The Dissertation Committee for Geetha Chandrasekaran
certifies that this is the approved version of the following dissertation:

Opportunistic Quality of Service Constrained Scheduling Algorithms for Wireless Networks

Committee:

Gustavo de Veciana, Supervisor

Hyeji Kim

John J Hasenbein

Sandeep Chinchali

Sanjay Shakkottai

Opportunistic Quality of Service Constrained Scheduling Algorithms for Wireless Networks

by
Geetha Chandrasekaran

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin
May 2024

Dedication

To my family – for all that I am and for all that I will ever be.

" தொட்டனைத் தூறும் மனக்கேணி மாந்தர்க்கு
கற்றனைத் தூறும் அறிவு ", திருக்குறள் 373.

“The flow of water in a well depends on the depth to which it is dug,
so is the flow of knowledge based on the depth of one’s learning”, Thirukkural 373.

Acknowledgments

The road to earning a PhD is filled with many twists and turns, dead ends, pit falls, some uphill rides, and some dangerous slides. My journey to a PhD would have been terrifying without a mentor who could walk alongside me every step of the way. This thesis wouldn't have been possible without the exemplary supervision and guidance of Prof. Gustavo de Veciana. He sets the bar really high with his zeal for perfection and his impeccable work ethic. I consider myself fortunate to have him as my PhD advisor, he has been very understanding with my busy schedule as a working mother. A pandemic two months into my graduate research assistantship was something neither of us were prepared for. Despite the limited connectivity/social circle during those first couple of years - some tough times to be supervising - he has been very encouraging and supportive. He has been the driving force that kept me focused and not thwarted by reviewers or rejections, instead channeling them into a good learning process – to write better and present compelling results. Thank you, professor, for believing in me even when it was hard for me to do so.

I am sure that Queueing theory as taught by Prof. Gustavo is one of the best courses offered at UT Austin. While I have had the pleasure of meeting with Prof. Gustavo every single week over the past four and half years, there are others who have had a strong impact on my research. I was fortunate to be taught by Prof. Sanjay for three semesters (well each coursework worth three semesters!), the tools that I have learned, some of which I have applied to develop proofs for my theorems in this thesis. I have also had the pleasure of being taught by the talented professors that WNCG has to offer, Prof. Constantine Caramanis, Prof. Robert Heath, Prof. Jeffrey Andrews, Prof. Ross Baldick and some from outside the department like Prof.

Peter Stone, and Prof. Gordan Zitkovic. I am grateful to Prof. Sriram Vishwanath for getting me admitted into the DICE program and a special note of thanks to Prof. Jeff Andrews for helping me navigate the first couple semesters of my PhD when I was still not clear on my research directions. I would like to express my gratitude to all my committee members: Prof. Sanjay, Prof. Hyeji Kim, Prof. Sandeep Chinchali and Prof. John Hasenbein for their time and effort in providing valuable feedback on my research. I would'nt have applied to UT Austin without the support and encouragement from my Masters advisor at IIT Madras, Prof. Sheetal Kalyani and her peers Prof. Radhakrishna Ganti and Prof. Krishna Jagannathan. My sincere thanks go out to all of you for being such wonderful role models!

Jamie and Apipol, you guys may have left WNCG, but your absence is still felt. Thank you, Melanie Gulick, Melody Singleton, and Thomas Achity for making administrative work easier these past few semesters. Thank you Erin Salcher and Leah Merritt, for helping me out with conference registrations and other administrative business. Thanks, Karen, for being you - the happy, cheerful, wear my feeling on my sleeve person, the go-to person for everything at WNCG and not to forget the only one to send funny email memes.

Thanks to all the WNCG affiliates whose contributions were instrumental to my research journey. Our sponsors from Samsung Research America – Dr. Hao Chen and Dr. Vishnu Rathnam for supporting my initial work on opportunistic delay constrained schedulers. I am also thankful to Dr. Harish Vishwanathan at Nokia Bell Labs and Dr. Simone Merlin at Qualcomm for my memorable research internships.

My PhD story wouldn't be complete without the friends that I made at UT Austin. Thank you Aditya, for all the sarcasm, wit and for just being there when I needed a friend. Our internship at Nokia Bell Labs would not have been so much fun without you. Thank you Anirudh for all the shuttle matches, the hour-long talks, the IIT Madras stories, the homemade delicacies, and much more. I was fortunate to work with Ahmad – although we never got to publish – I learned a lot from you. Thank you for just being there when I needed someone to talk to. Thanks to all my colleagues who were always up for some heated discussions over homework solutions or political ideologies - Nithin, Nihal, Kartik, Manan. To Ezgi and Eunsun, happy to have interacted with you over the years. Thanks to all my ASCEND group mates Jean, Sadallah, Agrim, Parikshit, Jianhan, Hassan, Heasung and Alejandro, for all the group outings, lunches, discussions, and celebrations.

I would also like to thank my friends outside of the university who are as much invested in my success, and who have been my lifeline throughout my time in Austin. In no specific order, Suman, Sakshi, Anshul, Prasanna, Gayathri, Ruchi, Paramjeet, Gayathri, Kalaivani, Meenakshi, Saranya, Shankar, and Amarnath, I would not have made it through all these years or the pandemic without you folks.

My parents are the epitome of hard work, perseverance, honesty, integrity, and unconditional love. They are always proud of my accomplishments, and encourage me to follow my heart (except for graduate studies). My mom went above and beyond her responsibilities to help me with child (infant) support during those crucial semesters at the start of my PhD. My dad always wanted me to be a doctor (the real kind), but he is happy that atleast I will have the honorary "Dr." prefix. My sister moving to Austin was one of the best things that has happened during the pandemic. She has been a life saver whenever I need some quality time without the kids around. I have the world's best teachers of patience, kindness, and time management – my kids. My son Naveen makes me proud with all his accomplishments (LISD QUEST, Symphony band, Science Olympiad and Math Olympiad), in turn challenging me to be my best in everything I do. Thanks to my very own personal therapist for all her love and affection – my adorable daughter Dharshana. She could turn my day around with all her hugs and kisses.

Behind every successful woman is a strong man who is very supportive of her goals. Well, I was fortunate to marry some one who can cook, take care of the kids, stock the refrigerator, and let me forget familial duties when there are demanding work deadlines. He moved to Austin, while I was still doing my masters at IIT Madras, because I thought it would be nice to do a PhD at UT Austin. He switched jobs during the pandemic to reduce my responsibilities at home. My summer internships felt like a breeze because he could be both a father and a mother to our kids. It is exceptional to be blessed with such unconditional love from someone who did not give birth to you! I am lucky to have Gokul to share my life with for he is my better half, he makes me a better person every single day – an incurable optimist for a clinical pessimist!

Abstract

Opportunistic Quality of Service Constrained Scheduling Algorithms for Wireless Networks

Geetha Chandrasekaran, PhD
The University of Texas at Austin, 2024

SUPERVISOR: Gustavo de Veciana

Support of next generation service categories such as Ultra Reliable Low Latency Communication (URLLC) and Enhanced Mobile BroadBand (eMBB) is expected to be critical towards enabling next generation wireless applications such as industrial automation, augmented and virtual reality, autonomous driving, remote diagnosis, and health care. The focus of this thesis is on wireless scheduling policies, and more broadly, resource allocation frameworks for settings in which one has multiple objectives associated with supporting multiple Quality of Service (QoS) requirements for users with heterogeneous traffic and channel characteristics. Particularly challenging is the need to deliver low latency traffic to heterogeneous users/devices with strict deadlines in a spectrally efficient manner. We develop spectrally efficient schedulers that ensure low latency and reliability constraints for URLLC users, while also maximizing the longer term throughput achievable by eMBB users.

We note that scheduling alone can not be called on to meet such complex objectives, particularly in heterogeneous settings with substantial uncertainty that

arises in wireless systems. Indeed, there is a need for complementary approaches to admission control if specific QoS requirements are to be met, and admission control itself presents substantial challenges. As part of the work, we propose relatively simple measurement based admission control policies. The general concept is to make admission control decisions based on monitoring the actual resource requirements of opportunistic schedulers that meet users' QoS requirements.

Base stations across wireless service areas typically have an unequal distribution of user load which is stochastic in nature and thus leads to a dynamic coupling among shared wireless resources. This along with a high network density makes it hard to predict interference from neighboring base stations or design a centralized algorithm for interference mitigation. We address the problem of distributed resource allocation in wireless systems in the presence of dynamic user traffic and coupling resulting from interference. In particular, we explore a setting where a stochastic game is set up among base stations to learn efficient frequency reuse patterns and solved using multi-agent RL given an underlying choice for user scheduling. We establish the existence and convergence to a Nash equilibrium of the proposed setting.

Our research focuses on three key areas – we first concentrate on designing delay constrained schedulers and admission control policies such that users' traffic will meet delay constraints with high reliability. Second, we focus on joint URLLC and eMBB scheduling, guaranteeing delay constraints to the former and minimum rate constraints to the latter. Finally, we focus on resource planning for dynamic interference across the network in the presence of dynamic load and user distribution across base stations – with special attention to service protection for delay constrained users.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1: Introduction

Next generation wireless networks are set to provide ubiquitous connectivity to diverse data traffic classes which have varied performance requirements. In order to meet the complexity of doing so academia and industry have proposed a framework which creates logical networks for each class of service type to enable virtual resource allocation, which can then be mapped to physical resources (e.g., time frequency blocks of Wireless spectrum) and/or compute resources to meet the Quality of Service (QoS) requirements of applications/users. 5G wireless systems aim to support three principle QoS categories: eMBB (enhanced Mobile Broadband), mMTC (massive Machine Type Communications) and URLLC (Ultra Reliable Low Latency Communication). A few significant differences among these use cases are tabulated in Table 1.1. For example, many works in literature assume perfect Channel State Information (CSI) for eMBB, no CSI for MMTC devices [83] and perfect/imperfect CSI for URLLC.

	eMBB	**mMTC**	**URLLC**
Traffic type	Best effort	Periodic	Latency sensitive
Error Tolerance	Medium	High	Very low
CSI	Perfect CSI	No CSI	Perfect/Delayed CSI
Rate requirement	Rate adaptation	Fixed	Low
Connection type	On demand	Periodic	Periodic/Aperiodic

Table 1.1: Differences across next generation communication devices.

A seamless user experience for future wireless applications, such as autonomous driving, online gaming, remote surgery, and augmented/virtual reality will require high speed data transfer with reliable information exchange and guaranteed resource provisioning. Increasing user density, mobility, and higher data demands together with stringent QoS requirements will lead to an ever increasing scramble for limited physical resources. It is therefore essential to design efficient resource allocation schemes that can support a large variety of user types with heterogeneity in QoS needs, traffic characteristics, and channel variations. Furthermore, wireless resource schedulers need to plan to support services in the face of substantial sources of uncertainty, e.g., stochastic user traffic, time varying channel conditions, and dynamic interference to provide reliable communications.

1.1 Background

Wireless networks operate over a limited set of radio frequencies with "regularly" deployed base station infrastructure serving as a central controller/transceiver connecting devices/users to the core network over the given frequency band. A considerable fraction of the wireless spectrum is dedicated to transmitting control and reference signals to users, in order to maintain connectivity and be aware of the channel quality of the users connected to the network. Wireless resources are shared across users connected to a base station over discrete time slots or frequency bands or a slice of both time and frequency, as is the case with Orthogonal Frequency Division Multiplexing Multiple Access (OFDMA) systems. Unlike wired communications, wireless communications are susceptible to severe degradation in the transmitted signals depending upon various factors such as:

- Pathloss – which depends upon the distance from the base station;
- Large scale shadowing – due to penetration loss caused by stationary (e.g., buildings) or non stationary objects (e.g., vehicles) in the environment;

- Small scale fading – as the user receives the wireless signal from multiple paths direct or reflected;
- Mobility – as the users' locations change;
- Interference – from neighboring base stations that operate over the same wireless resources.

Cellular network infrastructure: Previous generations of wireless network architectures had separate pipelines for voice and data, namely, circuit switched voice service and packet switched data service. This enabled simplifying the scheduling policy that the network employed to provide resources to users with different QoS requirements. However, the current cellular network architecture uses an Evolved Packet Core (EPC) framework to provide services to users. Voice or any other service that needs stringent QoS is treated as just another Internet Protocol (IP) service under the EPC's IP service architecture. An illustration of the network architecture is shown in Fig. 1.1 with some of the key components – Mobility Management Equipment (MME), Service Gateway (S-GW), Packet Gateway(P-GW), and radio base station.

Network control/data flow: The MME manages user sessions and authenticates and tracks a user across the network. Serving Gateway (S-GW) routes data packets through the access network. The Packet Gateway (P-GW) manages quality of service and acts as the interface between the Long Term Evolution (LTE) cellular network and other packet data networks. Another component not shown in the architecture, is the Policy and Charging Rules Function (PCRF) which supports service data flow detection, policy enforcement and flow-based charging. User data is broken down into packets which are then queued up at the base station for transmission over discrete

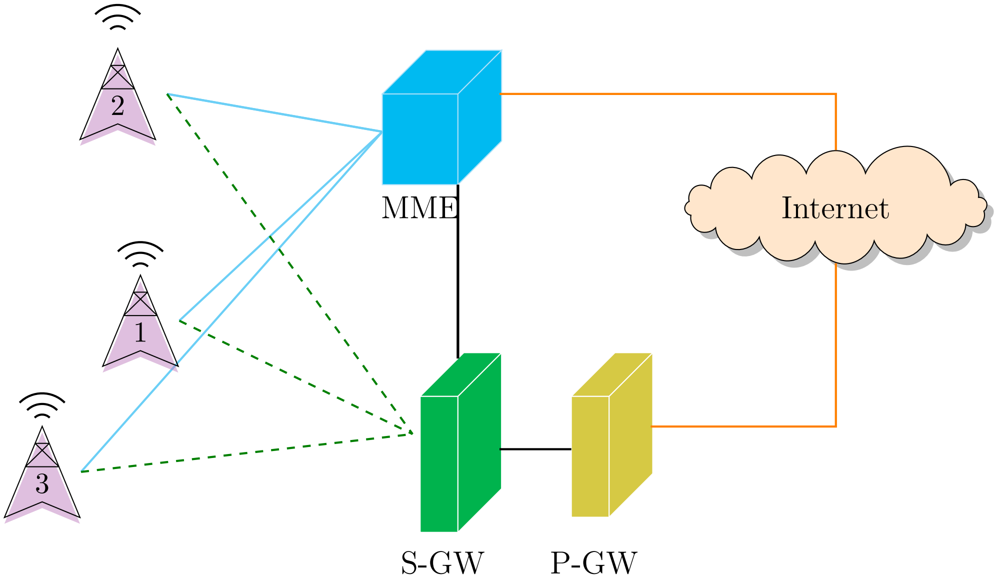


Figure 1.1: Wireless cellular network infrastructure.

time slots. The base station maintains a queue for each user with packets that are ready for downlink transmission waiting to be scheduled. The network has access to only finite transmission resources, so user packets wait for a random time in queue based on the availability of resources.

1.2 Quality of Service

User traffic can be broadly classified into elastic and inelastic traffic, based on their QoS requirements. Commonly known as best effort traffic, elastic traffic can adapt to the network conditions and is not as susceptible as inelastic traffic to variability in transmission errors, delay, and throughput. For instance, applications like web browsing, file download or electronic mail (e-mail) would be representative of best effort traffic with no guarantees on latency or priority in network resource allocation. By contrast, applications such as video conferencing or real time game streaming require guaranteed bit rate services that are highly sensitive to delay, jitter and overall network congestion. The quality of service received by such users can be quantified by key performance indicators such as latency, throughput, delay, fairness.

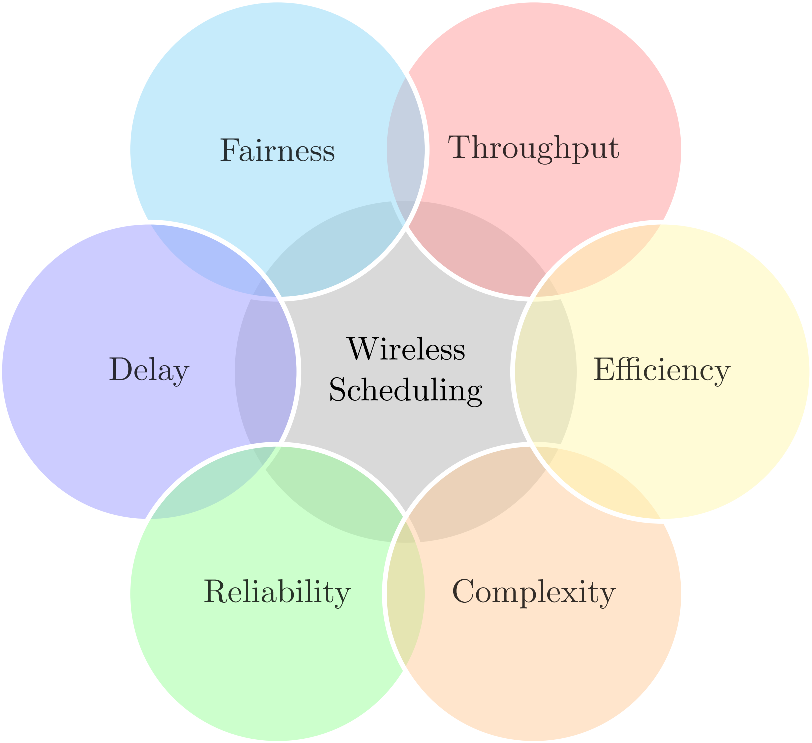


Figure 1.2: QoS requirements and challenges in scheduling wireless users.

Wireless scheduling involves making intelligent decisions about the assignment of resources such as frequency bands, time slots, power levels, and spatial dimensions to different users sharing the network. It is the responsibility of a wireless scheduler to meet the heterogeneous QoS requirements of users competing for the same set of resources as shown in Fig. 1.2. A wireless user's *throughput* refers to the rate at which data is transferred over the network within a given period. An opportunistic wireless scheduler could achieve better sum throughput (across users) by prioritizing users that currently have the best wireless channel conditions, as better channels correspond to higher data rates. However, this can cause queueing delays for users having poor channel conditions. As such, in practice, a certain extent of *fairness* is desirable to provide a satisfactory user experience by ensuring that no single user or device is allowed to hog network resources.

A wireless scheduler should in general maximize the spectral efficiency (i.e., bits per second per Hertz) to support as many users as possible, with an aim to balance the competing user QoS demands. The wireless scheduler should take into account the current information on each user's channel quality, traffic pattern, queue length, and the total number of active users to determine a subset of users to serve at any point in time. This may involve a considerable computational overhead on the network in terms of the scheduler complexity, transmission overhead for tracking channel quality information of all users, and significant protocol overhead to grant resources. The scheduler is also responsible for selecting the most efficient modulation and coding scheme as necessary that achieves a target block error rate associated with user's desired QoS. The interested reader can find a summary of wireless scheduling algorithms in [36, 44].

1.3 Challenges in Wireless Scheduling

QoS Constrained Scheduling: One of the key challenges of wireless communications is to allocate resources to users to meet their desired QoS requirements. In the case of best effort traffic, one could wait for a user to see favorable channel conditions, or simply allocate resources equally among users. However, real time traffic or mission critical applications operate on strict deadlines, where one cannot afford to wait for strong wireless channels. Furthermore, various aspects of the wireless environment such as channel uncertainty, and stochastic nature of user traffic make resource allocation even more challenging for such a scheduler. Packet loss, retransmission and non stationary wireless propagation conditions further complicate scheduling. Heterogeneity in user demands, channel variations, and traffic patterns demands resource schedulers to take into account more variables in their decision-making.

Interference Mitigation: Interference and the associated coupling of scheduling decisions across base stations is another particularly challenging aspect of wireless scheduling. The amount of interference seen by a user depends on various factors such as:

(a) Network topology influences interference from neighboring base stations.
(b) Spatial distribution of users can cause load differences across base stations.
(c) Stochastic traffic variations impact resource utilization.

The asymmetric distribution of user load across base stations further exacerbates the challenges of QoS constrained scheduling in wireless networks. When traffic dynamics and/or the user's locations change, predetermined interference mitigation policies and/or wireless schedulers may become less efficient. Dynamic interference also impacts the accuracy of channel state information and the effectiveness of adaptive modulation and coding leading to increased transmission errors or packet losses.

1.4 Thesis Outline

In this thesis, we introduce several classes of real-time wireless schedulers that exploit the temporal variability in users' channel capacity while maintaining QoS guarantees on user's packet delays or short term throughput. We then introduce a distributed learning algorithm that works in conjunction with a predetermined wireless scheduler to support delay constrained schedulers in the presence of interference.

This thesis is organized into various chapters each of which contributes to different aspects of QoS constrained wireless scheduling.

Chapter 2 presents a measurement-based Opportunistic Guaranteed Rate Scheduling (OGRS) algorithm and analyses its performance. In order to meet a pre-specified user delay deadline, it calculates the service rate for the user queue based on results from Deterministic Network Calculus, assuming leaky bucket constraint. In contrast, Chapter 3 introduces a measurement-based Opportunistic Guaranteed Deadline Scheduler (OGDS) that meets strict delay deadlines by scheduling packet transmissions whenever the current channel rate is better than that expected before packet deadlines expire.

Chapter 4 introduces a class of algorithms for *rate* constrained users that exploit statistical multiplexing gains across users while ensuring their respective *minimum rate* constraints are met. Finally, in Chapter 5 we investigate the problem of distributed resource allocation in wireless systems in the presence of dynamic user traffic, where the service rate of users' queues across base stations is coupled through interference. In particular, we explore a setting where a stochastic game is set up among base stations to learn frequency reuse patterns and solve using multi-agent Reinforcement Learning (RL) given an underlying choice for user scheduling.

Lastly, Chapter 6 summarizes key findings and takeaways derived from our research with a few directions for future exploration.

1.5 Publications

The following is a list of published conference papers and submitted journal versions related to the work presented in this thesis:

1. **G. Chandrasekaran**, G. de Veciana, V. Ratnam, H. Chen, C. Zhang, *"Measurement Based Delay and Jitter Constrained Wireless Scheduling with Near-Optimal Spectral Efficiency"*, Submitted to IEEE Trans. on Netwk., Mar 2024.

2. **G. Chandrasekaran**, G. de Veciana, *"Opportunistic Scheduling for Users with Heterogeneous Minimum Rate QoS Requirements"*, IEEE ICC, Jun 2024.

3. **G. Chandrasekaran**, G. de Veciana, V. Ratnam, H. Chen, C. Zhang, *"Delay and Jitter Constrained Wireless Scheduling With Near-Optimal Spectral Efficiency"*, IEEE PIMRC, Sep 2023.

4. **G. Chandrasekaran**, G. de Veciana, V. Ratnam, H. Chen, C. Zhang, *"Spectrally Efficient Guaranteed Rate Scheduling for Heterogeneous QoS Constrained Wireless Networks"*, IEEE WiOpt, Aug 2023.

5. **G. Chandrasekaran**, G. de Veciana, *"Distributed Reinforcement Learning based Delay Sensitive Decentralized Resource Scheduling"*, IEEE WMLC, Aug 2023.

Chapter 2: Opportunistic Guaranteed Rate Scheduling[*]

The support of Ultra Reliable Low Latency Communication (URLLC) is expected to be critical towards enabling next generation [3] wireless applications such as industrial automation, augmented and virtual reality, autonomous driving, remote diagnosis, and health care. The key challenge in supporting such applications is their stringent constraints on Quality of Service (QoS). The latency constraints for these applications range between 5 and 30 ms, with reliability requirements of 99.9 to 99.9999%, see e.g., [5]. Moreover, given the limited spectrum available and associated costs, it is also critical to deliver such URLLC based services in a spectrally efficient manner. In general, this is challenging, e.g., one must add substantial upfront redundancy to meet reliability requirements without delays associated with re-transmissions, or given low latency requirements one may not be able to exploit opportunism or wait for data to achieve more efficient modes of transmission.

In addition to dealing with the requirements of URLLC traffic, it is also critical to devise resource allocation and scheduling strategies that enable the support, of a mix of traffic, e.g., Enhanced Mobile Broadband (eMBB) and Machine-Type Communications (MTC) traffic, and possibly network slices provisioned to support different classes of applications. Our focus in this chapter will be on spectrally efficient scheduling of wireless user traffic with possibly heterogeneous delay deadlines, perhaps the most challenging traffic class, yet we aim to provide an approach that

[*] Publications based on this chapter: [23] Geetha Chandrasekaran, Gustavo de Veciana, Vishnu Ratnam, Hao Chen, and Charlie Zhang. Spectrally Efficient Guaranteed Rate Scheduling for Heterogeneous QoS Constrained Wireless Networks. In *2023 IEEE 21st International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2023

can be combined with other scheduling policies, e.g., proportionally fair or utility maximizing schedulers used to support eMBB traffic, to manage an assortment of services with diverse QoS requirements. Below we provide a brief summary of related work in this area, focused primarily on scheduling with delay based QoS constraints. We then introduce the key contributions of this chapter.

2.1 Related Work

Wireless scheduling can be based on the user's queue length, channel quality, history of past allocations, etc., and may have multiple objectives including Quality of Service (QoS) and fairness. In settings where users' queues are fully backlogged, perhaps the best known strategies are utility maximizing, i.e., maximizing the sum of users' utilities, which in turn is a function of each user's long term throughput. Perhaps the most popular wireless scheduling often used in practice is the proportionally fair scheduler, see [54], which maximizes the log utility of users' long term throughput. Such an approach results in users to *fair* long term rates with *opportunistic* scheduling in the short term when the channel rate is high. A heuristic algorithm to meet end to end delay constraints is considered in [113]. A more sensitive resource allocation that averts short term neglect of user allocation, see [31], maximizes the user's utility which is a function of the short term throughput. In general, such utility maximizing schedulers work best for elastic traffic with no hard deadlines. In general, utility-maximizing schedulers work best for elastic traffic with no hard deadlines. When hard delay deadlines are considered, either all users have a homogeneous/time-synchronized traffic model [49] or the problem can only be solved if the optimization problem is feasible, i.e., if all users are able to meet the delay deadlines [50]. Several questions such as how to choose the fairness criterion when user queues are not fully backlogged or when there are reliability constraints for meeting strict delays remain unanswered.

In settings where user queues are not fully backlogged but instead driven by stochastic arrivals, various channel state dependent throughput optimal policies (that guarantee queue stability when feasible) have been devised, e.g., [103], and may also achieve different types of delay objectives, e.g., roughly minimizing the max delays across users or overall average delays. While efficient for "best effort" type traffic, such scheduling policies do not deliver strict delay guarantees needed for real-time applications and/or URLLC based services. In [66] the authors consider design of throughput optimal scheduling under heavy-traffic regimes intending to identify an optimal trade off between mean delay and service regularity. Some of these schedulers also address other performance objectives, such as minimizing the max user queue length in [92], or minimizing the average delay as in [87]. Adaptations of throughput optimal schedulers to practical settings, like Modified Largest Weighted Delay First (MLWDF) [11] consider the channel state, head-of-line packet delays, and user weights reflecting QoS objectives for scheduling. Such schedulers offer a graceful degradation of service when there are insufficient resources to meet QoS of all users.

Another interesting line of research borrows ideas from wireline scheduling (e.g., traffic shaping and network calculus [65, 24]) to satisfy user QoS constraints under wireless channel variations. Weighted round robin [64] or weighted fair queueing [67] employ heuristic user weights or tokens [79] based on service deficit [57] to either minimize the average delay or provide a graceful degradation of service. Much of the above mentioned work focuses on scheduling one class of users, or traffic that is sensitive to packet delays. In practice, wireless systems need to be shared by heterogeneous user classes. Packet level deficit tracking for evaluating the QoS service deficit has been considered for each user in [106], however, such an approach is prohibitively expensive in complexity when there are a large number of users. In contrast, we employ cumulative service based techniques and queue based scheduling, avoiding the need to track packet level deadlines or control. Wireless scheduling for optimizing both service regularity and mean delay is considered in [66], but the emphasis is on graceful degradation rather than guaranteed latency. Although [28] considers scheduling with reliability for homogeneous user QoS requirements, it is assumed that only one user can be scheduled every time slot which is a severe limitation under practical scenarios.

Scheduling with guaranteed QoS is considered in [7], with no improvement in spectral efficiency for latency constrained users. Joint resource allocation for URLLC and eMBB traffic is proposed in [8] but opportunistic scheduling is limited to eMBB users. More recent literature on URLLC scheduling [33, 58, 115, 6, 71] include reliability guarantees, however, opportunistic scheduling has not been given much consideration apart from the perspective of energy efficiency [96] or the violation of deadline probability [112] or devising token based quality assurance [79] which may starve weaker users until one is forced to schedule close to their deadline. Previous research can only be considered a first step towards a more profound understanding of developing spectrally efficient algorithms for delay constrained traffic. To the best of our knowledge, a simple approach to opportunistic scheduling over temporal channel variations for deadline constrained traffic has not been considered in the existing literature. In this chapter, we propose a new class of opportunistic scheduling algorithms for URLLC users with heterogeneous traffic and disparate QoS requirements, to enhance the throughput/utility for eMBB traffic that also shares the overall network resources.

Studies on QoS provisioning cannot be considered complete without addressing the question of admission control and/or traffic shaping/policing. A closer look at existing literature reveals that much of the work on wireless scheduling does not solve this problem. Given the wireless channel uncertainty, it is infeasible to predict if a given scheduling policy will be able to meet the user's QoS constraints with high reliability. Given the uncertainty and heterogeneity associated with traffic, channels, and user requirements in a wireless system, it is virtually impossible to devise good models that would allow one to predict if the users' QoS requirements will be met under a given scheduling policy.

No practical wireless scheduling policy is complete without a complementary strategy for admission control and/or traffic shaping. Given the uncertainty and heterogeneity associated with traffic, channels, and user requirements in a wireless system, it is virtually impossible to devise good models that would allow one to predict if the users' QoS requirements will be met under a given scheduling policy. While there have been many works in literature that propose measurement based admission control [41, 42, 107], we note that meeting packet delay and loss targets in buffered systems is challenging [18]. In contrast to traditional Measurement Based Admission Control (MBAC) approaches, our approach directly measures the aggregate resource that our delay constrained schedulers are using thus indirectly capturing the impact of the users' traffic, channel variability and delay constraints.

2.2 Chapter contributions

We propose a class of wireless schedulers that under appropriate assumptions can meet heterogeneous delay deadlines and do so in a spectrally efficient manner such that the more relaxed the constraint to more efficient. We design a scheduler that evaluates the rate required for each user's QoS requirements and provision resources such that delay deadlines can be met. To do so, the scheduler needs information on each user's queue length and data traffic. If a user's traffic is leaky bucket constrained, one can determine a fixed service rate that will ensure a desired maximum delay. The assumed traffic shaping serves to constrain the peak *burstiness* of the arrival process. The key underlying idea is to leverage the flexibility of wireless systems, in terms of allocating a time varying number of Resource Blocks (RB) to overcome/exploit variations in wireless users' capacity per RB. This permits one to devise a scheduler, the Wireless Guaranteed Rate Service (WGRS), which will ensure a user will see a fixed service rate even with channels that have stochastic variations.

In fact, any scheduler which allocates at least as much cumulative service as the WGRS scheduler over busy periods is GRS *compliant*, and will thus also meet the user's delay deadlines. This observation suggests the possibility of opportunistically serving a user's data ahead of time when channel rates are good, relative to the GRS scheduler, and/or delay such service when channel rates are poor, as long as the scheduling is GRS compliant. We devise a class of Opportunistic GRS (OGRS) schedulers that take advantage of this relaxation along with knowledge of the statistics of the users' channel variations, to achieve better spectral efficiency while meeting users' strict delay constraints. At each time slot, the scheduler allocates resources to users with good channel quality. The scheduler determines the channel quality based on either a static threshold on the current channel's percentile or dynamic thresholds which further account for the extent to which the scheduler has served traffic ahead of the GRS scheduler.

By considering *oracle-aided* policies that have access to future channel capacity realizations, we show via extensive simulations that OGRS can be within 10% to 40%

of such policies as the delay constraint is relaxed. These gains translate to doubling the eMBB user's throughput even for the weakest user when URLLC and eMBB traffic share resources or an increase of upto 57% in the number of users admitted as long the arrival rates and channel strengths are similar for the newly admitted users.

Finally, we propose an MBAC strategy, that indirectly accounts for the heterogeneity in traffic, channel, and delay constraints by directly tracking resource usage statistics based on the resource allocation algorithm of our proposed OGRS scheduler. While this approach may be more robust to uncertainty, it may fail from time to time, unlike previously considered MBAC policies, and may have to resort to prioritizing a particular class of users.

This chapter is organized as follows. Section 2.3 describes the system model and traffic shaping assumptions. Sections 2.4 and 2.5 describe our proposed Guaranteed Rate Scheduling (GRS) algorithms for delay constrained scheduling with multiple options to select adaptive channel rate thresholds to identify the best time slots for transmission. Section 2.6 presents the main theoretical results of this chapter. Section 2.7 provides extensive simulations for some practical wireless network settings, and also evaluates the spectral efficiency of a natural class of delay constrained schedulers that use neural network based channel rate predictions. Finally, Section 2.8 includes some concluding remarks.

2.3 System Model

We consider discrete time downlink scheduling for a base station serving a set $\mathcal{U}$ of URLLC users with stochastic arrivals and possibly heterogeneous QoS requirements. We denote by $(A_n^u)_{n\in\mathbb{N}}$ the arrival process for user $u \in \mathcal{U}$, where A_n^u is a random variable denoting the number of bits that arrive and are available for service in time slot n with a transmission deadline of $n+d$. In general, it is not possible to ensure delay guarantees to a user without prior knowledge of its traffic statistics or of constraints on its traffic. A common approach for the latter is to establish and enforce (through traffic policing/shaping) apriori constraints on the user's traffic that can be used to design resource allocation mechanisms guaranteed to meet a user's QoS requirements. We will assume each user's traffic satisfies dual leaky bucket constraints [65] with parameters $(\rho^u, \sigma^u, \mu^u)$, where σ^u denotes the token bucket size in bits and ρ^u, μ^u denote the peak and mean bit arrival rate per time slot, respectively. The user's cumulative arrival process $A^u(\cdot, \cdot]$ is thus constrained as follows for all $\tau, n \in \mathbb{N}$,

$$A^u(\tau, \tau+n] = \sum_{k=\tau+1}^{\tau+n} A_k^u \leq \min\left[\rho^u n, \sigma^u + \mu^u n\right] . \tag{2.1}$$

The base station transmit resources are modeled as a sequence of frames/slots each comprising multiple Resource Blocks (RBs) which can be arbitrarily allocated to users on a per time slot basis by the scheduling policy. Each RB denotes a slice of time and frequency block available to the BS for resource allocation. We let the random variable $C_n^u \in \mathbb{R}^+$ denote the channel rate (bits per RB) that can be transmitted to user u if it is allocated a *single* RB on time slot n. A user may be allocated multiple RBs, but we assume a *flat fading* setting where the rate delivered to u is the same across RBs in a given time slot. Further, we assume $(C_n^u)_{n\in\mathbb{N}}$ are independent and identically distributed (i.i.d.) across time slots. A non zero transmission rate C_n can be viewed as a coverage/connectivity requirement for users.

Assumption 2.3.1. *(Connectivity Assumption) The BS can transmit data over an RB at a non-zero channel rate* $C_n^u > 0$ *with probability 1.*

Additionally, we also assume that a sufficiently large number of RBs are available every time slot to meet the users' QoS requirements. However, our goal is to devise schedulers for URLLC users that use a minimal number of RBs. In the sequel, we will introduce admission control strategies in the event that only limited resources are available to URLLC users.

We consider a system model where a scheduling policy, say π, decides the number of RBs be allocated to each user in each time slot. The decision of policy π at time n is assumed to be causal concerning knowledge of the current and past channel rates $(C^u_\tau)^n_{\tau=0}$, arrivals and queue lengths, allowing for opportunistic scheduling, i.e., taking advantage of capacity variations across time. In particular, we let $M^{u,\pi}_n \in \mathbb{R}^+$ denote the number of RBs allocated to user u on slot n by a policy π given the observed history. Such an allocation provides an overall service rate $S^{u,\pi}_n$ (total bits transmitted with potentially multiple RBs allocated) to the user u on time slot n given by,

$$S^{u,\pi}_n = M^{u,\pi}_n C^u_n,$$

and we define the cumulative service over an interval $(\tau, \tau+n]$ as follows,

$$S^{u,\pi}(\tau, \tau+n] = \sum_{k=\tau+1}^{\tau+n} S^{u,\pi}_k \,. \tag{2.2}$$

A user's data queue (in bits) is modeled as a First Come First Serve (FCFS) discrete time queue with arrivals A^u_n and service rate $S^{u,\pi}_n$ as shown in Fig. 2.1. We let $Q^{u,\pi}_{n+1}$ denote the number of bits in the user's queue at the start of slot $n+1$, then

$$Q^{u,\pi}_{n+1} = [Q^{u,\pi}_n - S^{u,\pi}_n]^+ + A^u_{n+1} \,. \tag{2.3}$$

2.4 Guaranteed Rate Scheduling

In this section, we will assume user traffic is leaky bucket constrained, whence assuming a user's data queue served in FCFS order, *it's delay requirement will be met*

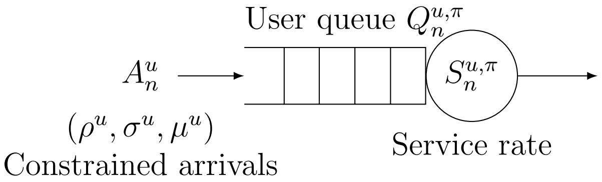


Figure 2.1: Leaky bucket constrained arrivals to a discrete time queue with a service rate controlled by scheduling policy π.

through a sufficiently high fixed service rate per slot. Note that in practice wireless capacity varies over time, yet we will start by introducing this example where the user's rate is *fixed* and later consider how to address user's channel variability across time. Without loss of generality, we shall henceforth present the analysis for a *single user* with traffic shaping parameters (ρ, σ, μ) and delay requirement d. Referring to the network calculus literature [65], the minimal service rate s required to meet the user's delay constraint d must satisfy,

$$d \leq \frac{[\rho - s]^+}{\rho - \mu} \frac{\sigma}{s} \implies s = \frac{\rho\sigma}{(\rho - \mu)d + \sigma}. \tag{2.4}$$

where $[x]^+ = \max[x, 0]$. *As long as a user is allocated enough resources to meet the service rate of s, it will meet the packet level delay requirement owing to leaky bucket constrained traffic.* This is easily visualized, see Fig. 2.2. The red curve is the worst case cumulative arrivals for leaky bucket constrained traffic, the blue line a fixed rate service, and the green interval the worst case delay a bit must wait until service.

Unfortunately, wireless channel strength can vary significantly over time and hence we return to the original channel rate construction as described in the system model, a leaky bucket constrained traffic with random transmission rate every time slot. Clearly, for a delay deadline d if the BS can provide a service rate of s every time slot, irrespective of the channel fluctuations, then the user delay constraint can be met with probability 1. This can be achieved by varying the number of RBs allocated

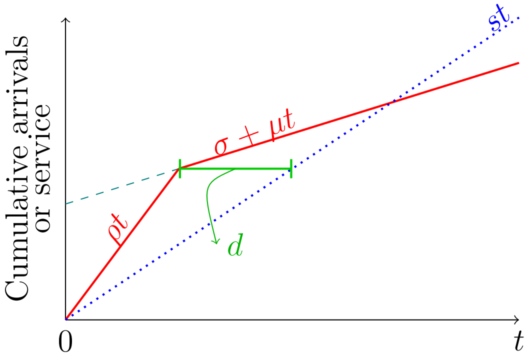


Figure 2.2: Leaky bucket constrained flow arrival and service curves for deterministic service rate.

to the user as a function of the channel quality. For instance, the service rate per RB available to the user is (the dash-dotted line) illustrated in Fig. 2.2, i.e., above the deterministic rate line st, then the delay deadlines will be guaranteed despite channel variability. Mathematically, the service rate over an interval $(\tau, \tau+n]$ for a user with variable channel rate C_k is given by,

$$S^{\pi}(\tau, \tau+n] = \sum_{k=\tau+1}^{n} S_k^{\pi}, \text{ where } S_k^{\pi} = M_k^{\pi} C_k \,, \tag{2.5}$$

where k denotes the discrete time index and M_k^{π} denotes the number of RBs (possibly fractional) allocated at time k.

