Towards low-latency and ultra-reliable services in wireless communication

Hao Yin

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:
Sumit Roy, Chair
Thomas R. Henderson
Akshay Gadre

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Towards low-latency and ultra-reliable services in wireless communication

Hao Yin

Chair of the Supervisory Committee:
Sumit Roy
Electrical and Computer Engineering

Ultra-Reliable and Low-Latency Communications (URLLC) services are critical for a wide range of applications such as industrial automation, intelligent transportation systems, gaming, and Virtual/Augmented Reality (VR/AR). The continuous evolution of wireless technologies, represented by 5G New Radio (NR), Wi-Fi 6 (IEEE 802.11ax), and Wi-Fi 7 (IEEE 802.11be), is driven by the goal of delivering URLLC services under a variety of scenarios. However, this goal comes with inherent challenges, such as optimizing scheduling and resource allocation strategies. This thesis addresses these challenges by introducing intelligent solutions for achieving low latency and ultra-high reliability in both 5G and Wi-Fi technologies.

Within the context of 5G NR, this thesis addresses some resource allocation problems - the main contributions detailed in Chapters 3 & 4. Chapter 3 explores multiplexing eMBB and URLLC traffic in 5G downlink transmission, that must satisfy the strict URLLC requirements while maximizing eMBB utility. The resource allocation problem is formulated as utility maximization for the eMBB while attaining proportional fairness and simultaneously satisfying URLLC constraints. This is formulated as an integer programming (IP) problem, offering two approaches for resolution: convex relaxation and a greedy algorithm, to attain superiority over basic round-robin in balancing the utility of eMBB users and meeting URLLC users' latency and reliability requirements. Further, it explores the Integrated Access and Backhaul (IAB) as a method for extending network coverage, offering a cross-layer design for routing and

resource allocation in IAB multi-hop networks under the current 3rd Generation Partnership Project (3GPP) 5G standards. We propose a novel entropy-based reinforcement learning integrated with a federated learning mechanism, which improves the overall performance of IAB networks and significantly accelerates convergence speed. Through simulation, the effectiveness of the proposed approach in outperforming baseline algorithms from both latency and reliability perspectives is confirmed.

Chapter 4 explores low latency applications in emerging Wi-Fi networks, particularly for gaming and Virtual/Augmented Reality (VR/AR) scenarios. The research addresses two critical aspects: adaptive rate control under fast-changing conditions and optimized resource allocation for the newest Wi-Fi 7 standard. The thesis first presents ADR-X, a novel reinforcement learning based wireless rate adaptation technique. ADR-X capitalizes on the power of online learning to respond effectively to rapidly fluctuating channel conditions, which sudden player movements can trigger during gameplay or variations in background interference. Unlike traditional rate adaptation mechanisms that often incur losses between 5-10%, ADR-X, by predictively choosing the appropriate data rates based on channel measurements, is able to limit packet losses to 10x lower levels. Further, we extend the focus to Wi-Fi 7 (IEEE 802.11be) and investigate the new Multi-Link Operation (MLO) feature introduced in this standard. MLO allows simultaneous transmission on multiple channels, enhancing throughput, lowering latency, and reducing collision probability. This work proposes an optimized cross-layer MLO resource allocation algorithm specifically designed for XR-type burst traffic. Compared to baseline MLO resource allocation based on proportional fairness, the proposed algorithm significantly reduces latency, enabling Wi-Fi networks to serve better applications demanding URLLC services.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

First and foremost, I would like to express my profound gratitude to my supervisor, Dr. Sumit Roy. His invaluable guidance, support, and unwavering faith in me have been instrumental in shaping my research journey. Dr. Roy's unique ability to connect academic and industrial relevance in his instruction has consistently illuminated my path, helping me select research topics that are not just academically rigorous, but also significant to industry. His continuous inputs and corrections have kept my research on the right track. Beyond research, Dr. Roy has also been a crucial mentor in providing life advice and assisting in my career choices, for which I am deeply appreciative.

I would also like to extend my heartfelt thanks to Dr. Thomas R. Henderson, whose expert advice on the ns-3 simulator significantly contributed to my research. Moreover, I am immensely grateful to my supervisory committee members, Dr. Thomas R. Henderson, Dr. Akshay Gadre, and Dr. Jingwei Hu for their time, effort, and guidance.

