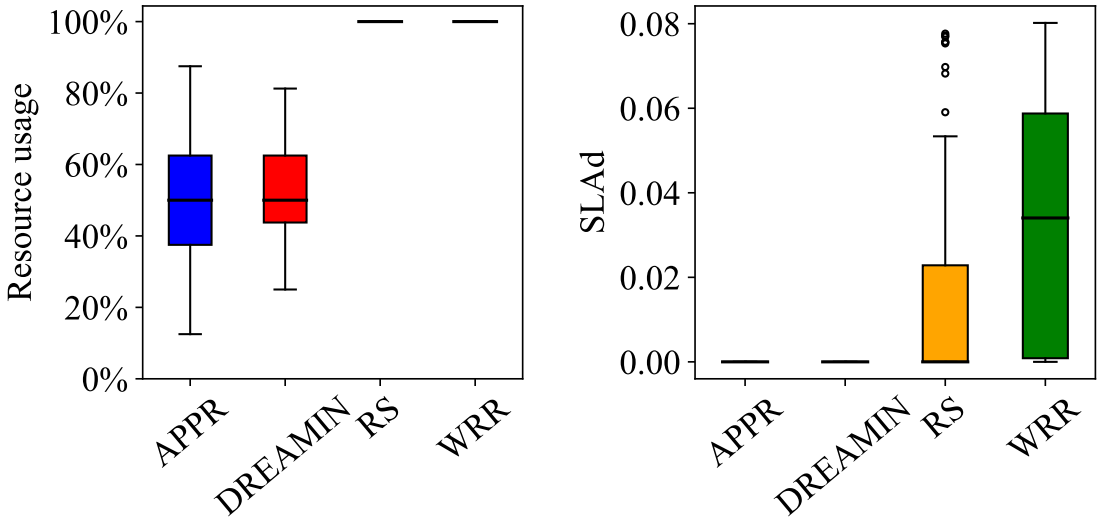
3.6.1 Small-scale scenarios

The problem from Section 3.4 is implemented using the Docplex Python library to solve it using the IBM CPLEX Constraint Programming Optimizer. All evaluations are done on a machine with an Intel i7-1255U (12 cores, 4.7 GHz), 40 GB of RAM, and Ubuntu 20.04. We limit the number of TTIs to 10 and set the RBG size as 32 RBs, totaling 16 available RBGs at the BS. To maintain the scale, both plentiful and scarce scenarios have three UEs, one per slice, but differ in SLA requirements. The requirements in Table 3.2 are enough for the plentiful scenario, whereas the scarce one is achieved by multiplying the capacity and LTC requirements by three and dividing the latency requirements by three. However, confirming the optimality of the solver's solution can take days even at this scale, thus we limit the solver time to one hour and evaluate the best-found solution.

[5]Full-HD bitrate at Section 5.1, 3GPP TR 26.925 v18.1.0 (2024-01-05)

[6]Interactive streaming applications downlink demand at Figure 3, https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/future-network-requirements-for-xr-apps

[7]Coop. collision avoid., Table 5.3-1, 3GPP TS 22.186 v18.0.1 (2024-04-05)



(a) *Resource usage distribution.* (b) *SLAd distribution.*

Figure 3.1: Boxplots for the small-scale plentiful scenario evaluation.

Figure 3.1 depicts the small-scale plentiful scenario results, where the APPR and DREAMIN schedulers achieve zero SLAd for all TTIs while allocating, respectively, an average of 50% and 52% resources, as shown in Figure 3.1(a). The same does not occur for RS and WRR, which disrespect SLAs at some TTIs despite always allocating 100% resources, as in Figure 3.1(b). This is due to RS and WRR following a non-channel-aware static proportion of resources for each slice, preventing slices from getting more resources when in worse channel conditions. For example, the most demanding slice, Non-GBR, receives 33% resources at every TTI from RS and WRR, while the APPR and DREAMIN allocate a similar value on average, but dynamically adapted.

Figure 3.2 shows the CDF for SLAd along all TTIs of the small-scale scarce scenario, where all schedulers allocate 100% resources. Note how there are two shapes of curves depending on whether the scheduler follows a static or dynamic resource proportion: DREAMIN approximates APPR's SLAd, while RS's higher values resemble WRR's. It is important to note that, due to its maximum capacity objective, RS achieves a higher total capacity than any other scheduler. Nevertheless, this does not benefit the SLAd metric since overprovisioning users have the same zero SLAd value as allocating only enough resources to fulfill all SLA requirements.

3.6.2 Large-scale scenarios

In the large-scale evaluation, we set the number of TTIs as 475 and the RBG size as 4 RBs, totaling 128 RBGs. Since there is no need to maintain scale, the scenarios

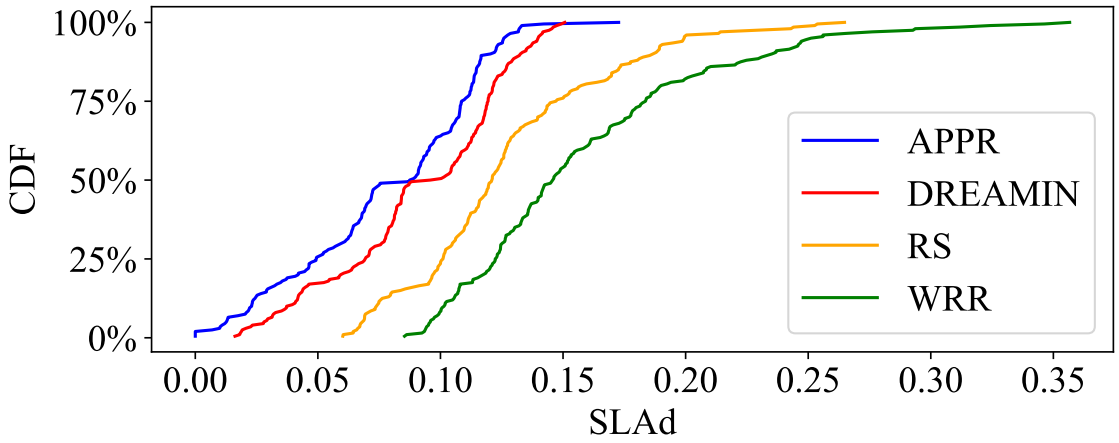
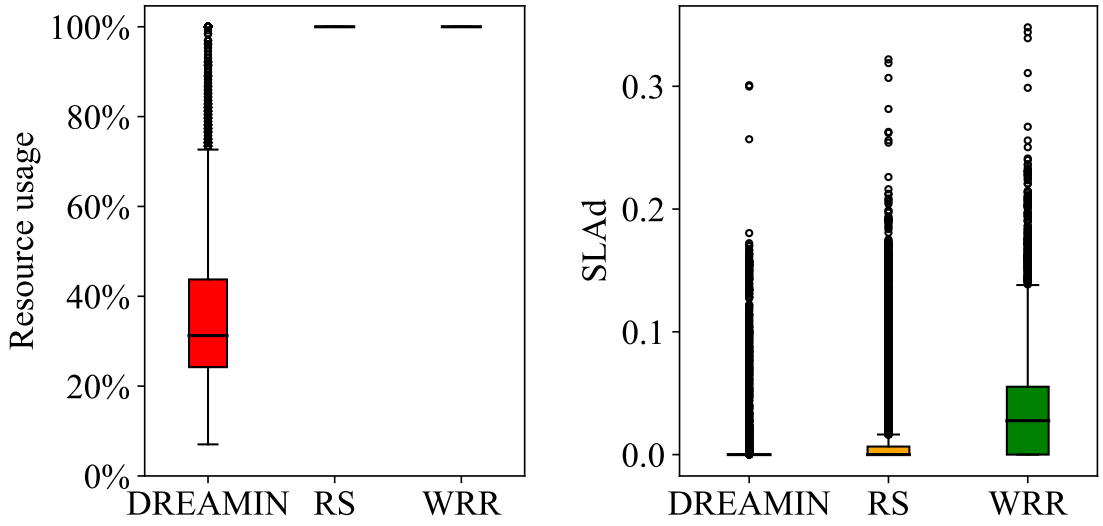