As long as we can ensure that $S^{\pi}(\tau, \tau+n]$ stays above sn throughout any interval $(\tau, \tau+n]$, the bound on the maximum delay d will still hold. In the following we propose causal scheduling policies that shall provide delay guarantees to the user with probability 1.

Definition 2.4.1. ***GRS(s)*** *We let the Guaranteed Rate Scheduler with service rate s, GRS(s), be the scheduling policy that guarantees a user data queue a service rate of at least s per slot whenever it is sufficiently backlogged.*

The above definition matches with that of a strict service curve in Deterministic Network Calculus [65].

Definition 2.4.2. ***WGRS(s)*** *A wireless GRS(s) scheduler for a user with time varying channel rate C_n bits per RB allocates a time varying number of RBs M_n to the user such that at time n,*

$$M_n = \min \left[\frac{s}{C_n}, \frac{Q_n}{C_n} \right], \tag{2.6}$$

where C_n is the channel rate at time n and Q_n denotes the number of bits in the user's queue, resulting in an overall service rate $S_n = M_n C_n$.

Under these policies, as long as the user's queue is sufficiently backlogged the user will see a service rate s. Since the GRS(s) scheduler satisfies the user's delay constraints for the appropriately selected s, so will the wireless version, although it may require the allocation of a large number of RBs if the user's channel capacity is low. Recall we consider a setting where there is a sufficiently large number of RBs available to users, and they are unlikely to require a lot of resources at the same time. Further note that although the scheduler is designed based on worst case analysis, resources are only be allocated if needed, i.e., only if the user queue is backlogged, hence no resources would be wasted, indeed they will be allocated to other users.

2.5 Opportunistic GRS schedulers

Following the setting in Section 2.4, we shall propose a new class of wireless schedulers that we refer to as Opportunistic GRS(s) schedulers that have additional flexibility to exploit temporal variations in users channel capacity, yet are guaranteed to meet the user's delay requirements. To that end, we first formally define a property that ensures the policy will meet the same delay requirements as GRS(s) when users' traffic is leaky bucket constrained.

Definition 2.5.1. ***GRS(s) compliant*** *A scheduling policy π is GRS(s) compliant if when subject to the* same *arrivals process for any busy cycle of the GRS(s) scheduler, say $(0, b]$, the cumulative service of π over the interval $(0, \tau]$ for $\tau = 1, 2, \ldots, b$ is greater than or equal to that of the GRS(s) scheduler. It follows that the user queue under GRS(s) compliant policy will empty out whenever that under the GRS(s) scheduler empties.*

Since a GRS(s) compliant policy's cumulative service (departures) is greater than that of the GRS(s) policy on any busy cycle, it is clear that it can only speed up departures and thus reduce delays in FCFS user queues. We note that GRS(s) compliance differs from the traditional service curve definition, see e.g., [65, 24]), in that it is defined via a coupling of π to the GRS(s) policy on busy cycles, and in particular it is not shift invariant, i.e., under π it is possible to have an interval in which queues are backlogged and there are no departures.

While opportunistic schedulers have been employed to efficiently allocate resources for best effort traffic, the role of opportunism in scheduling traffic with deadlines & reliability requirements is far less understood. Usually opportunism in the wireless literature is considered across users with varying rates, where the decision is to schedule users with high rates, long queues and/or best marginal utility. The design of GRS in the previous subsection is based on worst case delay analysis which is not opportunistic. In this section, we explore opportunism over a single user's temporal service rate variability, i.e., the wireless time varying channel could perhaps be better exploited by transmitting more data whenever the received signal strength is stronger.

Fig. 2.3 exhibits a sample realization. In the figure, the red curve shows the cumulative arrivals to a GRS(s) busy cycle beginning at time 0 – corresponds to the worst case cumulative arrivals associated with a dual leaky bucket. The dotted line represents the cumulative service at a fixed rate s. Meanwhile the blue cumulative service curve corresponds to a policy π. As can be seen, from the start of the busy cycle at time 0 to the end at time b, the cumulative service of policy π exceeds that of the fixed rate service and so GRS(s) compliant.

Fig. 2.3 also exhibits the perspective underlying Opportunistic GRS scheduling. *The key idea is to exploit temporal channel rate variability to improve spectral efficiency without impacting delay guarantees.* We observe that at times n_1 and n_3 the user's channels are particularly good, and the user has queued data significantly higher than s. The scheduler chooses to exploit these good user channels, by serving much more data at those times than the minimal service rate required by GRS(s) scheduling. In principle, since the user's channel is good at those times, the number of RBs the wireless scheduler would be allocating by doing so would be reduced as compared to the WGRS(s) scheduler introduced in the previous section. Next, we formally introduce a class of Opportunistic GRS(s) scheduling policies.

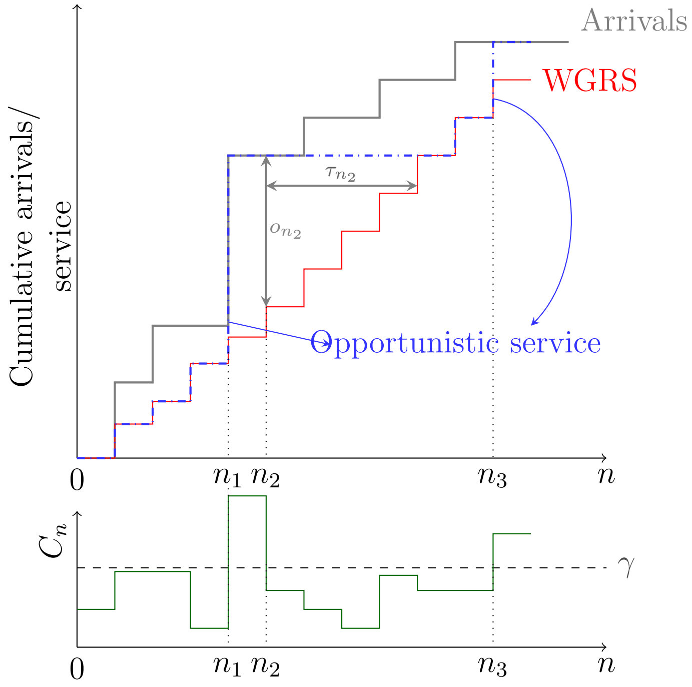


Figure 2.3: Temporal channel variations and opportunistic service based on bits in queue. The wireless channel rate variations are shown in the figure with a threshold γ to determine the channel quality.

Definition 2.5.2. ***OGRS(s)*** *An Opportunistic GRS(s) scheduling policy π subject to an arrival process $(A_n)_n$ simulates the GRS(s) scheduler and allocates RBs and thus service $S^\pi(\cdot,\cdot]$ to the user such that for any GRS(s) busy cycle, say $(0,b]$, we have*

$$A(0,\tau] \geq S^\pi(0,\tau] \geq S^{GRS}(0,\tau] \quad \text{for} \quad \tau = 1,2,\ldots,b,$$

where $A(0,\tau]$ denotes the cumulative arrivals and $S^{GRS}(0,\tau]$ the cumulative service allocated by GRS(s) since the beginning of the busy cycle.

By definition, OGRS(s) schedulers are GRS(s) compliant, and thus will satisfy the user's delay requirements if s is chosen appropriately, relative to the arrival processes' leaky bucket parameters. However, such schedulers have the additional freedom to decide when to allocate RBs to the user and in particular, to do so when the channels are particularly good.

Definition 2.5.3. ***Threshold-based Opportunistic GRS(s)*** *The basic principle underlying a threshold-based OGRS(s) scheduling policy π is as follows: if on time slot n the channel rate C_n exceeds a threshold γ_n^π, then a sufficient number of RBs are allocated by the scheduler to clear the queue backlog, i.e, $M_n^\pi = Q_n^\pi / C_n$. Otherwise, a minimal number of RBs are allocated so as to ensure the cumulative service allocated keeps up with that of the GRS(s) scheduler over its busy cycles.*

Note, that the threshold γ_n^π can be time/state dependent and controls how the algorithm exploits channel rate fluctuations – this will be explained in the sequel.

Algorithm 1 exhibits the details of the threshold based OGRS(s) scheduler which operates with respect to the cumulative service a virtual GRS scheduler would provide for the same arrival process. To start with, consider a GRS(s) busy cycle that without loss of generality begins at 0. Then one can express the user queue length and service for the GRS(s) at time n as follows,

$$Q_n^{\mathrm{GRS}} = [Q_{n-1}^{\mathrm{GRS}} - s]^+ + A_n,$$

$$S_n^{\mathrm{GRS}} = \min[Q_n^{\mathrm{GRS}}, s].$$

Therefore, the cumulative service provided by GRS(s) over an interval $(0, n]$ can be expressed as,

$$S^{\mathrm{GRS}}(0, n] = S^{\mathrm{GRS}}(0, n-1] + \min[Q_n^{\mathrm{GRS}}, s]\,.$$

Next, we have the OGRS policy which needs a metric that can measure the amount of service provided in excess of the guaranteed rate s per time. slot. Let O_n^π denote the amount of data that has been (opportunistically) sent ahead of time n relative to the GRS(s) scheduler, i.e.,

$$O_n^\pi = S^\pi(0, n-1] - S^{\mathrm{GRS}}(0, n-1].$$

We initialize $O_0^\pi = 0$ at the start of a busy cycle. Note that the duration of a busy cycle of a GRS(s) compliant scheduling policy with leaky bucket constrained arrivals is upper bounded [65] by $b_{\max}(s) = \frac{\sigma}{s-\mu}$, which bounds O_n^π and guarantees it will eventually return to 0. Note that any amount of opportunism O_n^π gained translates to the scheduler being $\tau_n = \left\lceil \frac{O_n^\pi}{s} \right\rceil$ time slots ahead of the service deadline, see $o_{n_2}^\pi$ and τ_{n_2} as marked in Fig. 2.3.

As mentioned in the policy Definition 2.5.3, if $C_n > \gamma_n^\pi$ then π serves all the data in the user queue, i.e., $S_n^\pi = Q_n^\pi$. Clearly, the metric O_n^π must be positive if $Q_{n-1}^\pi > s$, because π has served all the traffic that has entered the queue since the start of the busy cycle, while GRS(s) only the bare minimum service it guarantees.

When the channel rate is not so good, i.e., if $C_n \leq \gamma_n^\pi$ then OGRS can choose to not schedule any RBs if at least s bits had been transmitted in advance. Specifically, if the amount of excess service at time $n-1$ falls short of s then π only serves the minimum number of bits to ensure it keeps up with the GRS(s) scheduler, i.e.,

$$S_n^\pi = \left[\min[s, Q_n^\pi] - O_n^\pi\right]^+ .$$

2.5.1 OGRS Threshold selection

In this section, we propose various ways to design the thresholds $(\gamma_n^\pi)_n$ driving the behavior of the threshold-based OGRS scheduler. We shall assume that the scheduler has access to $F_C(\cdot)$, the CDF for the users' channel rate variations. We

Algorithm 1: Guaranteed Rate Scheduling with opportunism over temporal variations

```
initialize $O_0 = 0, S_0^\pi = 0, S_0^{\text{GRS}} = 0$ ;
while $n > 0$ do
    if $C_n > \gamma_n^\pi$ then
        $S_n^\pi = Q_n^\pi$ ;
    else
        $S_n^\pi = [\min[s, Q_n^\pi] - O_n^\pi]^+$ ;
    end
    $M_n^\pi = S_n^\pi / C_n$;
    $S_n^{\text{GRS}} = \min[s, Q_n^{\text{GRS}}]$ ;
    $Q_{n+1}^{\text{GRS}} = Q_n^{\text{GRS}} - S_n^{\text{GRS}} + A_{n+1}$ ;
    $Q_{n+1}^\pi = Q_n^\pi - S_n^\pi + A_{n+1}$ ;
    $O_{n+1}^\pi = S^\pi(0, n] - S^{\text{GRS}}(0, n]$ ;
end
```

also assume that $F_C^{-1}(\cdot)$ is an appropriately defined inverse CDF. As explained in the sequel, in practice the CDF can be inferred, as in [80] to possibly adapt to changes over time.

2.5.1.1 ST(α): Static threshold

A simple way to select channel quality threshold is to identify a quantile or percentile value above which the channel strength is considered high. Whenever the current channel quality is above the chosen quantile or percentile, the user can transmit all data in it's queue with RBs allocated accordingly. Our first threshold design is a static percentile, i.e., $\gamma_n^\pi = \gamma^\pi$ corresponding to the α-percentile of the channel rate CDF, where $\alpha \in (0, 1)$, so,

$$F_C(\gamma^\pi) = \alpha \implies \gamma^\pi = F_C^{-1}(\alpha). \tag{2.7}$$

For example, with a choice of $\alpha = 0.8$ the OGRS(s) triggers an opportunistic scheduling of the user's queued data only if the current channel rate has exceeded the 80^{th} percentile, i.e., $c_n > \gamma^\pi$. Note that the choice percentile α is a design parameter that can in principle be optimized to minimize the mean resources (RBs) allocated by the associated OGRS(s) scheduler.

2.5.1.2 DTP(δ): Dynamic threshold based on probability

Next we consider thresholds based on a dynamic percentile of the channel rate CDF $F_C(\cdot)$. Recall that $O_n^\pi = o_n^\pi$ denotes the amount of data that our OGRS(s) policy has delivered *ahead* of time as compared to GRS(s) at time n. Given that the GRS(s) must serve at least s bits per slot, an OGRS(s) policy could in principle wait for $\tau_n = \left\lfloor \frac{o_n^\pi}{s} \right\rfloor$ time slots before the GRS(s) scheduler catches up and is forced to schedule at time $\tau_n + 1$. Ideally the data should be scheduled on slot n if the current rate realization c_n is better than that to be observed in the next $\tau_n + 1$ time slots with high probability, i.e.,

$$\mathbb{P}\left(c_n > \max_{i=1,\ldots,\tau_n+1} C_{n+i}\right) \geq \delta. \tag{2.8}$$

The following lemma translates the above requirement to a threshold on c_n. Note that δ is a design parameter that needs to be carefully chosen so as to minimize the number of RBs required.

Lemma 2.5.1. *Let $(C_n)_n$ be i.i.d random variables with the same marginal distribution $F_C(\cdot)$ and appropriately defined inverse $F_C^{-1}(\cdot)$. If c_n exceeds the threshold $F_C^{-1}\left(\delta^{\frac{1}{\tau_n+1}}\right)$ then (2.8) is satisfied.*

Proof. See Appendix A.1. ■

2.5.1.3 DTE: Dynamic threshold based on expectation

A user with current channel rate c_n might choose not to schedule transmissions on the current slot in the hope of seeing a better channel in the next τ_n slots. The previous threshold design was based on the inequality in (2.8) being satisfied with high probability. Alternatively, the current channel rate c_n might be considered good if one can ensure the inequality holds on average. Taking the expectation of the inequality on the right hand side of and computing the expectation of the max of uniform random variables in (A.1) gives,

$$F_C(c_n) > \mathbb{E}\left[\max_{i=1,\ldots,\tau_n+1} U_i\right] = 1 - \frac{1}{\tau_n + 2}\,.$$

Under this rough approximation an associated threshold on c_n depends on τ_n which can be set to,

$$\gamma_n^{\pi} = F_C^{-1}\left(1 - \frac{1}{\tau_n + 2}\right). \tag{2.9}$$

This captures the key insight that with a larger number of slots τ_n, an OGRS(s) scheduler can choose to wait until the channel rate exceeds the $1 - 1/(\tau_n + 2)$ percentile. Furthermore, this threshold selection mechanism does not have any design parameter which makes it easier to implement in practice.

2.6 Stochastic Dominance of OGRS over WGRS

Consider a user with an arrival process $(A_n)_n$ and a time varying channel rate $(C_n)_n$ per resource block, whose traffic is subject to a delay constraint of at most d slots. The arrivals A_n are leaky bucket constrained (ρ, μ, σ) where bits arrive and are available for service at time n. Then the following theorem establishes how the number of RBs required for a given user by the WGRS policy stochastically dominates that of OGRS-DTE.

Theorem 2.6.1. *For a system in steady state, the mean RBs required by OGRS per time slot is stochastically dominated by that required by WGRS.*

$$\mathbb{E}[M^{WGRS}]\geq\mathbb{E}[M^{OGRS}]. \tag{2.10}$$

Proof. See Appendix A.2 for proof and Section 3.6.7 for simulation based results of the above theorem. ■

The above theorem establishes the superiority of the OGRS-DTE policy over the strict sense service policy WGRS in the ergodic sense, which directly implies that the OGRS-DTE enables the overall network scheduler to support existing eMBB users at a higher data rate than the WGRS policy.

2.7 Simulation results

We consider a BS serving a set of URLLC with each user's channel rate C_n per RB determined by the corresponding received Signal to Noise Ratio (SNR). The received SNR was modelled using the 3GPP Urban-Micro path loss model [3], with Rayleigh distributed small scale fading. We assume bounded channel realizations, where the SNR lies between -6.934 dB $\leq$ SNR $\leq$ 20 dB. Underlying this assumption is the notion that the network has been well engineered so that users are covered by at least one BS with reasonable SNR. reasonable as in practice users one would expect high overlapping coverage from multiple BSs where the user has non-negative rate support from at least one BS. For simplicity, we shall use Shannon capacity $B\log_2(1+\text{SNR}_n)$ to calculate the rate per RB C_n, where each RB is a time frequency slice of duration 1ms with bandwidth $B = 10$ KHz. We initially use Shannon capacity $B\log_2(1+\text{SNR}_n)$ to calculate the rate per RB C_n, where each RB is a time frequency slice of duration 1ms with bandwidth $B = 10$ KHz. Later on, we demonstrate our performance using realistic 3GPP modulation and coding scheme [3] taking one of 15 quantized values in -6.934 dB $\leq$ SNR ≤ 20 dB.

In our simulations, we assumed random numbers of packets of constant size (1KB) that can arrive each time slot, but they are shaped by a leaky bucket with parameters (in packets per time slot) $\sigma = 50, \rho = 10, \mu = 5$. We considered two kinds of traffic models: Bernoulli arrivals with parameter $\frac{\mu}{\rho}$ and ON-OFF burst arrivals where packets arrive at a peak rate ρ during the ON period. The ON, OFF cycles are of duration $\frac{\sigma}{\rho-\mu}, \frac{\sigma}{\rho}$, respectively. The guaranteed packet rate s per time slot for the given leaky bucket parameters are determined using (2.4) for the specified. The efficiency of a policy is measured in terms of the average number of RBs required to serve the user, subject to its delay deadline.

Henceforth, we shall refer to OGRS scheduling only by the threshold methods that we employ: OGRS with static percentile threshold as ST(α), OGRS with dynamic threshold in expectation as DTE and OGRS with dynamic threshold in probability as DTP(δ). The aim is to demonstrate the reduction in resource usage when we employ these OGRS policies as compared to the WGRS(s). We also explore the sensitivity of these efficiency gains to various system parameters such as: the number of past channel samples used to estimate the empirical distribution of the users channel quality, coding rates corresponding to 3GPP Modulation and Coding Scheme (MCS), wireless connectivity outages, and the delay deadline d.

Finally, to determine the channel quality thresholds, we need the CDF $F_C(\cdot)$ for the channel rate per RB on a given slot, for each user. We used the last 100 channel SNR realizations to determine the empirical CDF of the user's SNR at any given instant. Note that all plots in this section were generated over 10^6 slots, resulting in a ±0.1 error for the estimated mean number of allocated RBs per slot $\bar{M}^{\pi}$ with 99% confidence interval. Additionally, we will use the WGRS policy as a baseline and also the multicarrier version of MLWDF [11] policy as a benchmark for evaluating the performance of our proposed algorithm.

2.7.1 Spectral efficiency

For this subsection we consider three different URLLC users that are at distances 300, 500 and 700 meters and which we refer to as strong, medium and weak users, respectively. Also, we assume each URLLC user has stochastic arrivals with packet size of 1024 bits that arrive each time slot shaped by the leaky bucket with parameters (in packets per time slot) $\sigma = 50, \rho = 10, \mu = 5$. The guaranteed packet rate s per time slot is then determined using (2.4) for a common delay deadline of $d = 4$ ms for all the users. The efficiency of our proposed scheudling policy is measured in terms of the average number of RBs required to serve the user, subject to its delay deadline. Henceforth, we shall refer to the proposed OGRS scheduling policy by the threshold method that we employ to determine the channel rate quality.

Fig. 2.4 shows the percentage reduction in the average number of RBs required to serve all three types of users with ST(α) relative to that needed by WGRS under the same arrival and channel rate processes. It is interesting to note that the weak user sees the best gain, which can be attributed to the higher range of channel strength variations that a typical weak user would observe. Higher temporal variability should lead to higher opportunistic gains for such users. Also, it is clear that the percentile α that maximizes the spectral efficiency gain depends on the arrival pattern and user channel strength.

Finally, we also evaluated the efficiency gains of DTE vs WGRS(s) and the

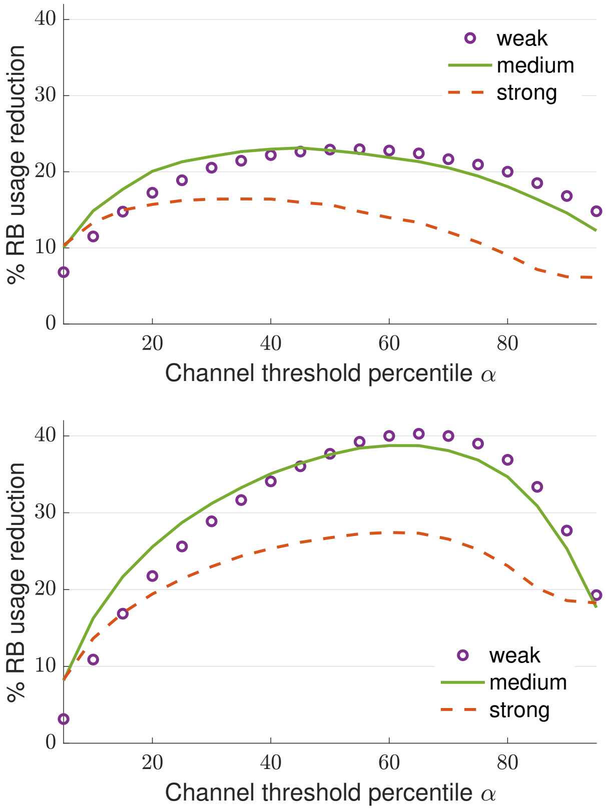


Figure 2.4: Percentage reduction in RB usage for ST(α) with respect to (w.r.t) that of WGRS. The top figure corresponds to efficiency gains for Bernoulli arrivals and the bottom for burst arrivals.

results were as follows: the percentage reduction in the number of resources required for each type of strong, medium, and weak, users were $26.79\%, 37.47\%, 37.67\%$ for the strong, medium and weak user, respectively. The DTE policy achieves at most 5% loss in efficiency as compared to DTP(δ) policy, thus providing a reasonable approach as it requires no parameter fine tuning.

2.7.2 Sensitivity to history

Next, we considered the impact of the number of channel samples used to estimate the empirical CDF evaluation on the spectral efficiency. Fig. 2.6 shows the

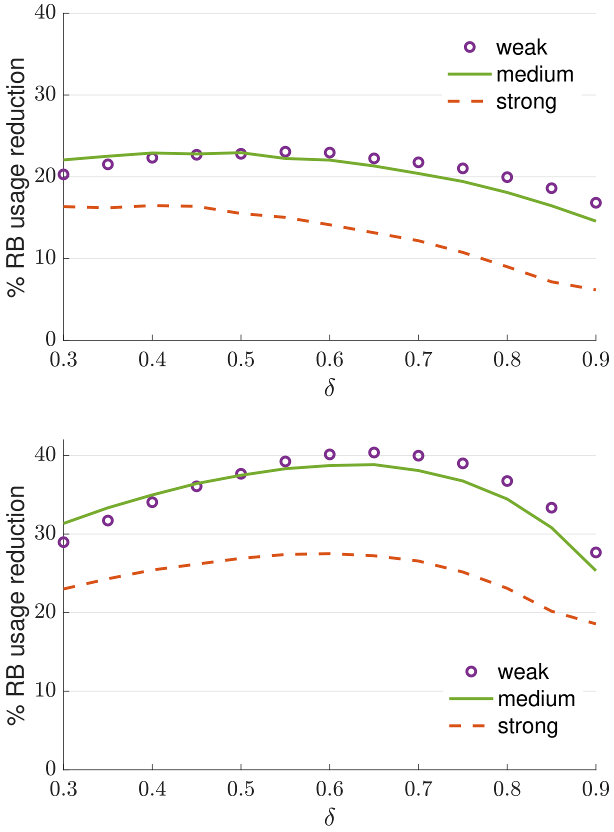


Figure 2.5: Resource reduction percent for DTP(δ) w.r.t. WGRS. The top figure corresponds to efficiency gains for Bernoulli arrivals and the bottom for burst arrivals.

spectral efficiency of the proposed OGRS-ST(α) algorithm, for two different of the number of past samples used to estimate the empirical distribution of the channel rate CDF.

We can see that for as little as 10 samples of the past channels, we are within a 5% of the spectral efficiency when using 100 samples for estimating the CDF. Clearly, a short history such as ten past samples is sufficient to track the wireless channel variations without significant loss in the efficiency of our scheduling algorithms.

2.7.3 Admission Control

We consider a set of 100 users with ON-OFF bursty traffic and leaky bucket constrained arrivals. The delay deadline and user location (distance from the BS) are drawn uniformly random from the sample spaces $\{200, 250, \ldots, 800\}$ and $\{2, 3, \ldots, 10\}$, respectively. This subsection and the remaining subsections in this chapter consider the bursty ON-OFF traffic model. The arrivals and channel variations are generated over 10^6 time slots to simulate the number of users that can be admitted for various system capacities $\overline{m}$, the total number of RBs available in the system. For the same set of users, we also use the Gaussian approximation for the aggregate resource requirement X_u to determine the number of admitted users, shown as dashed lines in Fig. 2.7. If Y is the random variable that denotes the resource requirement of a new user, then the probability that the total resource requirement $X_u + Y$ will exceed $\overline{m}$ is approximated using the following inequality,

$$\mathbb{P}\left(X_u + Y > \overline{m}\right) \leq \exp\left(-\frac{(\overline{m} - \mu)^2}{2\sigma^2}\right), \tag{2.11}$$

where $\mu = \mu_u + \hat{\mu}$ and $\sigma^2 = \sigma_u^2 + \hat{\sigma}^2$. Note that the inequality in (2.11) provides a computationally reasonable expression that can be used to decide if the new user can be admitted without exceeding the reliability requirement δ.

It can be seen that the Gaussian approximation provides a conservative estimate of the number of users that can be admitted into the system for both the OGRS and WGRS scheduling policies. Note that OGRS scheduling policy is able to admit more users when compared to the WGRS policy, which is interesting given that WGRS is more deterministic in resource provisioning, whereas the others are more bursty, as a function of the channel quality and arrivals.

2.7.4 Measurement error

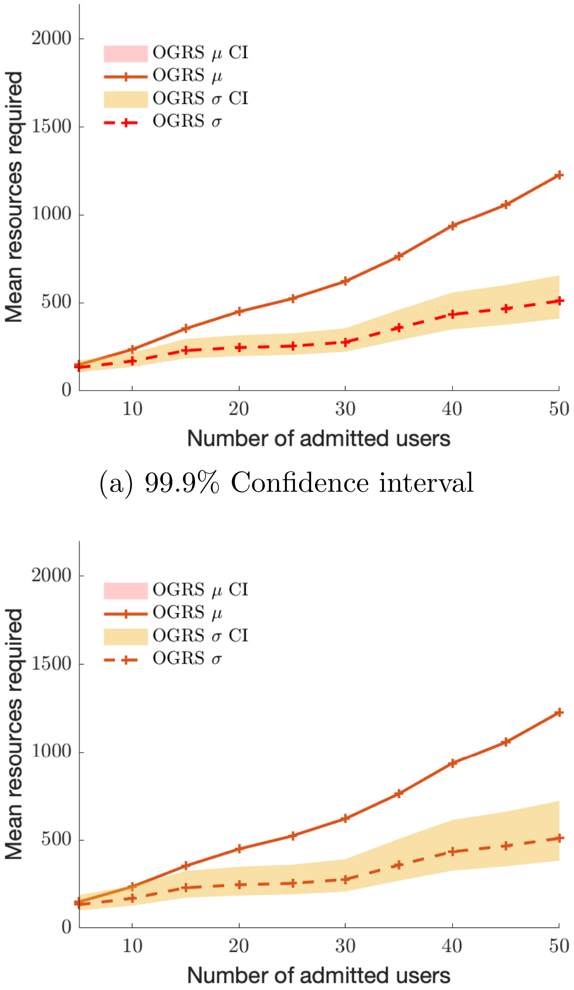
The admission control strategies discussed in [22, 23], rely on accurate knowledge of the users' aggregate resource usage statistics. In practice, however, one would need to measure the system's resource usage – which could potentially evolve based on the overall network traffic and user channel dynamics. Consequently, it is of interest to know the measurement errors in the mean and variance of aggregate resource usage that would impact performance.

We estimate the measurement error for the mean and standard deviation of the resources required by constructing a confidence interval based on 100 samples of the total RBs required for servicing the heterogeneous users. We determine the confidence interval for the mean, using the Gaussian confidence interval, $CI = \hat{\mu} \pm z_\alpha \hat{\sigma}$, where $z_\alpha \in \mathbb{R}$ is the value such that $\mathbb{P}(X \geq z_\alpha) \leq \alpha$. The confidence interval for the true standard deviation σ_{true} was obtained using the formula,

$$\sqrt{\frac{(k-1)\hat{\sigma}^2}{\chi^2_{\alpha/2}}} \leq \sigma_{\text{true}} \leq \sqrt{\frac{(k-1)\hat{\sigma}^2}{\chi^2_{1-\alpha/2}}} \tag{2.12}$$

where k is the number of samples used to obtain the sample variance and $\chi^2_\alpha = x : \mathbb{P}(X \geq x) = \alpha$, with X being χ^2 distributed with $k-1$ degrees of freedom.

Fig. 2.8 shows the confidence interval for the mean and standard deviation for a 99.9999% confidence interval on the resource requirement based on all three scheduling policies. The confidence interval around the mean is quite narrow and re-emphasizes the certainty equivalence of the measured mean. As for the standard deviation, Fig. 2.9 shows that one needs to account for the confidence interval of the variance to provide robust admission control to URLLC users that require high reliability.



(a) 99.9% Confidence interval

(b) 99.999% Confidence interval

Figure 2.9: Confidence interval for mean and standard deviation of the total number of resources allocated by OGRS-DTE to heterogeneous users admitted into the system.

2.7.5 Adaptation to transmission errors

Finally, we would like to consider a more practical wireless channel, unlike in Assumption 2.3.1, to discern how well our proposed adaptive algorithms can perform in the event of a channel outage. We need to adapt our OGRS algorithm to account for the uncertainty in wireless data transmissions. Suppose the user has a successful transmission with probability $1-\epsilon$, independent of all past and future transmissions, i.e., $\mathbb{P}(C_n = 0) = \epsilon$. Recall that the GRS($s$) compliant algorithms meet the delay deadline with probability 1 by assigning s RBs every time slot, under the assumption that $\mathbb{P}(C_n = 0) = 0$. Therefore, we propose a simple adaptation to address the transmission error at time slot n, by assigning $2s$ RBs in the subsequent time slot or $3s$ at $n+2$ when both $C_n = 0, C_{n+1} = 0$, and so forth. We demonstrate through extensive simulations that for $\epsilon \leq 0.1$, we are able to meet the delay deadline with very high probability.

Fig. 2.10 shows the probability of failure to meet the delay deadline when the channel has a non zero probability for outage, i.e., $\mathbb{P}(C_n = 0) = \epsilon$. We present results for both adaptive WGRS and adaptive OGRS scheduling algorithms proposed in Sec 2.5, where for OGRS we use the DTE threshold. It can be seen that the order of failure probability is at least one tenths less than ϵ for adaptive OGRS. Also note that the outage performance of DTE is superior to WGRS and demonstrates the robustness of DTE to channel outage. Since the failure probabilities of weak, medium and strong users are same up to three decimal points, Fig. 2.10 shows results for the medium user only.

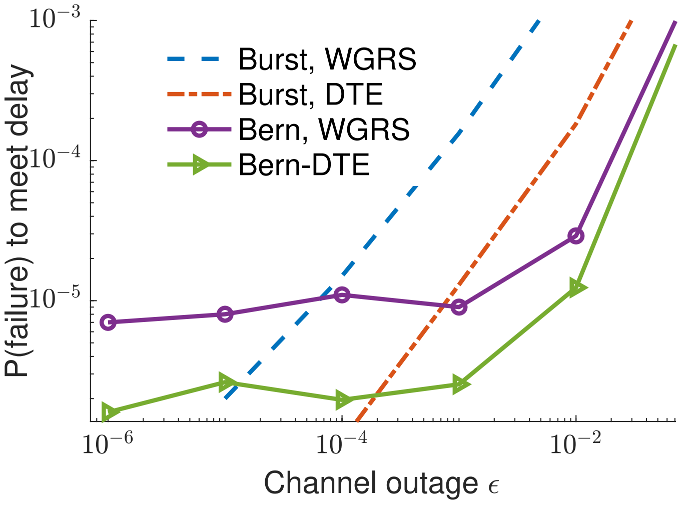


Figure 2.10: Probability of failure to meet the delay deadline for channel outage probability ϵ, using DTE policy.

In summary, the δ-based dynamic threshold has better spectral efficiency than either ST(α) (Sec. 2.5.1.1) or DTE (Sec. 2.5.1.3). However, ease of use with no design parameter gives DTE an added advantage over the other threshold selection mechanisms. Efficiency gains are higher if the traffic is bursty as the queues are usually much busier, which is desirable for video and real time applications. Efficiency gains are also higher if the user channel variations are large, as seen in the case of the weak 'cell edge' users. Furthermore, impact of number of samples for empirical estimation has been demonstrated, the higher the number of past channel realization samples, the better the empirical CDF estimate and hence overall spectral efficiency.

2.8 Chapter Summary

We have proposed a measurement based opportunistic wireless scheduler, which can meet heterogeneous users' hard delay deadlines while being spectrally efficient, i.e., minimizing the resources required, thus permitting the system to achieve additional throughput for other traffic sharing the network resources. The underlying design principle for OGRS policies is to ensure that the wireless scheduler meets or exceeds the service that a guaranteed rate scheduler with rate s would assign. Thus by design, OGRS policies can also be used to efficiently deliver a Guaranteed Bit Rate (GBR) service. Our proposed policy uses dynamic opportunistic thresholds to leverage the knowledge of the user's marginal channel quality rate distribution, which in practice would be measured and/or tracked based on a limited number, say 10, of the previous channel realizations.

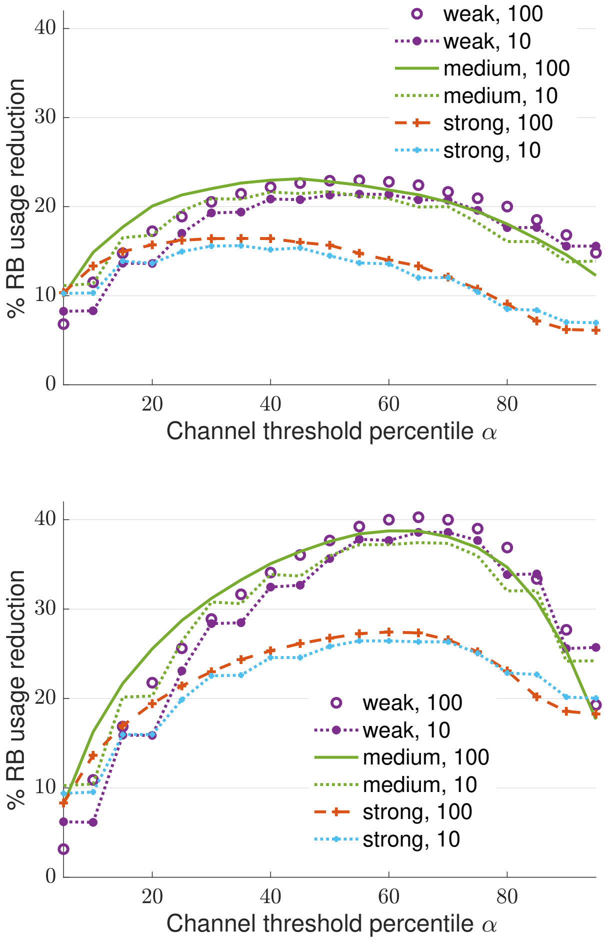


Figure 2.6: Reduction in RB usage for ST(α) as a function of the number of samples used for empirical CDF. The top figure corresponds to efficiency gains for Bernoulli arrivals and the bottom for burst arrivals.

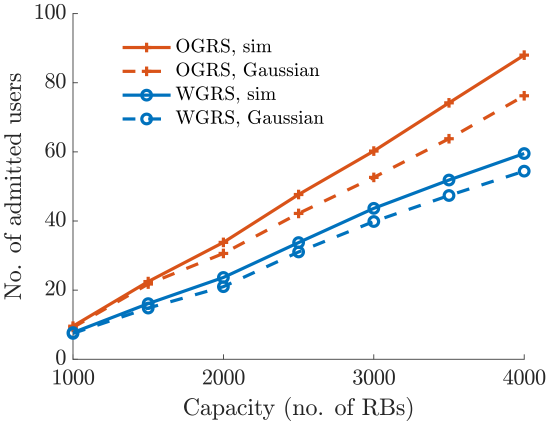


Figure 2.7: Admission control using Gaussian approximation on the total RBs required for all admitted URLLC users.

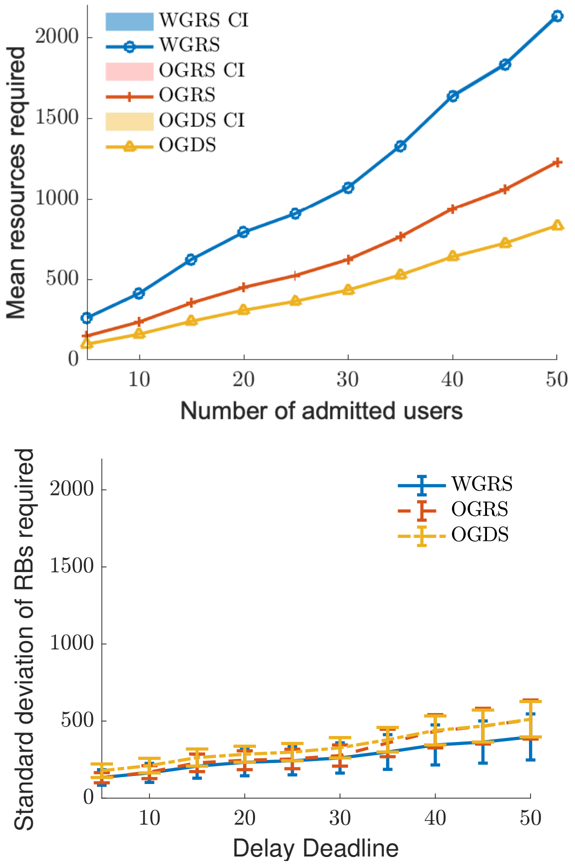


Figure 2.8: Confidence interval for the mean and standard deviation of the total number of RBs allocated by OGRS-DTE to heterogeneous users.