I owe a great deal of gratitude to my family - my parents and grandparents, who have been a consistent source of support during my doctoral studies. I also wish to express my deepest love and gratitude to Mingfei Chen for her unwavering companionship and support throughout this journey. I also want to express my thanks to all the professors during my undergraduate study at HUST: Dr. Xiaojun Hei, Yayu Gao, Guohui Zhong and Yu Liu.

My colleagues and friends, Lyutianyang Zhang, Liu Cao, Sian Jin, Xiangyu Gao, Ran Wei, Sachin Nayak, Collin Brady, Juan V Leon Rosas, Murali Ramanujam, Zhanke Zhou, Yihan Jiang, Kun Su, Yang Zheng, Yizhan Huang, Wenqiu Ma, Ruixuan Wan, Qinlin Yu, Yaohai Zhou, Dongyu Wei, among others, deserve my heartfelt appreciation for their encouragement, collaboration, and camaraderie.

I also want to acknowledge my industry mentors, Dr. Krishna Chintalapudi from Microsoft and Yixiang Li from Meta. Their expertise and advice have enriched my research significantly. I am also grateful to all my colleagues at Microsoft and Meta, including Ranveer Chandra, Yang Xi, Murali Ramanujam, Neelakantan Krishnan, and Eric Wu, whose contributions and support have been invaluable.

This thesis contains works that were supported in part by the NSF ICE-T Award (Grant Number: 1836725), the NIST Award (Grant Number: 70NANB20H179), and the NSF CCRI Award (Grant Number: 2016379). I would like to express their sincere gratitude to these organizations for their generous financial support and belief in this research. Additionally, I wish to thank the IEEE for granting permission to reproduce and adapt materials previously published by the organization. This support and collaboration have been instrumental in the realization of this thesis.

Reflecting upon my four years of doctoral studies, I recall the challenges I faced, the research failures I encountered, and the moments of self-doubt I experienced. These experiences, though tough, have shaped me into a stronger individual, well-prepared and confident to face future challenges. As I stand at the threshold of my new life, I feel an overwhelming sense of gratitude for my time at the University of Washington, a journey that has been enriching in every way.

DEDICATION

To my dear family

Chapter 1

INTRODUCTION

The growing demand for wireless access in support of new applications signifies the evolution of wireless networks. The upcoming generation of applications such as industrial automation, gaming, and Extended Reality (XR), including Virtual/Augmented Reality (VR/AR), heavily depend on ultra-low latency and high-reliability links. This, at times, includes high data rates as well. This evolution is a testament to the increasing network capacity and proof of lower application delays and higher reliability. In the realm of wireless technology, Wi-Fi and Cellular networks (4G and 5G) have coexisted for years. Wi-Fi, a local area network (LAN), is predominantly used indoors, such as in residences and workplaces. On the other hand, Cellular networks, wide area networks (WANs), are utilized over long distances, both indoors and outdoors. With the rise in the density of wireless networks, fulfilling the requirements of low latency and high reliability has become a challenge, especially in the absence of intelligent scheduling and resource allocation algorithms. This thesis focuses on optimizing these algorithms in the context of the contemporary 5G and Wi-Fi standards.

Emerging 5G networks are expected to support a variety of applications and services, notably the three main classes: (i) Enhanced mobile broadband (eMBB) services for longer duration flows requiring high network throughput; (ii) Ultra-reliable low-latency communication (URLLC) is to support mission-critical applications such as real-time telemedicine and autonomous vehicles; and (iii) massive machine type communications (mMTC) within the Internet of Things (IoT) [1]. For URLLC communications, the most stringent service target is one millisecond (ms) system latency with 99.999% reliability [2]. Simultaneously, multimedia flows utilizing eMBB desire bandwidth resources to achieve transmission rates of up to 10 Gbps. As illustrated in Fig. 1.1, for each base station, we need to allocate the wireless resource for different users and services. A key enabler is the emergence of *network*