Figure 3.2: CDF for the small-scale scarce scenario SLAd.

differ in the number of users while having the same SLA requirements from Table 3.2. The plentiful scenario includes one UE per slice, while the scarce scenario includes three.

Figure 3.3 displays the evaluation of the large-scale plentiful scenario. The higher number of RBGs in this scenario enables DREAMIN to perform more effective fine-grained allocations, approximating the capacity values for each UE to the minimum capacity required to achieve zero SLAd. Consequently, DREAMIN's average resource allocation in the plentiful scenario drops from 52% in the small-scale scenario to 38% in the large-scale scenario, as depicted in Figure 3.3(a). However, the extended number of TTIs makes room for outliers: moments when multiple users simultaneously have poor channel conditions, thus, zero SLAd cannot be achieved even allocating 100% resources. The same outliers are also present in DREAMIN's SLAd on Figure 3.3(b), but the distribution is still lower than the other schedulers, reducing RS's average SLAd by 78%.

Figure 3.4 shows the SLAd in the large-scale scarce scenario, where DREAMIN's average SLAd is 62% lower than RS while allocating the same 100% resources. Despite allocating the same number of RBGs for each slice and selecting RBGs with higher capacity, RS SLAd underperforms WRR. This occurs because RS predicts the intra-slice scheduler allocations while behaving like a maximum throughput scheduler, which results in RBGs that could have been allocated to users with poor channel conditions in one slice being assigned to another to increase total capacity. Therefore, in scenarios with significant channel disparity between UEs within the same slice, RS manipulates the intra-slice scheduler to over-provide UEs in better conditions while neglecting those in worse conditions, leading to high SLAd values. This behavior did not appear in the previous scenarios because there was only one UE per slice, so no discrepancy existed between users within the same slice. Figure 3.5 displays the Complementary



(a) *Resource usage distribution.* (b) *SLAd distribution.*

Figure 3.3: Boxplots for the large-scale plentiful scenario evaluation.

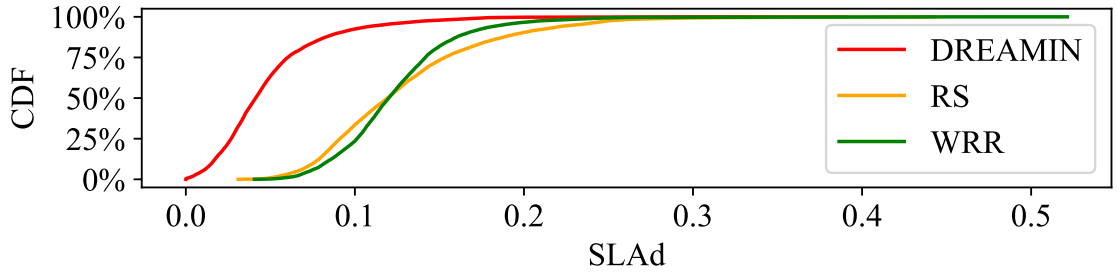


Figure 3.4: Large-scale scarce scenario SLAd CDF.

CDF (CCDF) for intra-slice fairness, calculated for each TTI as the weighted average of each slice's Raj Jain's fairness index for capacity between UEs. During 72% of the TTIs, the intra-slice fairness values for DREAMIN surpass 0.9, while the same happens in only 5% of RS's TTIs. The SLAd metric aligns with fairness indexes since all UEs within a slice have equal weight in the slice SLAd calculation. Therefore, it is expected that an SLAd-minimizing scheduler like DREAMIN achieves high fairness values, while unfair schedulers tend to have high SLAd values in scarce scenarios.

3.7 Conclusion and future work

In this work, we formulated the problem of channel-aware inter-slice radio resource scheduling for minimizing SLAd and resource usage, and we presented DREAMIN: a scalable polynomial-time heuristic. In our evaluation, DREAMIN approx-

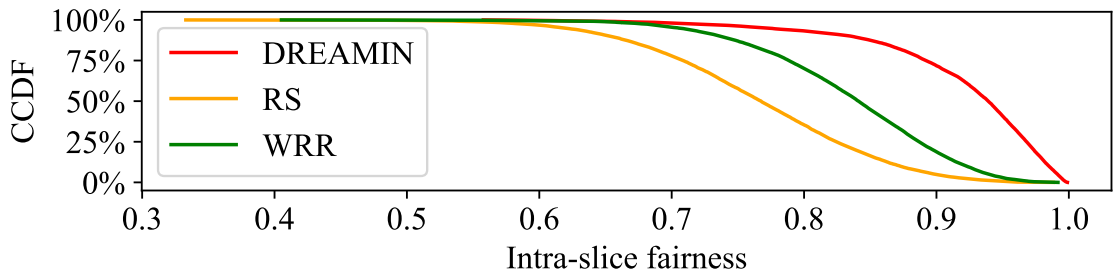


Figure 3.5: Large-scale scarce scenario fairness CCDF.

imates the optimal solution and outperforms the state-of-the-art RadioSaber scheduler by reducing SLAd and resource allocation to improve QoS and energy efficiency. We also showed how the SLAd metric aligns with fairness, as our SLAd-minimization scheduler's intra-slice fairness index is much higher than RadioSaber's, which has SLAd values even higher than a weighted round-robin due to unfairness in allocated capacity among UEs of the same slice. Lastly, future work includes adding channel prediction techniques to DREAMIN, so extra resources can be allocated to leverage the current channel quality before it drops, and expanding our formulation to consider transmission error, thus the packet error rate metric can be included as an SLA requirement.