Chapter 3: Opportunistic Guaranteed Deadline Scheduling[‡]

While traffic shaping or policing provides a handle on the worst case traffic arrival that the scheduler would see, it is often not convenient to represent real time data traffic. For instance, video data will require a large token bucket size in order to accommodate peak arrival rates corresponding to the packets of an *I frame*. In this chapter, we will look at a delay constrained scheduler that is not limited to the traffic being leaky bucket constrained. We propose an opportunistic scheduler that meets strict delay deadlines on users' packets by estimating the probability of seeing a good channel in the future, before the packet deadline expires.

3.1 Related Work

Many works have focused on a setting where users' data queues are *fully backlogged.* When this is the case, one can consider devising schedulers that maximize the sum of the users' utility of their allocated long term rate [99]. For example, Proportionally Fair (PF) wireless scheduling emerges when users have log utility functions, see e.g., [54], and results in a scheduler that realizes a good tradeoff between *opportunistically* scheduling users which have good channels versus achieving a *fair* long term allocation amongst the users. Variations on these ideas have been proposed where the users' utility is a function of the short term throughput, see e.g., [31]. This

[‡] Publications based on this chapter: [22] Geetha Chandrasekaran, Gustavo De Veciana, Vishnu Ratnam, Hao Chen, and Charlie Zhang. Delay and Jitter Constrained Wireless Scheduling with Near-Optimal Spectral Efficiency. In *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–7, 2023

leads to a more responsive allocation avoiding short term neglect of any user. In practice, PF, and other utility maximizing schedulers, provide a simple and effective strategy for best effort or enhanced Mobile Broadband (eMBB) traffic with no strict delay requirements. Still, questions remain as to what happens when user queues are not fully backlogged or how to choose the fairness criterion, i.e., utility functions when there are delay constraints that require high reliability.

In settings where users' queues are not fully backlogged, researchers have focused on devising queue and channel dependent wireless schedulers which are *throughput optimal*, i.e., ensure user queues' stability whenever feasible. These schedulers also address performance objectives, such as Max-Weight [103], which is delay optimal in the idealized symmetric case, Exponential rule [92] which attempts to minimize the max user queue, Log rule [87] which attempts to minimize the mean delay and a variant of Exponential rule that supports real time and non real time QoS [91]. Such schedulers have been adapted to more practical settings, such as the Modified Largest Weighted Delay First (MLWDF) [11] which schedules users based on head-of-line packet delays, current channels, and other hyperparameters (like queue lengths [32]) reflecting user QoS/allocation objectives. In practice, such schedulers do meet delay constraints (with high probability) if sufficient resources have been provisioned, yet it is difficult to verify when this is true, and as such provide a graceful degradation across users when this is not the case.

Another class of wireless schedulers was born from modifying/adapting ideas from wireline scheduling (e.g., traffic shaping and network calculus [65, 24]) to meet QoS requirements under wireless channel variations. For instance, weighted round robin [64] or weighted fair queueing [67] employ user weights drawn from heuristics or tokens [79] based on service deficit [57] to either minimize the average delay or provide a graceful degradation of service. Much of the above mentioned work focuses on scheduling one class of users, e.g., best effort users sensitive to throughput, or traffic that is sensitive to packet delays. In practice, wireless systems need to be shared by heterogeneous user classes.

While many schedulers in the existing literature address delay constraints for real time traffic, spectral efficiency is often neglected, leading to lesser resource availability for non real time traffic. In this chapter, we place such interplay front and center, with a focus on not only developing a scheduler that meets delay constraints but one that does so in a spectrally efficient manner.

No practical wireless scheduling policy is complete without a complementary strategy for admission control and/or traffic shaping. Given the uncertainty and heterogeneity associated with traffic, channels, and user requirements in a wireless system, it is virtually impossible to devise good models that would allow one to predict if the users' QoS requirements will be met under a given scheduling policy. While there have been many works in literature that propose Measurement Based Admission Control (MBAC) [41, 42, 107], we note that meeting packet delay and loss targets in buffered systems is challenging [18]. In contrast to traditional MBAC approaches, our approach directly measures the aggregate resource that our delay constrained schedulers are using thus indirectly capturing the impact of the users' traffic, channel variability and delay constraints. Building on [23, 22], we provide an in-depth analysis of proposed algorithms' spectral efficiency and a variety of practical considerations.

3.2 Chapter contributions

We propose several classes of wireless schedulers which under appropriate assumptions can meet heterogeneous delay deadlines and do so in a spectrally efficient manner such that the more relaxed the constraint the more efficient. The key underlying idea is to leverage the flexibility of wireless systems, in terms of allocating a time varying number of Resource Blocks (RB) to overcome /exploit variations in wireless users' capacity per RB. This permits one to devise a scheduler, the Guaranteed Rate Service (GRS), which will ensure a user will see a fixed service rate even with channels that have stochastic variations. If a user's traffic is leaky bucket constrained, one can determine a minimum fixed service rate which will ensure a desired maximum delay.

Any scheduler which allocates at least as much cumulative service as the GRS scheduler over busy periods is GRS *compliant*, and will thus also meet the user's delay deadlines. This observation suggests the possibility of opportunistically serving a user's data ahead of time when channel rates are good, relative to the GRS scheduler, and/or delay such service when channel rates are poor, as long as the scheduling is GRS compliant. We devise a class of Opportunistic GRS (OGRS) schedulers that take advantage of this relaxation along with knowledge of the statistics of the users' channel variations, to achieve better spectral efficiency while meeting users' strict delay constraints. While OGRS is opportunistic while meeting the delay constraints, the underlying requirement to provide a minimum rate for each time slot is limiting. We propose an alternative approach, denoted Opportunistic Guaranteed Delay Scheduling (OGDS) that schedules data opportunistically based on the statistics of its channel's temporal variations and the remaining time window until its deadline expires.

First, we establish a stochastic ordering between the resource requirements of OGRS and Wireless GRS scheduling algorithms. Next, by considering *offline policies* with access to future channel capacity realizations, we derive a bound on the spectral efficiency that any delay constrained schedulers could achieve. We show via extensive simulations that OGRS can be within 10% to 40% of the bound as the delay constraint is relaxed. Meanwhile, OGDS is within 10% of the bound for a range of delays that we have simulated so far, which is up to 10 ms. These gains translate to doubling the eMBB user's throughput even for the users with the weakest wireless channels when URLLC and eMBB traffic share resources. We also observe an increase of up to 57% in the number of users that can be supported. This chapter further explores the impact of various additional issues critical to wireless scheduling including transmission errors, Hybrid Automatic Repeat Request (HARQ), user mobility, and the time scales on which to estimate the empirical distribution of channel variations, to show how our proposed approach would fare in practice. Finally, we also compare the spectral efficiency of OGRS and OGDS with delay constrained schedulers leveraging neural network based forecasts of future channel rates. We demonstrate regimes where the spectral efficiency of schedulers using empirical statistics (OGRS, OGDS) is higher than those that employ neural network predictions and vice versa.

This chapter is organized as follows. Section 3.3 describes the system model. Section 3.4 describes our proposed algorithms for delay constrained scheduling. Section 3.5 presents the main theoretical results of this chapter. Section 3.6 provides simulation results that compares the spectral efficiency, jitter, error and admission control of the proposed OGDS scheduler with other baseline algorithms. Section 3.7 provides extensive simulations for some practical wireless network settings, and also evaluates the spectral efficiency of a natural class of delay constrained schedulers that use neural network based channel rate predictions. Finally, Section 3.8 includes some concluding remarks.

3.3 System Model

We consider discrete time downlink scheduling for a base station serving a variety of users with either real time or best effort traffic. We denote by set $\mathcal{U}$ the delay constrained users with stochastic arrivals and possibly heterogeneous QoS requirements. The base station also serves a set $\mathcal{E}$ of infinitely backlogged best effort traffic users. We denote by $(A_n^u)_{n\in\mathbb{N}}$ the arrival process for user $u \in \mathcal{U}$, where A_n^u is a random variable denoting the number of bits that arrive and are available for service in time slot n with a transmission deadline of $n + d^u$, where d^u is the delay constraint for user u. In general, it is not possible to ensure delay guarantees to a user without prior knowledge of its traffic statistics or of constraints on its traffic. A common approach for the latter is to establish and enforce (through traffic policing/shaping) apriori constraints on the user's traffic that can be used to design resource allocation mechanisms guaranteed to meet a user's QoS requirements. In this chapter, we devise a scheduler that meets packet delay constraints without directly relying on traffic shaping constraints, but assuming admission control is in place. Note that while OGRS (introduced in Chapter 2) exploits information on the traffic shaping parameters to design an opportunistic scheduling rule, such strict traffic shaping is not required for the OGDS policy, which only requires the peak bit arrival rate of the user be bounded.

The base station transmit resources are modeled as a sequence of frames/slots comprising multiple Resource Blocks (RBs) which the scheduler can allocate arbitrarily to users on a per time slot basis by. For each RB, i.e., slice of time and frequency, we let the random variable $C_n^u \in \mathbb{R}^+$ denote the channel rate (bits per RB) that can be transmitted to user u if it is allocated a *single* RB on time slot n. While C_n can be small we assume that each RB has a non-zero effective transmission rate, like we did in the previous chapter, reproduced below for ease of reference.

Assumption 3.3.1. *(Connectivity Assumption) The BS can transmit data over an RB at a non-zero channel rate $C_n^u > 0$ with probability 1.*

Remark. This can viewed as a coverage/connectivity requirement for URLLC users which is met using sufficiently strong coding and/or multiple antennas which is either met with probability 1 or with a probability sufficiently high to far exceed the desired reliability associated with users' QoS guarantees.

A user may be allocated multiple RBs, but we assume a *flat fading* setting where the rate delivered to user u is the same across RBs in a given time slot. Additionally, a single RB may be allocated to only one user in a given time slot. Further, we assume $(C_n^u)_{n\in\mathbb{N}}$ are independent and identically distributed (i.i.d.) across time slots. Additionally, we also assume that a sufficiently large number of RBs are available every time slot to meet each user's QoS requirements. In the sequel, we propose admission control techniques that will limit the total number of users in the system and thus ensure resource availability. Our goal is to devise schedulers for URLLC users that use a minimal number of RBs. The remaining resources can be used to serve backlogged eMBB users with no stringent delay requirements.

A scheduling policy π, decides the number of RBs to be allocated to each user in each time slot. For ease of exposition, we will assume that there are enough RBs to provision service to all users in the system, and in the sequel, we will introduce admission control to limit the number of users as needed. The decision of policy π at time n is assumed to be causal concerning knowledge of the current and past channel rates $(C_\tau^u)_{\tau=0}^n$, arrivals and queue lengths, allowing for *opportunistic scheduling*, i.e., taking advantage of capacity variations across time. In particular, we let $M_n^{u,\pi} \in \mathbb{R}^+$ denote the number of RBs allocated to user u on slot n by a policy π given the observed history. Such an allocation provides an overall service rate $S_n^{u,\pi}$ (total bits transmitted with potentially multiple RBs allocated) to the user u on time slot n given by,

$$S_n^{u,\pi} = M_n^{u,\pi} C_n^u,$$

and we define the cumulative service over an interval $(\tau, \tau + n]$ as follows,

$$S^{u,\pi}(\tau, \tau + n] = \sum_{k=\tau+1}^{\tau+n} S_k^{u,\pi} \,. \tag{3.1}$$

A user's data queue (in bits) is modeled as a First Come First Serve (FCFS) discrete time queue with arrivals A_n^u and service rate $S_n^{u,\pi}$ as shown in Fig. 3.1. We let $Q_{n+1}^{u,\pi}$ denote the number of bits in the user's queue at the start of slot $n+1$, then

$$Q_{n+1}^{u,\pi} = [Q_n^{u,\pi} - S_n^{u,\pi}]^+ + A_{n+1}^u \,. \tag{3.2}$$

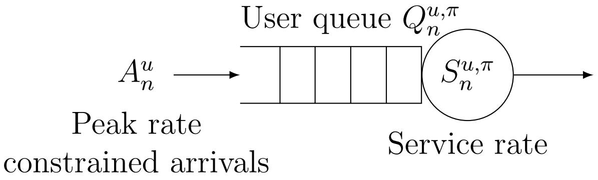


Figure 3.1: Peak rate constrained arrivals to a discrete time queue with a service rate controlled by scheduling policy π.

3.4 Opportunistic Guaranteed Deadline Scheduling

In this section, for the sake of brevity, we will drop the user index (marked by superscript u) as we consider per user scheduler. The user index will be reintroduced in the sequel when we introduce admission control. Suppose we start with an empty user queue at $t = 0$, then the arrival process $A(0, \tau]$ delayed by d would be such that $A(-d, \tau - d] = A(0, \tau - d]$. Fig. 3.2 depicts the cumulative arrivals $A(0, \tau]$ in blue and the corresponding delayed version in red $A(0, \tau - d]$. The delayed arrivals curve represents the worst case cumulative service that a server could provide without violating the delay constraint on each packet. Any cumulative service curve that lies within the arrivals and worst case departures curve will be delay compliant.

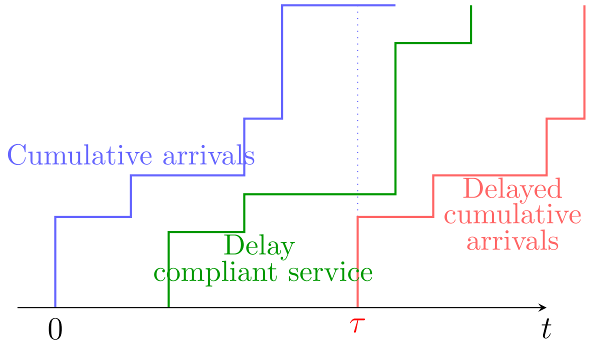


Figure 3.2: Delay compliant cumulative service along with worst case delayed service curve.

Definition 3.4.1. ***GDS(d)*** *We let the Guaranteed Deadline Scheduler with parameter d, GDS(d), be a scheduling policy that guarantees each bit in the user data queue be serviced within a delay of d since its arrival. Clearly,*

$$A(0, \tau] \geq S^{GDS}(0, \tau] \geq A(-d, \tau - d].$$

Definition 3.4.2. ***Opportunistic GDS(d).*** *A threshold based OGDS(d) scheduling policy π is as follows: whenever the channel rate C_n exceeds a threshold γ_n^π, a sufficient number of RBs are allocated by the scheduler to completely clear the queue backlog, i.e, $M_n^\pi = Q_n^\pi / C_n$. Otherwise, a minimal number of RBs are allocated so as to ensure that the cumulative service of π at slot n exceeds or matches that of the d delayed cumulative arrival curve.*

At each time slot, the number of slots τ_n over which there is flexibility to pick when to serve the data in the user queue depends on the residual time until the earliest deadline. Note that any data whose deadline is due to expire at a given time slot will be allocated resources in the same time slot. In case the current channel rate is expected to be better than those in the next τ_n time slots (i.e., current rate exceeds the threshold γ_n^π), the entire queue backlog is cleared.

Algorithm 2: Guaranteed Deadline Scheduling with opportunism over temporal variations.

```
initialize $S_0^\pi = 0$;
while $n > 0$ do
    $\tau_n = \min\,[k : k \geq n, A(0, k-d] \geq S^\pi(0, k]]$;
    if $C_n > \gamma_n^\pi$ then
        $S_n^\pi = Q_n^\pi$ ;
    else
        $S_n^\pi = [A(0, n-d] - S^\pi(0, n-1]]^+$ ;
    end
    $M_n^\pi = S_n^\pi / C_n$;
    $Q_{n+1}^\pi = Q_n^\pi - S_n^\pi + A_{n+1}$ ;
end
```

Algorithm 2 details the steps involved in OGDS(d) scheduling. When the channel rate is above a certain threshold $C_n > \gamma_n^\pi$, the OGDS policy π serves all data in the user queue,

$$S_n^\pi = Q_n^\pi.$$

Otherwise, the scheduler π allocates only the minimum number of RBs required to meet the worst case delayed service curve, i.e.,

$$S_n^\pi = [A(0, n-d] - S^\pi(0, n-1]]^+ .$$

Specifically, if the cumulative service provided by π until time $n-1$ is greater than that of the worst case delayed cumulative service at time n, then policy π can completely refrain from allocating any resources at time n if the channel rate is below the threshold.

OGDS Threshold selection: Define τ_n as the slack available to the scheduler before it is forced to schedule data to maintain delay guarantees, i.e.,

$$\tau_n = \min\left[k : k \geq n, A(0, k-d) \geq S^{\pi}(0,k)\right]. \tag{3.3}$$

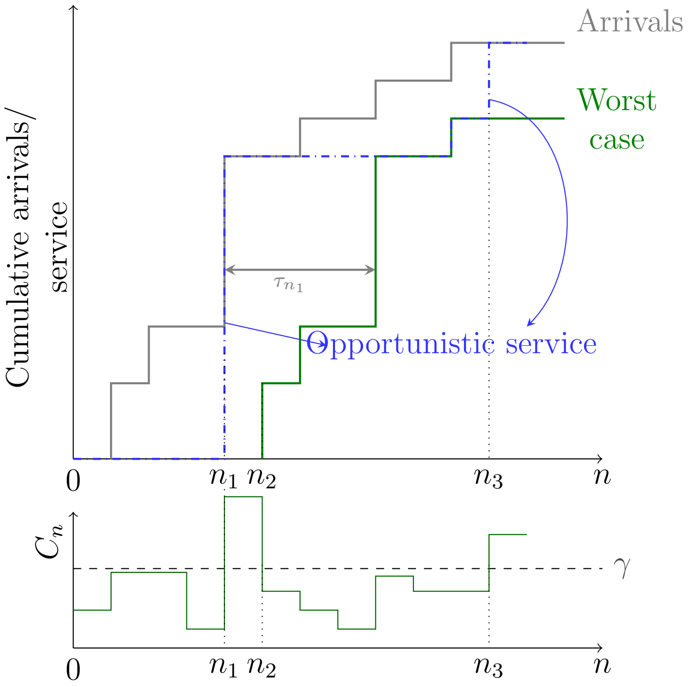


Figure 3.3: Illustration of the slack available to schedule cumulative arrivals based on the worst case service curve. The bottom figure shows the time varying nature of the wireless channel rates with a fixed threshold γ to determine the channel quality.

The threshold selection is illustrated in Fig. 3.3. At the time n_1, the OGDS scheduler has a slack of τ_{n_1} time slots before it is forced to start servicing the user queue. Therefore, for any particular channel rate realization c_n, the channel condition is considered good for opportunistic scheduling if,

$$\mathbb{E}\left[\max_{i=1,\ldots,\tau_n+1} U_i\right] < F_C(c_n)\,. \tag{3.4}$$

The left hand side is a maximum of $\tau_n + 1$ i.i.d uniform random variables which can be shown to be (see proof of Lemma 2.5.1),

$$\mathbb{E}\left[\max_{i=1,\ldots,\tau_n+1} U_i\right] = 1 - \frac{1}{\tau_n + 2} < F_C(c_n).$$

With this rough approximation an associated threshold on c_n depends on τ_n which can be set to,

$$\gamma_n^\pi = F_C^{-1}\left(1 - \frac{1}{\tau_n + 2}\right). \tag{3.5}$$

This captures the key insight that with a larger number of slots τ_n where π is not going to be forced to schedule user data, an OGRS(s) scheduler might choose to wait unless indeed it currently has a channel rate in the $1 - 1/(\tau_n + 2)$ percentile. Furthermore, the current threshold selection does not have any design parameter which makes it highly convenient for usage in practice. In the discussion above, we have assumed that the user's channel rate CDF is available. Typically, the serving BS tracks the user's Channel State Information (CSI) for adaptive modulation and coding, therefore, it is reasonable to assume that we can empirically estimate the channel rate CDF using CSI [80]. Also note that when the channel rates are discrete, we could use linear interpolation to invert the empirical CDF and compute the percentiles for the channel rate threshold.

3.4.0.1 Modified OGDS

While most scheduling policies for delay constrained traffic focus on improving key performance metrics such as energy efficiency [96], reliability [28] and delay, jitter is often neglected. It is a particularly important metric when transmitting periodic updates to networked real-time control and/or interactive AR/VR gaming applications. Disparate transmission delays across users can be undesirable/intolerable, especially in scenarios that need synchronization of updates across all users. There are multiple ways to measure the variability of transmission delay. We define jitter in terms of the standard deviation of delay for data transmissions that are periodic in nature.

We propose an elementary modification to the OGDS algorithm that provides a way to trade off between spectral efficiency, delay, and jitter, by carefully selecting a transmission window over which resources are allocated to the user. One could either wait for a predetermined number of transmit instants, say ζ, or artificially advance the targeted delay deadline to $d - \zeta$ to reduce packet jitter. A shorter window for transmission reduces the number of opportunities available for a user to be efficient, nevertheless, it reduces the variability in delay. Specifically, in Algorithm 2, step 10, the user queue update equation could be modified as follows,

$$Q^{\pi}_{n+1} = Q^{\pi}_{n} - S^{\pi}_{n} + A_{n+1+\zeta}\,, \tag{3.6}$$

where S^{π}_{n} denotes the service provided at time n, and $A_{n+1+\zeta}$ stands for the arrivals at time $n + 1 + \zeta$. In the sequel, we will refer to the parameter ζ as the jitter control parameter and demonstrate how the modified OGDS policy performs in terms of spectral efficiency and jitter.

3.5 Lower Bound on Spectral Efficiency

In this subsection, we state the theorem on a lower bound on the *minimum number of resource blocks* required by any wireless scheduler meeting the delay deadlines. The lower bound is based on considering an *offline* policy with complete knowledge of the future channel realizations and thus not achievable in an online setting, yet a good benchmark.

Consider a user with an arrival process $(A_n)_n$ and a time varying channel rate $(C_n)_n$ per resource block, whose traffic is subject to a delay constraint of at most d slots.

Theorem 3.5.1. *For any scheduling policy π meeting the delay constraint, let N_n^π denote the (possibly fractional) number of resource blocks used to serve the arrivals A_n, these RBs may be allocated at the earliest on slot n, but no later than the deadline $n+d$. Similarly, we let M_n^π denote the total number of RBs allocated on slot n. It then follows that,*

$$N_n^\pi \geq A_n \min_{0\leq j\leq d} \left[\frac{1}{C_{n+j}}\right] \quad a.s.. \tag{3.7}$$

Furthermore, if the arrivals and channel rate processes are stationary and independent of each other and the policy π is such that,

$$\lim_{n\to\infty} \frac{1}{n}\sum_{\tau=1}^{n} N_\tau^\pi = \bar{N}^\pi,$$

then the time average of $(M_n^\pi)_n$ also converges to a limit $\bar{M}^\pi$, which satisfies

$$\bar{M}^\pi = \bar{N}^\pi \geq \mathbb{E}[A_1]\,\mathbb{E}\left[\frac{1}{\max\limits_{0\leq j\leq d} C_{1+j}}\right]. \tag{3.8}$$

Proof. See Appendix B.1 for proof. ■

The lower bound on spectral efficiency is illustrated below using Fig. 3.4. An *oracle-aided* policy that has access to future channel rate realizations can schedule a packet that arrives by n_1 for transmission at time slot n_5 when the user's rate is the best before the packet deadline expires.

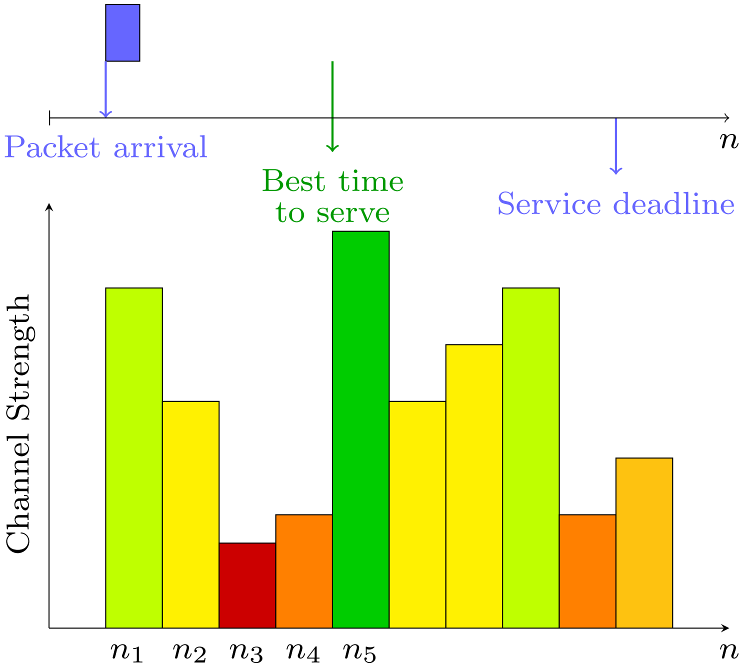


Figure 3.4: Illustration of the spectral efficiency lower bound when future channel rate realizations are known.

3.6 Simulation results

We consider a BS serving a set of URLLC with each user's channel rate C_n per RB determined by the corresponding received Signal to Noise Ratio (SNR). The received SNR was modelled using the 3GPP Urban-Micro path loss model [3], with Rayleigh distributed small scale fading. We assume bounded channel realizations, where the SNR lies between -6.934 dB $\leq$ SNR ≤ 20 dB. For simplicity, we shall use the 3GPP MCS Table 3.1 (see [2, Table 5.2.2.1-2]) to determine the rate obtained per RB C_n, where each RB is a time frequency slice of duration 1ms with bandwidth $B = 10$ KHz.

The traffic model is stochastic arrivals with a packet size of 1024 bits that arrive each time slot shaped by the leaky bucket with parameters (in packets per

MCS Index	Modulation	Coding Rate (bits/sym)	Spectral Efficiency	SNR (dB)
1	QPSK	78	0.1523	-6.934
2	QPSK	120	0.2344	-5.147
3	QPSK	193	0.3770	-3.18
4	QPSK	308	0.6016	-1.254
5	QPSK	409	0.8770	0.761
6	QPSK	602	1.1758	2.697
7	16-QAM	378	1.4766	4.697
8	16-QAM	490	1.9141	6.528
9	16-QAM	616	2.4063	8.576
10	64-QAM	466	2.7305	10.37
11	64-QAM	567	3.3223	12.3
12	64-QAM	666	3.9023	14.18
13	64-QAM	772	4.5234	15.89
14	64-QAM	872	5.1152	17.82
15	64-QAM	948	5.5547	19.83

Table 3.1: 3GPP Modulation and Coding

time slot) $\sigma = 50, \rho = 10, \mu = 5$, unless otherwise specified. Finally, to determine the channel quality thresholds, we need the CDF $F_C(\cdot)$ for the channel rate per RB on a given slot, for each user. We used the last 100 channel SNR realizations to determine the empirical CDF of the user's SNR at any given instant. Note that all plots in this section were generated over 10^6 slots, resulting in a ± 0.1 error for the estimated mean number of allocated RBs per slot $\bar{M}^\pi$ with 99% confidence interval. Additionally, we will use Wireless GRS (WGRS), OGRS policies introduced in the previous chapter as a baseline and also other benchmark policies such as the multicarrier version of the MLWDF [11] policy and schedulers that use neural network based forecasts of future channel rates to evaluate the promise of our proposed algorithms.

3.6.1 Spectral efficiency and jitter

We consider real time video streaming applications to evaluate the spectral efficiency and jitter performance of our modified OGDS policy. Let traffic arrivals be periodic, with 50 payloads of size 1 KB each that arrive once every 10 milliseconds (ms), over a duration of 10^6 ms for a total rate of 5 Mbps. A range of $4 - 10$ ms delay deadlines are considered for each payload. Fig. 3.6 exhibits how jitter reduces for each user with higher ζ at the cost of lower spectral efficiency as shown in Fig. 3.5. The plots for modified OGDS are labeled as "ζ-OGDS", where ζ is the jitter control parameter. Also, note that as the delay deadline increases we see a fall in the total number of RBs required for all policies.

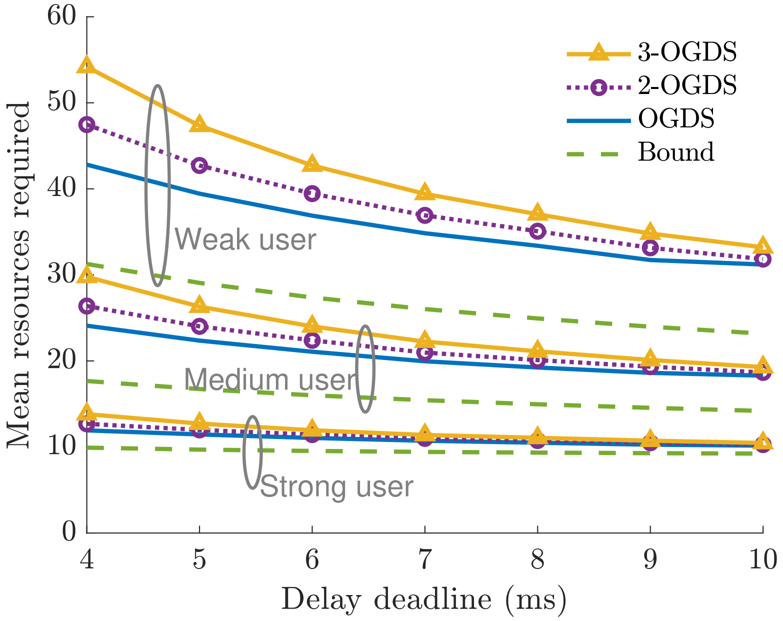


Figure 3.5: Spectral efficiency for various jitter control parameter ζ.

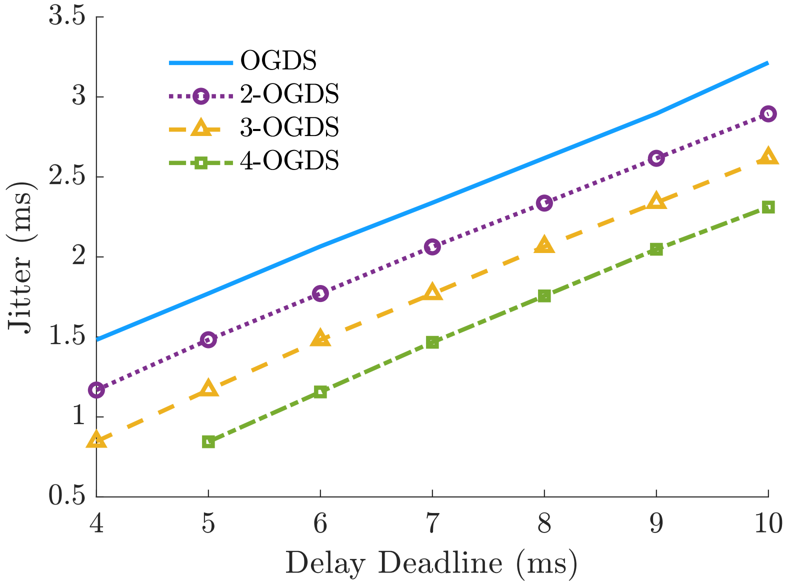


Figure 3.6: Jitter performance.

3.6.2 Improvement on eMBB throughput

In this section, we demonstrate the throughput improvement for eMBB users where the BS supports multiple heterogeneous users, 3 URLLC and 5 eMBB users. We consider ON-OFF bursty arrivals for URLLC users, where packets arrive at a peak rate ρ during the ON period. The ON, OFF cycles are of duration $\frac{\sigma}{\rho-\mu}, \frac{\sigma}{\rho}$, respectively. Distance from the BS and leaky bucket parameters for the 3 URLLC users are tabulated below:

Table 3.2: Leaky bucket parameters for multiple users.

User	distance (m)	Delay(ms)	ρ	μ	σ
1	300	5	10	5	50
2	500	3	20	10	50
3	700	7	10	5	50

The eMBB users are located at distances $250, 560, 650, 720, 800$ meters from the BS. Note that we assume that eMBB users are infinitely backlogged and do not have any stringent deadlines. Furthermore, we use proportionally fair scheduling to select one eMBB user from the set of eMBB users at each time slot that gets allocated all the leftover RBs.

A total of 6000 RBs are available to all the users connected to the BS and the URLLC users are allocated resources with priority. After allocating resources to all active URLLC users, the leftover RBs are used to serve eMBB users. The throughput performance of the oracle-aided policy is also included to provide a *bound* on the best feasible spectral efficiency for URLLC users, which translates to higher throughput for eMBB users.

Fig. 3.7 showcases the throughput gains for eMBB users for the various algorithms. When compared to the baseline and benchmark scheduling policies, OGRS policy is indeed closer to the throughput gain bound set by the oracle-aided scheduling policy with access to future channel rates.

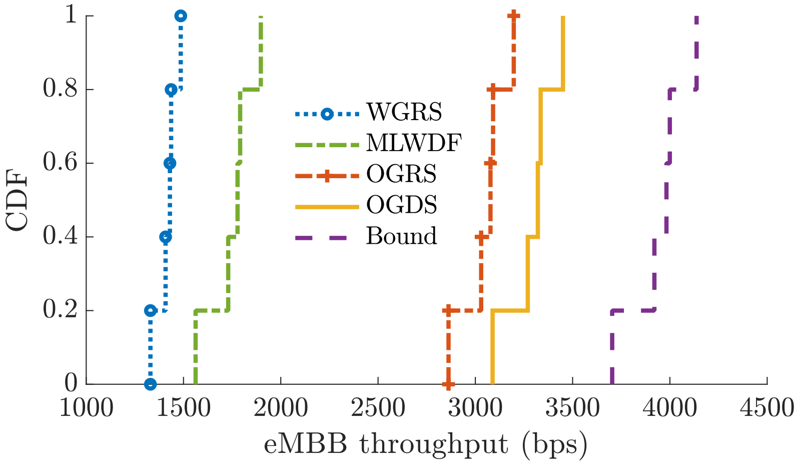


Figure 3.7: Long term throughput distribution for eMBB users.

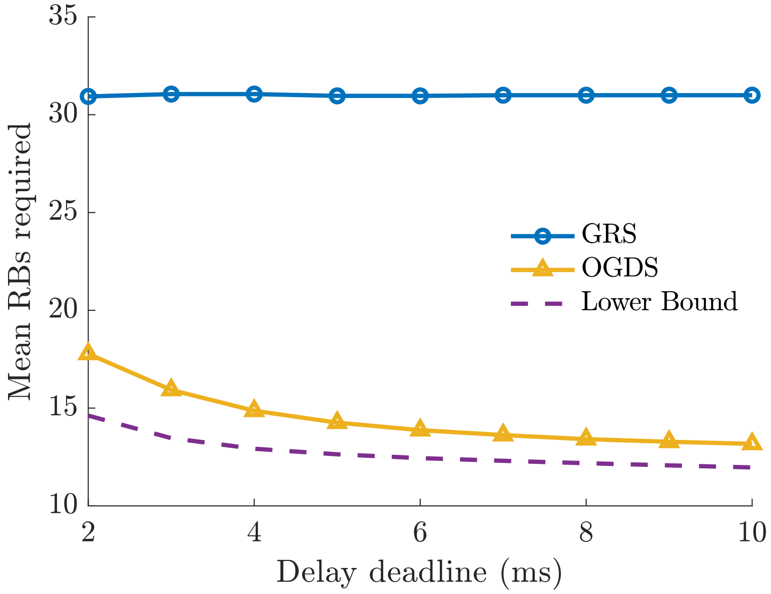


Figure 3.8: Non-stationary environment.

3.6.3 Robust performance under nonstationary environment

Our design of the dynamic threshold that triggers packet scheduling was based on the assumption that channel rate variations are i.i.d across time slots. However, in practice, wireless channel rates are often nonstationary, depending on user mobility and other propagation dynamics. To evaluate the performance of our scheduler on non-stationary wireless channels we used a trace [46] driven simulation for all three policies. In our simulation, we used 15 samples of past channel realizations to track the empirical CDF of the wireless channel. Fig. 3.8 demonstrates that OGDS is very close to the offline lower bound in a practical real world wireless environment.

3.6.4 Admission Control

We consider a set of 100 users with ON-OFF bursty traffic and leaky bucket constrained arrivals. The ON OFF duration is set to $\frac{\sigma}{\rho-\mu}, \frac{\sigma}{\rho}$, with parameters (in packets per time slot) $\sigma = 50, \rho = 10, \mu = 5$. The delay deadline and user location (distance from the BS) are drawn uniformly random from the sample spaces {200, 250, ..., 800} and {2, 3, ..., 10}, respectively. The arrivals and channel variations are generated over 10^6 time slots to simulate the number of users that can be admitted for various system capacities $\overline{m}$ (the total number of RBs available in the system). If $Y \sim \mathcal{N}(\hat{\mu}, \hat{\nu})$ denotes the random resource requirement of a new user, then the probability that the total resource requirement $X_u + Y$ will exceed $\overline{m}$ is approximated using the following inequality, see [26],

$$\mathbb{P}\left(X_u + Y \geq \overline{m}\right) \leq \exp\left(-\frac{\overline{m} - \mu}{2\nu}\right), \tag{3.9}$$

where $X_u \sim \mathcal{N}(\mu, \nu)$, $\mu = \mu_u + \hat{\mu}$ and variance $\nu = \nu_u + \hat{\nu}$. Note that the inequality in (3.9) provides a computationally reasonable expression that can be used to decide if the new user can be admitted without exceeding the reliability requirement δ. Therefore, for the same set of users, we also use the Gaussian approximation for the aggregate resource requirement X_u to determine the number of users that can be admitted, shown as dashed lines in Fig. 3.9.

Recall, it was seen in Fig. 2.7 from Chapter 2 that the Gaussian approximation provides a conservative estimate of the number of users that can be admitted into the system for both the OGRS and WGRS scheduling policies. However, for OGDS, there is a cross-over point where the CLT estimate is above the simulation based users admitted until a system capacity of 3500 RBs, after which the CLT estimate becomes conservative. This can be attributed to the more bursty nature of OGDS scheduling as compared to the other scheduling policies, making the CLT approximation for OGDS reasonable only at higher system capacities than OGRS/WGRS. Also interesting to note is that both OGRS and OGDS scheduling policies admit more users as compared

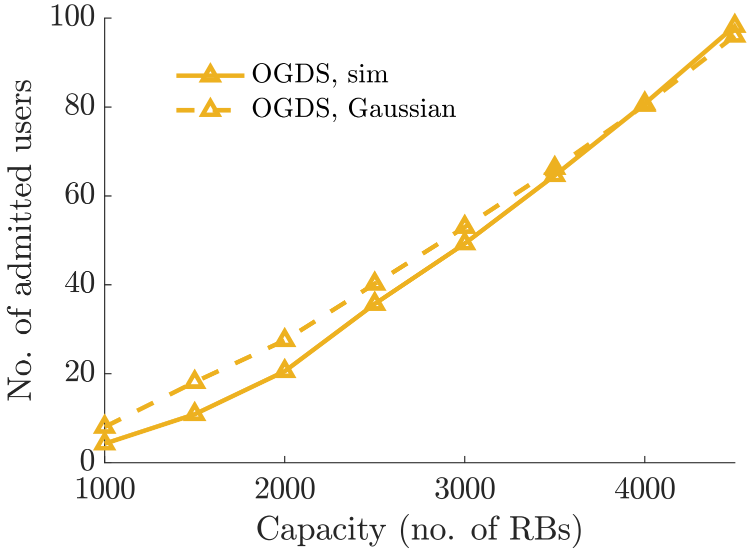


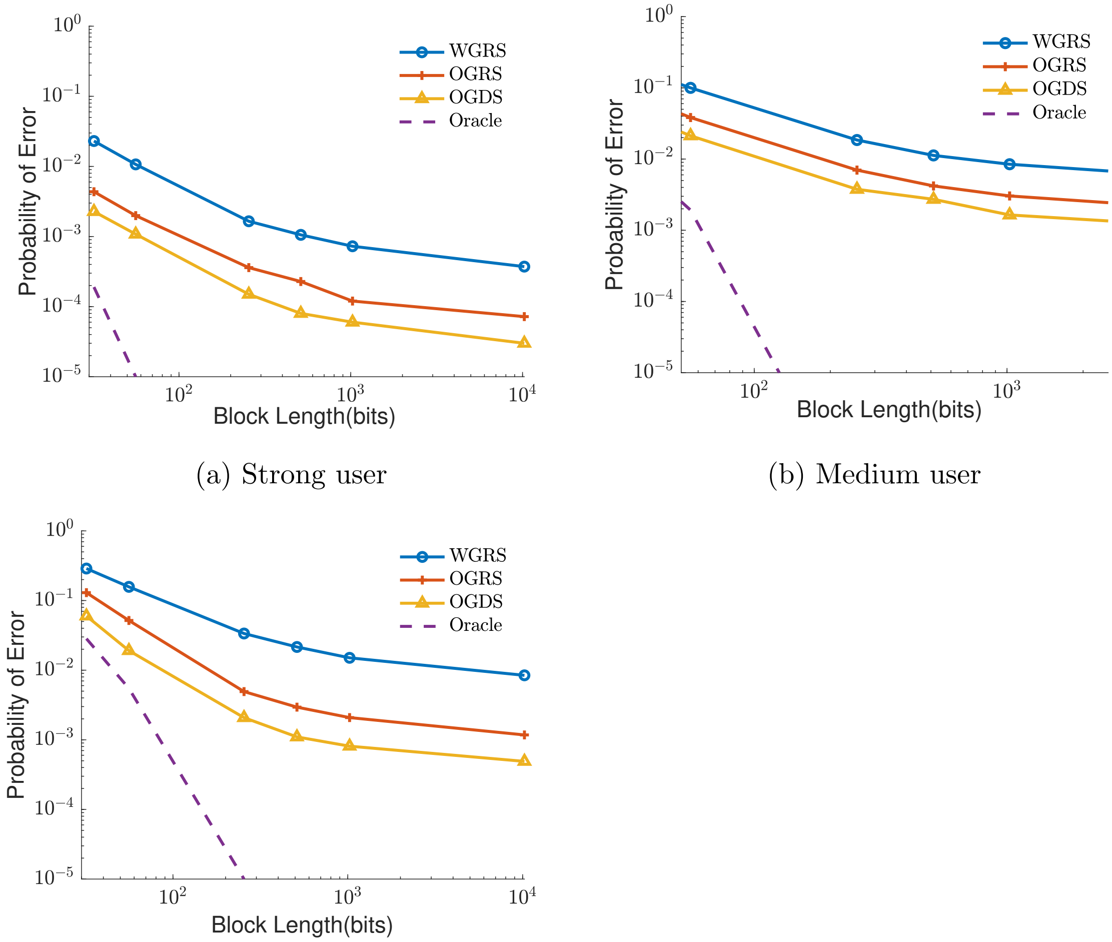
Figure 3.9: Admission control assuming CLT approximation on the total RBs required for all admitted URLLC users.

to the WGRS policy, given that WGRS is more deterministic in resource provisioning, whereas the others are more bursty.

3.6.5 Transmission Error

So far we have discussed delay constrained scheduling based on the *connectivity* assumption (see Assumption 3.3.1) that a user's packet transmissions are successful at all times. In practice, user transmissions are bound to see errors due to the nature of wireless channel uncertainty. The probability with which errors in wireless transmission occur depends on the size of the data packets, channel strength, and the amount of redundancy added to the original data for error detection and/or correction. In this subsection, we evaluate the probability of error for all algorithms over various transport block lengths (which is higher for larger packets).

Recall that the modulation and coding scheme is chosen using the 3GPP MCS table ([2, Table 5.2.2.1-2]) based on the instantaneous channel strength. It can be seen that the performance of all our proposed algorithms results in a ten-fold decrease in the probability of transmission errors as compared to the WGRS, with the OGDS algorithm being the best among all feasible (future agnostic) algorithms considered. The proposed algorithms schedule transmissions when the channel has a higher probability of being better than future channels before the deadline expires, which in turn leads to a lower probability of transmission errors.



(a) Strong user

(b) Medium user

(c) Weak user

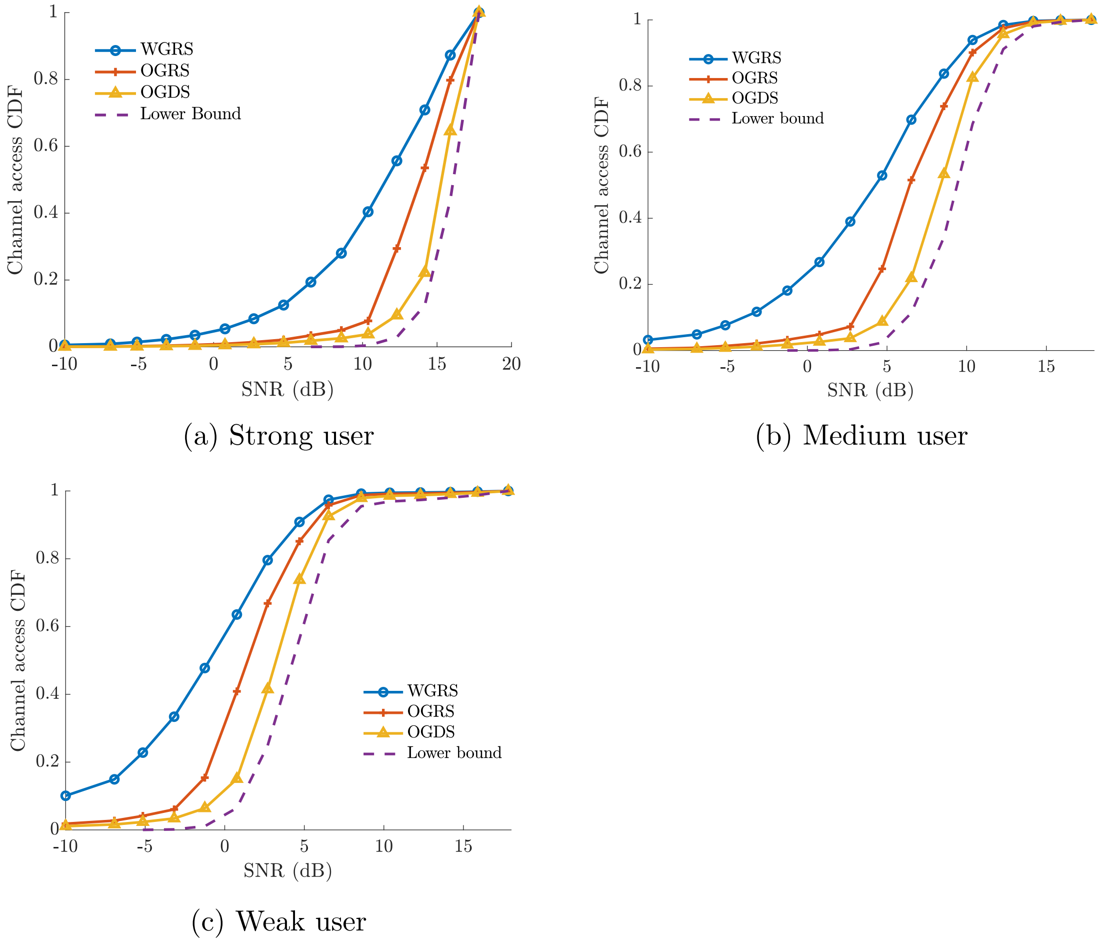
Figure 3.10: Probability of packet transmission error for various user types.

3.6.6 Better Reliability

In this subsection, we illustrate the improvement in resource allocation by having more transmissions scheduled for better channel quality. We plot a weighted distribution of the channel strength per channel use, where the weights are directly proportional to the number of bits scheduled for transmission at that channel strength. This enables us to compare and contrast all the proposed scheduling policies according to the efficiency with which each policy is able to identify the best time slot for data transmission in terms of channel quality. Clearly, Fig. 3.11 validates the superior performance of OGDS policy across all users, irrespective of the average channel strength of the received signal. Furthermore, it also shows that GDS policy schedules URLLC traffic for transmission at a relatively better channel strength which corresponds to greater reliability that is crucial for URLLC traffic. While OGDS is the best performing scheduler we have proposed, it achieves these gains through a potentially more bursty service and thus poor for jitter sensitive traffic. In certain settings OGRS might be preferred as it provides a guaranteed service rate as long as there is data in the queue.

3.6.7 Stochastic dominance of WGRS policy

Through extensive simulations, we were able to observe first-order stochastic dominance of the number of resource blocks allocated during a WGRS busy cycle over both OGRS-DTE policy and OGDS. We observed that on average OGRS-DTE required lesser resources than WGRS during a WGRS busy cycle, i.e., $27\%, 47\%, 22\%$ for strong, medium, and weak users, respectively. Similarly, OGDS required fewer resources than WGRS during a WGRS busy cycle, i.e., $43\%, 62\%, 54\%$ for strong, medium, and weak users, respectively.



(a) Strong user

(b) Medium user

(c) Weak user

Figure 3.11: The weighted distribution of channel strength per channel use for data transmission under all three policies.

3.7 Practical Considerations

We list a few practical considerations which have not been addressed rigorously in the theory developed so far. To start with, we assume that all transmissions on the wireless channel are reliable, which may not be always possible [82], especially with small payloads/block lengths [95]. It is not very clear as to which layer of the network stack is better suited to support reliability, with the physical and MAC layer being potential candidates. Moreover, it is not clear if typical Medium Access

Control (MAC) layer Hybrid Automatic Repeat Requests (HARQ) can be adapted to provide spectrally efficient resource allocation with guaranteed delay deadlines or whether spatial diversity will be sufficient to achieve the target reliability without any changes to the MAC layer.

Next, we do not consider an upper limit on the number of RBs that can be allocated to a user in a time slot. Typically BSs support multiple type of users, and possibly users with best effort traffic that do not operate under tight latency constraints. Therefore, it seems reasonable to assume that there shall be enough RBs available at all times. Most part of our analysis takes a single user perspective, but we would like to emphasize that the scheduler could in practise collate users with similar QoS requirements into the same class and provision resources based on the requirements of each class. Each class could in principle also perform admission control on latency constrained users to limit the total number of users as per availability of spectrum.

It would also be more desirable to carefully consider the fractional RB setting that we have used to measure the efficiency. Short block lengths call for mini-slot scheduling, so it might be prudent to allow only integer values for M_n^π. One could use the mini slot size [5] to determine the number of RBs required, perhaps corresponding to discrete coding rates that are used in practice, refer Table 3.1. In this section, we consider some additional practical considerations affecting our proposed schedulers including HARQ, user mobility, and how confidence levels on the measured resource usage statistics impact the effectiveness of our proposed admission control strategy.

3.7.1 Transmission Error

So far we have discussed delay constrained scheduling based on the *connectivity* assumption that a user's packet transmissions are successful at all times. In practice, user transmissions are bound to see errors due to the nature of wireless channel uncertainty. The probability with which errors in wireless transmission occur depends on the size of the data packets, channel strength, and the amount of redundancy added to the original data for error detection and/or correction. In this subsection, we evaluate the probability of error for all algorithms over various transport block lengths (which is higher for larger packets).

Fig. 3.10 shows the packet transmission error probability of our proposed algorithms for various user channel strengths as a function of the transport block length. The probability of transmission error ϵ was modeled based on the Polyanski bound [76] [93]. For transmission of block length m, coding rate r and channel SNR γ, the error bound is given by,

$$\epsilon = Q\left(\sqrt{\frac{m}{(\log_2 e)^2\left(1-\frac{1}{1+\gamma^2}\right)}}\left(\log_2(1+\gamma)-r\right)\right). \tag{3.10}$$

Recall that the modulation and coding scheme is chosen using the 3GPP MCS table ([2, Table 5.2.2.1-2]) based on the instantaneous channel strength. It can be seen that the performance of all our proposed algorithms results in a ten-fold decrease in the probability of transmission errors as compared to the WGRS, with the OGDS algorithm being the best among all feasible (future agnostic) algorithms considered. The proposed algorithms schedule transmissions when the channel has a higher probability of being better than future channels before the deadline expires, which in turn leads to a lower probability of transmission errors.

3.7.2 HARQ

Wireless networks typically use automatic retransmit requests whenever there are transmission errors that lead to packet losses. A one-shot retransmission would typically suffice if the modulation and coding scheme were carefully chosen so as to maximize the likelihood of successful retransmission. So a simple modification to the proposed algorithms is to reduce the target delay deadline by one slot and perform a one-shot retransmission of packets lost due to transmission error. Note that the proposed modification assumes that the delay constraints exceed several time slots (more than 2 slots).

Incorporating the modified target deadline to allow for HARQ one-shot retransmissions leads to successful packet deliveries for all types of users with an associated loss in spectral efficiency. Setting an earlier deadline for strong, medium, and weak users leads to a loss in spectral efficiency of at most $9\%, 14\%$, and 15%, respectively.

3.7.3 User mobility

The adaptive rate thresholds (refer to Equation (3.5)) depend on the accuracy with which the distribution of the channel rate variations can be empirically estimated. A sufficient number of past channel values are required to estimate the channel variation statistics, but short enough to track non stationary changes and exclude obsolete channel data. This calls for selecting the number of past channel samples that are used to determine the channel rate distribution, which could potentially depend upon the user's mobility speed.

To evaluate this we use the Random Way Point model (RWP) to model user mobility with a constant speed in the range of $5-40$m/s. Using the user's location given by RWP model, the wireless channel variations are modeled based on the 3GPP channel model with correlated log-normal Shadow fading based on the distance traveled. Specifically, the correlation factor for shadowing is given by, $R_x(\cdot) = \exp\left(-\frac{\text{distance}}{d_{corr}}\right)$, where the parameter d_{corr} depends on the presence or absence of Line of Sight (LoS) signal, i.e.,

1. LoS shadowing : Log $\mathcal{N}(0, 4)$, $\quad d_{corr} = 10$m.
2. nLoS shadowing : Log $\mathcal{N}(0, 7.82)$, $d_{corr} = 13$m.

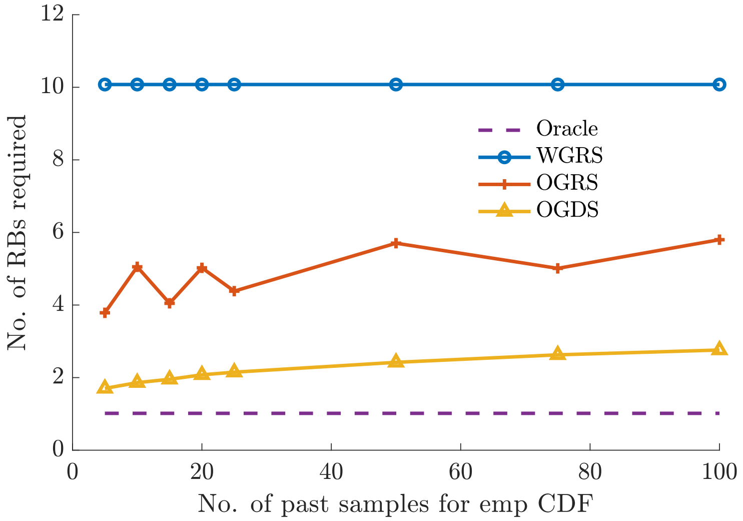


Figure 3.12: Resource requirement for a mobile user moving at a speed of 40 m/s.

Fig. 3.12 shows the resource requirement for various algorithms as a function of the number of past samples used to estimate the CDF. We find that for various mobility speeds in the range of 5-40 m/s, the number of samples (channel history) required to efficiently track the user's nonstationary channel distribution is 5. As can be seen, using more samples leads to obsolete channel information being included in the empirical estimate, resulting in the *adaptive rate* threshold being irrelevant to the current wireless channel – and hence a loss in spectral efficiency.

3.7.4 Delay constrained schedulers based on based channel prediction

Recall that in Section 3.5 we proposed an offline/genie based policy that given *perfect* knowledge of future channel realizations achieved the best possible spectral efficiency subject to the packet delay constraints. It is thus natural to attempt to implement such a policy based on predicted future channel rates given the the observed channel history. To explore the potential of this approach we considered various possible predictors introduced in [19] and followed their methodology for training our predictions for the wireless channel data set in [46] using Adam/SGD* optimization based Feed-forward Neural Networks (FNN) and state-of-the-art machine learning architectures including Long Short-Term Memory (LSTM) [47], Recurrent Neural Network (RNN) [39], Convolutional Neural Network (CNN) [63]. We evaluated the spectral efficiency of the prediction based delay constrained schedulers, for traffic having deadlines from 2 to 10ms. We considered a scenario with bursty traffic arrivals with packets of size 1024 bits, that arrive according to an ON and OFF process with duration 10 and 5 time slots respectively, with an ON rate of 10 pkts/slot.

Fig. 3.13 and Fig. 3.14 show the spectral efficiency (as measured by the mean resource requirements to support the delay constrained traffic) that the prediction based schedulers and our proposed schedulers achieve for the wireless trace data in [46] and [35], respectively. The normalized Root Mean Square Error (RMSE) for neural network predictions lies in the range of $0.081 - 0.137$ for "Berlin data" in [46] and $0.057 - 0.127$ for "Austria data" in [35]. Later on, we will see how this small difference in prediction accuracy leads to large variations in performance.

The overall improvement in spectral efficiency saturates beyond a certain delay

*Stochastic Gradient Descent

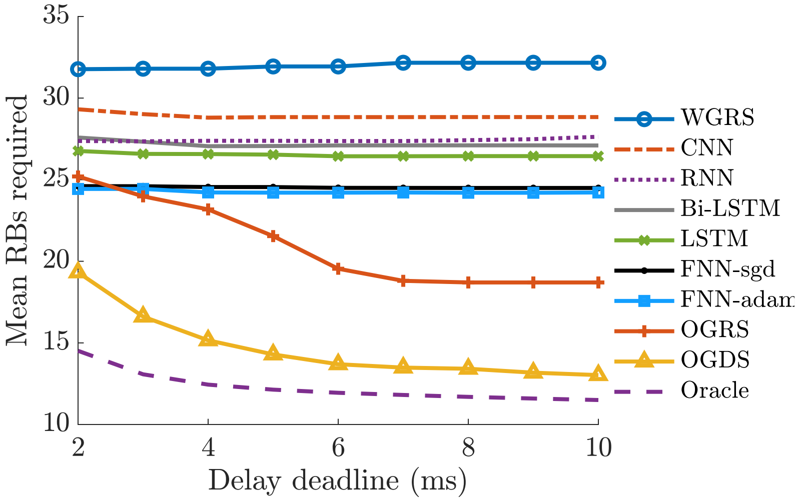


Figure 3.13: Resource requirement for non-stationary “Berlin data”, wireless trace data in [46], across proposed scheduling algorithms.

deadline due to the diminishing value of the flexibility that each additional time slot provides. To see this clearly, we provide the prediction accuracy results for neural network prediction for both “Berlin data” and “Austria data”. Figures 3.15, 3.16 show the Root Mean Square Error (RMSE) and 95% confidence interval for channel rate predictions for the wireless trace data given in [46] and [35], respectively.

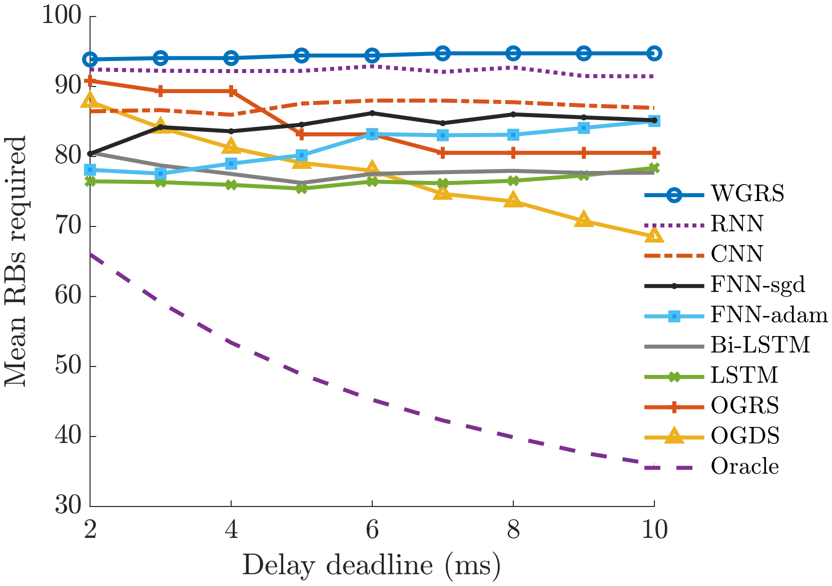


Figure 3.14: Resource requirement for non-stationary "Austria data", wireless trace data [35], across proposed scheduling algorithms.

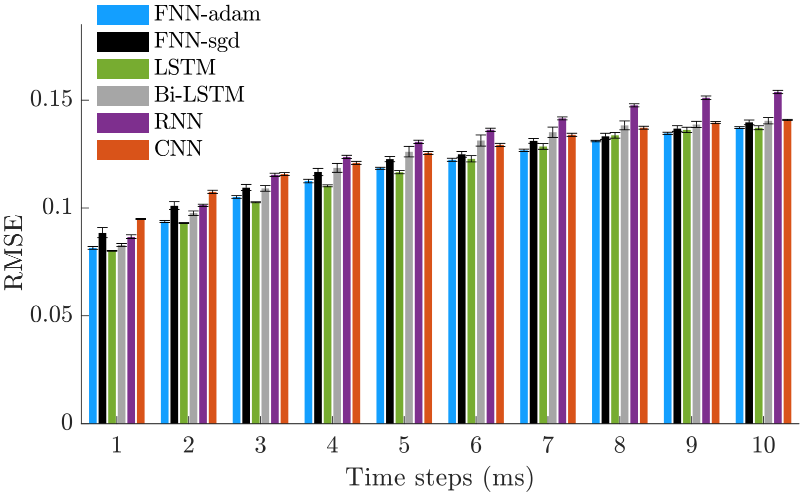


Figure 3.15: RMSE across various neural network architectures for channel rate prediction [46].

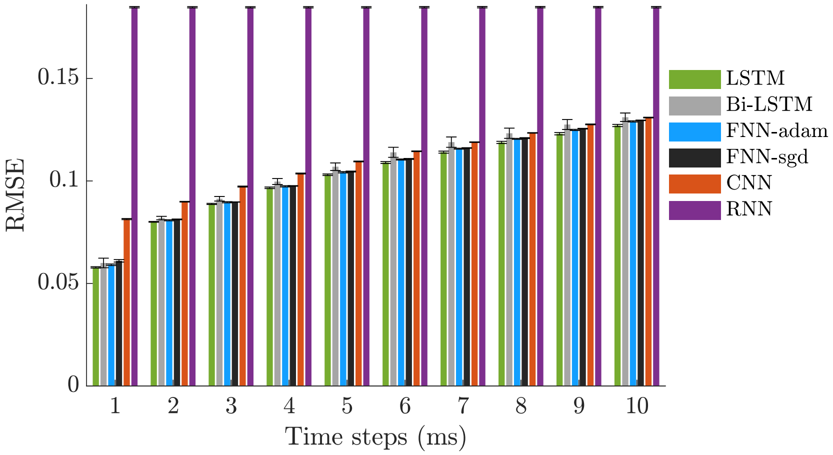


Figure 3.16: RMSE across various neural network architectures for channel rate prediction [35].

As one would expect, the architecture delivering the lowest prediction error, FNN-Adam for “Berlin data” in [46] and LSTM for “Austria data” in [35], leads to the most spectrally efficient scheduling for prediction based delay constrained schedulers. Our proposed measurement-based scheduler, OGDS outperforms all the prediction based schedulers for “Berlin data”, with OGRS only slightly worse than FNN-Adam based scheduler when delay deadlines are less than 2 ms. However, for “Austria data”, a higher prediction accuracy for shorter delays ($\leq$ 6 ms) leads to better spectral efficiency for prediction based schedulers than both OGDS and OGRS. It appears that the proposed ML-based schedulers are more sensitive to prediction errors when deadlines are relaxed ($>$ 6ms), resulting in lesser spectral efficiency than OGDS.

It should be noted that all neural networks were trained using thousands of samples of data before being deployed on test data. This would mean the neural network having adequate training on actual rate variations in the wireless environment and a prediction phase where the user remains stationary, which is unrealistic! In summary, we have proposed measurement-based schedulers that appear robust for real-world traces and have much lower computational complexity than schedulers that use NNs. *Online* time series predictors that can learn to predict with fewer learning samples and quickly adapt to non-stationary variations could be a promising avenue for improving spectral efficiency.

3.8 Chapter Summary

In the previous chapter, we have shown that OGRS policy can be within 10% to 40% of the spectral efficiency *oracle-aided* policies, whereas OGDS can be upto within 10% of such *oracle-aided* policies, as the delay constraint is relaxed. These gains translate to doubling the eMBB user's throughput even for the weakest user when URLLC and eMBB traffic share resources or an increase of upto 57% in the number of users admitted as long the arrival rates and channel strengths are statistically distributed very similar to the *typical*[†] admitted user.

The underlying design principle for OGRS policies in the previous chapter was to ensure that the wireless scheduler meets or exceeds the service that a fixed rate scheduler designed based on leaky bucket constrained delay analysis would assign. In contrast, our proposed OGDS policies allow for more aggressive opportunistic scheduling, which depending on the delay constraints can achieve within 10% of the spectral efficiency of optimal offline scheduling. Both policies use dynamic opportunistic thresholds to leverage the knowledge of the user's marginal channel quality rate distribution which in practice would be measured and/or tracked, based on a

[†] The resource requirement to meet the QoS constraints of a *typical* user will be Gaussian distributed, with mean and variance equal to the average resource usage statistics (mean, variance) of all admitted users.

limited number, say 10, of the previous channel realizations. However, the choice of a particular scheduler will depend on the availability of resources and the desired spectral efficiency. When the network resources are limited, OGRS supports more users making it better suited than OGDS. Whereas OGDS might be preferred when there are enough resources and spectral efficiency is the key to support more users.

Chapter 4: Opportunistic Minimum Rate Scheduling[§]

So far, we have focused on delay-constrained users with small packet sizes and limited delay requirements (of the order of a few milliseconds). Future wireless applications, such as online gaming, live video streaming, and augmented/virtual reality need high fidelity data transfer with reliable control information exchange to provide an immersive user experience. Developing wireless scheduling algorithms that can guarantee a minimum data rate for users over a few hundred milliseconds with possibly heterogeneous requirements and/or channel quality is crucial. In this chapter, we will look at such a special class of QoS, namely the Guaranteed Bit Rate (GBR) service, where users have minimum rate constraints over time windows that are of the order of a few hundred milliseconds.

According to a recent study, more than 86% of the world population owns a smartphone with more than 78% of data usage [88] coming from digital media. Wireless communication, once an infrastructure that provided cellular connectivity for mobile users, has evolved into an urbane network that provides various data intensive services such as GPS navigation, mobile gaming, and streaming digital content from anywhere in the world. Due to current trends in live streaming and the prevalence of social media, more real-time data communication is required, with stricter Quality of Service (QoS) standards for a better user experience. Such applications require predictable and stable data transfer rates in order to maintain smooth and uninter-

[§] Publications based on this chapter: [21] Geetha Chandrasekaran and Gustavo de Veciana. Opportunistic Scheduling for Users with Heterogeneous Minimum Rate QoS Requirements. In *ICC 2024 - IEEE International Conference on Communications*, 2024 rupted operations. Wireless spectrum is limited which bounds the physical resources available for user allocation. It may be possible to provide better QoS for more users by designing an adaptive algorithm that can efficiently allocate resources based on network load and user target QoS requirements.

Consistency in network resource provision is indispensable to maintain the overall quality of user experience for real time applications, especially those that require interactive control features. Reliable QoS provision for users with heterogeneous wireless channel variability makes it impossible for a causal scheduler to be able to accurately predict the fidelity with which it can support all users. As user loads, data traffic, and wireless channel strength vary in the system, it is necessary to provide a robust resource allocation to meet QoS of all users.

4.1 Related Work

Research interest in dynamic bandwidth allocation started with Asynchronous Transfer Mode (ATM) networks [15] to provide differentiated quality of service to users. The main objective was to optimize the network's performance by adapting to changing traffic conditions and ensuring fair allocation among competing users or connections while maintaining QoS requirements. Then came wireline algorithms, see [37, 27, 94, 34], that were specifically designed to ensure that different types of traffic receive the appropriate level of service to meet their QoS guarantees. However, meeting QoS requirements with high reliability is challenging in wireless networks due to channel uncertainty.

Numerous early studies in the wireless literature offer differentiated quality of service by optimizing the user's utility function, such as overall network throughput (MaxRate [108]), or maintaining bounded user queues whenever feasible (MaxWeight

[103]), or minimizing average delay (Log rule [87]) or weighted queue lengths (EXP [92]). Schedulers of this type achieve *throughput optimality*, meaning they have the capability to stabilize user queues if that is possible. However, *throughput optimal* schedulers may not provide the best performance and/or require parameter tuning when the traffic flow dynamics are nonstationary [110] or when users see heterogeneous channel variations. Proportional Fair (PF) scheduling [54] was an attempt to balance between maximizing the sum throughput versus fairness in resource allocation across users but without any minimum rate constraints. There have been numerous algorithms proposed to provide a *minimum rate* QoS while maintaining fairness among users through QoS dependent weighted user selection, namely, QoS aware PF [38], EXP rule for QoS constraints [91], and weighted maxQuantile [90] for fair rate constrained resource allocation. *Minimum rate* QoS constrained scheduling for differentiated services based on largest weighted time based debt [48] may unfairly prioritize weak users with very poor channel conditions that have a lot of backlog in their queues.

The majority of recent research on QoS constrained wireless scheduling has focused on latency, refer [9, 28, 56, 23], with an aim to support mission critical applications. Often such schedulers consider reliable transmission of short packets with very low delay and designed to achieve better reliability through redundancy, neglecting spectral efficiency. One could argue that it may be easy to extend delay constrained schedulers that measure/track the wireless channel distribution [22], or apply machine learning techniques to predict future rate realizations [52], however, that is not the case.

It is quite challenging to provide the required amount of reliability and fair distribution of excess resources under heterogeneous *minimum rate* QoS requirements and non identically distributed user channel rates. Algorithms such as DRUM [31, 78] provide a more customizable time window over which weighted α fairness could possibly be met. However, such algorithms call for fine tuning a variety of design parameters/user weights that need to be chosen in a non-trivial fashion. Furthermore, they may not be well suited for the heterogeneous setup, i.e., if one were to design max-min fair resource allocation with *heterogeneous* minimum rate guarantees (where user time windows may be different or nonoverlapping) it is not clear how to formulate the optimization problem parallel to the one proposed in [31].

Considering a network's fixed capacity, it is crucial to develop an algorithm that can support as many users as possible at their minimum QoS standards. Furthermore, algorithms proposed for *minimum rate* QoS usually target fully *backlogged* (full buffer) users with best-effort traffic but do not focus on guaranteed resource availability during every QoS measuring window. This makes it really hard to directly adopt such algorithms for the next generation applications that need VBR services with very high reliability. Thus, it is necessary to develop an efficient spectrally efficient algorithm paired with a complementary admission control strategy that assists in guaranteeing a minimum level of QoS for all admitted users while also distributing excess resources fairly. The literature on *minimum rate* QoS scheduling falls into two categories: heuristic algorithms that may not work well in the case of heterogeneous users and theoretical algorithms that are optimal under asymptotic/steady state conditions. In practice, such algorithms fail due to the uncertainty of wireless channels and the unpredictability of satisfying QoS requirements within the specified time window. Novel applications such as Augmented Reality (AR) need service protection with a guaranteed minimum bit rate over a duration of a few hundred milliseconds. Such applications require the transmission of bursty traffic with large packets which often have periodic/quasi-periodic arrival times. This calls for a measurement based approach to learn and/or predict future channel variations and online service tracking to monitor timely *minimum rate* QoS delivery.

4.2 Chapter Contributions

In this chapter, we consider wireless scheduling algorithms designed to meet minimum rate constraints over pre-specified time windows, e.g., on the order of a few hundred milliseconds, for each user. We propose and evaluate a class of measurement-based wireless schedulers that are based on tracking the empirical distribution of users' wireless channel capacity variations and on tracking the users' current QoS service deficit. The key underlying idea, and challenge, is to find a balance between opportunistically scheduling users which are currently experiencing the best channels and prioritizing the scheduling of users that are running a deficit with respect to meeting their QoS requirements. A good scheduler thus might have two goals, on the one hand, that of minimizing the probability of failing to meet users' QoS requirements, and on the other that of maximizing the overall sum rate.

We first consider a simpler setting where users' QoS requirements and channel variations are homogeneous. The proposed scheduler denoted *Adaptive Slack for Greedy Opportunism*, simply estimates if there is sufficient time to meet users' requirements, assuming they are served at the mean rate. If so, the scheduler chooses amongst *all* users opportunistically, otherwise, it chooses opportunistically *only amongst users with a deficit*. We then propose a second scheduler denoted *Service Deficit Based Opportunism* which addresses heterogeneity and further improves upon reliability-sum throughput tradeoff. The second algorithm chooses users to schedule based on their estimated quantile of the current channel and the normalized deficit. We also formulate an *oracle-aided* linear optimization for the *minimum rate* QoS constrained scheduling problem to determine a bound on the overall sum rate. Through extensive simulations, we show that our proposed algorithm provides a much higher user rate than state-of-the-art schedulers such as QoS-PF [101], EXP-QoS [91], and weighted MaxQuantile [80] with better reliability when the network is critically loaded. We observe at least a two-fold increase in the mean rate for each user when compared to the EXP-QoS [91] algorithm.

This chapter is organized as follows. Section 4.3 describes the system model and sets up the bound on the best that one could do by formulating an *oracle-aided* linear optimization problem. Section 4.4 describes our proposed scheduling algorithms for achieving the minimum rate QoS for both homogeneous and heterogeneous user settings. Section 4.5 provides simulation results that compares the spectral efficiency of the proposed algorithms with other baseline algorithms. Finally, Section 4.6 includes some concluding remarks.

4.3 System Model

We consider discrete time downlink scheduling for a base station serving a set $\mathcal{U}$ of users with heterogeneous rate QoS requirements. We assume that each user has a fully *backlogged queue* and that a user i requires a minimum rate guarantee of $r^i_{\min}$ over a time window of τ^i. Such QoS requirements can be found in LTE specifications, known as Guaranteed Bit Rate (GBR) services (refer C). The base station transmit resources, i.e., Resource Blocks (RB) are modeled as a sequence of time-frequency slices that can be arbitrarily allocated to users on a per time slot basis. At any given time slot each RB can be assigned to at most one user and there are a total of $\overline{M}$ resources available to service all users.

Let the random variable $R^i_n \in \mathbb{R}^+$ denote the channel rate (bits per RB) that can be transmitted to user i if it is allocated a *single* RB on time slot n. A user may be allocated multiple RBs, but we assume a *flat fading* setting where the rate delivered to i is the same across RBs in a given time slot. Further, we assume $(R^i_n)_{n\in\mathbb{N}}$ are independent and non identically distributed (i.n.i.d) across time slots. Additionally, we also assume that a sufficiently large number of RBs are available every time slot so that the users' QoS requirements can be met. However, our goal is to devise schedulers for users that use a minimal number of RBs so that each user can eventually get a much higher rate than benchmark scheduling algorithms.

We consider a system model where resource allocation decisions are made every time slot, which is typically 1 ms in an LTE system. The decision of the scheduler at time n is assumed to be causal concerning knowledge of the current and past channel rates $(R^i_k)^n_{k=0}$ and user resource allocation allowing for opportunistic scheduling, i.e., taking advantage of channel rate variations across slots. In particular, we let $M^i_n \in \mathbb{R}^+$ denote the number of RBs allocated to user i on slot n by the proposed scheduler. Note that for ease of mathematical analysis, we consider fractional resource allocations for users possible. Such an allocation provides a service rate of S^i_n (total bits transmitted with potentially multiple RBs allocated) to the user i on time slot n given by, $S^i_n = M^i_n R^i_n$, and we define the cumulative service over any interval $[(l-1)\tau+1, l\tau]$ as follows,

$$S^i[(l-1)\tau+1, l\tau] = \sum_{n=(l-1)\tau+1}^{l\tau} S^i_n \,. \tag{4.1}$$

A user is *minimum rate* QoS compliant if,

$$S^i[(l-1)\tau^i+1, l\tau^i] \geq r^i_{\min}, \forall l \in \mathbb{N} \,. \tag{4.2}$$

In the next subsection, we develop a lower bound on the *number of resource blocks* required by any policy that guarantees the users' respective rate requirements are met. This will serve as a baseline or bound for the best spectral efficiency that one could achieve under *minimum rate* QoS constrained resource allocation.

4.3.1 Oracle-aided Scheduling

Typically user downlink data transfer requests come in at random times, add in the QoS window heterogeneity, we need a scheduler that can meet heterogeneous QoS requirements over disparate time windows $(\tau^i)_{i\in\mathcal{U}}$. Suppose the resource scheduler has complete information about the future channel variations of each user $R^i_n = r^i_n$ for the next τ' slots, where $\tau' = \text{lcm}(\tau^i,\, i \in \mathcal{U})$, where $\text{lcm}(\cdot)$ is the least common multiple of $\tau^i, i \in \mathcal{U}$. Then one could solve the following combinatorial integer optimization problem every τ' slots, to find the optimal resource allocation that maximizes the overall network sum rate while meeting the QoS constraints of all users, whenever feasible.

$$\begin{aligned}
\max_{M^i_n,\, i\in\mathcal{U}, n\in[1,\tau']} \quad & \sum_{i\in\mathcal{U}} \sum_{n=1}^{\tau'} r^i_n M^i_n \\
s.t. \;\; & S^i[(l-1)\tau^i+1, l\tau^i] \geq r^i_{\min},\ \forall i \in \mathcal{U}, \forall l \in \mathbb{N} : ((l-1)\tau, l\tau] \subseteq [1,\tau'], \\
& \sum_{i\in\mathcal{U}} M^i_n \leq \overline{M},\ \forall n \in [1,\tau'], \\
& M^i_n \geq 0, i \in \mathcal{U}, n \in [1,\tau'].
\end{aligned} \tag{4.3}$$

Although the optimization problem in (4.3) is NP-hard, we can simplify it by relaxing the integer value condition on M^i_n to allow fractional values. Note that the resulting optimal value for resource allocation is the best any scheduling policy can achieve (as it uses *oracle-aided* future information for all users).

4.4 Proposed algorithms

We first introduce a scheduler for the idealized *homogeneous setting* in order to motivate our scheduler design for the more practical heterogeneous case. We note that QoS constrained schedulers will realize an inherent trade off. If they focus on maximizing the sum rate, then this would impact the reliability/fairness with which the QoS requirements are met. Alternatively, if they focus on simply meeting the QoS requirements, they compromise the overall sum rate. The proposed algorithms achieve a balance between spectral efficiency and meeting the QoS requirements.

4.4.1 Adaptive Slack Algorithm

Suppose users have i.i.d channel realizations, denoted by the random variables $R_n^i, i \in \mathcal{U}$, and the same *minimum rate* QoS constraint (i.e., a minimum rate of $r_{\min}$ over a window τ). We consider synchronized QoS measuring windows across users for ease of exposition. Assuming that each user sees a channel rate that is around its mean, we can roughly estimate the number of time slots τ_{est} that are necessary to meet all of the users' rate QoS requirements. Based on this estimate, two scheduling phases will be considered over the QoS time window τ.