3.8 Acknowledgements

This work was supported by CAPES, MCTIC/CGI.br/São Paulo Research Foundation (FAPESP) through the Smart 5G Core And MUltiRAn Integration (SAMURAI) project under Grant 2020/05127-2 and the Slicing Future Internet Infrastructures (SFI2) project under Grant 2018/23097-3, by RNP/MCTIC through the Brasil 6G project under Grant 01245.020548/2021-07 and the OpenRAN@Brasil Program.

CHAPTER 4

Conclusion

This thesis researched the problem of channel-aware inter-slice Radio Resource Scheduling (RRS) oriented to maximizing Service Level Agreement (SLA) assurance and minimizing resource allocation. Two papers were written based on our investigation into this problem. The first one assumed some simplifications to achieve an initial comprehension of the scheduling dynamics, while the second one expanded the scope to consider all characteristics of the proposed problem. Both papers formulated the research problem as optimization models, proposed fast algorithms to solve them in a reasonable time, and evaluated their performance in resource usage and SLA assurance compared with state-of-the-art works in the literature and the optimal/approximated solutions. As a subproduct of this thesis, the code developed for the experiments of both papers is publicly available for reproducibility, study, and improvement in https://github.com/LABORA-INF-UFG/paper-DGWCAMK-2024 and https://github.com/LABORA-INF-UFG/paper-DGMK-2024. Moreover, they implement a modular simulation to facilitate integrating and evaluating other inter-slice RRS algorithms from the literature, such as the DRL agent from [Nahum et al. 2024] and RadioSaber [Chen et al. 2023], which are already present in the published repositories. We also note that the present master's student contributed to other works in the area of wireless networks during his research, resulting in the co-authorship of the Open Radio Access Network (Open-RAN) xApp development tutorial paper [Santos et al. 2025].

We initiated our studies by comprehending a simplified problem in our first paper. First, we limited the formulation to be stepwise, so we solved each Transfer Time Interval (TTI) independently. Second, we removed channel-awareness, ignoring differences in channel quality between Resource Block Groups (RBGs) for the same User Equipment (UE). Third, we only evaluated scenarios where radio resources were plentiful, so we could assume all SLAs are ensurable and focus only on minimizing allocated RBGs. This also removed the need to prioritize slices, since they will always receive enough resources to fulfill their requirements. The three simplifications we considered made it possible to formulate a linear programming problem whose optimal solution could be achieved by calculating the Minimum Throughput Necessary (MCN) for each UE, which

has $O(1)$ complexity, and then, for each slice, allocating one RBG at a time until all its UEs' MCNs are achieved. We called this method the Stepwise Optimal Algorithm (SOA). Our simulated results showed how SOA could ensure SLAs at all TTI and save an average of 62% resources on the evaluated scenario. On the other hand, the Deep Reinforcement Learning (DRL) agent, built and trained as specified in [Nahum et al. 2024] on the same scenario, violated SLAs at some TTIs despite always allocating 100% of the RBGs.

In our second paper, we removed the simplifications from the first one, and constructed a more realistic scenario with (*i*) slices and SLAs defined based on 3rd Generation Partnership Project (3GPP) specifications, (*ii*) UE capacity values based on a dataset of Channel Quality Indicator (CQI) traces per RBG [Chen et al. 2023] to enable channel-aware evaluations, and (*iii*) the widely used Proportional Fair (PF) algorithm as intra-slice scheduler, replacing the simplistic Round-Robin. Since the new problem formulation approached multiple TTIs in a single problem, the buffer needed to be modeled to calculate latency metrics. Furthermore, adding channel-awareness turned the integer decision variables that represented the number of RBGs for each UE into binary variables spanning three dimensions, UEs, TTIs, and RBGs, representing whether or not an RBG is allocated to the UE in the TTI. The new variables were constrained to be distributed among the UEs of the same slice following the PF scheduler, so inter-slice RRS predicts the intra-slice RRS as in [Chen et al. 2023]. Also, to support scarce scenarios, we leveraged the intent-drift from [Nahum et al. 2024] to define the SLA-drift (SLAd) metric as a method of quantifying SLA non-assurance. Thus, the scheduling goal is to minimize the overall SLAd in the base station, calculated as a weighted average assuming the network operator provides weights for slices and their SLA requirements. Lastly, formulating a general problem for both plentiful and scarce resources scenarios implied in two hierarchical objectives: primarily minimizing SLAd and, if all SLAs are assured, minimizing resource allocation. The new additions to the formulation turned it into a non-linear problem, which is solved with a constraint programming optimizer. Despite evaluating the optimization model only on small scenarios, due to its high complexity and low scalability, the obtained solution is not optimal, but the best one found within 1 hour of search. Such limitations justify using a heuristic to approximate the optimal solution in a reasonable time. Hence, we proposed the Drift and REsource Allocation MINimization (DREAMIN) scheduler, which makes one allocation at a time, always choosing the one that most reduces SLA-drift. As baseline for DREAMIN, we evaluate RadioSaber [Chen et al. 2023], an also channel-aware heuristic, but that works as a maximum throughput algorithm, choosing the allocation with higher capacity to fulfill each slice's static quota of RBGs given by its weight. In all evaluated scenarios, DREAMIN closely matched the approximated-optimal solution and outperformed RadioSaber both on resource usage and SLAd and outperformed RadioSaber. While RadioSaber always uses 100% resources, DREAMIN

used an average of 38% in the evaluated plentiful scenario. Nevertheless, in the scarce scenario, DREAMIN's SLAd was 62% lower than RadioSaber's despite both allocating 100% resources. The discrepancy between the two is explained by RadioSaber's fixed quota of RBGs per slice and unfair allocation. The latter is a result of its objective being to maximize capacity, resulting in UEs in poor channel conditions having their SLAs not assured. As the SLAd of a slice considers all UEs as equal, SLAd-oriented solutions like DREAMIN tend to fairer allocations within the same slice and thus higher SLA assurance.