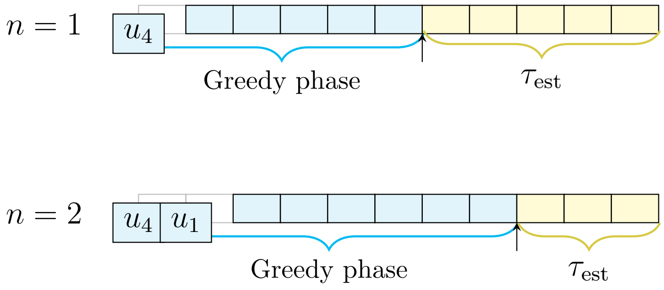


Figure 4.1: Illustration of the adaptive slack algorithm.

The algorithm's initial phase begins at the start of the QoS time window and persists until time slot $\tau - \tau_{\text{est}}$ (refer to blue shaded region in Fig. 4.1, also associated

Algorithm 3: Adaptive Slack for greedy Opportunism

```
1  while $n > 0$ do
2      if $n \pmod{\tau} = 0$ then
3          $\tau_{\text{est}} = |\mathcal{U}| \left\lceil \frac{r_{\min}}{\overline{M}\mathbb{E}[R]} \right\rceil$ ;
4          $D_n^i = r_{\min}$;
5      end
6      if $n \pmod{\tau} < \tau - \tau_{est}$ then
7          $\mathcal{U}' = \mathcal{U}$;
8      else
9          $\mathcal{U}' = \{i \mid D_n^i > 0\}$;
10     end
11     $i^* = \arg\max_{i \in \mathcal{U}'} R_n^i$;
12     $D_n^{i^*} = \min\left(0,\ D_{n-1}^{i^*} - R_n^{i^*} M_n^{i*}\right)$ ;
13     $M_n^{i*} = \overline{M}/R_n^{i*}$ ;
14     $\tau_{\text{est}} = \max\left(0, \tau_{\text{est}} - \frac{D_n^{i^*}}{\overline{M}\mathbb{E}[R]}\right)$ ;
15     $n = n + 1$ ;
16 end
```

with the *if condition* between Lines 6-8 in Algorithm 3). The scheduler picks user i^* with the highest wireless channel rate for resource allocation during the first phase on every time slot, i.e.,

$$i^* = \arg\max_i R_n^i, \tag{4.4}$$

where R_n^i is the instantaneous channel rate of user i in bits per RB at time n. Given that we have assumed i.i.d user channel variations, the *adaptive slack* algorithm is expected to serve each user equally likely during the greedy opportunistic phase. Whenever a user is scheduled, the algorithm (Line 12 in Algorithm 3) updates the variable D_n^i that tracks the amount of service required to meet the *minimum rate* QoS of user i.

The dynamic estimate for τ_{est} (the number of time slots required to meet the

QoS needs of all users) is updated every time slot according to the service already provided to each user. Therefore, we have access to an online estimate of the *slack* available for greedy opportunistic scheduling before the algorithm is forced to service users that need QoS constraints to be met (the yellow portion in Fig. 4.1). Given that the algorithm is highly opportunistic during the initial phase, one of two things can happen in every slot:

1. The algorithm has exhausted all *slack*, in which case the algorithm will be forced to only service those users that have QoS deficits.

2. All users' *minimum rate* QoS requirements have been met, allowing the scheduler to remain in the greedy opportunistic phase until the end of τ time slots.

We will refer to the second phase as the *QoS deficit* phase where the algorithm restricts its service to users with service deficit until all their QoS requirements are met. During the QoS deficit phase, the scheduler still picks the user with the maximum rate but from the reduced set of users $\mathcal{U}'$ with QoS deficit (refer Line 9 Algorithm 3). Every time slot the set $\mathcal{U}'$ is updated to reflect the users that still fall short of their rate requirements. This process repeats until the end of the QoS window of τ time slots, or when all the users' QoS deficits are met. At this point, if there are still slots left over within the current QoS window, the algorithm falls back to the greedy opportunistic phase.

In practice, the *Adaptive Slack* scheduler needs to be modified to address realistic settings where users' channel statistics and QoS requirements are heterogeneous. We introduce such a modified scheduler in the next subsection.

4.4.2 Service Deficit based Scheduler

In the heterogeneous channel setup, a user with the highest quantile channel rate [80] is considered the *best* concerning favorable wireless channel conditions. Mathematically, the highest quantile user index i^* is given by, $i^* = \arg\max_{i \in \mathcal{U}} F_{R^i}(R_n^i)$, where R_n^i is the instantaneous channel rate of user i on time slot n and $F_{R^i}(\cdot)$ is the Cumulative Distribution Function (CDF) of the channel rate realizations. For each time slot, the algorithm calculates a weight for each user that has a non-zero service deficit (i.e., $D_n^i > 0$) which is the sum of the instantaneous channel rate percentile and normalized service deficit as shown in Line 11 of the Algorithm 4. The *best user* is the one with the highest weight and will be allocated all the resources in the current slot if need be.

If the *best user* needs fewer resources than the total available, then the user with the second best weight will be served in the current time slot. If there are any residual resources after serving the *best* two users with service deficit, they are all allocated to the user $u \in \mathcal{U}$ that has the best channel rate percentile (not necessarily with service deficit), see Lines $14 - 17$ in Algorithm 4.

Algorithm 4 Complexity: The algorithm's run time complexity depends on the number of rate constrained users M and the number of past channel samples K used to estimate empirical channel rate distributions. A user's channel rate percentile is determined by sorting K past channel samples, which is of $O(K \log K)$ complexity, and then using their current rate value to determine where they are in the sorted array. An efficient implementation of empirical estimation requires each user to initialize the channel rate distribution with $O(K \log K)$ complexity. As the algorithm iterates, the oldest channel value is removed from the sorted list and the most recent one is inserted with a complexity of $O(\log K)$. A lookup table can then be used to determine the

Algorithm 4: Service Deficit based scheduler

```
while $n > 0$ do
    if $n \pmod{\tau^i} = 0$ then
        $D_n^i = r_{\min}^i$;
    end
    $k = 1$;
    $\mathcal{U}' = \{i \,|\, D_n^i > 0\}$;
    while $\mathcal{U}' \neq \emptyset \;\&\; k \leq 2 \;\&\; \sum_{i\in|\mathcal{U}|} M_n^i < \overline{M}$ do
        $k^* = \arg\max_{i\in\mathcal{U}'} \left(F_{R^i}(R_n^i) + \frac{D_n^i}{r_{\min}^i}\right)$;
        $M_n^{k^*} = \min\left(\frac{D_{n-1}^{k^*}}{R_n^{k^*}}, \overline{M} - \sum_{i\in|\mathcal{U}|} M_n^i\right)$ ;
        $D^{k_n^*} = \min\left(0,\; D_{n-1}^{k^*} - R_n^{k^*} M_n^{k^*}\right)$ ;
        $k = k + 1$ ;
        $\mathcal{U}' = \{i \,|\, D_n^i > 0\}$;
    end
    if $\sum_{i\in|\mathcal{U}|} M_n^i < \overline{M}$ then
        $i^* = \arg\max_{i\in\mathcal{U}} F_{R^i}(R_n^i)$;
        $M_n^{i^*} = \overline{M} - \sum_{i\in|\mathcal{U}|} M_n^i$ ;
    end
    $n = n + 1$ ;
end
```

percentile based on the position at which the current channel rate is inserted. All other steps in the algorithm are of $O(1)$ complexity for each user, resulting in an overall complexity of $O(M \log K)$. This algorithm is $O(MK)$ space-complex, mainly due to storing past channel values, and maintaining each user's QoS deficits.

Remark: We have so far assumed fully backlogged users. However, the proposed schedulers can handle real time users with any type of arrival pattern. For instance, if user i has periodic arrivals, we can maintain a virtual service token queue with tokens added at the start of each QoS time window τ^i. In the case of stochastic arrivals, one could add tokens to the virtual queue every time a packet arrives, with an appropriate bound on the maximum number of tokens that can be added (based on the maximum bit rate permissible by design).

4.5 Simulation results

We shall consider a BS serving ten users with a *minimum rate* QoS constraint over a fixed time window, where user i's channel rate on slot n is R_n^i per RB based on its received Signal to Noise Ratio (SNR). The received SNR was modelled using the 3GPP Urban-Micro path loss model [3], with flat Rayleigh distributed small scale fading. We will simulate both the *symmetric* user setting – with homogeneous user rates and QoS constraints, and the *heterogeneous* case where the users have heterogeneous channel rate distributions and different QoS requirements. Each user's distance from the BS is as indicated in Table 4.2 for the heterogeneous setting. Since we discuss both the symmetric and heterogeneous user settings, the users' distance (from the BS) for the symmetric setting is mentioned in the relevant subsection. Furthermore, we assume bounded channel realizations, where the SNR lies between -6.934 dB $\leq$ SNR $\leq$ 20 dB. The total number of RBs available for resource allocation to rate constrained users is set to 60 for all our simulations unless otherwise specified.

We consider a practical Modulation and Coding Scheme (MCS) mapping SNR to a coding rate in order to determine the user's transmission rate R_n^i using the MCS table in [2, Table 5.2.2.1-2]. Recall that our schedulers require $F_{R^i}(\cdot)$ – the CDF for each user's transmission rate. We estimate these using the empirical distribution of the respective user's past channel rate realizations. At each time instant we use the last 100 channel rate realizations to determine the empirical distribution of each user's rate variations. As stated in our system model, we assume fully backlogged user queues with packets of fixed size (1 KB) that can be allocated resources each time slot.

Scheduler	W_n^i
Oracle-aided (Sec. 4.3.1)	Not Applicable
QoS PF [101]	$\tau^i r^i_{\min} \dfrac{R_n^i}{T_i}$
EXP QoS [91]	$a_n^i \exp\left(\dfrac{a_n^i W_n^i - \overline{aW}}{1+\sqrt{\overline{aW}}}\right)$
Weighted MaxQuantile [80]	Tokens based MaxQuantile $F_{R^i}(\cdot)$

Table 4.1: Benchmark scheduling schemes.

We shall employ the proposed *Adaptive Slack* algorithm for scheduling in the symmetric user setting (homogeneous rates, QoS requirements), and the proposed *Service Deficit* based scheduling algorithm for the heterogeneous setting. We use a set of extensive benchmark scheduling algorithms listed in Table 4.1 to compare and contrast with the performance of our proposed scheduling algorithms. Minimum rate targets for the EXP scheduling algorithm [91] are achieved by maintaining virtual token queue V_n^i for each user i, where $r^i_{\min}$ tokens arrive every τ_i time slots. The values for a_n^i, W_n^i and $\overline{aW}$ are given in Table. 4.1, which were selected as recommended in [91], are as follows: $a_n^i = \frac{R_n^i}{\overline{R^i}}$, $W_n^i = \frac{V_n^i}{r_{\min}^i}$ and $\overline{aW} = \frac{1}{k}\sum_{i=1}^{k} a^i W_n^i$. Here $\overline{R^i}$ denotes the statistical mean of the channel strength of user i, which we assume is known to the scheduler.

One of the foremost important metrics for a minimum rate constrained scheduler is the number of users that can be supported with the guaranteed minimum rate for all. Also crucial for any network is the amount of additional data that can be provided to each of its users, i.e., the *excess rate* over the minimum rate requirement. Finally, it is also important to have algorithms that are robust to load variations in the network. A network is said to be *critically loaded* when one or more benchmark algorithms fail to support the minimum rate requirements of one or more users. But when there are fewer users in the system, the amount of the excess rate that a scheduling algorithm can provide its users and the level of fairness with which it can distribute the resources to users matter. In the sequel, we address all these key metrics and demonstrate the robustness of our proposed algorithms through extensive simulations.

We demonstrate the robustness of our proposed algorithms through extensive simulations using the mean rate achieved by each user as the performance metric. We simulated 10^7 slots to generate each of the plots in this section, resulting in a ± 0.1 error for the estimated mean excess rate provided to each user with 99% confidence interval. The maximum RBs available to the network is only 60, and therefore the network is *critically loaded* (because one or more benchmark algorithms are not able to meet the QoS needs of all users).

4.5.1 Symmetric case

For this subsection, all users are located at a distance of 500 m from the BS. Also, they have the same *minimum rate* QoS requirement of 20.48 KB where the time window τ was set to 100 time slots for all users. Fig. 4.2 shows the CDF for the mean rate provided to each of the users under the symmetric network setup. As can be seen, in Figure. 4.2 the mean excess rate provided to users is the highest for the oracle-aided policy followed by the *Adaptive slack* algorithm. Some important points of note are listed below:

- *Adaptive slack* algorithm achieves around a two-fold improvement in the mean rate when compared to the EXP QoS [91] scheduling algorithm.
- *Adaptive slack* algorithm provides a marginally higher mean user rate than the QoS-PF based technique.

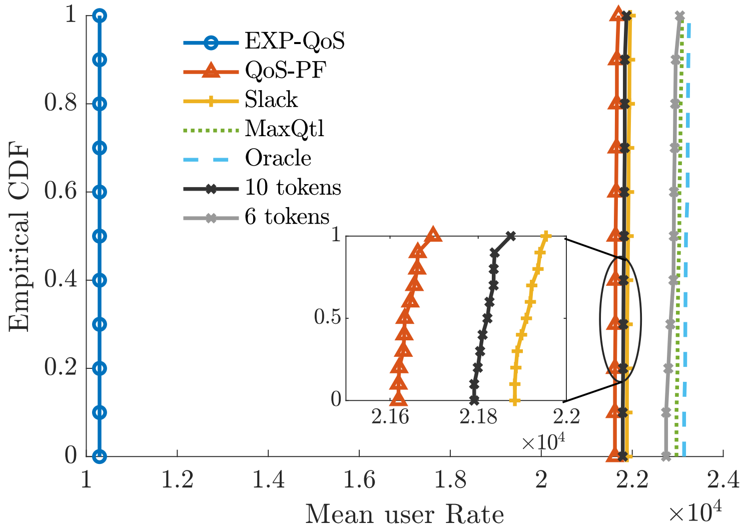


Figure 4.2: Mean user rate over $\tau = \mathbf{100}$ time slots when 60 RBs are available every time slot to all users in the network.

Note that while the weighted Max Quantile [80] provides user mean rates comparable to the *oracle-aided* algorithm, it does not meet the user minimum rates with high reliability like the other algorithms. This has been observed across all values for the token design parameter provided in [80].

4.5.2 Heterogeneous case

For the results presented in this subsection, the users' distances from the BS are as specified in Table 4.2. The time window τ for the *minimum rate* QoS of 20.48 KB is 50 time slots for users with odd indices. Users with even indices also have the same *minimum rate* QoS but over a time window of $\tau = 100$ time slots. The *Slack Deficit* algorithm (see Algorithm 4) is used for resource allocation in this heterogeneous setting. Fig. 4.3 shows the mean rates achieved for all the users, while the probability of meeting the rate requirement is tabulated in Table. 4.2.

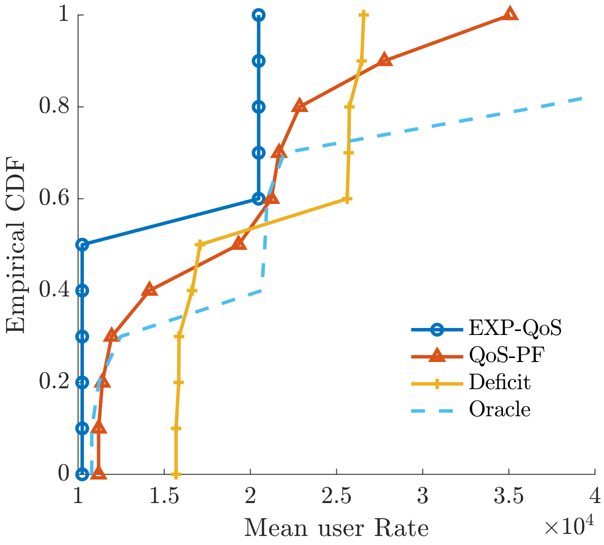


Figure 4.3: Heterogeneous case: Mean user rate when 60 RBs are available every time slot to all users in the network.

The key takeaways from the figure and table are as follows. *Service deficit*

User Index	Distance	EXP	QoS-PF	Deficit
1	300	0.3710	0	0.0044
2	320	0.3599	0	0
3	350	0.3869	0	0.0063
4	400	0.3738	10^{-5}	0
5	450	0.4077	10^{-4}	0.0160
6	500	0.3818	10^{-5}	0
7	550	0.4287	0.0597	0.0528
8	600	0.4027	10^{-5}	0
9	620	0.4583	0.3419	0.0987
10	680	0.4174	10^{-4}	0

Table 4.2: Probability of not meeting the minimum rate requirement.

algorithm provides the best mean rates to users without compromising reliability - note that the QoS-PF algorithm does not handle heterogeneous settings very well. As can be seen from Table. 4.2, QoS-PF fails to meet the rate requirements for the weakest user that has only 50 time slots to meet its QoS constraint. (Note that the other weak users have 100 time slots and hence meet their rate requirement with very high reliability.) It is interesting to note that our *Service deficit* algorithm is able to meet the minimum rate requirement of all users that have a longer time window with probability 1, irrespective of their respective channel strengths! Fig. 4.4 illustrates a comprehensive performance summary of the various algorithms considered in terms of the sum rate achieved by each algorithm versus it's average probability of failure to meet the QoS. Clearly, our proposed *Service Deficit* algorithm for heterogeneous users provides the best sum rate and the least probability of failure to meet QoS both on average and absolute terms (see Table. 4.2).

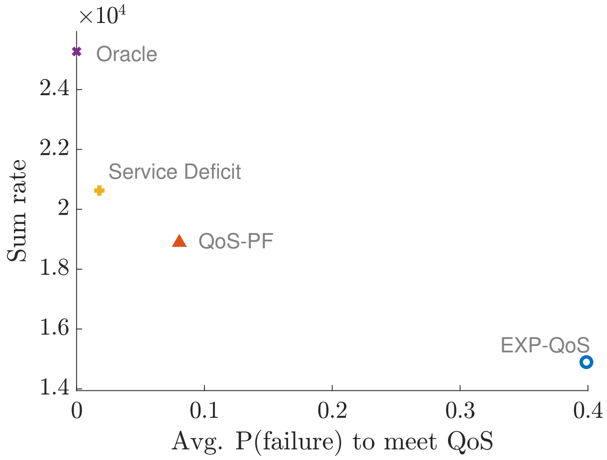


Figure 4.4: Comprehensive performance

In order to demonstrate the mean user rates when the system is **not** *critically loaded*, we include simulation results for the same set of users when the total number of resources available is set to 100 RBs. Fig. 4.5 shows that our proposed algorithms provide a fair distribution of excess resources across users with similar QoS, while meeting the minimum rate QoS of each of these users.

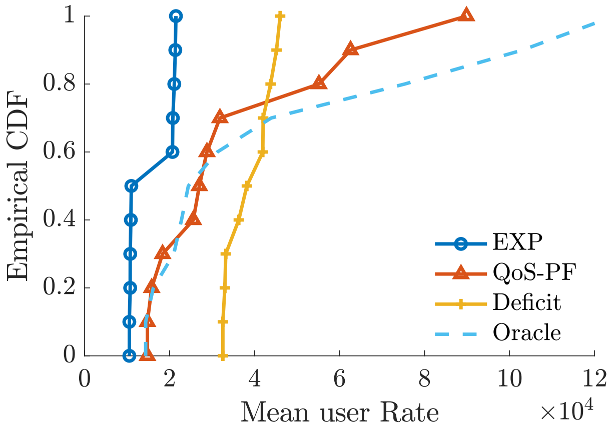


Figure 4.5: Mean rates provisioned to users with a network capacity of 100 RBs.

4.5.3 User capacity

In this subsection, we will show the number of users that our proposed *slack deficit* algorithm can support as a function of the reliability with which minimum rate QoS requirements are met for each user. We consider the heterogeneous setting where the users' distances are as specified in Table. 4.2. All the users have the same minimum rate QoS of 20.48 KB but the time window for measuring QoS is set to 50 for all users with odd indices and 100 for those with even indices. Figure. 4.6 demonstrates how our proposed *Slack Deficit* algorithm supports more users than other *causal** benchmark algorithms. We will explain the intuition behind what enables our proposed algorithm to support more users in the subsequent subsections. While the number of users that meet QoS is similar for slack deficit and QoS-PF, in the sequel we show how this is not the case when we evaluate reliability of both these algorithms.

4.5.4 Weighted Fairness index

One of the most important performance criteria for a scheduler is the fairness with which it allocates resources to its users. In an environment where users' QoS/channel rates are heterogeneous, fairness is important. A scheduler might find it easy to simply meet the QoS needs of users with a strong channel, thereby severely impacting the QoS performance of weak users. Some of the most popular criteria for measuring the fairness of scheduling algorithms are: max-min fairness, proportional fairness, throughput fairness, and time fairness. Note that all these fairness indices were introduced with best-effort traffic in mind. However, what we need is an index

*The *oracle-aided* algorithm is non-causal as it requires future user channel rate information rendering it impractical.

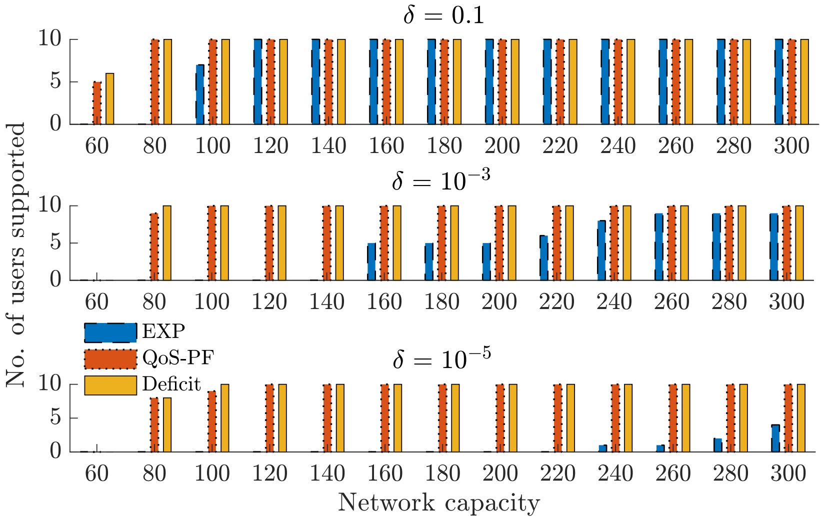


Figure 4.6: Number of users supported for various reliability values as a function of the network capacity.

that can measure the fairness with which resources are allocated among users with similar QoS requirements, i.e., how well an algorithm can provide meet minimum rate QoS that is not impacted by weak channel strength. Towards this end, we introduce a new fairness criterion for QoS-constrained scheduling and show that our proposed *Service deficit* algorithm offers the best fairness index among all benchmark algorithms that have comparable reliability performance.

Let $r^1, r^2, \ldots, r^k$ denote the sequence of mean rates achieved by all users in the system. Then the Jain's fairness index [53] for these users is given by,

$$\mathcal{J}(r^1, r^2, \ldots, r^k) = \frac{(\sum_{i=1}^{k} r^i)^2}{k \cdot \sum_{i=1}^{k} (r^i)^2}. \tag{4.5}$$

In the case of heterogeneous users, we introduce the following metric, namely the QoS fairness index. Let c^i denote the windowed mean rate achieved by user i, i.e., the average rate allocated to user i (by any scheduling algorithm) measured over disjoint windows of length τ_i, which is as defined by the QoS constraint of the user. Then the

QoS fairness index for these users is given by,

$$\mathcal{Q}(c^1, c^2, \ldots, c^k) = \frac{\left(\sum_{i=1}^{k} \dfrac{c^i \tau^i}{r^i_{\min}}\right)^2}{k \cdot \sum_{i=1}^{k} \left(\dfrac{c^i \tau^i}{r^i_{\min}}\right)^2}. \tag{4.6}$$

Homogeneous QoS fairness: First we consider the scenario where the users all have homogeneous QoS requirement of $20.48KB$ over 50 time slots, but with heterogeneous channel rate variations where user locations are as specified in Table. 4.2. We then calculate the QoS fairness index for each of the non-causal scheduling algorithms under different load conditions – light, critical, and overload. (The network is considered to be *overloaded* when the *oracle-aided* policy is not able to find a feasible resource allocation solution that can meet the QoS constraints of all users with probability 1.) Table. 4.3 shows the QoS fairness indices for all the algorithms as a function of the network capacity (in RBs). It can be seen that except for the overloaded condition (network capacity = 40 RBs) and the *critically loaded* conditions, all algorithms have similar fairness performance when the QoS constraints are homogeneous.

	Network Capacity (RBs)			
Algorithms	40	60	80	100
Service Deficit	0.9656	0.9981	0.9995	1
QoS-PF	0.9001	0.9928	0.9998	1
EXP	0.7640	0.9351	1	1

Table 4.3: QoS fairness across algorithms for the Homogeneous case.

Heterogeneous QoS fairness: Next, we consider the heterogeneous setting with user distances as specified in Table. 4.2. The time window τ for the uniform minimum QoS rate of 20.48 KB is 50 time slots for users with odd indices, while the rest have

$\tau = 100$ time slots. Figure. 4.7 shows the QoS fairness indices for all users as a function of the network capacity (the number of RBs available every time slot). It can be seen that the QoS fairness index of our proposed *Service Deficit* algorithm exceeds that of other benchmark algorithms that we have considered in this paper. Also, note that the QoS fairness index of our proposed algorithm is quite robust to network load conditions[†]. It can be seen that EXP and *Service Deficit* algorithms have relatively better fairness in resource allocation than the other algorithms. To better visualize the fairness of these algorithms when there are more *slack* resources in the system, we present in Figure. 4.5 the mean rates allocated to the users when the network capacity is 100 RBs.

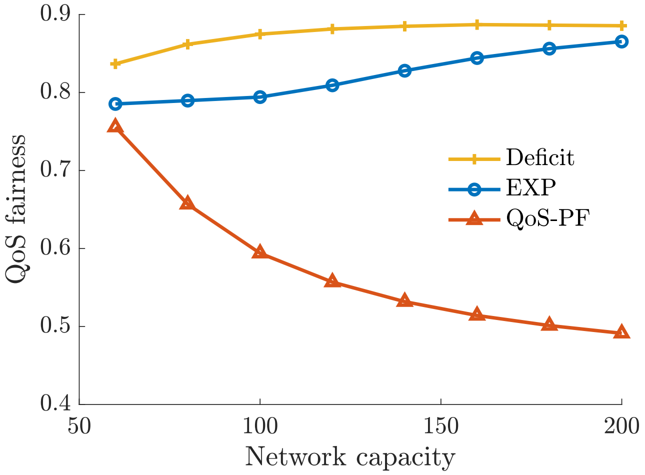


Figure 4.7: QoS fairness indices for benchmark algorithms as a function of the network capacity.

In summary, our proposed algorithms perform better than existing scheduling algorithms in terms of reliability and being able to support more users under similar

[†] We simulate varying network load conditions by fixing the number of users in the network and changing the number of resources to share among these users.

wireless channel and load conditions. Intuitively, one could explain the reason for the better performance as follows. The total resources available to the scheduler over the duration of τ time slots can be categorized into two portions - the fraction of *bare minimum* resources that are needed to meet the QoS requirement of all users and then the *slack* resources (those which are available after meeting the QoS needs of all users). It is important to perform opportunistic scheduling decisions (as it improves spectral efficiency) while meeting the QoS needs in order to make room for additional resources that can later be exploited through greedy opportunism. We conjecture that our proposed algorithms strive for a good balance between conservative QoS-constrained provisioning and greedy opportunism in order to maximize the overall excess rate that the scheduler is able to provide all users without violating their QoS requirements.

4.6 Chapter Summary

We have proposed a new class of *slack* based opportunistic wireless schedulers designed to meet disparate QoS requirements of users with possibly heterogeneous channel variations. Our study shows that our proposed schedulers can provide higher mean rates to users than existing QoS-aware schedulers that also meet their minimum rate requirements. Furthermore, for wireless schedulers to be able to effectively support more users under QoS-constrained scheduling, they have to find a sweet spot between maximizing long-term rates and ensuring fair resource allocation. Finally, we have shown that our algorithms are able to handle graceful degradation of services under *critical load* conditions, with better user rates than other algorithms with comparable fairness indices and/or reliability.

Chapter 5: Distributed Reinforcement Learning for Interference Mitigation[†]

Wireless networks have undergone tremendous change over the past decades, going through various technology generations supporting higher data rates, improved network coverage, and better user experience. A major factor enabling this progression has been network densification. While network densification improves users' data rates, it also may lead to reduced user traffic aggregation increasing the likelihood of bursty interference from neighboring base stations rendering static frequency reuse techniques less effective. Channel uncertainty, dynamic traffic, high interference, transmit power constraints and limited availability of spectrum make resource allocation a challenging task in dense wireless networks. This makes a dynamic frequency reuse scheme that can self tune to the network and traffic conditions highly desirable.

Resource allocation and power adaptation are problems in wireless systems where Reinforcement learning (RL) has proven to have some promise. Two key settings have been considered: cooperative Markov games and non-cooperative distributed games. Cooperative games typically draw on a more centralized decision making approach that uses extensive information sharing posing practical limitations [70]. By contrast, non-cooperative games typically involve distributed decision making based on local information, typically leading to sub-optimal but more scalable approaches suitable to adapt to dynamic interference and user traffic.

[†] Publications based on this chapter: [20] Geetha Chandrasekaran and Gustavo de Veciana. Distributed Reinforcement Learning based Delay Sensitive Decentralized Resource Scheduling. In *2023 Proceedings IEEE WiOpt Workshop on Machine Learning in Wireless Communications (WMLC)*, 2023

5.1 Related work

Reinforcement Learning: Numerous works have proposed an RL based approach to solve the wireless resource allocation problem. For example,[73] proposes a centralized learning system that periodically updates BS's policies (neural network model) giving it a partially centralized learning architecture. By contrast, we consider a completely distributed learning approach with variants that do not need any information exchange across BSs in the network. The work in [13], considers a heterogeneous network (HetNet) with macro and femto BSs sharing the spectrum in the same area and proposes a Q-learning based algorithm for carrier selection and power allocation, but does not account for users' channel variability or interference from other HetNets. Most of the current literature based on a distributed RL approach to resource allocation, see [116, 25, 114, 74], ignores the impact of user traffic dynamics and/or resource scheduling on the effective network throughput. In our work, we show that user traffic dynamics can be learned and devise resource allocation strategies that leverage this information.

RL algorithms have also been applied to mitigate interference in wireless networks. Techniques such as dynamic Q-learning [97], neural network [55], actor-critic RL [111], Deep Q-network (DQN) [89] have been used in a variety of settings such as HetNets, Cognitive radio and vehicular communication, with the goal of resource allocation that either minimizes or mitigates interference. However, these articles fail to acknowledge and/or do not consider opportunistic scheduling, which we believe cannot be ignored when building a solution to distributed resource allocation.

Stochastic games and scheduling: A centralized algorithm for resource allocation for interference mitigation and scheduling is considered in [86], but the approach results in increased computational complexity for larger networks. In contrast, we design autonomous learning at each BS that is scalable and of the same complexity irrespective of the network size. A downlink power control stochastic game between a macro BS and its co-located small cell BSs has been considered in [105] without considering the impact of inter cell interference from other macro BSs. A resource allocation game among users using a Code-Division Multiple Access (CDMA) system, with power and rate control for each user, has been considered in [72]. Note that interference from neighboring CDMA networks has not been considered, this impacts the feasible range of power and transmit rate values. Distributed learning for resource selection and/or power allocation have also been considered in [43, 75, 84, 77], but one or more of the following are not considered: interference from neighboring BSs, user traffic dynamics, and channel uncertainty due to time varying interference. For a more comprehensive survey of existing distributed learning approaches, the interested reader is directed to [12].

Coupled queues: It is fundamentally challenging to share wireless spectrum among users of different BSs because the transmission rate declines for all users scheduled over the same resources. As a result, the service rate at each of these users is now influenced by interfering transmissions from other BSs, creating a *coupling* between the queues in terms of the maximum throughput that these users can see and the associated delay at each of the user queues. The stability region for such coupled users has been investigated in the literature, see [85, 102, 109], where any amount of interference causes a complete loss of packets. The vast majority of *scheduling policy* literature on coupled queues [30, 81, 62, 16, 61] deal with the case where both queues can be serviced by the same server.

Parallel queues with coupled service rates in the presence of channel aware scheduling have been discussed in [17], nevertheless, cooperative strategies for interfering BSs have not been investigated. It provides a detailed analysis of convergence and the 'capacity' region associated with the proposed policies. Some recent work, with realistic queue aware transmission schemes such as [29, 40] mainly focuses on stability with results that depend on the assumed system model. While [14] investigates a more realistic network with coupled queues that are not saturated, the results are mainly applicable to a single shared resource. There is also a study of the stability region of two parallel queues with coupled service rates in [17], however, this study does not consider any interference mitigation policy between the queues. This chapter shows that appropriate coordination between coupled users, with a carefully chosen resource scheduler, can help improve the sum user throughput when multiple resources are shared by coupled users.

5.2 Chapter Contributions

We consider the design of a resource allocation algorithm with dynamic user traffic for BSs coupled through interference. We propose a novel approach based on two coupled sub problems which permit us to explore RL based interference mitigation, having chosen a state-of-the-art (opportunistic) throughput optimal scheduler. This in turn reduces the state-space of the RL resource allocation problem leading to a quick training time (order of a couple of minutes in real time) and a resulting improvement in system capacity and performance. The main contributions of this chapter are as follows:

1. We propose a systematic decomposition approach to optimizing frequency reuse under a *predetermined* dynamic user scheduling policy geared at making distributed RL techniques possible. To the best of our knowledge, this has not been previously considered.

2. We propose and validate a proxy metric (reward) that enables distributed RL agents (base stations) to learn the interference-driven coupling amongst BSs.

3. Multiagent RL is a non-stationary stochastic game, existence and convergence to a Nash Equilibrium (NE) under specific conditions has been established in [51]. We show that our proposed algorithm satisfies these conditions, and hence show the existence of and convergence to an NE.

4. We analytically show that our proposed distributed learning policy converges to a policy that has a higher capacity region when compared to the full frequency reuse policy.

5. We show the existence of a distributed frequency reuse policy with a larger stability region than the full frequency reuse by considering scheduling coordination between two coupled queues.

6. Finally, we show that the choice of Max weight [103] policy as the pre-determined resource scheduler at each BS is *throughput optimal*.

7. We evaluate and compare through detailed simulation the potential of our proposed distributed approach vs aggressive and more centralized baseline algorithms. Our results exhibit capacity gains of 5-25% over full frequency reuse as well as associated improvements in delay performance with improved energy efficiency on the order of 9-34%.

This chapter is organized as follows. Section 5.3 describes the system model and sets up the non cooperative Markov game among the BS agents. Section 5.4 describes our proposed distributed RL algorithm for learning a frequency reuse policy at each BS. Section 5.5 establishes the existence of Nash Equilibrium and additionally provides ordering of the capacity/stability regions of the various Markov game settings. Section 5.6 presents the main theoretical results of this chapter by analyzing a pair coupled user queues. Section 5.7 provides simulation results that compares the capacity and stability region of the proposed RL based frequency reuse policies with other baseline algorithms. Finally, Section 5.8 includes some concluding remarks.

5.3 System Model

We consider downlink transmissions from a set of wireless BSs $\mathcal{B}$ of cardinality $|\mathcal{B}| = B$, serving a set of users/devices $\mathcal{U}$ such that $|\mathcal{U}| = U$, as shown in Fig. 5.1. The dynamics of the system evolve in discrete time, corresponding to transmission frames that are synchronized across BSs. Each frame consists of N Resource Blocks (RBs) each corresponding to a slice of subcarriers and time slots within the frame. Each RB can be assigned by a BS b to serve at most one of its set of associated users $\mathcal{U}^b$. Let $A_u(t)$ be a random variable denoting the arrivals (in packets) for user u during time slot t and thus available for transmission at $t+1$. We shall assume that a user's arrivals across time slots are independent and identically distributed (i.i.d). Let $\lambda_u = E[A_u(t)]$ denote the mean packet arrivals per time slot for user u and let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_U)$. In the sequel, $Q_u(t)$ will denote the queue length (in packets) of user u, i.e., the data available for transmission in time slot t and $\boldsymbol{Q}^b(t) = (Q_u(t) : u \in \mathcal{U}^b)$ the queues at BS b, while $\boldsymbol{Q}(t) = (\boldsymbol{Q}^b(t) : b \in \mathcal{B})$ the overall queue state of the system.

The channel gain between BS b and user u in slot t is modeled by a random variable $G_u^b(t)$ and assumed i.i.d across time. Let $\mathbf{G}(t) = (G_u^b(t) : b \in \mathcal{B}, u \in \mathcal{U})$ denote the gains amongst all BSs and users, while $\mathbf{G}^b(t) = (G_u^b(t) : u \in \mathcal{U}^b)$ denotes

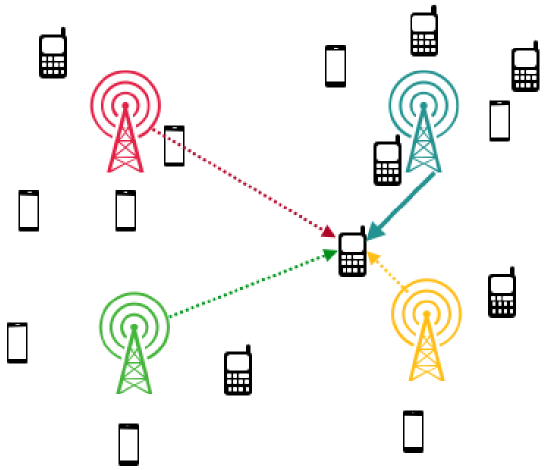


Figure 5.1: Network Model with dotted lines showing the interference caused by neighboring base stations.

solely those between BS b and its associated users. For simplicity, we will assume flat fading i.e., the channel gains for all the subcarriers within a resource block are the same. We denote the mean channel gains by $\bar{g}_u^b = E[G_u^b(t)]$ along with associated vector notations $\bar{\boldsymbol{g}}^b$ and $\bar{\boldsymbol{g}}$. The allocation of RBs to users is modeled as a two-step process. First, a frequency reuse decision is made which determines the subset of RBs *available* for user allocation at each BS. A set of binary decisions are made at each base station b for an RB k on slot t: $S_k^b(t)$ is such that if $S_k^b(t) = 1$ if RB k is available for use by the BS, and if $S_k^b(t) = 0$ it is not to be used. Second, a scheduling decision is made determining which (if any) users are scheduled to transmit on the *available* subset of RBs. We let $\mathbf{S}^b(t) = (S_k^b(t) : k = 1, \ldots, N)$ denote the frequency reuse state of BS b at time t and $\mathbf{S}(t) = (\mathbf{S}^b(t) : b \in \mathcal{B})$ the overall frequency reuse state of the network. It will be convenient to let $\mathcal{S}^b(t) = \{k : S_k^b(t) = 1\}_{k=1,\ldots,N}$ denote the set of available RBs at base station b.

Generally, a scheduling policy $\mathbf{h}$ is an assignment h^b for each BS b of the available RBs to its users. The assignment may depend on the available information denoted $\mathbf{I}^b(t)$, so a scheduling policy h^b for BS b is a mapping,

$$h^b(\,\cdot\,;\mathbf{I}^b(t)) : \mathcal{S}^b(t) \to \mathcal{U}^b \cup \{0\}\,, \tag{5.1}$$

assigning each RB made available by the frequency reuse policy $\mathcal{S}^b(t)$ to one of its users $\mathcal{U}^b$ or none at all, represented by user 0. Typically a BS scheduler will only have local information such as its users' channel gains and queue lengths, e.g., $\mathbf{I}^b(t) = (\mathbf{S}^b(t), \mathbf{G}^b(t), \mathbf{Q}^b(t))$. For simplicity we shall equivalently represent the result of scheduling via binary variables $\mathbf{h}(t) = (h^b_{uk}(t) : b \in \mathcal{B}, u \in U^b, k = 1, \ldots N)$ where $h^b_{uk}(t) = 1$ if the scheduler allocated an available RB $k \in \mathcal{S}^b(t)$ to user $u \in \mathcal{U}^b$ of BS b, otherwise it is 0.