As findings of both papers, we list some important characteristics for inter-slice RRS algorithms:

- **Channel-awareness** - Due to frequency-selective fading, RBGs in different frequencies have also different channel qualities for the same UE. Therefore, it is not only important to decide the quantity, but also which RBGs are allocated.
- **Predicting intra-slice RRS** - Enables the inter-slice RRS to take UE-level decisions, such as channel-aware allocations, improving the performance.
- **Dynamic slice resource proportion** - Since the Radio Access Network (RAN) demands may significantly fluctuate due to the heteronegeneity of slices' applications and their traffic patterns, the proportion of allocated resources should follow similar dynamics. Having a fixed quota or number of RBGs to be allocated at all TTIs would instead lead to one slice wasting resources while other starves.
- **SLAd minimization** - Inter-slice RRS algorithms guided by SLAd metric tend to higher intra-slice fairness and SLA assurance indexes when resources are scarce (i.e., the network demand can not be fully served). Moreover, the SLAd metric avoids overprovisioning by treating all allocations achieving or surpassing the SLA requirements as equal, which also reduces resource allocation.
- **Resource minimization** - When resources are plentiful (i.e., when the network demand is relatively low), RBGs can be spared to reduce power consumption in the base station, increasing energy efficiency and reducing carbon footprint.

In conclusion, we suggest improvements for future work approaching inter-slice RRS algorithms. First, intelligent solutions running in the base station, like DREAMIN, can improve the performance of xApps that define Radio Resource Management (RRM) policies on the Open-RAN architecture. Such integration should result in a framework uniting fast channel-aware scheduling with predictive data-driven solutions. Second, new optimization strategies (e.g., stochastic and multi-stage formulations) may improve the problem scalability, enabling comparisons with solutions certified as optimal. Third, the problem can be expanded to tackle more advanced 3GPP scenarios, considering MIMO channels, UEs associated with multiple slices, and base stations with mixed numerologies, for instance. Lastly, designing intra-slice RRS algorithms tailored to work together with SLAd-oriented inter-slice RRS could also improve the overall SLAd in a joint solution.

Bibliography

[5G Slicing Association 2020]5G Slicing Association. *5G Network Slicing Self-Management White Paper*. [S.l.], 2020. Disponível em: <https://www-file.huawei.com/-/media/corporate/pdf/news/5g-network-slicing-self-management-white-paper.pdf?la=en-us>.

[Balasingam, Kotaru e Bahl 2024]BALASINGAM, A.; KOTARU, M.; BAHL, P. {Application-Level} service assurance with 5g {RAN} slicing. In: *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. [S.l.: s.n.], 2024. p. 841–857.

[Bonati et al. 2021]BONATI, L. et al. Colosseum: Large-Scale Wireless Experimentation Through Hardware-in-the-Loop Network Emulation. In: *Proc. of IEEE Intl. Symp. on Dynamic Spectrum Access Networks (DySPAN)*. Virtual Conference: [s.n.], 2021.

[Boutiba et al. 2023]Boutiba et al. Optimal radio resource management in 5G NR featuring network slicing. *Computer Networks*, v. 234, p. 109937, out. 2023. ISSN 1389-1286. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1389128623003821>.

[Boutiba et al. 2022]BOUTIBA, K. et al. NRflex: Enforcing network slicing in 5G new radio. *Computer Communications*, Elsevier, v. 181, p. 284–292, 2022. ISSN 0140-3664.

[Campos et al. 2024]CAMPOS, D. et al. Stepwise Optimal Inter-Slices Radio Resource Scheduling for Service-Level Agreement Assurance. In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. [S.l.]: SBC, 2024. p. 840–853. ISSN 2177-9384.

[Chen et al. 2023]CHEN, Y. et al. Channel-Aware 5G RAN Slicing with Customizable Schedulers. In: *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. [S.l.: s.n.], 2023. p. 1767–1782.

[Cheng et al. 2024]CHENG, H. et al. ORANSlice: An open-source 5G network slicing platform for O-RAN. *arXiv preprint arXiv:2410.12978*, 2024.

[Dai et al. 2024]DAI, J. et al. O-RAN-enabled intelligent network slicing to meet service-level agreement (SLA). *IEEE Transactions on Mobile Computing*, IEEE, n. 01, p. 1–17, 2024. ISSN 1558-0660.

[ETSI 2020]ETSI. *5G; NR; Physical channels and modulation (3GPP TS 38.211 version 16.2.0 release 16)*. [S.l.], 2020.

[ETSI 2020]ETSI. *ETSI TS 136 213. LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures (3GPP TS 36.213 version 15.10.0 Release 15)*. [S.l.], 2020.

[Jaeckel et al. 2014]JAECKEL, S. et al. Quadriga: A 3-d multi-cell channel model with time evolution for enabling virtual field trials. *IEEE transactions on antennas and propagation*, IEEE, v. 62, n. 6, p. 3242–3256, 2014.

[Jr e Lozano 2018]JR, R. W. H.; LOZANO, A. *Foundations of MIMO communication*. [S.l.]: Cambridge University Press, 2018.

[Khodapanah et al. 2020]KHODAPANAH, B. et al. Framework for Slice-Aware Radio Resource Management Utilizing Artificial Neural Networks. *IEEE Access*, v. 8, p. 174972–174987, 2020. ISSN 2169-3536. Disponível em: <https://ieeexplore.ieee.org/document/9204709/>.

[Kokku et al. 2012]KOKKU, R. et al. NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks. *IEEE/ACM Transactions on Networking*, v. 20, n. 5, p. 1333–1346, out. 2012. ISSN 1063-6692, 1558-2566. Disponível em: <http://ieeexplore.ieee.org/document/6117098/>.

[Li et al. 2024]LI, J. et al. Hierarchical intelligent radio access network slicing for differential service level agreement guaranteeing. *IEEE Transactions on Industrial Informatics*, IEEE, v. 20, n. 3, p. 4124–4136, 2024.

[Lotfi, Afghah e Ashdown 2023]LOTFI, F.; AFGHAH, F.; ASHDOWN, J. *Attention-based Open RAN Slice Management using Deep Reinforcement Learning*. arXiv, jun. 2023. ArXiv:2306.09490 [cs, eess]. Disponível em: <http://arxiv.org/abs/2306.09490>.

[Mei et al. 2021]MEI, J. et al. Intelligent Radio Access Network Slicing for Service Provisioning in 6G: A Hierarchical Deep Reinforcement Learning Approach. *IEEE Transactions on Communications*, v. 69, n. 9, p. 6063–6078, set. 2021. ISSN 0090-6778, 1558-0857. Disponível em: <https://ieeexplore.ieee.org/document/9459763/>.

[Mondal et al. 2015]MONDAL, B. et al. 3D channel model in 3GPP. *IEEE Communications Magazine*, IEEE, v. 53, n. 3, p. 16–23, 2015.

[Nahum et al. 2025]NAHUM, C. et al. Intent-based radio scheduler for ran slicing: Learning to deal with different network scenarios. *arXiv preprint arXiv:2501.00950*, 2025.

[Nahum et al. 2024]NAHUM, C. V. et al. Intent-aware radio resource scheduling in a RAN slicing scenario using reinforcement learning. *IEEE Transactions on Wireless Communications*, v. 23, n. 3, p. 2253–2267, 2024.

[Navidan et al. 2024]NAVIDAN, H. et al. Radio Resource Management for Intelligent Neutral Host (INH) in Multi-Operator Environments. *IEEE Open Journal of the Communications Society*, IEEE, v. 5, p. 1975–1986, 2024.

[Polese et al. 2022]POLESE, M. et al. ColO-RAN: Developing Machine Learning-based xApps for Open RAN Closed-loop Control on Programmable Experimental Platforms. *IEEE Transactions on Mobile Computing*, p. 1–14, 2022. ISSN 1536-1233, 1558-0660, 2161-9875. Disponível em: <https://ieeexplore.ieee.org/document/9814869/>.

[Polese et al. 2023]POLESE, M. et al. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Communications Surveys & Tutorials*, v. 25, n. 2, p. 1376–1411, 2023. ISSN 1553-877X, 2373-745X. Disponível em: <https://ieeexplore.ieee.org/document/10024837/>.

[Rana et al. 2024]RANA, M. K. et al. A QoS improving downlink scheduling scheme for slicing in 5G radio access network (RAN). *IEEE Transactions on Vehicular Technology*, v. 73, n. 3, p. 4219–4233, 2024.

[Santos et al. 2025]SANTOS, J. F. et al. Managing o-ran networks: xapp development from zero to hero. *IEEE Communications Surveys & Tutorials*, IEEE, 2025.

[Sherif, Ahmed e Kotb 2025]SHERIF, H.; AHMED, E.; KOTB, A. M. Towards green networking: Efficient dynamic radio resource management in open-ran slicing using deep reinforcement learning and transfer learning. *Computer Communications*, Elsevier, v. 236, p. 108126, 2025.

[Tong e Zhu 2021]TONG, W.; ZHU, P. (Ed.). *6G: The Next Horizon: From Connected People and Things to Connected Intelligence*. 1. ed. Cambridge University Press, 2021. ISBN 978-1-108-98981-7 978-1-108-83932-7. Disponível em: <https://www.cambridge.org/core/product/identifier/9781108989817/type/book>.

[Xie, Yi e Jamieson 2020]XIE, Y.; YI, F.; JAMIESON, K. PBE-CC: Congestion control via endpoint-centric, physical-layer bandwidth measurements. In: *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*. [S.l.]: ACM, 2020. p. 451–464.

[Yang et al. 2024]YANG, K. et al. Advancing ran slicing with offline reinforcement learning. In: IEEE. *2024 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. [S.l.], 2024. p. 331–338.

[Zhu et al. 2021]ZHU, Q. et al. 3GPP TR 38.901 channel model. In: *the wiley 5G Ref: the essential 5G reference online*. [S.l.]: Wiley Press, 2021. p. 1–35.

APPENDIX A

Publication webpage of the first paper

ANAIS DO SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS (SBRC)

Stepwise Optimal Inter-Slices Radio Resource Scheduling for Service-Level Agreement Assurance

Daniel Campos
UFG

Gabriel M. F. de Almeida
UFG

William T. P. Junior
UFG

Cleverson V. Nahum
UFPA

Aldebaro Klautau
UFPA

Mohammad J. Abdel-Rahman
Princess Sumaya University for Technology / Virginia Tech

Kleber V. Cardoso
UFG

DOI: https://doi.org/10.5753/sbrc.2024.1482

RESUMO

In 5G networks and beyond, radio access networks (RANs) must be able to support multiple services with different service level agreements (SLAs). Network slicing is a critical concept in this context and it depends on an efficient approach for radio resource scheduling (RRS). Inter-slices RRS is responsible for allocating resource block groups (RBGs) to the slices to ensure their SLAs. Mainly motivated by the O-RAN initiative, several works in the literature have presented proposals based on machine learning (ML) to solve this problem. However, there is still a lack of problem formalization and an optimal strategy, which are both introduced in this work. Through simulations, we compare our approach with a state-of-the-art deep reinforcement learning (DRL) agent. The results show the excess resources employed by the agent when they are plentiful, suggesting an unnecessary increase in energy consumption. Additionally, we show the relevant gap between solutions when the resources are scarce. Finally, we discuss guidelines on how to improve ML-based approaches to the inter-slices RRS problem.

REFERÊNCIAS

Chen, Y., Yao, R., Hassanieh, H., and Mittal, R. (2023). Channel-Aware 5G RAN Slicing with Customizable Schedulers. In 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23), pages 1767–1782.

ETSI (2020a). 5G; NR; Physical channels and modulation (3GPP TS 38.211 version 16.2.0 release 16). Technical report, European Telecommunications Standards Institute (ETSI).

ETSI (2020b). ETSI TS 136 213. LTE; Evolved Universal Terrestrial Radio Access (EUTRA); Physical Layer

Procedures (3GPP TS 36.213 version 15.10.0 Release 15). Technical report, European Telecommunications Standards Institute (ETSI).

Heath Jr, R. W. and Lozano, A. (2018). Foundations of MIMO communication. Cambridge University Press.

Jaeckel, S., Raschkowski, L., Börner, K., and Thiele, L. (2014). Quadriga: A 3-d multicell channel model with time evolution for enabling virtual field trials. IEEE transactions on antennas and propagation, 62(6):3242–3256.

Khodapanah, B., Awada, A., Viering, I., Barreto, A. N., Simsek, M., and Fettweis, G. (2020). Framework for Slice-Aware Radio Resource Management Utilizing Artificial Neural Networks. IEEE Access, 8:174972–174987.

Kokku, R., Mahindra, R., Zhang, H., and Rangarajan, S. (2012). NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks. IEEE/ACM Transactions on Networking, 20(5):1333–1346.

Lotfi, F., Afghah, F., and Ashdown, J. (2023). Attention-based Open RAN Slice Management using Deep Reinforcement Learning. arXiv:2306.09490 [cs, eess].

Mei, J., Wang, X., Zheng, K., Boudreau, G., Sediq, A. B., and Abou-Zeid, H. (2021). Intelligent Radio Access Network Slicing for Service Provisioning in 6G: A Hierarchical Deep Reinforcement Learning Approach. IEEE Transactions on Communications, 69(9):6063–6078.