In practice, a BS's scheduler has access to the Channel Quality Indicator (CQI) as well as estimates of previously observed interference and/or success/failure of transmissions for each of its associated users, based on which it estimates the users' current Signal to Interference and Noise ratio (SINR). For simplicity, we assume an adaptive modulation and coding scheme at the transmitter that can make use of this information to achieve a data rate close to the Shannon capacity. We understand that the Shannon capacity serves as a *rough* upper bound to the achievable rate, nevertheless, to keep things simple we use Shannon capacity as a rate metric to compare various algorithms proposed in this chapter.

Due to possible interference from neighboring BSs, the transmission user rate under a given resource schedule is a complex function of all scheduled users. An idealized model might be as follows: if $h^b_{uk}(t) = 1$ the SINR for user u of BS b on RB k is

$$\mathrm{SINR}^b_{uk}(t) = \frac{PG^b_u(t)}{\sum\limits_{b':b'\neq b}\sum\limits_{u'\in\mathcal{U}^{b'}} Ph^{b'}_{u'k}(t)G^{b'}_u(t) + N_0}, \tag{5.2}$$

where the numerator corresponds to the received transmit power, and the denominator is the sum of intercell interference and noise. The downlink transmission rate to user $u \in \mathcal{U}_b$ on resource k at time t is given by,

$$c^b_{uk}(t) = n\mu \frac{W}{2} \log(1 + \mathrm{SINR}^b_{uk}(t)) \text{ bits}, \tag{5.3}$$

where n is the number of subcarriers per RB and μ is the time duration of an RB. Thus aggregating across the RBs of BS b we denote the total transmissions to user u in slot t as,

$$c_u(t) = \sum_{k=1}^{N} h^b_{uk}(t) c^b_{uk}(t) \text{ bits}\,, \tag{5.4}$$

where we suppressed the superscript b in $c^b_u(t)$ since each user is served by only one BS. Hence, under such a scheduling policy the queue dynamics for user u are given by

$$Q_u(t+1) = [Q_u(t) - f(c_u(t))]^+ + A_u(t)\,, \tag{5.5}$$

where $[x]^+ = \max[0, x]$ and $f(x)$ is an integer valued non-decreasing function on x modeling the packet departures at a user queue as a function of the SINR.

In the sequel, we will find it useful to introduce the following notation. Note that given a frequency reuse state $\mathbf{S}(t) = \mathbf{s}$, a scheduler (5.1), channel and queue states $\mathbf{G}(t) = \boldsymbol{g}$, $\mathbf{Q}(t) = \boldsymbol{q}$, the service to user u can be written as,

$$c_u(t) = c_u(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q}). \tag{5.6}$$

Note that cellular networks can determine the *interference-free* signal to noise ratio (SNR) based on the user location through state-of-the-art machine learning techniques [59]. For a given user $u \in \mathcal{U}^b$, the SNR at time t is denoted,

$$\mathrm{SNR}_u(t) = \frac{P G^b_u(t)}{N_0}, \tag{5.7}$$

and a user's *effective "interference free" capacity* per RB for a channel strength $G_b^u(t) = g_b^u$ is denoted,

$$\kappa_u(g_b^u) = n\mu \frac{W}{2} \log\left(1 + \frac{P g_u^b}{N_0}\right) \text{ bits}. \tag{5.8}$$

The three main sources of variability in the network are due to user packet arrivals, channel variations, and dynamic interference across resource blocks. Each BS allocates resources to its users based on the channel strength and user queue lengths in the presence of dynamic interference from neighboring BSs. In principle, one could design a centralized entity that collects channel strength and user queue length across all BSs to perform joint resource allocation. Alternatively, one could fix a scheduler (say MaxWeight) and let each BS figure out its resource selection policy independently. We propose a separation of concerns by allowing solving for a frequency reuse policy independent of user resource scheduling, i.e., fix a throughput optimal scheduler and let each BS act as an independent agent to learn a resource selection strategy that minimizes interference. In these circumstances, one can think of a Markov game among the BSs for resource selection. It is challenging to select a good set of resources because the user queues in the network are coupled through interference.

The design of an optimal frequency reuse and scheduling policy for this stochastic network system with queues that are coupled through interference is an exceedingly challenging problem. In this chapter, we propose a separation of concerns where the underlying BS schedulers are fixed, e.g., to a state-of-the-art opportunistic scheduler based on local information. Given the underlying scheduler, we propose to have BSs learn how to manage frequency reuse so as to reduce the impact of inter-cell interference.

5.3.1 Markov Game: Learning frequency reuse policies

We formulate the problem of determining an overall frequency reuse policy across BSs as a Markov game [68] where each BS decides on it's own reuse policy so as to either (a) maximize its own reward, or (b) maximize a shared network reward. The rewards are a result of the BSs' frequency reuse decisions, and the underlying BS scheduling policies as well as the underlying environment/dynamics.

More formally, we consider a B-player Markov game

$$\langle \mathcal{S}^1, \ldots, \mathcal{S}^B; \mathcal{A}^1, \ldots, \mathcal{A}^B; p^1, \ldots, p^B; r^1, \ldots, r^B \rangle \tag{5.9}$$

including the following elements.

- $\mathcal{S}^b = \{0,1\}^N$ denotes the set of possible frequency reuse states for for BS b i.e., values $\mathbf{S}^b(t)$ can take.
- $\mathcal{A}^b$ denotes the set of all possible actions BS b can take.
- $p^b(\boldsymbol{s}'^b|\boldsymbol{s}^b, \boldsymbol{a}^b)$ models the transition probabilities to the next state $\boldsymbol{s}' \in \mathcal{S}$, given that the current state and action pair given by $(\boldsymbol{s}^b, \boldsymbol{a}^b)$.
- $r^b(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q})$ corresponds to a reward associated with users scheduled at BS b on a given time slot conditional on the *overall* frequency reuse state $\boldsymbol{S}(t) = \boldsymbol{s}$, channel gains $\boldsymbol{G}(t) = \boldsymbol{g}$ and user queues $\boldsymbol{Q}(t) = \boldsymbol{q}$.

Below we describe several approaches to defining the rewards and action space for this game. Note that the frequency reuse game is such that BSs do not have access to the entire network state, in particular to the frequency reuse state, channels and queues of *other* BSs.

5.3.2 Actions and Rewards: Non cooperative Markov game

An action $\boldsymbol{a}^b \in \mathcal{A}^b$ determines the next frequency reuse state for BS b. We consider two possible models for the action space $\mathcal{A}^b$, $\{0,1\}^N$ or $\{1,\ldots,N\}$, depending on the admissible action state complexity. When $\mathcal{A}^b = \{0,1\}^N$, the frequency reuse state for the next frame is deterministically set to $\boldsymbol{s}'^b = \boldsymbol{a}^b$. In the second model where the action space is $\mathcal{A}^b = \{1,\ldots,N\}$, an action $a^b = k$ corresponds to a decision to transmit only on k RBs in the subsequent frame, with the RB positions chosen uniformly at random. Fig. 5.2 illustrates how a probabilistic action would be taken when the total number of resources is 5 and the action action $a^b = 4$.

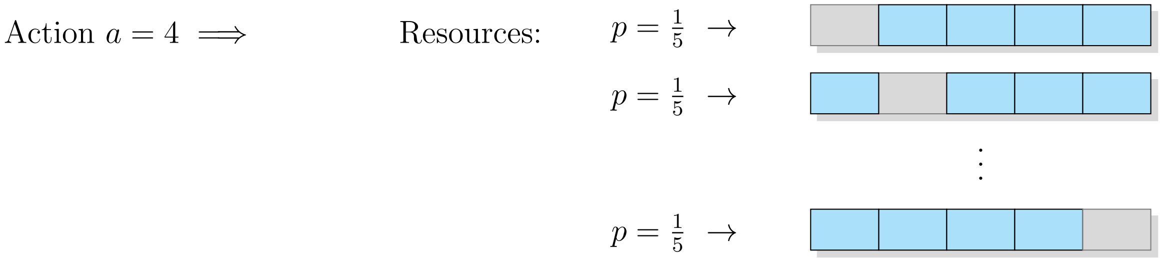

Action $a = 4 \implies$
Resources:
$p = \frac{1}{5} \rightarrow$
$p = \frac{1}{5} \rightarrow$
$\vdots$
$p = \frac{1}{5} \rightarrow$

Figure 5.2: Illustration of the probabilistic action space in the case of $\mathcal{A}^b = \{1,\ldots,5\}$.

We design the per slot reward for each BS b to capture both the amount of data transmitted and the "efficiency" of such transmissions. In particular, given a frequency reuse state $\boldsymbol{s}$, and the scheduling decisions associated with channel gains $\boldsymbol{g}$ and queue lengths $\boldsymbol{q}$, the reward at BS b is modeled by,

$$r^b(\boldsymbol{s},\boldsymbol{g},\boldsymbol{q}) = \sum_{u \in \mathcal{U}^b} \frac{c_u(\boldsymbol{s},\boldsymbol{g},\boldsymbol{q})}{\kappa_u(\boldsymbol{g})}, \tag{5.10}$$

where $c_u(\cdot)$, defined earlier in (5.6) is the overall bits delivered to user u and κ_u defined in (5.8) is the *effective interference free capacity* of user u. This rewards the transmission of data to users at the BS, but penalizes transmissions experiencing excessive interference. Note that each agent in the Markov Game only sees its own frequency reuse state $\boldsymbol{s}^b$, whence it sees a reward $r^b((\boldsymbol{s}^b, \boldsymbol{s}^{-b}), \boldsymbol{g}, \boldsymbol{q})$ that depends on the frequency reuse actions of other players denoted $\boldsymbol{s}^{-b}$, the stationary distribution of the networks channel gains $\boldsymbol{G}(t)$ and possibly not stationary distribution of the network queues $\boldsymbol{Q}(t)$.

We consider a non cooperative Markov game where each BS learns a policy based on rewards either generated by its own users or all users in the network. The learned frequency reuse policy $\boldsymbol{\pi} \triangleq (\pi^b,\ b \in \mathcal{B})$ induces a set of transition probabilities on the frequency reuse states $(\mathcal{S}^b,\ b \in \mathcal{B})$ such that the expected long term rewards are maximized. We consider three different game settings based on the rewards and/or action space.

(G1) **Global reward game**: Each BS trains on the sum reward $\sum_{b\in\mathcal{B}} r^b(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q})$ generated by all BSs. Each BS b has an action space $\mathcal{A}^b = \{0, 1\}^N$.

(G2) **Local reward game**: Each BS trains on its own *local* reward $r^b(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q})$. Each BS has an action space $\mathcal{A}^b = \{0, 1\}^N$.

(G3) **Random action game**: Each BS trains on its own *local* reward $r^b(\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{q})$. Each BS has an action space $\mathcal{A}^b = \{1, \ldots, N\}$.

We can thus model the frequency reuse state transitions as a Markov chain induced on the frequency reuse state space by policy $\boldsymbol{\pi}$ and scheduling rule $\boldsymbol{h}$. With a slight abuse of notation, we use $(\pi^b(\boldsymbol{s}) : \boldsymbol{s} \in \mathcal{S}^b)$ to also denote the steady state distribution of the induced Markov chain at BS b. Note that the frequency reuse policy $\boldsymbol{\pi}$ in conjunction with a scheduling rule $\boldsymbol{h}$ determines the users' queue length distributions.

5.4 Probabilistic Frequency Reuse (PFR)

Given a traffic load $\boldsymbol{\lambda}$, one would like to pick a set of frequency reuse policies $\boldsymbol{\pi}$ from the set of all feasible policies $\mathcal{P}$ for interference mitigation and scheduler $\boldsymbol{h} \in \mathcal{H}$ that can either stabilize the user queues or maximize some network utility. Consider the B-player Markov game summarized in (5.9), we fix the scheduler $\boldsymbol{h}$ at each BS and learn interference management policies $\boldsymbol{\pi}$ using one of three game settings (G1), (G2) or (G3) based on our carefully chosen proxy metric (5.10).

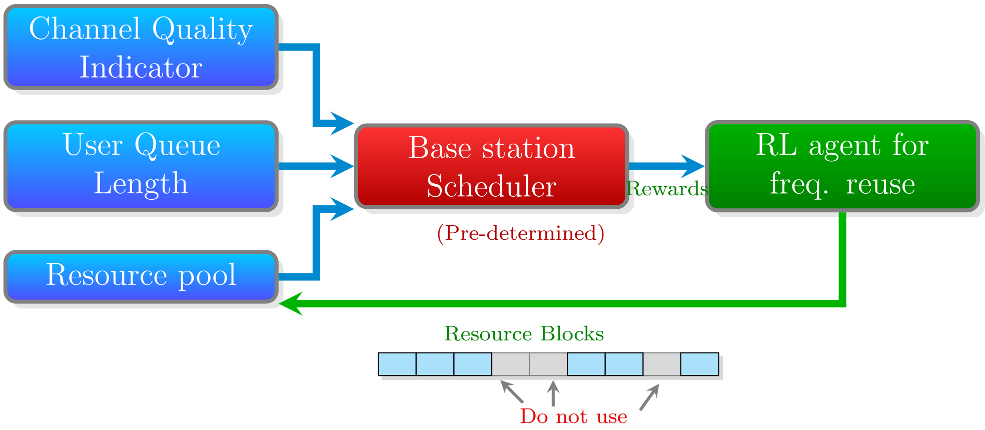


Figure 5.3: Block diagram representation of our proposed system architecture at base station b.

We propose that each BS use an efficient algorithm to learn it's own frequency reuse policy in a distributed manner. Multi agent Q learning [51] is a model free learning algorithm for non cooperative Markov games. A schematic representation of our learning algorithm is depicted in Fig. 5.3. An agent at each BS b learns* its frequency reuse policy π^b using the rewards generated by the underlying BS scheduler. The scheduler then allocates RBs made available by the frequency reuse policy to users based on their channel quality and queue length. Finally the learning agent at each

*After a random initialization of the reuse policy, the rewards generated by the scheduler is used to iteratively improve the policy.

BS trains on the reward metric for each resource selection configuration $\boldsymbol{s}^b$ based on which the reuse policy is updated to maximize discounted future rewards. We will refer to this distributed method of learning as Probabilistic frequency reuse where each BS learns the fraction of time it should spend on a particular frequency reuse state to alleviate interference.

Let $\boldsymbol{\pi}$ be the multi agent frequency reuse policy learnt by the distributed algorithm, then for a given initial state $\boldsymbol{s}^b$, the learning agent at each BS maximizes it's value function $v^{\pi^b}(\cdot)$

$$v^{\pi^b}(\boldsymbol{s}^b) = \sum_{k=1}^{\infty} \gamma^k \mathbb{E}^{\boldsymbol{\pi},\chi} \left[r^b(\mathbf{S}^b_{t+k}, \mathbf{G}_{t+k}, \mathbf{Q}_{t+k}) | \boldsymbol{\pi}, \mathbf{S}^b_t = \boldsymbol{s}^b \right]. \tag{5.11}$$

Note $r^b(\cdot)$ is the reward of BS b at time t and k is time step to capture future rewards. We start training with $\mathbf{Q}_0 = \mathbf{0}$ and after sufficiently long training time, the frequency reuse policy converges to a stationary distribution π^b which induces a distribution χ on the queue length. Also, each BS has access to the reward of other BSs either directly (as in global reward setting) or indirectly through the interference that each BS sees. Furthermore, the actions of each BS in the network are either indirectly observable at each BS through interference or irrelevant if there is no interference. We will use $v^{\boldsymbol{\pi}}(\boldsymbol{s})$ to denote the sum of the value functions of all BSs under the frequency reuse policy $\boldsymbol{\pi}$.

Definition 5.4.1. *In a stochastic game a Nash equilibrium (NE) is a set of policies* $\boldsymbol{\pi}^* = (\pi^{*1}, \ldots, \pi^{*b}, \ldots, \pi^{*B})$ *such that for all* $\boldsymbol{s} \in \mathbf{S}$, $\forall \pi^b \in \mathcal{P}^b$ *(*$\mathcal{P}^b$ *is the set of all feasible frequency reuse policies for BS b) and* $b = 1, \ldots, B$,

$$v^{\boldsymbol{\pi}^*}(\boldsymbol{s}) \geq v^{\boldsymbol{\pi}'}(\boldsymbol{s}), \;\; \textit{where } \boldsymbol{\pi}' = (\pi^{*1}, \ldots, \pi^b, \ldots, \pi^{*B}). \tag{5.12}$$

We will next establish the existence of and convergence to a Nash equilibrium for our non cooperative Markov game among the BSs.

5.4.1 Existence of and convergence to a Nash equilibrium for our proposed non cooperative Markov game

Theorem 5.4.1. *Consider a non cooperative Markov game where each BS in the network is autonomously learning a frequency reuse policy to mitigate interference. There exists a Nash equilibrium, possibly not unique, for the game under all three reward modes (G1), (G2) and (G3), with each BS's agent converging to an NE frequency reuse policy.*

Proof. See Appendix. D.1. ■

5.5 Main Results

We shall first introduce a few definitions needed to present our main theoretical results. Next we define a notion of the *capacity region* of our proposed Probabilistic Frequency Reuse (PFR) for the non cooperative Markov game. By *capacity region* we refer to the set of all user arrival rate vectors that the network is able to support with stable queues. Next we establish a capacity order among the three game settings (G1), (G2) and (G3) according to their learnt value functions.

5.5.1 Network stability under interference mitigation polices

We begin by defining a notion of capacity for the network given a frequency reuse policy $\boldsymbol{\pi} = (\pi^b, b \in \mathcal{B})$ which characterizes the set of possible long term downlink transmission rates which are achievable under two assumptions (a) all users' transmit queues are backlogged, and (b) all BSs make use of the all resources made available by their respective frequency reuse policies, and thus offer the worst case interference according to their frequency reuse policy.

Recall that $\boldsymbol{\pi}(\mathbf{s})$ denotes the probability that the frequency reuse policy across all the BSs is in state $\boldsymbol{s}$. Further, given the frequency reuse and channel states of the network $\mathbf{s}, \mathbf{g}$ and $s_k^b = 1$ (resource k is available at base station b), $\phi_{uk}^b(\boldsymbol{s}, \boldsymbol{g})$ denotes the fraction of time resource k is allocated to user $u \in \mathcal{U}^b$, and thus under the assumption (b) in the paragraph above, we have that $\sum_{u \in \mathcal{U}^b} \phi_{uk}^b(\boldsymbol{s}, \boldsymbol{g}) = 1$. This corresponds to a static splitting resource allocation policy across the network when the network is in state $\boldsymbol{s} \in \mathbf{S}, \boldsymbol{g} \in \mathbf{G}$. Let $\mathcal{F}$ denote the set of such feasible splittings for all possible network states,

$$\mathcal{F} = \left\{ \boldsymbol{\phi} : \forall b \in \mathcal{B}, \forall \boldsymbol{s}, \boldsymbol{g}, \text{if } s_k^b = 1 \text{ then} \sum_{u \in \mathcal{U}^b} \phi_{uk}^b(\boldsymbol{s}, \boldsymbol{g}) = 1 \right\}. \tag{5.13}$$

Suppose $\boldsymbol{\phi}$ is a feasible static splitting, then one could come up with a lower bound on the downlink rate under a frequency reuse $\boldsymbol{\pi}$ for each user $u \in \mathcal{U}^b$ given by,

$$\mu_u^\infty(\boldsymbol{\pi}, \boldsymbol{\phi}) = \mathbb{E}^{\boldsymbol{\pi}, \chi} \left[\sum_{k=1}^{N} \phi_{uk}^b(\mathbf{S}, \mathbf{G}) c_{uk}(\mathbf{S}, \mathbf{G}) \right]. \tag{5.14}$$

where $\boldsymbol{\pi}$ and χ correspond to the distributions of the network's frequency reuse and channel states $\mathbf{S}$ and $\mathbf{G}$ respectively. We further let $\mu^\infty(\boldsymbol{\pi}, \boldsymbol{\phi}) = (\mu_u^\infty(\boldsymbol{\pi}, \boldsymbol{\phi}), u \in \mathcal{U})$.

Definition 5.5.1. *Given a frequency reuse policy $\boldsymbol{\pi}$, we define the* ***saturated network capacity*** *region $\mathcal{C}_\pi^\infty$ as follows*

$$\mathcal{C}_{\boldsymbol{\pi}}^\infty \triangleq \left\{ \boldsymbol{r} : \mathbf{0} \preceq \boldsymbol{r} \preceq \boldsymbol{\mu}^\infty(\boldsymbol{\pi}, \boldsymbol{\phi}), \quad \boldsymbol{\phi} \in \mathcal{F} \right\}.$$

Notation: $\mathcal{C}_{\boldsymbol{\pi}}^{\infty,b}$ denotes the *saturated capacity* region for BS b, $(\cdot)^\circ$ denotes *interior* of a set and $\boldsymbol{\lambda}^b$ denotes the user arrival rate vector of the arrival rate for BS b. $\boldsymbol{\lambda}^{-b}$ denotes the arrival rate vector of users at all BSs in the network except BS b.

We further define the *capacity region* for saturated networks under all possible Markovian frequency reuse polices $\mathcal{P}$ as

$$\mathcal{C}^\infty = \bigcup_{\boldsymbol{\pi} \in \mathcal{P}} \mathcal{C}_{\boldsymbol{\pi}}^\infty \quad . \tag{5.15}$$

Remark. It is easy to show that $\mathcal{C}^{\infty}_{\boldsymbol{\pi}}$ is convex owing to the convexity of the set of all possible static splits $\mathcal{F}$. However, $\mathcal{C}^{\infty}$ need not be convex. Indeed one could design $\boldsymbol{\pi}_1$, $(\boldsymbol{\pi}_2)$ which allocate resources only to b_1, (b_2), respectively. In this case it is not feasible to achieve a convex combination of $\mathcal{C}_{\boldsymbol{\pi}_1}$ and $\mathcal{C}_{\boldsymbol{\pi}_2}$ as that calls for a scheduler aware frequency reuse policy which goes against our separation of concerns paradigm.

Lemma 5.5.1. *Consider a network where the users' queues across the BSs have iid arrivals with mean $\boldsymbol{\lambda}$ such that there exists a π and $\boldsymbol{\phi}$ such that $\boldsymbol{\lambda} < \boldsymbol{\mu}^{\infty}(\boldsymbol{\pi}, \boldsymbol{\phi}) \in \mathcal{C}^{\infty}$, then the network is stable under the reuse policy $\boldsymbol{\pi}$ with static splitting rule $\boldsymbol{\phi}$.*

Proof. See Appendix. D.2. ■

Lemma 5.5.2. *For a given frequency reuse policy $\boldsymbol{\pi}$, if the arrival rate $\boldsymbol{\lambda}^b$ at BS b is such that $\boldsymbol{\lambda}^b \in (\mathcal{C}^{\infty,b}_{\boldsymbol{\pi}})^{\circ}$, then assuming base station b employs max weight scheduling while all other BSs are saturated, the system is stable.*

Proof. See Appendix. D.3. ■

Intuitively from a BS's perspective, saturation of neighboring BSs' users' queues corresponds to a worst case in terms of interference and thus the capacity that can be achieved for its users. One would expect if one relaxes the saturation assumption that one would still achieve stability. The following result shows that this is the case for a network where the arrivals at all BS/users lie in the interior of $\mathcal{C}^{\infty}$.

Theorem 5.5.1. *For a given frequency reuse policy $\boldsymbol{\pi}$, if the arrivals to the user queues in the network satisfy $\boldsymbol{\lambda} \in (\mathcal{C}^{\infty}_{\boldsymbol{\pi}})^{\circ}$, then max weight schedulers at each BS will stabilize it's users' queues.*

Proof. See Appendix. D.4. ■

Remark. Local reward based (as in (G2), (G3)) distributed learning of frequency reuse policy $\boldsymbol{\pi}$ is *effective* only when the metric represents the relative performance of each user with reference to the no interference scenario.

Theorem 5.5.2. *For a given scheduling policy $\boldsymbol{h}$, let $\boldsymbol{\pi}_g^*$, $\boldsymbol{\pi}_l^*$ denote possibly not unique frequency reuse policies which are Nash Equilibria learnt using the global (G1) and local (G2) reward games, respectively. For every local frequency reuse policy $\boldsymbol{\pi}_l^*$ there exists a global frequency reuse policy $\boldsymbol{\pi}_g^*$ such that $v^{\boldsymbol{\pi}_g^*}(\boldsymbol{s}) \geq v^{\boldsymbol{\pi}_l^*}(\boldsymbol{s})$.*

Proof. See Appendix. D.5. ■

5.5.2 Higher capacity region when compared to FR1

We start by defining what we call a network with BSs *tightly coupled* in the sense of interference caused by downlink transmissions. Then we state and prove our main theoretical results. The next step will be to solve for the NP hard combinatorial optimization problem in (5.24).

Definition 5.5.2. *We say that the BSs in a network are* ***tightly coupled*** *for some scheduling policy $\boldsymbol{h}$ if there exists a non empty subset of BSs $A \subset \mathcal{B}$ such that,*

$$E^{\gamma}\left[\sum_{b\in A^c}\sum_{u\in\mathcal{U}_b} c_{uk}(\mathbf{S},\mathbf{G},\mathbf{Q})\right] > E^{\gamma}\left[\sum_{b'\in A}\sum_{\tilde{u}\in\mathcal{U}_{b'}} c_{\tilde{u}k}(\mathbf{S},\mathbf{G},\mathbf{Q})\right]. \tag{5.16}$$

For instance, a network is called tightly coupled, if it is possible to turn off a single BS b on resource k and have the sum marginal rate increase of all the other BSs on resource k exceed the loss in rate for BS b on resource k.

Theorem 5.5.3. *Let $\mathcal{C}_1$ be the saturated network capacity region for the frequency reuse 1 policy denote by $\boldsymbol{\pi}'$, which has all its mass only on the all ones state, i.e., $\pi'^{b^i}(\boldsymbol{s}) = 1\ \ \forall i,$ if $\boldsymbol{s} = \{1\}^N$ and 0 otherwise. Fix $\boldsymbol{h}$ to be the Max-Weight Scheduler.*

When the BSs are ***tightly coupled*** *as in* ***Definition 5.5.2****, for any* $\boldsymbol{\lambda} < \boldsymbol{\mu} \in \mathcal{C}_1$*, we can find a policy* $\boldsymbol{\pi}$ *such that* $\alpha\boldsymbol{\lambda} \in \mathcal{C}$*, where* $\alpha > 1$*, i.e., we can find a frequency reuse policy that stabilizes the system for a higher arrival rate vector.*

Proof. See Appendix. D.6 ■

Corollary 5.5.1. *Every RL agent converges to a frequency reuse policy* π^b *based on the rewards generated by the proxy metric such that the arrival rate vector of* $\mathcal{C}_{\pi^b}$ *strictly dominates* $\mathcal{C}_{\pi'}$*, where* π' *is the frequency reuse 1 policy that has all its mass only on the all ones state ,i.e.,* $\pi'(\boldsymbol{s}) = 1,$ *iff* $\boldsymbol{s} = \{1\}^N$*.*

Proof. See Appendix. D.7. ■

5.6 Parallel queues with coupled service rates

We have seen how neighboring BSs can learn a frequency reuse policy by playing a Markovian game, but it's unclear if such a policy exists. Is there a feasible frequency reuse policy with a larger capacity region and if so, what scheduler should be used at each of the BSs? In this section, we demonstrate how coupled queues with coordination (appropriate frequency reuse policy) can help improve overall system throughput when the resource scheduler at each BS is carefully chosen. In general, the interference from the three strongest interference causing BSs is sufficient to capture the impact on the SINR of the user, as shown in [4, Section 9.2]. However, for simplicity of analysis, we will only consider the interference from the strongest interference causing BS. This leads to a system where the service rates of two users are coupled through interference.

Consider two parallel queues with coupling in service rates such that, if both queues are busy servicing packets, then the service rate is assumed to be r_I for each of the queues. However, when one of the queues is idle, the other queue gets to service at rate r_0, where $r_0 > r_I$. We will construct the stability region for two queues with coupled service rates under interference mitigation, with and without the presence of *coordination* between the two queues. Specifically, when a queue backs off from servicing its queue for a fraction of time, irrespective of the backlog in it's queue, there is an improvement in the sum total service rate at both queues.

Let π_{δ_i} be the policy where Q_i backs off from transmission for a fraction of time $\delta_i \in (0, 1)$, irrespective of the backlog in its queue. Define π' as the baseline policy without interference mitigation, i.e., a queue is busy servicing whenever there are packets to send. Moreover, by *coordination* we refer to the minimal information exchange between the queues such that they do not *choose*[†] to go idle simultaneously. Note that the global reward Markov game involves coordination and the local reward and random action game settings are uncoordinated.

5.6.1 Queues Q_1 and Q_2 with infinitely backlogged buffers

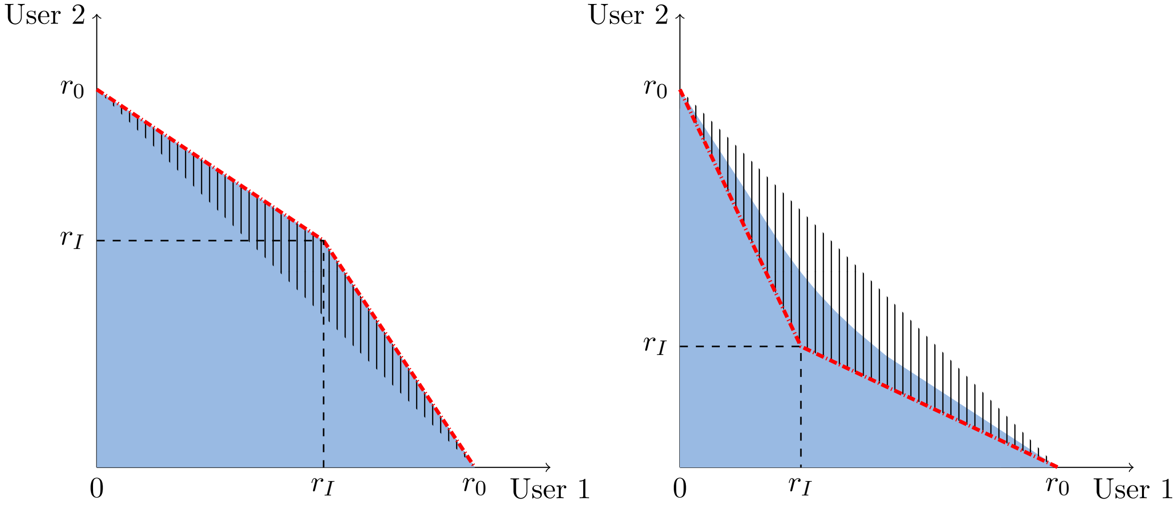
- **Case 1: No back off**

  Under the no backing off policy π', the sum service rate at both queues cannot exceed $2r_I$. The corresponding stability region is given by the dashed rectangular region in Fig. 5.4.

- **Case 2: Q_1 or Q_2 backs off**

  Suppose we have π_{δ_1}, where Q_1 backs off by a fraction δ_1, then maximum rate at which Q_1 can service the packets in its queue is given by, $(1 - \delta_1)r_I$. For Q_2, the maximum rate at which packets can be serviced is $\delta_1 r_0 + (1 - \delta_1)r_I$.

[†] Since we want to design an efficient frequency reuse policy, it is desirable to avoid having both queues go idle simultaneously



(a) $r_0 < 2r_I$

(b) $r_0 > 2r_I$

Figure 5.4: Stability region for parallel queues with coupled service rates. The shaded regions correspond to the stability region for uncoordinated back off, and striped region corresponds to the stability region for coordinated back off with $\delta_1 + \delta_2 \leq 1$.

Adopting the frequency reuse policy π_{δ_1} as compared to π' in terms of total service rate will be beneficial only when,

$$\begin{aligned} (1-\delta_1)r_I + (1-\delta_1)r_I + \delta_1 r_0 &> 2r_I \\ \delta r_0 &> 2\delta_1 r_I \qquad (5.17) \\ r_0 &> 2r_I \end{aligned}$$

Similarly, one can show that under π_{δ_2} the maximum service rates at Q_1 & Q_2 are $\delta_2 r_0 + (1-\delta_2)r_I$ and $(1-\delta_2)r_I$, respectively. If $r_0 > 2r_I$, then adopting π_{δ_2} is beneficial and the corresponding stability region is given by the dash dotted red line in Fig. 5.4(b).

- **Case 3**: $\boldsymbol{Q_1}$ **&** $\boldsymbol{Q_2}$ **coordinated back off**

  Consider the case $\pi_{\delta_1} \times \pi_{\delta_2}$, where Q_1 & Q_2 back off by a fraction of time δ_1 & δ_2, respectively, such that the time over which they back off does not overlap.

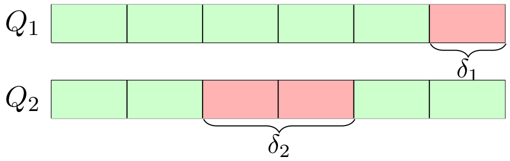


Figure 5.5: Parallel queues with coupled service rates and coordinated back off in a slotted time system.

Also, with infinitely back logged queues it is not beneficial for either of the queues if $\delta_1 + \delta_2 > 1$. From the joint distribution shown in Fig. 5.5, it is clear that the maximum service rate at Q_1 & Q_2 is $(1-\delta_1-\delta_2)r_I + \delta_2 r_0$ and $(1-\delta_1-\delta_2)r_I + \delta_1 r_0$, respectively.

	Q_1 ON	Q_1 OFF
Q_2 ON	$1-\delta_1-\delta_2$	δ_1
Q_2 OFF	δ_2	0

Table 5.1: Joint state distribution of saturated parallel queues with coupled service rates and **coordinated** back off.

Adopting the frequency reuse policy $\pi_{\delta_1} \times \pi_{\delta_2}$ as compared to π' in terms of total service rate will be beneficial only when,

$$\begin{aligned} 2(1-\delta_1-\delta_2)r_I + (\delta_1+\delta_2)r_0 &> 2r_I \\ (\delta_1+\delta_2)r_0 &> 2(\delta_1+\delta_2)r_I \qquad (5.18) \\ r_0 &> 2r_I \end{aligned}$$

The stability region for this case is represented by the area covered by the vertical lines in Fig. 5.4(b).

- **Case 4**: $\boldsymbol{Q_1}$ **&** $\boldsymbol{Q_2}$ **random uncoordinated back off**

  Consider the case $\pi_{\delta_1} \cap \pi_{\delta_2}$, where Q_1 & Q_2 back off by a random fraction of time δ_1 & δ_2, respectively. From the joint distribution in Table 5.2, it is clear that the maximum service rates at Q_1 and Q_2 are given by $(1-\delta_1)\left[\delta_2 r_0 + (1-\delta_2) r_I\right]$ and $(1-\delta_2)\left[\delta_1 r_0 + (1-\delta_1) r_I\right]$, respectively. The stability region is given by the blue shaded region in Fig. 5.4(b).

	Q_1 ON	Q_1 OFF
Q_2 ON	$(1-\delta_1)(1-\delta_2)$	$\delta_1(1-\delta_2)$
Q_2 OFF	$(1-\delta_1)\delta_2$	$\delta_1\delta_2$

Table 5.2: Joint distribution of the states of saturated parallel queues with coupled service rates and uncoordinated back off.

5.6.2 Unsaturated queues

If the arrival rates at each of the queues, $\lambda_1, \lambda_2 \in (0, r_I)$ then both the queues are stable, because they can be treated as two independent queues with constant service rates r_I. Define $\rho_i = \frac{\lambda_i}{\mu_i}$, which denotes the utilization factor of Q_i. We would like to find the arrival rates at each of the queues for which the system can be stabilized.

Suppose one of the queues, say Q_2 is busy such that the utilization rate $\rho_2 \to 1$, then the max service rate at Q_1 can at most be r_I. For an arrival rate λ_1 at Q_1, $\rho_1 = \frac{\lambda_1}{r_I}$ let the arrival rate at Q_2 be expressed as $\lambda_2 = r_I + \epsilon$. The system is stable if

$$\begin{aligned}\lambda_2 &< \rho_1 r_I + (1-\rho_1) r_0 \\ &= \lambda_1 + \left(\frac{r_I - \lambda_1}{r_I}\right) r_0 \\ &= r_0 + \left(\frac{r_0}{r_I} - 1\right)\lambda_1\end{aligned} \tag{5.19}$$

Therefore as long as $\epsilon < (r_0 - r_I) + \left(\frac{r_0}{r_I} - 1\right)\lambda_1$, the system is stable.

5.6.3 Impact of frequency reuse policy π_δ on stability region

Consider the system when there is no interference mitigation employed, i.e., policy π' where all BSs transmit packets whenever the queues are non-empty. Let Q_1 be busy such that $\rho'_1 \to 1$ and $\lambda'_2 < r_I$, then $\mu'_1 = (1-\rho'_2)r_0 + \rho'_2 r_I$ and $\mu_2 \to r_I$.