Mondal, B., Thomas, T. A., Visotsky, E., Vook, F. W., Ghosh, A., Nam, Y.-H., Li, Y., Zhang, J., Zhang, M., Luo, Q., et al. (2015). 3D channel model in 3GPP. IEEE Communications Magazine, 53(3):16–23.

Nahum, C. V., Lopes, V. H., Dreifuerst, R. M., Batista, P., Correa, I., Cardoso, K. V., Klautau, A., and Heath, R. W. (2023). Intent-aware Radio Resource Scheduling in a RAN Slicing Scenario using Reinforcement Learning. IEEE Transactions on Wireless Communications, pages 1–1.

Polese, M., Bonati, L., D'Oro, S., Basagni, S., and Melodia, T. (2022). ColO-RAN: Developing Machine Learning-based xApps for Open RAN Closed-loop Control on Programmable Experimental Platforms. IEEE Transactions on Mobile Computing, pages 1–14.

Polese, M., Bonati, L., D'Oro, S., Basagni, S., and Melodia, T. (2023). Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. IEEE Communications Surveys & Tutorials, 25(2):1376–1411.

Zhu, Q., Wang, C.-X., Hua, B., Mao, K., Jiang, S., and Yao, M. (2021). 3GPP TR 38.901 channel model. In the wiley 5G Ref: the essential 5G reference online, pages 1–35. Wiley Press.

PDF (ENGLISH)

PUBLICADO

20/05/2024

COMO CITAR

SELECIONE UM FORMATO

ABNT
ACM
APA
BibTeX
CBE
EndNote - formato Macintosh & Windows
IEEE
MLA
ProCite - formato RIS (Macintosh & Windows)
RefWorks
Turabian
Reference Manager - formato RIS (somente para Windows)

CAMPOS, Daniel; ALMEIDA, Gabriel M. F. de; JUNIOR, William T. P.; NAHUM, Cleverson V.; KLAUTAU, Aldebaro; ABDEL-RAHMAN, Mohammad J.; CARDOSO, Kleber V.. Stepwise Optimal Inter-Slices Radio Resource Scheduling for Service-Level Agreement Assurance. *In*: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS (SBRC), 42. , 2024, Niterói/RJ. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024 . p. 840-853. ISSN 2177-9384. DOI: https://doi.org/10.5753/sbrc.2024.1482.

Artigos mais lidos do(s) mesmo(s) autor(es)

- Phelipe A. de Souza, Elivelton F. Bueno, Kleber V. Cardoso, Alocação e posicionamento de recursos para redes de acesso virtualizadas com diferentes níveis de centralização , Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC): 2018: Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos
- Henrique V. Lima, Elivelton F. Bueno, Kleber V. Cardoso, Alocação de recursos em redes LTE utilizando bandas não-licenciadas , Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC): 2018: Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos
- Jefferson Morais, Yomara Pires, Fernando dos Santos, Waldemar Neto, Aldebaro Klautau, Seleção de Parâmetros para Classificação de Faltas do Tipo Curto-Circuito em Linhas de Transmissão , Anais da Escola Regional de Sistemas de Informação do Rio de Janeiro (ERSI-RJ): 2014: Anais da I Escola Regional de Sistemas de Informação do Rio de Janeiro
- Antonio Oliveira-Jr, Ciro Macedo, Gabriel M. F. de Almeida, Leonardo Rodrigues, Marcos Abreu, Sand Correa, Kleber Cardoso, Experimentos da Proposta de Integração SOFTWAY4IoT e FIWARE-Lab@RNP para Internet das Coisas (IoT) , Anais do Workshop do Testbed FIBRE (WFIBRE): 2021: Anais do VI Workshop do testbed FIBRE
- Lucas Nóvoa, Virgínia Tavares, Cleverson Nahum, Silvia Lins, Aldebaro Klautau, Middleware implementation for RYU SDN Controller to manage switches in a C-RAN scenario , Anais do Seminário Integrado de Software e Hardware (SEMISH): 2021: Anais do XLVIII Seminário

Integrado de Software e Hardware

- Pedro H. P. Castro, Sand Corrêa, Kleber V. Cardoso, Uma Abordagem Baseada no Consumo de CPU e RAM para a Eficiência Energética em Centros de Dados para Computação em Nuvem , Anais do Simpósio em Sistemas Computacionais de Alto Desempenho (SSCAD): 2013: Anais do XIV Simpósio em Sistemas Computacionais de Alto Desempenho
- Gabriel M. F. de Almeida, Victor Hugo L. Lopes, Cristiano B. Both, Sand Corrêa, Aldebaro Klautau, Kleber V. Cardoso, MPP-RAN: Posicionamento de funções de rede virtualizadas em redes de acesso de nova geração com divisão de fluxos , Anais do Workshop de Redes 6G (W6G): 2021: Anais do I Workshop de Redes 6G
- Ciro J. A. Macedo, Hudson P. Romualdo, Cristiano B. Both, Antonio Oliveira-Jr, Kleber V. Cardoso, NWDAF habilitando Inteligência Artificial em operações de busca e salvamento assistidas por VANTs , Anais do Workshop de Redes 6G (W6G): 2021: Anais do I Workshop de Redes 6G
- Henrique V. de Lima, Rogério S. Silva, Cristiano B. Both, Antonio Oliveira-Jr, Kleber V. Cardoso, Sand L. Corrêa, Orquestração Inteligente de Network Slicing: Revisão da Literatura e Prospecção para Redes 6G , Anais do Workshop de Redes 6G (W6G): 2021: Anais do I Workshop de Redes 6G
- Heitor Scalco Neto, Antonia Vanessa D. Araujo, Cristiano B. Both, Antonio Oliveira-Jr, Kleber V. Cardoso, Uma Visão de Arquitetura para Redes 6G , Anais do Workshop de Redes 6G (W6G): 2021: Anais do I Workshop de Redes 6G

1 2 > >>

IDIOMA

Português (Brasil)

English

O conteúdo publicado neste portal representa exclusivamente a opinião de seus autores e não necessariamente a posição da Sociedade Brasileira de Computação – SBC, seus colaboradores e associados. A SBC poderá adotar a qualquer tempo, e sem a necessidade de prévio aviso, a cobrança de uso e disponibilização da plataforma e seu conteúdo para não associados.