Suppose Q_1 has a saturated queue, if Q_1 backs off by $\delta > 0$, then the service rate at Q_2 is $\mu_2 = (1-\delta)r_I + \delta r_0 > \mu'_2 \implies \rho'_2 < \rho$. The service rate at Q_1 can be found using the expression $\mu_1 = (1-\delta_1)\left[(1-\rho_2)r_0 + \rho_2 r_I\right]$. In order to construct π_{δ_1} such that we can support a higher arrival rate for both queues, we need to pick δ_1 such that

$$\begin{aligned} \mu_1 = (1-\delta_1)\left[(1-\rho_2)r_0 + \rho_2 r_I\right] &> (1-\rho'_2)r_0 + \rho'_2 r_I \\ 1-\delta_1 &> \frac{r_0 - \rho'_2(r_0 - r_I)}{r_0 - \rho_2(r_0 - r_I)} \\ \delta_1 &< 1 - \frac{r_0 - \rho'_2(r_0 - r_I)}{r_0 - \rho_2(r_0 - r_I)} \\ &= \frac{(\rho'_2 - \rho_2)(r_0 - r_I)}{r_0 - \rho_2(r_0 - r_I)} \end{aligned} \tag{5.20}$$

Suppose Q_1 has an unsaturated queue, then we have $\rho'_1 = \frac{\lambda'_1}{\mu'_1} < 1$, and hence the maximum service rate at Q_2 is given by $(1-\rho'_1)r_0 + \rho'_1 r_I$. Assuming that Q_1 and Q_2 coordinate to not be idle on the same slot, the system of queues are stable when,

$$\begin{aligned} \lambda'_1 &< (1-\rho'_2)r_0 + \rho'_2 r_I \\ \lambda'_2 &< (1-\rho'_1)r_0 + \rho'_1 r_I \end{aligned} \tag{5.21}$$

Suppose Q_1 backs off by a fraction δ_1 (coordinated back off happens only when Q_2 is active), the maximum service rate at Q_2 is $\delta_1 r_0 + (1-\delta_1)\left[(1-\rho_1)r_0 + \rho_1 r_I\right]$, which can also be expressed as $\left[\delta_1 + (1-\delta_1)(1-\rho_1)\right] r_0 + (1-\delta_1)\rho_1 r_I$. Since Q_1 backs off by a fraction δ_1, we can expect that $\rho'_1 < \rho_1$. We want $\mu_2 > \mu'_2$, for which we need to choose δ_1 such that

$$(1-\rho'_1) \leq \delta_1 + (1-\delta_1)(1-\rho_1) \qquad \text{and} \quad \rho'_1 < (1-\delta_1)\rho_1 \tag{5.22}$$

We simply pick $\delta_1 = \dfrac{\rho_1 - \rho_1'}{\rho_1}$, using which the maximum service rate at Q_2 is given by, $(1-\delta_1)\left[\rho_2 r_I + (1-\rho_2)r_0\right]$. Since we want π_{δ_1} to stabilize higher arrival rates at the queues,

$$\begin{aligned}(1-\delta_1)\left[\rho_2 r_I + (1-\rho_2)r_0\right] &> \rho_2' r_I + (1-\rho_2')r_0 \\ \delta_1 &< 1 - \frac{\rho_2' r_I + (1-\rho_2')r_0}{\rho_2 r_I + (1-\rho_2)r_0} \\ &= \frac{(\rho_2' - \rho_2)(r_0 - r_I)}{\rho_2 r_I + (1-\rho_2)r_0}\end{aligned} \tag{5.23}$$

Problem Formulation

The first step to solve for $\boldsymbol{\pi}, \boldsymbol{h}$ is to select an appropriate metric function $f(\cdot)$, for instance, sum user rate $f(\boldsymbol{\mu}) = \mathbf{1}^T\boldsymbol{\mu}$, proportional fairness $f(\boldsymbol{\mu}) = \mathbf{1}^T \log(\boldsymbol{\mu})$, weighted user rate $f(\boldsymbol{\mu}) = \mathbf{w}^T\boldsymbol{\mu}$ or the transmission delay associated with each user packet.

$$\boldsymbol{\pi}, \boldsymbol{h} = \underset{\boldsymbol{\pi}\in\mathcal{P}, H\in\mathcal{H}}{\arg\max} \;\; f(\boldsymbol{\mu}) \;\; s.t. \;\; \boldsymbol{\lambda} < \boldsymbol{\mu}\,. \tag{5.24}$$

Suppose Q_1 and Q_2 do not coordinate, then they could be idle during the same time slot with probability $(1-\rho_1)(1-\rho_2)$, for a detailed joint distribution check Table 5.3. The maximum transmission rate for a user in Q_1 and Q_2 will then be $\rho_1(\rho_2 r_I + (1-\rho_2)r_0)$ and $\rho_2(\rho_1 r_I + (1-\rho_1)r_0)$, respectively. Analyzing this case just like the coordinated case results in the corresponding bounds for a π that aligns the idle state of one queue with the active state of the other.

	Q_1 ON	Q_1 OFF
Q_2 ON	$\rho_1\rho_2$	$\rho_2(1-\rho)$
Q_2 OFF	$(1-\rho_2)\rho_1$	$(1-\rho_1)(1-\rho_2)$

Table 5.3: Joint state distribution of unsaturated parallel queues with coupled service rates and coordinated back off.

Definition 5.6.1. *Let $\mathcal{F}$, as in* (5.13)*, denote the set of feasible splittings for all possible network states. The mean downlink rate under policy $\boldsymbol{\pi} \in \mathcal{P}$ and a feasible static splitting rule $\boldsymbol{\phi}$ for user $u \in \mathcal{U}^b$ is given by,*

$$\mu_u(\boldsymbol{\pi}, \boldsymbol{\phi}) = \mathbb{E}^{\boldsymbol{\pi},\gamma,\nu}\left[\sum_{k=1}^{N} \phi_{uk}^b(\mathbf{S}, \mathbf{G}, \mathbf{Q}) c_{uk}(\mathbf{S}, \mathbf{G}, \mathbf{Q})\right],$$

where $\boldsymbol{\pi}$, γ correspond to the distributions of the network's frequency reuse, channel strength $\mathbf{S}, \mathbf{G}$, respectively and ν is the policy induced distribution on the queue sizes $\mathbf{Q}$. We define the network capacity *region $\mathcal{C}_\pi$ given the frequency reuse policy π as follows*

$$\mathcal{C}_{\boldsymbol{\pi}} \triangleq \{\mathbf{r} : \mathbf{0} \preceq \mathbf{r} \preceq \boldsymbol{\mu}(\boldsymbol{\pi}, \boldsymbol{\phi}), \quad \boldsymbol{\phi} \in \mathcal{F}\}.$$

Theorem 5.6.1. *For a given frequency reuse policy $\boldsymbol{\pi} \in \mathcal{P}$, if the arrivals to user queues in the network can be stabilized i.e., $\boldsymbol{\lambda} \in (\mathcal{C}_{\boldsymbol{\pi}})^\circ$, then max-Weight Scheduler at BS b will stabilize it's user queues.*

Proof. See Appendix. D.8. ■

To summarize, a coordinated back off from resource scheduling across coupled queues results in a larger sum capacity region for both saturated and unsaturated queues. Furthermore, given any feasible frequency reuse policy, Max-Weight scheduler is *throughput optimal*, i.e., it will stabilize queues if there exists any scheduler that can stabilize queues.

5.7 Simulations

A simple network as in Fig. 5.6 with four BSs is considered. Each BS serves 5 users and has access to 5 resources (3.5-3.55 GHz with RB bandwidth 10 MHz). Users are dropped uniformly at random over each square region and associated to the nearest BS. Other network parameters are as listed in the Table 5.4. For all three game settings (G1), (G2) and (G3) a DQN is employed at each BS to learn its frequency reuse policy π^b based on the rewards generated. The DQN learns a Q table [100] which keeps track of the mean discounted reward as a function of the (*state*, *action*) pair of each BS. The *state* of every BS agent is given by the binary vector corresponding to the frequency reuse state. The *action* corresponds to the resources selected by each BS for transmission for the next time slot, with 1(0) to denote whether a resource can be (cannot be) assigned to a user. During the training phase each DQN performs exploration and exploitation to learn a policy π^b that maximizes the long term discounted rewards. The Keras Adam optimizer [1] was used to implement the DQN with the following hyperparameters: exploration rate $\epsilon \in [1, 0.01]$, exploration decay factor 0.995, a learning rate of 0.001 and a reward discount factor $\gamma = 0.95$. Algorithm 11 gives the DQN training logic used at each of the BSs. Open AI gym [69] was used to simulate the wireless network environment for multi-agent RL.

Table 5.4: Simulation Parameters

Parameter	Value
Base station tx power	2 W
User tx power	100 mW
Carrier frequency	3.5 - 3.55 GHz
Resource Bandwidth	10 MHz
Bounded path loss with $\alpha = 3$	$\max\{1, r^{-\alpha}\}$
Noise power	-104 dB

The SINR of a user scheduled to transmit, is calculated using (5.2). The transmit power of the BS (user) is 2 W (100 mW) with noise power set to -104 dB. $\hat{c}_u$ is the estimated data rate for user u per RB, based on channel quality (fading

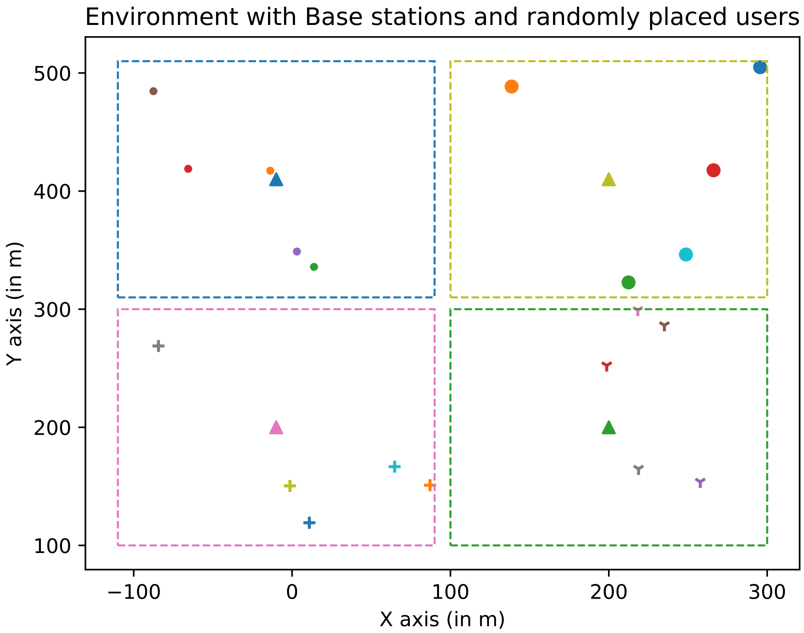


Figure 5.6: The network configuration used for simulations with four BSs (triangles), each associated with 5 users (dots).

and path loss). We use bounded standard path loss $\max[1, r^{-\alpha}]$ with exponent $\alpha = 3$ to model channel gains. Note that we use $|h_i|^2 = 1$ for the simulations, since we assume that the Channel State Information (CSI) is available at the transmitter. This is a reasonable assumption given that the BSs can evaluate the CSI for each of its associated users and hence the channel fading $|h|^2$ is not an unknown quantity under flat fading with CSI.

Algorithm 5: Training DQN of each RL agent at each base station

```
initialize policy $\pi^b$;
while training do
    generate random arrivals;
    schedule resources using policy $\pi^b$;
    save training data (state, action, reward, next state) to batch;
    if batch memory full, train DQN using batch data ;
    update policy $\pi^b$, choose action with max $Q$-value ;
    if iterations > max Iterations then
        done = True;
    end
end
```

At each BS the DQN provides a list of available resources to the scheduler that can be allocated to users. Each BS uses a Max weight scheduler (predetermined scheduler $\boldsymbol{h}$) to determine the users to be assigned available RBs. The weight for each user is calculated based on both the current user queue size N_u and the estimated downlink rate $\hat{c}_u$ as, $w_u = N_u \hat{c}_u$. A BS assigns RBs iteratively to its users as follows. The user with maximum weight is assigned the best channel available and then user weights are reevaluated based on updated queue size accounting for potential packet transmissions. Specifically, suppose user u was assigned a channel with rate $\hat{c}_u$ for transmission, then the user weight w_u is calculated with an updated queue size $N_u = N_u - f(\hat{c}_u)$, where $f(\cdot)$ is a non decreasing function that denotes the number of packets transmitted as a function of the downlink rate $\hat{c}_u$. We use a piece wise linear function $f(x) = \lfloor \log_2(1 + x) \rfloor$ to determine the number of packets transmitted, as a function of the rate, $\lfloor \log_2(1 + \text{SINR}) \rfloor$, where $\lfloor \cdot \rfloor$ is the floor operator. The minimum SINR threshold below which no packets can be reliably transmitted is set to 0dB.

The training algorithm for the DQN agent at each BS is shown in Algorithm

11. We use batch training for the DQNs, hence, a single iteration in the DQN training is equivalent to multiple time steps in real time resource scheduling. We use max-weight scheduling with frequency reuse 1 as a bench mark to gauge the performance of our multi agent RL framework. In order to find the "best" that one could do with a greedy strategy, we also include plots for an *oracle-aided* centralized benchmark where each BS completely knows the interference that will be seen as resources are allocated in the network. For the *centralized* benchmark, we use full frequency reuse at each BS in the network and a sequential scheduling order, wherein the n^{th} BS uses max weight scheduling of resources but is completely aware of the exact interference caused by the scheduling of the previous $n-1$ BSs.

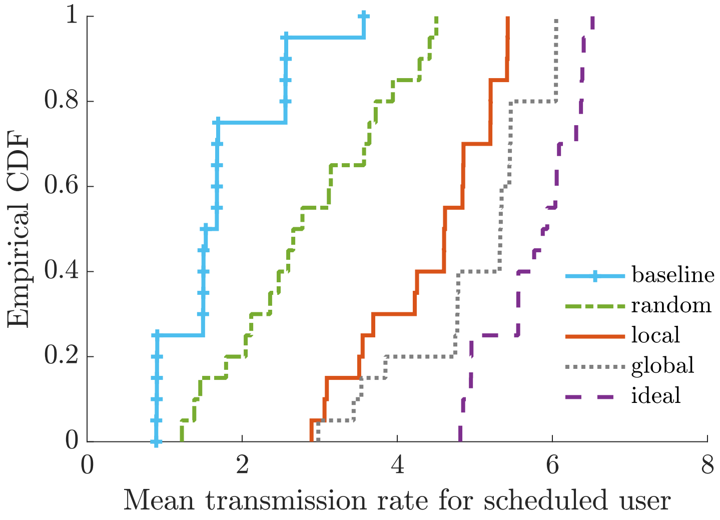


Figure 5.7: User mean rate CDF for all policies.

The empirical CDF of the users' mean rate is shown in Fig. 5.7. It can be seen that even the simplest and most practical random action game (G3) results in better mean rate for users. Specifically, the global reward, local reward and random action games show a 4.8%, 14.8% and 25% improvement in the total rate delivered to all users, when compared to the baseline.

5.7.1 Resource usage across BSs

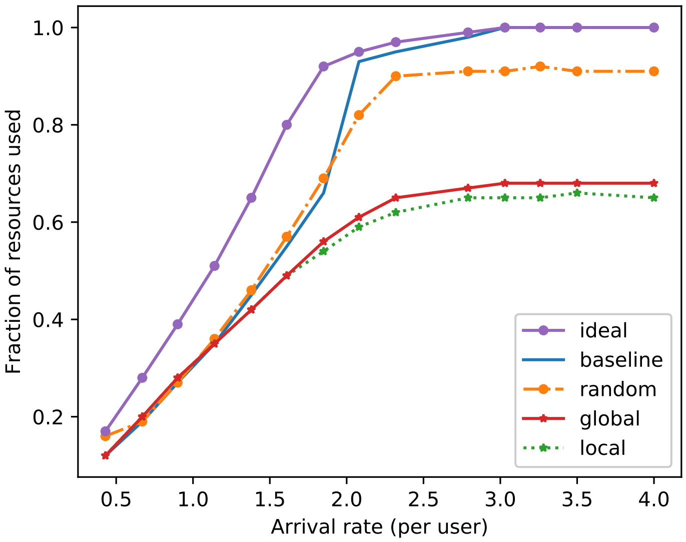


Figure 5.8: Fraction of resources used on average across the network, as a function of the arrival rate per user.

Fig. 5.8 shows the mean fraction of resources selected for downlink transmission in the network as a function of the arrival rate at each of the users' queues. Observe that when compared to the random action game, the global and local reward based policies learn to use a better frequency reuse strategies, which helps them achieve better throughput Fig. 5.7. Also, note that in all three settings considered, initially the fraction of resources allocated to users increases with an increase in the user packet arrival rate. However, beyond a critical level, the fraction of resources used almost saturates to a constant which is not 100%. While the global (G1) and local reward (G2) games demonstrate a 32% and 34% improvement in energy efficiency through better resource utilization (resource positioning), the random action game (G3) still shows a 9% improvement in energy efficiency.

5.7.2 Correlation in learnt policy across BSs

Each frequency reuse policy induces a distribution on the transitions across various frequency reuse states of the underlying markov chain that models the policy.

Definition 5.7.1. *A frequency reuse policy π^b, defines the conditional probability of choosing a frequency reuse state $\boldsymbol{s}' \in \boldsymbol{S}_b$, given that the current frequency reuse state is $\boldsymbol{s} \in \boldsymbol{S}_b$, such that the long term discounted rewards of the network is maximized.*

We illustrate the correlation in frequency reuse patterns across BSs in the network in spite of a distributed learning paradigm. One would expect to see negatively correlated state transitions (intuitively the BSs would not use the same RBs simultaneously), however, Fig. 5.9 shows both positively and negatively correlated states.

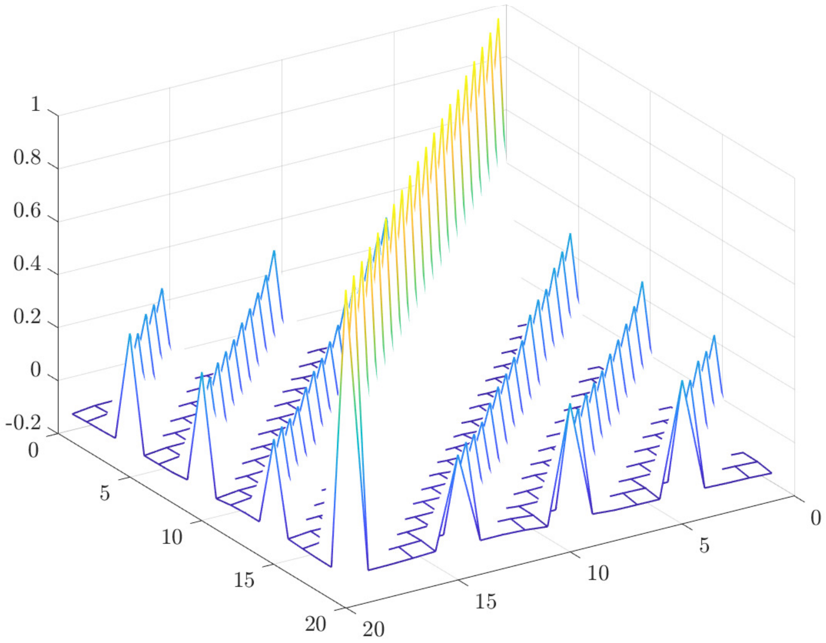


Figure 5.9: Correlation across the frequency reuse states of all BSs under the global reward game.

As the user arrival rates increase, more resources are assigned for downlink transmission at each of the BSs. Note that although each BS uses at least two thirds of its resources at high user arrival rates, a correlation coefficient of at most 0.3 strength indicates that the BSs are trying to learn resource allocation patterns that are complementary. Also, compared to global reward game (G1), the local reward game (G2) (see Fig. 5.10) has weaker correlations across states, leading to relatively poor down link rates. Clearly, the frequency reuse states learned across BSs based on $\boldsymbol{\pi}_g$ are more strongly correlated than those for $\boldsymbol{\pi}_l$. This can be attributed to the fact that the global reward reflects complete information about the network performance as compared to local rewards. Consequently, resource selection patterns learnt using the global reward (G1) better mitigate interference as compared to (G2) or (G3).

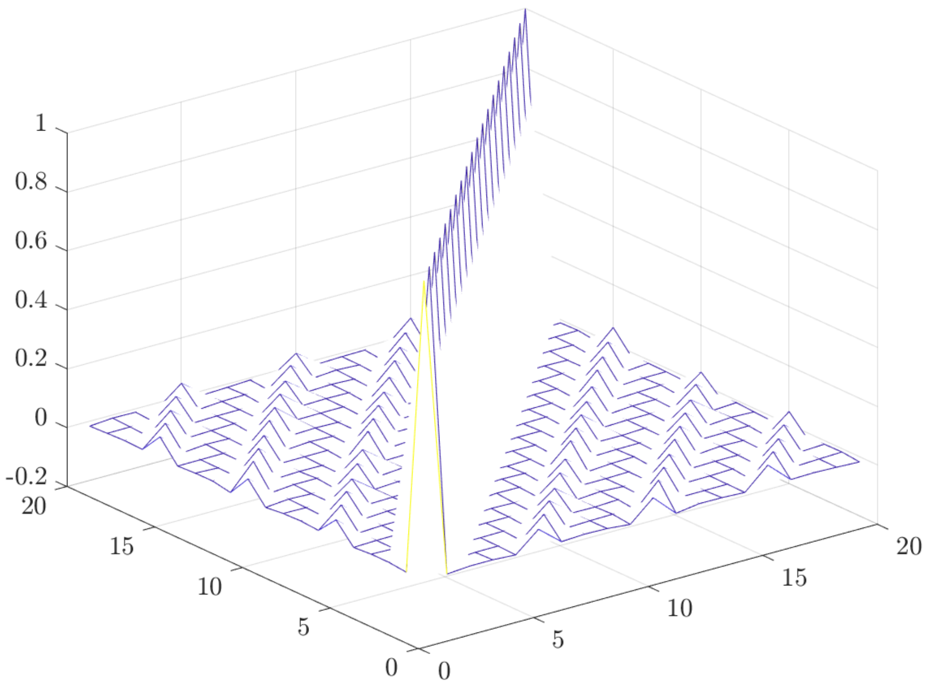


Figure 5.10: Correlation across the frequency reuse states of all BSs under the local reward game.

A further *interesting* result from experimenting with our proposed architecture is as follows. When all the BSs in the network except one, say BS 1, were configured to act greedily, that is employ frequency reuse 1, then using the proxy metric in (5.10) to train the DQN of BS 1 results in the agent to learn **not** to backoff but simply use all its RBs (frequency reuse 1). This behaviour demonstrates that our proposed algorithm is capable of learning the right policy in an adversarial setup. Another experiment to test the proposed architecture involved moving the BSs further away from each other (including their user locations), and it was observed that the DQN agents at each of the BS learn to use frequency reuse 1 as expected. Additional simulation results were not included due to lack of space.

5.8 Chapter Summary

In this chapter, we have proposed two key concepts: First a separation of concerns where one fixes the base station scheduler and optimizes a frequency reuse policy for the given scheduler. Second, the use of a proxy reward metric that accounts for the interference coupling among base stations during the learning process. The key insight is that the proposed learning algorithm is able to learn a frequency resuse policy without any explicit information exchange on the network topology, interference graph or user traffic dynamics. Furthermore, we have demonstrated that the training duration can be substantially reduced by a simplified action space in the *random action game*.

Chapter 6: Conclusion

Our goal in this thesis was to develop wireless scheduling algorithms that can meet QoS constraints of a variety of services for heterogeneous users, e.g., URLLC, eMBB, mMTC. We examined (1) delay constrained opportunistic wireless scheduling, (2) rate constrained opportunistic wireless scheduling, and (3) learning algorithms for distributed wireless scheduling which accounts for the dynamics of interference. We developed measurement-based online scheduling algorithms that can provide spectral efficiency on par with scheduling algorithms that use computationally intensive neural network based channel rate predictions. Our proposed algorithms were still effective for higher packet delay deadlines, whereas we found neural net based schedulers suffer from poor accuracy in this regime. We also proposed and evaluated a new class of *minimum rate* QoS constrained schedulers where relatively large packets are to be delivered to users over user-specific time windows that are possibly not synchronous. Our research also examined the impact of dynamic interference on delay constrained users to develop frequency reuse planning based on decentralized multi-agent reinforcement learning.

6.1 Key takeaways

Some of the key takeaways from this thesis are as follows:

1. It is difficult to tune traditional wireless scheduling policies such as MaxWeight, EXP rule, and Log rule, to meet strict packet deadlines for *heterogeneous* users (channels and delay requirements), moreover, such solutions tend to translate to *spectrally inefficient* resource allocation policies.

2. One can build opportunistic delay constrained wireless schedulers that are based on Neural network predictions of future wireless channels. Such schedulers can be near optimal when delay constraints are tight and predictions are accurate, but, in such scenarios the opportunistic gains are limited. By contrast, when delay constraints are more relaxed, and one would hope to exploit channel variations over longer periods of time, the noise in the neural network's prediction appears to have a deleterious impact on the scheduler's spectral efficiency.

3. Admission control based on directly measuring resource needs of wireless delay constrained schedulers is an effective way to account for heterogeneity in users' channels, traffic, and QoS requirements.

4. In general, when designing opportunistic delay constrained schedulers, there is a trade off between the average spectral efficiency and variability in the per-user resource requirement, which in turn impacts the number of users that can be supported. Given a limited amount of wireless resources, one must decide how to prioritize spectral efficiency versus the number of users that can be supported.

5. Traditional approaches to minimum rate QoS constrained scheduling focus on prioritizing users either based on their QoS deficit or their completion time to satisfy QoS. Our approach provides a balanced trade-off by exploiting good wireless channel quality as long as there is enough slack available to meet the aggregate QoS deficit of all users.

6.2 Future Work

A few interesting directions for future research on wireless scheduling algorithms are listed below.

6.2.1 Intelligent coordination under stochastic interference

Our proposed opportunistic delay constrained schedulers rely on accurate empirical estimation of the wireless channel rate distribution for achieving better performance. However, stochastic interference from neighboring BSs can lead to obsolete or erroneous CQI reports, potentially leading to increased transmission errors. While elastic traffic can recover from such errors through HARQ, this may not be feasible for real time traffic with low latency constraints. A more systematic analysis is required to understand the amount of coordination required across BSs to isolate (if feasible) or protect delay constrained users across the network.

6.2.2 Mitigating URLLC outages due to mobility

We have considered packet level delay constrained scheduling when neighboring base stations produce dynamic interference from stochastic user loads. Users can move from one base station to another, which can cause outages in connectivity and change the spatial distribution of load across the network leading to heterogeneity in interference across BSs. An approach that proactively predicts mobility based drops in signal strength and makes delay constrained transmission decisions anticipating such outages could provide a more seamless user experience. In addition, a future research direction might involve power control of base stations with redistribution of cell edge users efficiently across base stations when the spatial distribution of user load is not homogeneous.

6.2.3 Joint resource optimization across Heterogeneous QoS classes

This thesis addresses the problem of joint resource allocation for URLLC and eMBB users on the downlink with appropriate admission control techniques. However, massive machine type communications on the uplink are characterized by a large number of users/devices relaying short packets of critical/periodic information to the base station. The key metric of importance is the number of devices that can be supported (unlike URLLC or eMBB devices) with little or no knowledge of the channel state information. It would be interesting to see how efficiently resources can be shared among all three classes of users (URLLC, eMBB, mMTC).

Appendix A: Chapter 2 Proofs

A.1 Proof of Lemma 2.5.1

The current channel rate realization c_n is considered good for opportunistic scheduling if,

$$\begin{aligned} \max_{i=1,\dots,\tau_n+1} C_{n+i} &< c_n, \\ \iff F_C\left(\max_{i=1,\dots,\tau_n+1} C_{n+i}\right) &\overset{(a)}{<} F_C(c_n), \\ \iff \max_{i=1,\dots,\tau_n+1} F_C\left(C_{n+i}\right) &\overset{(b)}{<} F_C(c_n), \\ \iff \max_{i=1,\dots,\tau_n+1} U_i &\overset{(c)}{<} F_C(c_n). \end{aligned} \tag{A.1}$$

Where step (a) follows from the monotonicity of the cumulative distribution function (CDF) $F_C(\cdot)$ of the wireless channel strength and step (b) follows from the commutative property of the max function with CDF $F_C(\cdot)$. Step (c) follows from the fact that $F_C(C_{n+i}) \sim U_i$ are i.i.d. Uniform$[0,1]$.

One could design a dynamic threshold so as to ensure that the probability of *not* seeing a better channel rate realization in the next $\tau_n + 1$ time slots is greater than a pre-specified $\delta \in (0,1)$. Such a design criterion would lead to the following,

$$\begin{aligned} \mathbb{P}\left(\max_{i=1,\dots,\tau_n+1} F_C\left(C_{n+i}\right) < F_C(c_n)\right) &\geq \delta\,, \\ \mathbb{P}\left(\max_{i=1,\dots,\tau_n+1} U_i < F_C(c_n)\right) &\geq \delta\,, \\ \left(F_C(c_n)\right)^{\tau_n+1} &\overset{(a)}{\geq} \delta\,, \\ (\tau_n+1)\log F_C(c_n) &\geq \log\delta\,, \\ \implies F_C(c_n) &\geq \delta^{\frac{1}{\tau_n+1}} \end{aligned} \tag{A.2}$$

where step (a) follows from the CDF of the maximum of τ_n+1 independent uniformly distributed random variables. Consequently, the threshold is chosen to be,

$$\gamma_n^\pi = F_C^{-1}\left(\delta^{\frac{1}{\tau_n+1}}\right). \tag{A.3}$$

A.2 Proof of Theorem 2.6.1

Let M_n^{WGRS} and M_n^{OGRS} denote the (possibly fractional) number of resource blocks used to serve the user queue at time n, under WGRS(s) and OGRS(s)-DTE scheduling policies, respectively. Without loss of generality, let the system start with an empty queue and let $(0, N]$ denote a busy cycle of the WGRS policy. We compare the performance of WGRS and OGRS schedulers under a coupled queueing system, where both queues see the same arrival and channel rate processes but one is serviced by schedulingdling policy and the other by OGRS. First, we will show that in any WGRS busy cycle, the resource requirement for WGRS stochastically dominates that of OGRS, i.e.,

$$\sum_{n=1}^{N} M_n^{\text{WGRS}} \geq_{st} \sum_{n=1}^{N} M_n^{\text{OGRS}}. \tag{A.4}$$

Then we will prove that,a in steady state, the average resource requirement under WGRS is greater than that required by OGRS using the stochastic dominance result.

As long as the user queue is sufficiently backlogged, WGRS provides a deterministic service rate s throughout it's busy cycle. The only nondeterministic part of the WGRS scheduling policy is at the end of it's busy cycle N, when there might not be enough data in the queue to utilize service rate s fully. Let us partition the interval $(0, N]$ based on time instants when the channel rate exceeds the adaptive threshold. Define $T_1 \in (0, N]$ as the first time the channel rate exceeds the threshold γ_{T_1} of the OGRS policy, i.e.,

$$T_1 = \min\left(N, \min_{t>0}(t : C_t > \gamma_t)\right), \tag{A.5}$$

where γ is the OGRS-DTE threshold as previously defined in 2.5.1.3. Note by definition, the user queue length and the number of RBs utilized to schedule data under both WGRS and OGRS policies will be the same until T_1, i.e.,

$$\begin{aligned} M_n^{\text{WGRS}} &= M_n^{\text{OGRS}} \text{ a.s. } , \forall n \in (0, T_1), \text{ and} \\ Q_n^{\text{WGRS}} &= Q_n^{\text{OGRS}} \text{ a.s. } , \forall n \in (0, T_1]. \end{aligned} \tag{A.6}$$

Consider a particular realization of $T_1 = t_1$. Denote by Q_{t_1} (same for WGRS and OGRS) the amount of data available to be transmitted at time t_1 and note that this is the same for both policies.

Now we shall compare the number of RBs that will be used by both the policies to service Q_{t_1}. Since the channel rate exceeds the DTE threshold at time t_1, OGRS will use $M_{t_1}^{\text{OGRS}} = \dfrac{Q_{t_1}}{C_{t_1}}$ RBs to clear the entire queue. However, the WGRS policy will require $\Delta_1 = \left\lceil \frac{Q_{t_1}}{s} \right\rceil$ time slots to service the same amount of bits in the queue Q_{t_1}. Clearly, $Q_{t_1} \leq s\Delta_1$, therefore, one can conclude that,

$$M_{t_1}^{\text{OGRS}} \leq s\frac{\Delta_1}{\gamma_{t_1}} \text{ a.s.} \tag{A.7}$$

We will show that the number of RBs allocated by OGRS-DTE policy in the interval $(0, t_1]$ is stochastically dominated by that allocated by WGRS policy in the interval $(0, t_1 + \Delta_1)$. Since the number of RBs utilized by both the WGRS and OGRS policies is the same in the interval $(0, t_1)$, it is sufficient to compare their resource allocations in the interval $[t_1, t_1 + \Delta_1)$.

Recall Lemma A.2.1 to obtain,

$$\mathbb{P}(C_{t_1} > C_{t_1+i} | C_{t_1} \geq \gamma_{t_1}) \geq \mathbb{P}(C_{t_1} < C_{t_1+i} | C_{t_1} \geq \gamma_{t_1}) \,, \tag{A.8}$$

where $i = 1, \ldots, \Delta_1 - 1$. Now consider the distribution of the number of RBs that WGRS requires to clear the same queue. The only way that WGRS could require fewer RBs than $s\Delta_1, 1 \leq \Delta_1 \leq d$ is if all the channel realizations between $[t_1+1, t_1+\Delta_1)$ are greater than C_{t_1}, i.e., for any particular realization of the random variable $\Delta_1 = k$,

$$\begin{aligned}
&\mathbb{P}\left(\sum_{n=0}^{k-1} M_{t_1+n}^{\text{WGRS}} \leq s\frac{k}{\gamma}\middle| C_{t_1} \geq \gamma_{t_1}\right) \\
&= \mathbb{P}\left((C_{t_1+1} \geq C_{t_1}) \cap \ldots (C_{t_1+k} \geq C_{t_1} | C_{t_1} \geq \gamma_{t_1})\right), \\
&= \prod_{i=1}^{k-1} \mathbb{P}\left(C_{t_1+i} \geq C_{t_1} | C_{t_1} \geq \gamma_{t_1}, \Delta_1 = k\right), \\
&\overset{(a)}{\leq} \prod_{i=1}^{k-1} \mathbb{P}\left(C_{t_1+i} < C_{t_1} | C_{t_1} \geq \gamma_{t_1}, \Delta_1 = k\right), \\
&\overset{(b)}{=} q^{k-1} < 1 = \mathbb{P}\left(M_{t_1}^{\text{OGRS}} \leq s\frac{k}{\gamma}\middle| C_{t_1} \geq \gamma_{t_1}\right),
\end{aligned} \tag{A.9}$$

where inequality (a) follows from equation (A.8). By definition of t_1 note that $C_{t_1} \geq \gamma_{t_1}$, where the channel rate threshold is at least as large as the median, i.e., $\gamma_{t_1} \geq F_C^{-1}(1/2)$ always, since $\gamma_{t_1} = F_C^{-1}(1 - \frac{1}{\Delta_1+2})$ and $\frac{1}{x+2} \leq \frac{1}{2}$, $\forall x \geq 0$. So it follows that each of the probabilities in the product of step (a) has a value of $q \leq \frac{1}{2}$. Therefore, based on equations (A.6) and (A.9), we have, $\forall t > 0$,

$$\mathbb{P}\left(\sum_{n=0}^{\Delta_1-1} M_{t_1+n}^{\text{WGRS}} \leq t\middle| \Delta_1 = k\right) \leq \mathbb{P}\left(M_{t_1}^{\text{OGRS}} \leq t \middle| \Delta_1 = k\right). \tag{A.10}$$

Consequently, we draw the following conclusion applying theorem [60, Theorem 1.2.15] about the preservation of the stochastic order of two random variables, if there exists an order when conditioned on a dependent random variable, i.e.,

$$M_{t_1}^{\text{OGRS}} \leq_{st} \sum_{n=0}^{\Delta_1-1} M_{t_1+n}^{\text{WGRS}}. \tag{A.11}$$

Note that the above equation holds for any realization t_1 of the random variable T_1, so it must hold for all realizations of T_1. Moreover, recall (A.6) where the number of resources utilized by both policies are equal until T_1, so summarizing the previous equation and (A.6) we have,

$$\sum_{n=0}^{T_1} M_n^{\text{OGRS}} \leq_{st} \sum_{n=0}^{T_1+\Delta_1-1} M_n^{\text{WGRS}} . \tag{A.12}$$

Finally, note that if $T_1 = N$, we are done with the proof. Otherwise, we shall partition the WGRS busy cycle $(0, N]$ for each of the scheduling policies based on the number of occurrences of the channel rate exceeding the threshold,

$$\begin{aligned} \text{OGRS} :&(0, T_1], (T_1, T_2] \ldots (T_P, N], \\ \text{WGRS} :&(0, T_1 + \Delta_1), [t_1 + \Delta_1, T_2 + \Delta_2) \ldots [T_P + \Delta_p, N]. \end{aligned} \tag{A.13}$$

Here the times T_i are defined as follows,

$$T_i = \min\left(N, \min_{t > T_{i-1}} (t : C_t > \gamma_t)\right) . \tag{A.14}$$

For the subsequent WGRS cycle $(t_1, T_2]$, if $T_2 > t_1 + \Delta_1 - 1$ then by OGRS algorithm design, the number of RBs allocated by OGRS in the interval $(t_1, t_1 + \Delta_1)$ is zero and we can repeat the same analysis as we did for interval $(0, t_1]$ to establish stochastic dominance. In case $T_2 \leq t_1+\Delta_1-1$, we know that $\gamma_t \geq F_C^{-1}(1-\frac{1}{3}), \forall t \in [t_1+1, t_1+\Delta_1)$. Therefore we have $\gamma_{T_2} > \gamma_{t_1}$, and the occurrence of any future rate realizations for WGRS to be better than C_{T_2} will be governed by,

$$q = \mathbb{P}\left(C_{T_2+i} > C_{T_2} | \, C_{T_2} > \gamma_{T_2}\right) < 1/3 , \tag{A.15}$$

which has to be satisfied by WGRS over multiple time slots according to $\Delta_2 = \left\lceil \frac{Q_{T_2}}{s} \right\rceil$ a.s., in order to utilize lesser resources than OGRS-DTE. Stochastic dominance of the number of resources allocated by WGRS over that allocated by OGRS can be derived as in equation (A.9), now using q as in (A.15).

Finally, to complete the proof let us define the following random variables,

$$X_i = \sum_{n=1+T_i}^{T_{i+1}} M_n^{\text{OGRS}} \text{ and } Y_i = \sum_{n=T_i+\Delta_i}^{T_{i+1}+\Delta_{i+1}-1} M_n^{\text{OGRS}} , \tag{A.16}$$

where $T_0 = 0$ and $\Delta_0 = 0$. We have shown that $X_i \leq_{st} Y_i, 1 \leq i \leq I$, where I is the number of time instances that the channel rate exceeds the adaptive threshold over the interval $(0, N]$. The result in (A.4) follows from [60, Theorem 1.2.17], because

$$\sum_{i=1}^{I} X_i = \sum_{n=1}^{N} M_n^{\text{WGRS}} \text{ and } \sum_{i=1}^{I} Y_i = \sum_{n=1}^{N} M_n^{\text{OGRS}}. \tag{A.17}$$

We provide simulation results in Sec.3.6.7 that establish the stochastic dominance of the number of resource blocks allocated by both OGRS-DTE policy over WGRS, during a WGRS busy cycle.

If we let the time $n \to \infty$, and $(N_i)_{i\in\mathbb{N}}$ denote the WGRS busy cycle lengths, then the average number of RBs per time slot required by WGRS is given by,

$$\begin{aligned}\mathbb{E}[M^{\text{WGRS}}] &= \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} M_k^{\text{WGRS}} \\ &= \lim_{n\to\infty} \frac{1}{n} \sum_{i:N_i \leq n} \sum_{n=1}^{N_i} M_n^{\text{WGRS}} \\ &\geq_{st} \frac{1}{n} \sum_{i:N_i \leq n} \sum_{n=1}^{N_i} M_n^{\text{OGRS}} = \mathbb{E}[M^{\text{OGRS}}].\end{aligned} \tag{A.18}$$

Lemma A.2.1. *For any two positive random variables X_1, X_2 and a constant $\gamma >$ median(X_2), the following inequality holds,*

$$\mathbb{P}(X_1 \geq X_2 | X_1 \geq \gamma) \geq \mathbb{P}(X_1 < X_2 | X_1 \geq \gamma), \tag{A.19}$$

as long as X_1 has a non-zero probability of taking values higher than γ, i.e., $\mathbb{P}(X_1 \geq \gamma) > 0$.

Proof. The Left Hand Side (LHS) of equation (A.19) can be written as,

$$\mathbb{P}(X_1 \geq X_2 | X_1 \geq \gamma) = \mathbb{E}[\mathbb{1}_{(X_1 \geq X_2)} | X_1 \geq \gamma], \tag{A.20}$$

where $\mathbb{1}_E$ is the indicator function of the event E. Similarly expressing the right-hand side of the equation as an expectation and then finding the difference yields,

$$\begin{aligned}
&\mathbb{P}(X_1 \geq X_2 | X_1 \geq \gamma) - \mathbb{P}(X_1 < X_2 | X_1 \geq \gamma) \\
&\quad = \mathbb{E}[\mathbb{1}_{(X_1 \geq X_2)} | X_1 \geq \gamma] - \mathbb{E}[\mathbb{1}_{(X_1 < X_2)} | X_1 \geq \gamma] \qquad \text{(A.21)} \\
&\quad = \mathbb{E}[\mathbb{1}_{(X_1 \geq X_2)} - \mathbb{1}_{(X_1 < X_2)} | X_1 \geq \gamma].
\end{aligned}$$

Whenever a realization of X_1 is below γ, the RHS above is 0. Otherwise, the right hand side of (A.21) becomes,

$$\begin{aligned}
\mathbb{E}[\mathbb{1}_{(x \geq X_2)} - \mathbb{1}_{(x < X_2)}] &= \mathbb{E}[\mathbb{1}_{(x \geq X_2)}] - \mathbb{E}[\mathbb{1}_{(x < X_2)}] \\
&= \underbrace{\mathbb{P}[(x \geq X_2)]}_{\geq \frac{1}{2}} - \underbrace{\mathbb{P}[(x < X_2)]}_{< \frac{1}{2}} \geq 0. \qquad \text{(A.22)}
\end{aligned}$$

From (A.21) and (A.22), it is clear that for any realization of X_1, the LHS of (A.21) is greater than or equal to 0, leading to the result in (A.19). It should be noted that while we have provided proof for only OGRS-DTE, the same proof would hold for any fixed percentile threshold $\alpha \geq 0.5$. ■

Appendix B: Chapter 3 Proofs

B.1 Proof of Theorem 3.5.1

For any policy π satisfying the delay constraint d, it must be the case that the A_n bits arriving to the user queue at time n are served within the next d slots. Thus, in particular, if we let $S^{\pi,n}_{n+j}$ denote the number of bits of A_n that are served on slot $n+j$, we have that the possibly fractional number of resource blocks required must satisfy,

$$\begin{aligned} N^{\pi}_{n} &= \sum_{j=0}^{d} \frac{S^{\pi}_{n+j}}{C_{n+j}}, \text{ where } \sum_{j=0}^{d} S^{\pi}_{n+j} = A_n, \\ &\geq A_n \min_{0 \leq j \leq d} \left[\frac{1}{C_{n+j}} \right] \quad \text{a.s.} \end{aligned} \tag{B.1}$$

It is easy to establish the following inequalities,

$$\frac{1}{n} \sum_{\tau=1}^{n-d} N^{\pi}_{\tau} \leq \frac{1}{n} \sum_{\tau=1}^{n} M^{\pi}_{\tau} \leq \frac{1}{n} \sum_{\tau=1}^{n} N^{\pi}_{\tau}, \tag{B.2}$$

because on the one hand, the total RBs allocated across the first n time slots is lower bounded by the total number that was allocated to serve the traffic that arrived within $(0, n-d]$; indeed given the delay constraint, all such traffic should be served prior to time n. On the other hand, the total number of RBs allocated in the first n time slots can at most be the total RBs used to serve all the traffic that arrived within $(0, n]$. Taking the limit as $n \to \infty$ in (B.2) and the additional assumptions stated in the theorem, it is clear that the time average of $(M^{\pi}_{n})_n$ converges to $\bar{M}^{\pi}$ and $\bar{M}^{\pi} = \bar{N}^{\pi}$. The lower bound in (3.8) then follows from (3.7) under the assumptions on arrivals and channels being stationary and independent.

Appendix C: Chapter 4

C.1 QoS classes in LTE

4G LTE identifies the following classes of QoS types, with each Guaranteed Bit Rate (GBR) type service associated with a minimum and maximum flow bit rate.

QCI	Type	Priority	PDB	Error	Example Services
1	GBR	2	100 ms	10^{-2}	Conversational voice
2		4	150 ms	10^{-3}	Conversational video (Live streaming)
3		3	50 ms	10^{-3}	Real Time Gaming
4		5	300 ms	10^{-6}	Non-Conversational Video
5	Non-GBR	1	100 ms	10^{-6}	IMS Signalling
6		6	300 ms	10^{-6}	Video (Buffered Streaming)
7		7	100 ms	10^{-3}	Video (Buffered Streaming)
8		8	300 ms	10^{-6}	Video (Buffered Streaming)
9		9			TCP-based (e-mail, chat, FTP, p2p)

Table C.1: QoS Class Indicator for various heterogeneous user types in LTE.

Appendix D: Chapter 5 Proofs

D.1 Proof for Theorem 5.4.1

We have a non cooperative Markov game where each BS agent determines its optimal policy in response to the either the sum reward as in (G1) or a proxy as in (G2), in (G3) that reflects the reward of all BSs in the network. This is an B-player general sum stochastic game known as the Nash Q learning algorithm [51]. The convergence of Nash Q learning has been established in [51, Sec 3.2], when the following three conditions are satisfied: (i) each action state is visited infinitely often, (ii) the learning rate step size satisfies $0 \leq \alpha_t < 1, \sum \alpha_t = \infty, \sum \alpha_t^2 < \infty$ and (iii) the game has either a global optimum or a saddle point. While the first two conditions can be easily satisfied by the choice of learning hyper parameters, the last condition follows from the fact that an n-player game with finite actions has at least one Nash equilibrium with mixed strategy [45]. Note that both (G2) and (G3) use the *relative* downlink rate in (5.10) as the training reward which indirectly reflects on how well other BSs in the network are doing. Specifically, a smaller *relative* downlink rate over a prolonged time duration implies that the neighboring BSs are causing more interference due to their users' queues being active.

D.2 Proof of Lemma. 5.5.1

For a frequency reuse policy $\boldsymbol{\pi}$ and static split rule $\boldsymbol{\phi}$ consider the standard Lyapunov function $V(\mathbf{Q}) = \sum_{b \in \mathcal{B}} (\boldsymbol{q}^b)^T \boldsymbol{q}^b$. Note that the Lyapunov drift of this network is at least as the large as that of the case where all user queues in the network are infinitely backlogged. One can easily check for Foster's stability criterion (in the infinitely backlogged users case) to show that $V(\mathbf{Q})$ has a negative drift when $\exists \boldsymbol{\mu}(\boldsymbol{\pi}, \boldsymbol{\phi}) \in \mathcal{C}^{\infty}$ such that $\boldsymbol{\lambda} < \boldsymbol{\mu}(\boldsymbol{\pi}, \boldsymbol{\phi})$.

D.3 Proof of Lemma 5.5.2

Max weight scheduling algorithm [104] is throughput, i.e., stabilizes the user queue lengths, whenever feasible. Stability of max weight scheduler for a single base station has been established for iid arrivals and Markovian channel variations in [98]. The channel variations as seen at BS b in our setting are iid across all BSs in the network and the frequency reuse policy $\boldsymbol{\pi}$ which determines the availability of resources at each BS is Markovian. Therefore, the channel variations seen at BS b are Markovian. Furthermore arrivals at user queues are assumed independent throughout the network, allowing us to invoke result [98], to show stability.

D.4 Proof of Theorem 5.5.1

Suppose $\boldsymbol{\lambda} \in (\mathcal{C}_{\boldsymbol{\pi}}^{\infty})^{\circ}$, by Lemma 5.5.2 we know that each base station operating under max weight with the neighbors saturated would be stable, as shown by defining a quadratic Lyapunov function [10, Section 5] and showing it has negative drift outside a finite set of queue states. We note that if neighboring BSs are not saturated the drift at the base station would only be larger, because this would reduce the interference seen at the base station. To prove stability of the overall network we can consider a sum of the Lyapunov functions across BSs, since the state of the network as a whole is Markovian. Note that the sum of the Lyapunov functions has a larger service rate and thus a larger negative drift as compared to that of the same network with infinitely backlogged users. Thus it should be clear that user queues at each BS are stable even when the neighboring BSs are not saturated.

D.5 Proof of Theorem 5.5.2

Suppose we can pick a specific local Nash Equilibrium frequency reuse policy $\boldsymbol{\pi}_l$ for our network, let us choose a particular BS in the network. This selected BS is then allowed to learn a frequency reuse policy based on the global reward of the network. Clearly, this can only improve the rewards of the BS and hence the overall network, since the BS agent is now able to learn a policy based on rewards for it's actions using the overall network reward, i.e., $v^{\boldsymbol{\pi}_g^*}(\boldsymbol{s}) \geq v^{\boldsymbol{\pi}_l^*}(\boldsymbol{s})$. Therefore, for every $\boldsymbol{\pi}_l^*$ one can construct a global policy $\boldsymbol{\pi}_g^*$ that achieves better rewards.

D.6 Proof of Theorem 5.5.3

We ignore the trivial case where none of the BSs are interfering with each other (because we would not need an interference mitigation policy) and instead consider a ***tightly coupled*** network. More specifically, we assume that $\exists \tilde{u} \in \mathcal{U}_{b_1}$ such that $c_{\tilde{u}k}(\boldsymbol{s}_1, \boldsymbol{g}, \boldsymbol{q}) < \bar{c}_{\tilde{u}}$, where $\boldsymbol{s}_1 = \{1\}^N$ and $\bar{c}_{\tilde{u}}$ is the data rate that can be supported for user $\tilde{u}$ under no interference. Without loss of generality let $k = N$ and $\tilde{u} \in \mathcal{U}_{b_1}$.

In order to prove this theorem, let us construct a frequency reuse policy vector $\boldsymbol{\pi}^\delta = (\pi^{b_1}\pi^{b_2} \ldots \pi^B)$, such that $\pi'^{b^i}(\boldsymbol{s}) = 1 \quad \forall i \neq 2$ and $\pi^{b_2}(\boldsymbol{s}_2) = \delta$ and $\pi^{b_2}(\boldsymbol{s}_1) = 1-\delta$ for some $\delta \in (0, 1)$. The frequency reuse state $\boldsymbol{s}_2$ denotes that BS b_2 can choose to transmit on all resources except resource N. We denote by $c_{uN}(\boldsymbol{s}_1)$ the rate for user u on resource N under frequency reuse 1, i.e., state $\boldsymbol{s}_1$ at all BSs. Let ϵ_{uk} be the additional downlink data rate for a scheduled user $\tilde{u}$ at BS b_1 when BS b_2 abstains from using resource N, note that $\epsilon_{uN} > 0$ due to lesser interference on resource N from BS b_2. Then for any $\tilde{u} \in \mathcal{U}_{b_1}$ if $\phi_{uk}^{h^b}(\cdot)$ is the fraction of time resource k is allocated to user u then,

$$\begin{aligned}
&\mu_{\tilde{u}}(\boldsymbol{h}, \boldsymbol{\pi}') \\
&= \mathbb{E}^{\nu,\gamma}\left[\sum_{k=1}^{N} \phi_{uk}^{h^b}(\boldsymbol{s}_1, \mathbf{G}, \mathbf{Q}) c_{\tilde{u}k}(\boldsymbol{s}_1, \mathbf{G}, \mathbf{Q})\right] (\because \ \ {\pi'}^{b_1}(\boldsymbol{s}_1) = 1) \\
&= \mathbb{E}^{\nu,\gamma}\Bigg[(1-\delta)\sum_{k=1}^{N} \phi_{uk}^{h^b}(\boldsymbol{s}_1, \mathbf{G}, \mathbf{Q}) c_{\tilde{u}k}(\boldsymbol{s}_1, \mathbf{G}, \mathbf{Q}) \\
&\qquad\qquad + \delta \sum_{k=1}^{N} \phi_{uk}^{h^b}(\boldsymbol{s}_1, \mathbf{G}, \mathbf{Q}) c_{\tilde{u}k}(\boldsymbol{s}_1, \mathbf{G}, \mathbf{Q})\Bigg] \\
&< (1-\delta)\mathbb{E}^{\nu,\gamma}\left[\sum_{k=1}^{N} \phi_{uk}^{h^b}(\boldsymbol{s}_1, \mathbf{G}, \mathbf{Q}) c_{\tilde{u}k}(\boldsymbol{s}_1, \mathbf{G}, \mathbf{Q})\right] \\
&\quad + \mathbb{E}^{\nu,\gamma}\left[\delta\sum_{k=1}^{N-1} \phi_{uk}^{h^b}(\boldsymbol{s}_1, \mathbf{G}, \mathbf{Q}) c_{\tilde{u}k}(\boldsymbol{s}_1, \mathbf{G}, \mathbf{Q})\right] \\
&\quad + (c_{\tilde{u}N}(\boldsymbol{s}_1) + \epsilon_{\tilde{u}N}) \\
&= \mu_{\tilde{u}}(\boldsymbol{h}, \boldsymbol{\pi}_\delta)
\end{aligned} \tag{D.1}$$

It remains to be shown that $u \in \mathcal{U}_{b_2}$ also have increased downlink rates as a result of adopting $\boldsymbol{\pi}_\delta$. Note that user queues at BSs b_1 and b_2 have coupled downlink rates. Preliminaries on queues with coupled service rates have been provided in Appendix 5.6. It has also been shown there that a frequency reuse policy $\boldsymbol{\pi}_\delta$ helps to stabilize higher arrival rates at both queues, when one of the queues backs off from transmission for a fraction of time.

D.7 Proof of Corollary 5.5.1

The results for global reward game follows from the fact that each BS has access to the rewards of all BS in the network. When the RL agents use local reward as a training metric, it has to be noted that the local metric indirectly reflects on the potential future rewards of other BSs in the network. For example, a low proxy metric (read *effective capacity*) indicates that the other BSs in the network are causing very high interference, which in turn implies poor interference coordination among the BSs. A low reward also indicates heavily loaded user queues and longer network delays. Hence the local reward serves as proxy metric for the expected rewards of other BSs in the network, leading to a frequency reuse policy that performs better than π'. For the random action game setting, we use the same local reward with the only change being a reduced action space. Hence the capacity region of (G3) also dominates $\mathcal{C}_{\pi'}$.

D.8 Proof of Theorem 5.6.1

Consider the case where one of the BSs (say b) has stable user queues under Max Weight scheduling. Users at BSs $b' \in \mathcal{B} \setminus b$ will fall into one of the following two categories:

1. *High impact* users who see interference from BS b will achieve higher downlink rates whenever BS b is idle.
2. *No impact* users who are not impacted by BS b, and hence see no change in downlink rates

Hence, the overall network stability holds, in spite of the fact that $\boldsymbol{\lambda} > \boldsymbol{\mu}(\boldsymbol{\pi}, \boldsymbol{\phi})\ \forall \boldsymbol{\mu}(\boldsymbol{\pi}, \boldsymbol{\phi}) \in (\mathcal{C}_{\boldsymbol{\pi}}^{\infty})^{\circ}$, because of the larger Lyapunov drift as long as $\boldsymbol{\lambda} \in (\mathcal{C}_{\boldsymbol{\pi}})^{\circ}$.

A similar sequence of arguments can be applied for each BS in the network, establishing that a Max Weight scheduler can stabilize users if $\boldsymbol{\lambda} \in (\mathcal{C}_{\boldsymbol{\pi}})^{\circ}$.

Bibliography

[1] Keras Optimizers. https://keras.io/api/optimizers/.

[2] 3GPP. 5G NR physical layer procedures for data. Technical Specification (TS) 38.214, 3rd Generation Partnership Project (3GPP), 04 2018. Version 15.2.0.

[3] 3GPP. 5G Study on Channel Models for frequencies from 5 to 100 GHz Release 14. Technical Report (TR) 38.901, 3rd Generation Partnership Project (3GPP), 04 2018. Version 14.3.0.

[4] 3rd Generation Partnership Project (3GPP). Feasibility Study on Interference Cancellation for UTRA FDD User Equipment, Technical Report 25.963, (Release 17). Technical report, 3GPP, 2022.

[5] 5G Americas. New Services & Applications with 5G Ultra-Reliable Low Latency Communications. White Paper, 2018.

[6] Madyan Alsenwi, Nguyen H. Tran, Mehdi Bennis, Anupam Kumar Bairagi, and Choong Seon Hong. eMBB-URLLC Resource Slicing: A Risk-Sensitive Approach. *IEEE Communications Letters*, 23(4):740–743, 2019.

[7] Pablo Ameigeiras, Jorge Navarro-Ortiz, Pilar Andres-Maldonado, Juan M Lopez-Soler, Javier Lorca, Quiliano Perez-Tarrero, and Raquel Garcia-Perez. 3GPP QoS-based scheduling framework for LTE. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):1–14, 2016.

[8] Arjun Anand, Gustavo De Veciana, and Sanjay Shakkottai. Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 1970–1978, 2018.

[9] Arjun Anand, Gustavo De Veciana, and Sanjay Shakkottai. Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks. *IEEE/ACM Transactions on Networking*, 28(2):477–490, 2020.

[10] Matthew Andrews, Krishnan Kumaran, Kavita Ramanan, Alexander Stolyar, Rajiv Vijayakumar, and Phil Whiting. Scheduling in a Queuing System with Asynchronously Varying Service Rates. 18(2):191?217, April 2004.

[11] Matthew Andrews, Krishnan Kumaran, Kavita Ramanan, Alexander Stolyar, Phil Whiting, and Rajiv Vijayakumar. Providing Quality of Service Over a Shared Wireless Link. *IEEE Communications Magazine*, 39(2):150–154, 2001.

[12] Arash Asadi and Vincenzo Mancuso. A Survey on Opportunistic Scheduling in Wireless Communications. *IEEE Communications Surveys & Tutorials*, 15(4):1671–1688, 2013.

[13] Mehdi Bennis and Dusit Niyato. A Q-learning Based Approach to Interference Avoidance in Self-Organized Femtocell Networks. In *2010 IEEE Globecom Workshops*, pages 706–710, Dec 2010.

[14] Rubem Toledo Bergamo, Plínio Santini Dester, and Paulo Cardieri. Throughput-Delay Tradeoff in Coupled Queues in Wireless Networks: Priority and Concurrent Transmissions Regimes. *IEEE Transactions on Wireless Communications*, 20(8):5423–5433, 2021.

[15] R. Bolla, F. Danovaro, F. Davoli, and M. Marchese. An Integrated Dynamic Resource Allocation Scheme for ATM Networks. In *IEEE INFOCOM '93 The Conference on Computer Communications, Proceedings*, pages 1288–1297 vol.3, 1993.

[16] Sem Borst, Onno Boxma, and Miranda van Uitert. The Asymptotic Workload Behavior of Two Coupled Queues. *Queueing systems*, (1-2), 2003.

[17] Sem Borst, Matthieu Jonckheere, and Lasse Leskelä. Stability of Parallel Queueing Systems with Coupled Service Rates. *Discrete Event Dynamic Systems*, 2008.

[18] Lee Breslau, Sugih Jamin, and Scott Shenker. Comments on the Performance of Measurement-Based Admission Control Algorithms. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, volume 3, pages 1233–1242 vol.3, 2000.

[19] Rohitash Chandra, Shaurya Goyal, and Rishabh Gupta. Evaluation of Deep Learning Models for Multi-Step Ahead Time Series Prediction. *IEEE Access*, 9:83105–83123, 2021.

[20] Geetha Chandrasekaran and Gustavo de Veciana. Distributed Reinforcement Learning based Delay Sensitive Decentralized Resource Scheduling. In *2023 Proceedings IEEE WiOpt Workshop on Machine Learning in Wireless Communications (WMLC)*, 2023.

[21] Geetha Chandrasekaran and Gustavo de Veciana. Opportunistic Scheduling for Users with Heterogeneous Minimum Rate QoS Requirements. In *ICC 2024 - IEEE International Conference on Communications*, 2024.

[22] Geetha Chandrasekaran, Gustavo De Veciana, Vishnu Ratnam, Hao Chen, and Charlie Zhang. Delay and Jitter Constrained Wireless Scheduling with Near-Optimal Spectral Efficiency. In *2023 IEEE 34th Annual International Sympo-*

sium on Personal, Indoor and Mobile Radio Communications (PIMRC), pages 1–7, 2023.

[23] Geetha Chandrasekaran, Gustavo de Veciana, Vishnu Ratnam, Hao Chen, and Charlie Zhang. Spectrally Efficient Guaranteed Rate Scheduling for Heterogeneous QoS Constrained Wireless Networks. In *2023 IEEE 21st International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2023.

[24] Cheng-Shang Chang. *Performance Guarantees in Communication Networks.* Springer-Verlag, Berlin, Heidelberg, 2000.

[25] Xianfu Chen, Zhifeng Zhao, and Honggang Zhang. Stochastic Power Adaptation with Multiagent Reinforcement Learning for Cognitive Wireless Mesh Networks. *IEEE Transactions on Mobile Computing*, Nov 2013.

[26] Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications.* Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg, 2009.

[27] Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *Proceedings of the ACM SIGCOMM '89 Workshop on Communications Architectures and Protocols*, pages 1–12. ACM, 1989.

[28] Apostolos Destounis, Georgios S. Paschos, Jesus Arnau, and Marios Kountouris. Scheduling URLLC Users with Reliable Latency Guarantees. In *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 1–8, 2018.

[29] Ioannis Dimitriou and Nikolaos Pappas. Stable Throughput and Delay Analysis of a Random Access Network With Queue-Aware Transmission. *IEEE Transactions on Wireless Communications*, 17(5):3170–3184, 2018.

[30] Ekaterina Evdokimova, Koen De Turck and Dieter Fiems. Coupled Queues with Customer Impatience. *Performance Evaluation*, 118:33–47, 2018.

[31] Atilla Eryilmaz and Irem Koprulu. Discounted-Rate Utility Maximization (DRUM): A Framework for Delay-Sensitive Fair Resource Allocation. In *2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 1–8, 2017.

[32] Atilla Eryilmaz and Rayadurgam Srikant. Fair Resource Allocation in Wireless Networks Using Queue-Length-Based Scheduling and Congestion Control. *IEEE/ACM Transactions on Networking*, 15(6):1333–1344, 2007.

[33] Ali A. Esswie and Klaus I. Pedersen. Opportunistic Spatial Preemptive Scheduling for URLLC and eMBB Coexistence in Multi-User 5G Networks. *IEEE Access*, 6:38451–38463, 2018.

[34] Bo Fang, Srinivas V Bovelli, and Randy H Katz. Admission Control and Scheduling for QoS Support in Packet-Switched Networks. In *Proceedings of the IEEE INFOCOM '99. Conference on Computer Communications. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, volume 2, pages 620–628. IEEE, 1999.

[35] Stefan Farthofer, Matthias Herlich, Christian Maier, Sabrina Pochaba, Ju-

lia Lackner, and Peter Dorfinger. *CRAWDAD srfg/lte-4g-highway-drive-tests-salzburg*. IEEE Dataport, 2022.

[36] Hossam Fattah and Cyril Leung. An Overview of Scheduling Algorithms in Wireless Multimedia Networks. *IEEE Wireless Communications*, 9(5):76–83, 2002.

[37] Sally Floyd and Van Jacobson. Connections with Multiple Classes of Service. *IEEE/ACM Transactions on Networking*, 1(5):585–590, 1993.

[38] Ioannis G. Fraimis and Stavros A. Kotsopoulos. QoS-Based Proportional Fair Allocation Algorithm for OFDMA Wireless Cellular Systems. *IEEE Communications Letters*, 15(10):1091–1093, oct 2011.

[39] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 1999.

[40] Mohammad Gharbieh, Hesham ElSawy, Ahmed Bader, and Mohamed-Slim Alouini. Spatiotemporal Stochastic Modeling of IoT Enabled Cellular Networks: Scalability and Stability Analysis. *IEEE Transactions on Communications*, 65(8):3585–3600, 2017.

[41] Richard J Gibbens and Frank P Kelly. Measurement-based Connection Admission Control. In *15th International Teletraffic Congress*, volume 2, pages 879–888, 1997.

[42] Matthias Grossglauser and David NC Tse. A Framework for Robust Measurement-based Admission Control. *IEEE/ACM Transactions on networking*, 7(3):293–309, 1999.

[43] Zhu Han and K.J. Ray Liu. Noncooperative Power-Control Game and Throughput Game over Wireless Networks. *IEEE Transactions on Communications*, 53(10):1625–1629, 2005.

[44] Md. Emdadul Haque, Faisal Tariq, Muhammad R. A. Khandaker, Kai-Kit Wong, and Yangyang Zhang. A Survey of Scheduling in 5G URLLC and Outlook for Emerging 6G Systems. *IEEE Access*, 11:34372–34396, 2023.

[45] Joseph E. Harrington Jr. Essays on the Foundations of Game Theory. *Managerial and Decision Economics*, 12(4):329–334, 1991.

[46] Rodrigo Hernangomez, Philipp Geuer, Alexandros Palaios, Daniel Schäufele, Cara Watermann, Khawla Taleb-Bouhemadi, Mohammad Parvini, Anton Krause, Sanket Partani, Christian Vielhaus, Martin Kasparick, Daniel F. Külzer, Friedrich Burmeister, Slawomir Stanczak, Gerhard Fettweis, Hans D. Schotten, and Frank H. P. Fitzek. *Berlin V2X*. IEEE Dataport, 2022.

[47] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[48] I-Hong Hou and P. R. Kumar. Admission control and scheduling for qos guarantees for variable-bit-rate applications on wireless channels. In *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '09, page 175–184, New York, NY, USA, 2009. Association for Computing Machinery.

[49] I-Hong Hou and P. R. Kumar. Utility-Optimal Scheduling in Time-varying Wireless Networks with Delay Constraints. In *Proceedings of the Eleventh ACM*

International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '10, page 31–40, New York, NY, USA, 2010. Association for Computing Machinery.

[50] I.-Hong Hou and P. R. Kumar. Real-Time Communication Over Unreliable Wireless Links: a Theory And Its Applications. *IEEE Wireless Communications*, 19(1):48–59, 2012.

[51] Junling Hu and Michael P. Wellman. Nash Q-Learning for General-Sum Stochastic Games. *J. Mach. Learn. Res.*, 4(null):1039?1069, December 2003.

[52] Pihe Hu, Ling Pan, Yu Chen, Zhixuan Fang, and Longbo Huang. Effective multi-user delay-constrained scheduling with deep recurrent reinforcement learning. MobiHoc '22, page 1–10, New York, NY, USA, 2022. Association for Computing Machinery.

[53] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 21, 1984.

[54] Ahmad Jalali, Roberto Padovani, and Rajesh Pankaj. Data Throughput of CDMA-HDR a High Efficiency-High Data Rate Personal Communication Wireless System. In *VTC2000-Spring. 2000 IEEE 51st Vehicular Technology Conference Proceedings (Cat. No.00CH37026)*, volume 3, pages 1854–1858 vol.3, 2000.

[55] Jonggyu Jang, Hyun Jong Yang, and Sunghyun Kim. Learning-Based Distributed Resource Allocation in Asynchronous Multicell Networks. In *Interna-*

tional Conference on Information and Communication Technology Convergence (ICTC), Oct 2018.

[56] Juan Jose Jaramillo, R. Srikant, and Lei Ying. Scheduling for Optimal Rate Allocation in Ad Hoc Networks With Heterogeneous Delay Constraints. *IEEE Journal on Selected Areas in Communications*, 29(5):979–987, 2011.

[57] Juan José Jaramillo and Rayadurgam Srikant. Optimal Scheduling for Fair Resource Allocation in Ad hoc Networks with Elastic and Inelastic Traffic. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. IEEE, 2010.

[58] Ali Karimi, Klaus I. Pedersen, Nurul Huda Mahmood, Guillermo Pocovi, and Preben Mogensen. Efficient Low Complexity Packet Scheduling Algorithm for Mixed URLLC and eMBB Traffic in 5G. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–6, 2019.

[59] Parham Kazemi, Hanan Al-Tous, Christoph Studer, and Olav Tirkkonen. SNR Prediction in Cellular Systems based on Channel Charting. In *2020 IEEE Eighth International Conference on Communications and Networking (ComNet)*, pages 1–8, 2020.

[60] Peter Kiessler. Comparison Methods for Stochastic Models and Risks:Comparison Methods for Stochastic Models and Risks. *Journal of The American Statistical Association - J AMER STATIST ASSN*, 100:704–704, 06 2005.

[61] Charles Knessl and John Morrison. Asymptotic Analysis of Two Coupled Queues with Vastly Different Arrival Rates and Finite Customer Capacities. *Studies in Applied Mathematics*, 2012.

[62] Charles Knessl and John A Morrison. Two Coupled Queues with Vastly Different Arrival Rates: Critical Loading Case. *Advances in operations research*, 2011:1–26, 2011.

[63] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[64] L.B. Le, E. Hossain, and A.S. Alfa. Service Differentiation in Multirate Wireless Networks with Weighted Round-Robin Scheduling and ARQ-based Error Control. *IEEE Transactions on Communications*, 54(2):208–215, 2006.

[65] Jean-Yves Le Boudec and Patrick Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003.

[66] Bin Li, Ruogu Li, and Atilla Eryilmaz. Wireless Scheduling Design for Optimizing Both Service Regularity and Mean Delay in Heavy-Traffic Regimes. *IEEE/ACM Transactions on Networking*, 24(3):1867–1880, 2016.

[67] P. Lin, B. Benssou, Q.L. Ding, and K.C. Chua. CS-WFQ: a Wireless Fair Scheduling Algorithm for Error-Prone Wireless Channels. In *Proceedings Ninth International Conference on Computer Communications and Networks*, pages 276–281, 2000.

[68] Michael L. Littman. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *Machine Learning Proceedings*. 1994.

[69] Richard Lowe and Other Author Names. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[70] Angel Lozano, Robert W. Heath, and Jeffrey G. Andrews. Fundamental Limits of Cooperation. *IEEE Transactions on Information Theory*, 59(9):5213–5226, 2013.

[71] Aunas Manzoor, S. M. Ahsan Kazmi, Shashi Raj Pandey, and Choong Seon Hong. Contract-Based Scheduling of URLLC Packets in Incumbent EMBB Traffic. *IEEE Access*, 8:167516–167526, 2020.

[72] Meshkati, Farhad and Poor, H. Vincent and Schwartz, Stuart C. and Balan, Radu V. Energy-efficient Resource Allocation in Wireless Networks with Quality-of-Service Constraints. *IEEE Transactions on Communications*, 57(11):3406–3414, 2009.

[73] Navid Naderializadeh, Jaroslaw Sydir, Meryem Simsek, and Hosein Nikopour. Resource Management in Wireless Networks via Multi-Agent Deep Reinforcement Learning, 2020.

[74] Y. S. Nasir and D. Guo. Multi-Agent Deep Reinforcement Learning for Dynamic Power Allocation in Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 37(10):2239–2250, 2019.

[75] Qiang Ni and Charilaos C. Zarakovitis. Nash Bargaining Game Theoretic Scheduling for Joint Channel and Power Allocation in Cognitive Radio Systems. *IEEE Journal on Selected Areas in Communications*, 30(1):70–81, 2012.

[76] Gozde Ozcan and M. Cenk Gursoy. Throughput of Cognitive Radio Systems with Finite Blocklength Codes. *IEEE Journal on Selected Areas in Communications*, 31(11):2541–2554, 2013.

[77] Yunus Ozcan and C. Rosenberg. Uplink Scheduling in Multi-Cell OFDMA Networks: A Comprehensive Study. *IEEE Transactions on Mobile Computing*, 2020.

[78] D. Park and B.G. Lee. QoS Support by Using CDF-Based Wireless Packet Scheduling in Fading Channels. *IEEE Transactions on Communications*, 54(5):955–955, 2006.

[79] Shailesh Patil and Gustavo de Veciana. Managing Resources and Quality of Service in Heterogeneous Wireless Systems Exploiting Opportunism. *IEEE/ACM Transactions on Networking*, 15(5):1046–1058, 2007.

[80] Shailesh Patil and Gustavo de Veciana. Measurement-Based Opportunistic Scheduling for Heterogenous Wireless Systems. *IEEE Transactions on Communications*, 57(9):2745–2753, 2009.

[81] Jamol Pender. An analysis of nonstationary coupled queues. *Telecommunication Systems*, 61(4):823–838, 2016.

[82] Petar Popovski, Cedomir Stefanovic, Jimmy J. Nielsen, Elisabeth de Carvalho, Marko Angjelichinoski, Kasper F. Trillingsgaard, and Alexandru-Sabin Bana. Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC). *IEEE Transactions on Communications*, 67(8):5783–5801, 2019.

[83] Petar Popovski, Kasper Fløe Trillingsgaard, Osvaldo Simeone, and Giuseppe Durisi. 5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View. *IEEE Access*, 6:55765–55779, 2018.

[84] S. Ramakrishnan and Venkatesh Ramaiyan. Completely Uncoupled Algorithms for Network Utility Maximization. *IEEE Transactions on Networking*, 2019.

[85] R. Ramesh Rao and Anthony Ephremides. On the Stability of Interacting Queues in a Multiple-Access System. *IEEE Transactions on Information Theory*, 34(5):918–930, 1988.

[86] Balaji Rengarajan and Gustavo de Veciana. Architecture and Abstractions for Environment and Traffic-Aware System-Level Coordination of Wireless Networks. *IEEE/ACM Transactions on Networking*, 2011.

[87] Bilal Sadiq, Seung Jun Baek, and Gustavo de Veciana. Delay-Optimal Opportunistic Scheduling and Approximations: The Log Rule. *IEEE/ACM Transactions on Networking*, 19(2):405–418, 2011.

[88] Anika Schwind, Cise Midoglu, Özgü Alay, Carsten Griwodz, and Florian Wamser. Dissecting the Performance of YouTube Video Streaming in Mobile Networks. *International Journal of Network Management*, 30(3), jan 2019.

[89] Fatemeh Shah-Mohammadi and Andres Kwasinski. Deep Reinforcement Learning Approach to QoE-Driven Resource Allocation for Spectrum Underlay in Cognitive Radio Networks. In *2018 IEEE International Conference on Comm. Workshops (ICC Workshops)*, May 2018.

[90] Gustavo de Veciana Shailesh Patil. Managing Resources and Quality of Service in Heterogeneous Wireless Systems Exploiting Opportunism. *IEEE/ACM Transactions on Networking*, 15(5):1046–1058, oct 2007.

[91] Sanjay Shakkottai and Alexander L Stolyar. Scheduling Algorithms for a Mixture of Real-time and Non-real-time Data in HDR. In *Teletraffic Science and Engineering*, volume 4, pages 793–804. Elsevier, 2001.

[92] Sanjay Shakkottai and Alexander L Stolyar. Scheduling for Multiple Flows Sharing a Time-Varying Channel: The Exponential Rule. *Translations of the American Mathematical Society-Series 2*, 207:185–202, 2002.

[93] Changyang She, Chenyang Yang, and Tony Q. S. Quek. Joint Uplink and Downlink Resource Configuration for Ultra-Reliable and Low-Latency Communications. *IEEE Transactions on Communications*, 66(5):2266–2280, 2018.

[94] Jing Shi and Jonathan S Turner. Implementation and performance of integrated queueing in a router. *IEEE/ACM Transactions on Networking*, 6(2):169–181, 1998.

[95] Mahyar Shirvanimoghaddam, Mohammad Sadegh Mohammadi, Rana Abbas, Aleksandar Minja, Chentao Yue, Balazs Matuz, Guojun Han, Zihuai Lin, Wanchun Liu, Yonghui Li, Sarah Johnson, and Branka Vucetic. Short Block-Length Codes for Ultra-Reliable Low Latency Communications. *IEEE Communications Magazine*, 57(2):130–137, 2019.

[96] David I. Shuman and Mingyan Liu. *Opportunistic Scheduling with Deadline Constraints in Wireless Networks*, pages 127–155. Springer New York, New York, NY, 2011.

[97] Meryem Simsek, Mehdi Bennis, and İsmail Güvenç. Learning Based Frequency- and Time-Domain Inter-Cell Interference Coordination in HetNets. *IEEE Transactions on Vehicular Technology*, Oct 2015.

[98] Alexander L. Stolyar. MaxWeight Scheduling in a Generalized Switch: State Space Collapse and Workload Minimization in Heavy Traffic. *Ann. Appl. Probab.*, 14(1):1–53, 02 2004.

[99] Alexander L Stolyar. Maximizing Queueing Network Utility Subject to Stability: Greedy Primal-Dual Algorithm. *Queueing Systems*, 50:401–457, 2005.

[100] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[101] Patrick Svedman, Sara K. Wilson, and Björn Ottersten. A QoS-aware Proportional Fair Scheduler for Opportunistic OFDM. In *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*, volume 1, pages 558–562 Vol. 1, 2004.

[102] Wojciech Szpankowski. Stability Conditions for Some Distributed Systems: Buffered Random Access Systems. *Advances in Applied Probability*, 26(2):498–515, 1994.

[103] Leandros Tassiulas and Anthony Ephremides. Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks. In *29th IEEE Conference on Decision and Control*, pages 2130–2132. IEEE, 1990.

[104] Leandros Tassiulas and Anthony Ephremides. Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput

in Multihop Radio Networks. *IEEE Transactions on Automatic Control*, (12), Dec 1992.

[105] Tran Kien Thuc, Ekram Hossain, and Hina Tabassum. Downlink Power Control in Two-Tier Cellular Networks With Energy-Harvesting Small Cells as Stochastic Games. *IEEE Transactions on Communications*, 2015.

[106] Christos Tsanikidis and Javad Ghaderi. On the Power of Randomization for Scheduling Real-Time Traffic in Wireless Networks. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 59–68, 2020.

[107] David Tse and Matthias Grossglauser. Measurement-based call admission control: Analysis and simulation. In *Proceedings of INFOCOM'97*, volume 3, pages 981–989. IEEE, 1997.

[108] Boris S. Tsybakov. File Transmission Over Wireless Fast Fading Downlink. *IEEE Transactions on Information Theory*, 48(8):2323–2337, 2002.

[109] Boris Solomonovich Tsybakov and Viktor Alexandrovich Mikhailov. Ergodicity of a Slotted ALOHA System. *Problemy Peredachi Informatsii*, 15(4):73–87, 1979.

[110] Peter van de Ven, Sem Borst, and Seva Shneer. Instability of MaxWeight Scheduling Algorithms. In *IEEE INFOCOM 2009*, pages 1701–1709, 2009.

[111] Yunfei Wei and Other Author Names. User Scheduling and Resource Allocation in HetNets With Hybrid Energy Supply: An Actor-Critic Reinforcement Learning Approach. *IEEE Transactions on Wireless Communications*, Jan 2018.

[112] Huasen Wu, Xiaojun Lin, Xin Liu, and Youguang Zhang. Application-Level Scheduling With Probabilistic Deadline Constraints. *IEEE/ACM Transactions on Networking*, 24(3):1504–1517, 2016.

[113] Qian Xu, Jianping Wang, and Kui Wu. Learning-based Dynamic Resource Provisioning for Network Slicing with Ensured End-to-end Performance Bound. *IEEE Transactions on Network Science and Engineering*, 7(1):28–41, 2018.

[114] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Fred Juang. Deep Reinforcement Learning Based Resource Allocation for V2V Communication. *IEEE Transactions on Vehicular Technology*, 68(4), April 2019.

[115] Hao Yin, Lyutianyang Zhang, and Sumit Roy. Multiplexing URLLC Traffic Within eMBB Services in 5G NR: Fair Scheduling. *IEEE Transactions on Communications*, 69(2):1080–1093, 2021.

[116] Ning Zhao and Other Author Names. Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Cellular Networks. *IEEE Transactions on Wireless Communications*, 18(11), Nov 2019.

Vita

Geetha Chandrasekaran received her B.E., M.S., and Ph.D degrees in Electronics and Communication Engineering from the College Of Engineering Guindy, the Indian Institute of Technology Madras, and the University of Texas at Austin, respectively. She has over five years of prior experience in the industry including design, development and maintenance of embedded firmware for various Consumer Electronic products. She has worked on a wide variety of wireless projects as part of her research internships: *"Decentralized interference mitigation"* at Nokia Bell Labs, NJ (2019), *"WiFi breath rate and heart rate monitoring"* at Qualcomm, SD (2020) and *"Cell breathing with dynamic load balancing"* at Qualcomm, SD (2022). During her Ph.D, she was part of the Wireless Networking and Communications Group (WNCG) in the Chandra Department of Electrical and Computer Engineering, University of Texas at Austin. Her thesis focuses on the design, analysis, optimization and control of wireless scheduling algorithms with attention to differentiated Quality of Service (QoS) across heterogeneous users. Her research interests lie at the intersection of applying statistical inference and machine learning tools to improve the overall performance of next generation wireless applications.

Address: geethac@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.