Av. Bento Gonçalves, 9500 | Setor 4 | Prédio 43.412 | Sala 219 | Bairro Agronomia
Caixa Postal 15012 | CEP 91501-970
Porto Alegre - RS
CNPJ: 29.532.264/0001-78
Fone: (51) 99252-6018
sbc@sbc.org.br

PKP | PUBLIC KNOWLEDGE PROJECT powered by OJS | Open Journal Systems

APPENDIX B

Acceptance letter of the second paper

Daniel Campos <dante_campos@discente.ufg.br>

IEEE ICC'25 - MWN Symposium - Congratulations for 1571083537: DREAMIN: Channel-Aware Inter-Slices Radio Resource Scheduling for Efficient SLA Assurance

1 mensagem

ieeeicc25-mwnsymposium-chairs@edas.info <ieeeicc25-mwnsymposium-chairs@edas.info> 17 de janeiro de 2025 às 21:30
Para: Daniel Campos <dante_campos@discente.ufg.br>, Gabriel Almeida <gmfaria6@gmail.com>, "Mohammad J. Abdel-Rahman" <mo7ammad@vt.edu>, Kleber V Cardoso <kleber@inf.ufg.br>

Dear Mr. Campos:

Congratulations! We are pleased to inform you that your paper #1571083537 ('DREAMIN: Channel-Aware Inter-Slices Radio Resource Scheduling for Efficient SLA Assurance') has been **accepted** for presentation at the 2025 IEEE International Conference on Communications (ICC): Mobile and Wireless Networks Symposium.

Your paper's review reports are shown at the end of this email or can be found on EDAS at 1571083537.

Accepted and presented papers will be published in the IEEE ICC 2025 Conference Proceedings and submitted to IEEE Xplore® (subject to the fulfillment of conditions outlined below - see also full COMSOC policy at the end of the message).

PLEASE READ ALL OF THE IMPORTANT INFORMATION BELOW THOROUGHLY.

Registration

In order to be published in the IEEE ICC 2025 Proceedings and IEEE Xplore®, at least one author (including students) of an accepted paper is required to register for the conference at either the All-inclusive, Full or Limited author registration rate, and the paper must be presented in person by an author or authors named on the paper. Non-refundable registration fees must be paid prior to uploading the final IEEE formatted, publication-ready version of the paper. Author registration will open shortly on the IEEE ICC 2025 website at https://icc2025.ieee-icc.org/registration, and an email will be sent when it is open.

For authors with multiple accepted papers, one Author - All-Inclusive, Full or Limited registration is valid for up to 3 (THREE) papers. Although the registration confirmation code enables submission of up to THREE papers, it can only be used once, by you, to create an author account for your submissions. Please do not share your registration confirmation code with anyone else.

Paper Revision and Final Manuscript Upload

Each accepted paper is limited to 6 pages (or no more than 7 pages with USD 100 over-length charge for the extra page). You are strongly recommended to revise your paper in consideration of the reviewers' comments. Your Symposium/Track Chairs may contact you if they are not satisfied with your final version of your paper. Please note that the paper title as well as the list and order of authors cannot be changed after the paper is accepted. Failure to abide by this policy may result in the removal of your paper from the final conference program.

The final camera-ready manuscript is due by **28 February, 2025**, but you must be registered for IEEE ICC 2025 to upload your paper. Once you have registered for the conference, you will receive a confirmation code that you will use when you upload the paper. More details on the final camera-ready manuscript upload process will be given later and will be posted in the “Information for Authors/Speakers” page of the IEEE ICC 2025 website at https://icc2025.ieee-icc.org/authors/information-authors .

Presentation Format and Session

IEEE ICC 2025 includes both oral lecture-style presentation and interactive poster-style presentation sessions. This distinction has no relationship with the quality of the accepted papers whatsoever. You will receive a separate email informing you of the format (oral or interactive) of your presentation, and the session to which your paper is assigned. Please note that by submitting a manuscript, all authors have implicitly agreed to present their accepted papers, regardless of format, in sessions organized by the Conference Program Committee. Authors may not request to change the presentation format of their accepted papers. To assist you with your presentation we

suggest you review the presentation guide provided under the ICC 2025 website.

No-Show Policy

The organizers of IEEE ICC 2025 as well as attendees expect an accepted paper to be presented in person at the conference by an author of that paper. ICC 2025 will be **an in-person conference**, and online live presentations will not be possible. **Authors may be granted the possibility to upload a recorded presentation of their paper on the conference repository only under extreme circumstances and only after they obtain approval of the chairs.** A paper that fails to meet this requirement will be removed from the final conference proceedings before uploading to IEEE Xplore®. No refund will be made to the authors of these papers. Please read the full policy below under "IEEE COMSOC POLICY" carefully.

Registration

In order to be published in the IEEE ICC 2025 Conference Proceedings and IEEE Xplore®, an author (including students) of an accepted paper is required to register for the conference at the Author All-Inclusive, Full or Limited (member or non-member) rate and the paper must be presented by an author named on the paper at the **conference unless the TPC Chair grants permission for a substitute presenter.** Non-refundable registration fees must be paid prior to uploading the final IEEE formatted, publication-ready version of the paper. Author registration will open on the IEEE ICC2025 web site shortly.

For authors with multiple accepted papers, one full or limited registration is valid for up to 3 (THREE) papers. Accepted and presented papers will be published in the IEEE ICC 2025 Conference Proceedings and in IEEE Xplore®.

Although the registration confirmation code enables submission of up to THREE papers, it can only be used once, by you, to create an author account for your submissions. Please do not share your registration confirmation code with anyone else. Registration will open soon at: https://icc2025.ieee-icc.org/registration .

Transfer Policy:

All requests for registration transfers must be provided in writing and email to m.a.torres@comsoc.org by 26 May 2025. Registration rate difference will apply depending on the membership status of the new registration. All transfer requests received AFTER 26 May 2025 will be charged an additional US$100.

PRESENTER SUBSTITUTION POLICY

If an author or co-author is NOT available to present the paper at the conference, the TPC Chair can grant permission in some extreme cases to have the paper presented by a qualified substitute presenter. If you need to make such a request, please fill out a Substitute Presenter Form no later than 26 May 2025. A copy of the form will go directly to the TPC Chair for approval.

Logistics

We suggest you make your airline and hotel reservations as early as possible. All sleeping room reservations are open at: https://icc2025.ieee-icc.org/venue/hotels. The IEEE Communications Society has contracted a favorable rate with both the Westin and the InterContinental hotels directly across the street from the main entrance of the Palais Convention Centre.

Visas

Check to see if you require a visa to travel to Canada. More details can be found at: https://icc2025.ieee-icc.org/hotel-travel/visa-letter-request.
For authors needing assistance, please contact Melissa Torres at m.a.torres@comsoc.org.

IEEE COMSOC POLICY

To be published in the 2025 IEEE ICC Conference Proceedings and to be eligible for publication in IEEE Xplore®, an author of an accepted paper is required to register for the conference at the AUTHOR (member or non-member) rate, and the paper must be presented by an author of that paper at the conference unless the TPC Chair grants permission for a substitute presenter arranged in advance of the event and who is qualified both to present and answer questions. Non-refundable registration fees must be paid prior to uploading the final IEEE formatted, publication-ready version of the paper. For authors with multiple accepted papers, one full registration is valid for up to 3 papers. Accepted and presented papers will be published in the 2025 IEEE ICC Conference Proceedings and submitted to IEEE Xplore® as well as other Abstracting and Indexing (A&I) databases.

Congratulations again for having your paper accepted in IEEE ICC 2025 - IEEE International Conference on Communications, a flagship conference of the IEEE Communications Society. We look forward to welcoming you to Montreal, Canada in May 2025.

Thank you and best regards,

Hossam Hassanein, Soumaya Cherkaoui, and Dusit (Tao) Niyato

IEEE ICC 2025 Technical Program Chairs

ICC review 1

Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.

Excellent (5)

Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.

Solid work of notable importance. (4)

Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.

A pioneering piece of work. Striking novel ideas or results. (5)

Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.

Well written. (4)

Strong aspects: Comments to the author: what are the strong aspects of the paper

1. The authors formulate a constraint programming problem for channel-aware Radio Resource Scheduling (RRS) oriented to Service Level Agreement Drift (SLAD) minimization and resource usage optimization. This approach is both novel and impactful, addressing critical aspects of network slicing in 5G and beyond.
2. The authors demonstrate an excellent grasp of the field with a thorough and up-to-date literature review. This provides a solid foundation for the proposed approach and contextualizes its contributions relative to existing work.
3. The language and format of the paper are smooth and professional, with clearly articulated arguments and logical organization. The clarity of the presentation makes the work accessible to readers.
4. The simulation results are comprehensive, covering a variety of scenarios. The detailed comparisons with the state-of-the-art RadioSaber (RS) scheduler highlight the strengths of the proposed approach, DREAMIN.

Weak aspects: Comments to the author: what are the weak aspects of the paper?

1. While the simulation results show significant performance improvements over RadioSaber (RS), the paper lacks a theoretical analysis of DREAMIN’s optimality. It would strengthen the work to explain why DREAMIN outperforms RS and provide theoretical insights beyond simulation results.
2. Considering the rapid channel fading in real-world scenarios, it is essential to compare the algorithm's complexity with RS and discuss potential challenges for practical deployment. AI-based methods might be worth exploring for channel-aware scheduling.
3. The full name of DREAMIN should appear earlier in the abstract or introduction to avoid confusion.

Recommended changes: Please indicate any changes that should be made to the paper if accepted.

1. Provide a theoretical analysis of the proposed algorithm’s optimality. This could include approximation bounds or guarantees to complement the simulation results. Additionally, explain the mechanisms behind DREAMIN’s significant performance improvements over RS.
2. Compare the computational complexity of DREAMIN with RS and discuss its feasibility in practical

scenarios, especially under rapid channel fading conditions. Consider addressing the challenges of applying DREAMIN in real-time systems.

3. Give the full name of DREAMIN in the abstract or introduction to enhance clarity.

Submission Policy: Does the paper list the same author(s), title and abstract (minor wording differences in the abstract are okay) in its PDF file and EDAS registration? (yes/no)

yes

ICC review 2

Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.

Good (4)

Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.

Valid work but limited contribution. (3)

Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.

Some interesting ideas and results on a subject well investigated. (3)

Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.

Well written. (4)

Strong aspects: Comments to the author: what are the strong aspects of the paper

The paper formulates a constraint programming problem for channel-aware RRS oriented toward SLAD and resource usage minimization and proposes DREAMIN as a scalable approximation of the solution. Simulation results show DREAMIN outperforming the state-of-the-art RadioSaber.

Weak aspects: Comments to the author: what are the weak aspects of the paper?

Some of the symbols used are not defined, e.g., the symbol C(sub(r,t),sup(u)) in equation (1). While being channel-aware, it is not clear what channel model and whether UE mobility are considered in the simulation evaluation.

Recommended changes: Please indicate any changes that should be made to the paper if accepted.

Concerns raised in the Weak Aspects need to be addressed as much as possible. In particular, the paper must ensure that all symbols and acronyms are clearly defined when they first appear. The channel model and mobility model adopted should be described. Computation time should be included in the results comparison, and the feasibility should be discussed. The reference to the 'RadioSaber scheduler' should be cited when it first appears on Page 2. Additionally, explain why Reference [4] is not included in the comparison, as it appears highly relevant to this work. “Inter-Slices” in the title should be changed to “Inter-Slice”.

Submission Policy: Does the paper list the same author(s), title and abstract (minor wording differences in the abstract are okay) in its PDF file and EDAS registration? (yes/no)

Yes

ICC review 3

Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.

Good (4)

Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.

Solid work of notable importance. (4)

Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.

Significant original work and novel results. (4)

Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.

Well written. (4)

Strong aspects: Comments to the author: what are the strong aspects of the paper

The problem is relevant, and the solution is sound.

Real-world traces are used in the performance evaluation.

Real-world aspects of 5G are considered, along with real-world 3GPP documents.

Weak aspects: Comments to the author: what are the weak aspects of the paper?

The paper seems to solely focus on packet delivery and scheduling, and higher-level aspects of networks are ignored.

The fact that network slices and SLAs include computing as well as networking aspects is ignored.

Improvements could be done to presentation and writing quality.

Recommended changes: Please indicate any changes that should be made to the paper if accepted.

More clearly position this paper within the wider context of 5G networks. The contribution of the paper is worthwhile, however, the authors should clarify which aspects of 5G networks they leave out of the scope of their work.

Submission Policy: Does the paper list the same author(s), title and abstract (minor wording differences in the abstract are okay) in its PDF file and EDAS registration? (yes/no)

yes

ICC review 4

Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.

Excellent (5)

Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.

Solid work of notable importance. (4)

Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.

Significant original work and novel results. (4)

Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.

Excellent. (5)

Strong aspects: Comments to the author: what are the strong aspects of the paper

This paper formulates the constraint programming problem of channel-aware RRS, which aims to minimize the LAD and resource usage, and proposes DREAMIN as a scalable approximation of the solution. It is very logically organized and easy to read.

Weak aspects: Comments to the author: what are the weak aspects of the paper?

Please refer to the recommendation points in the following.

Recommended changes: Please indicate any changes that should be made to the paper if accepted.

The reviewers thought authors should explain why they need to focus on SLAD rather than SLA, and the relationship between the target QoS values and the target values in SLAD.

Submission Policy: Does the paper list the same author(s), title and abstract (minor wording differences in the abstract are okay) in its PDF file and EDAS registration? (yes/no)

yes