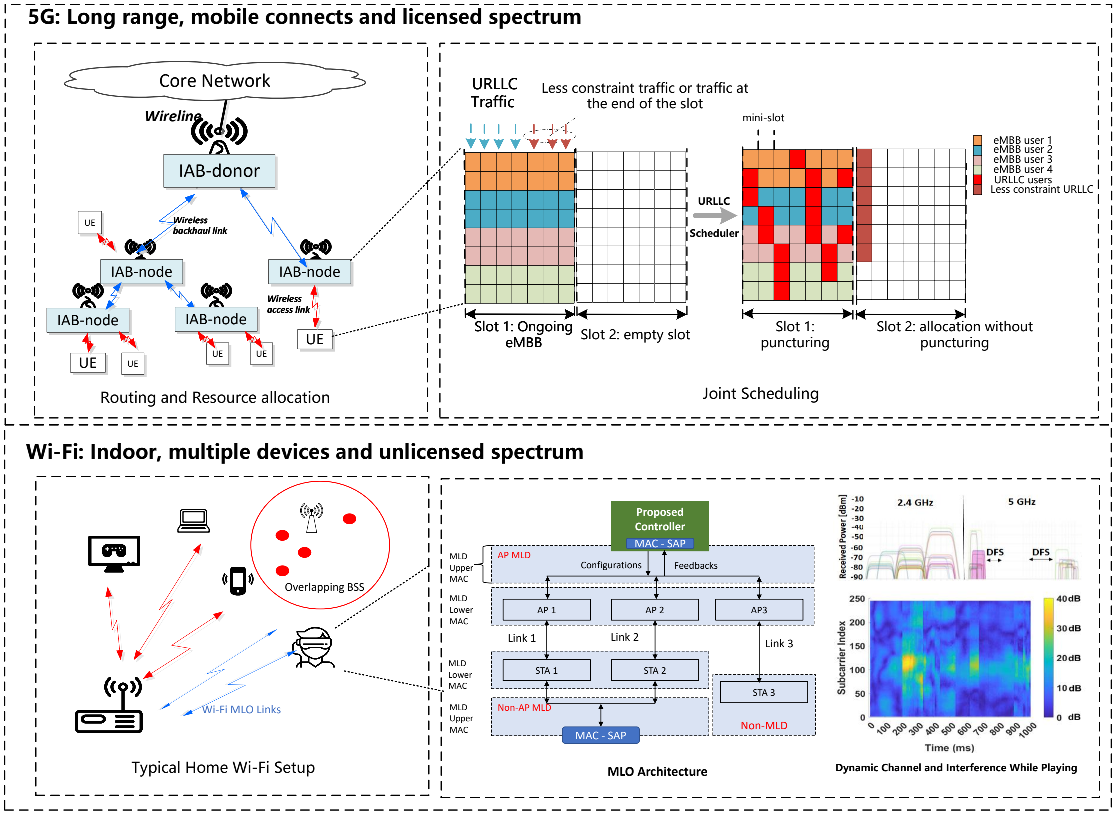


Figure 1.1: Illustration of URLLC services and resource allocation in 5G and Wi-Fi

slicing concepts that allocate specific radio access network (RAN) functionality to each of the supported services within the same network infrastructure. In our work, RAN slicing is implemented at the packet level, where the scheduler decides the physical resource allocation [3]. To satisfy delay and reliability constraints, resources originally allocated to eMBB users are re-allocated over mini-slot duration by 'puncturing', i.e., setting the transmission power of eMBB traffic to zero in the mini-slot. Alternatively, choosing non-zero transmission powers for both eMBB and URLLC traffic may be used to achieve superposition. In either case, the consequence is a reduction of eMBB user throughput. We investigate a scheduler design that maximizes the original eMBB users' utility while guaranteeing the QoS of URLLC users.

In pursuit of higher network capacity, frequencies above 24 GHz have been identified for Radio Access Networks (RANs) - so-called Frequency Range 2 (FR2) or millimeter-wave (mmWave) band - for 5G networks, for meeting the demands from traffic growth that is challenging the capacity of access networks below 6 GHz. However, mmWaves exhibit unfavorable propagation characteristics such as high isotropic losses and marked susceptibility to blockages and signal attenuation[4]. Indeed, mmWave deployments are typically coverage-limited, leading to denser deployments for hot-spot (high-demand) style scenarios [5]. In order to provide a technically effective and economically viable solution to the required network densification, wireless backhaul solutions for 5G networks have recently emerged as a viable strategy. Notably, 3GPP Release 16 specifications introduced a new multi-hop wireless access architecture, Integrated Access and Backhaul (IAB), a wireless backhaul solution in which the access and backhaul links share the same hardware, protocol stack, and spectrum. As illustrated in Fig. 1.1, IAB uses relaying among infrastructure nodes (IAB-nodes) to extend the coverage for the mobile edge users to the base station (IAB-donor) that is connected by high bandwidth wireline to the 5G core. This work explores the resource allocation schemes for the URLLC traffic under the IAB structure and design.

Because of the lower deployment, maintenance and expansion costs, Wi-Fi remains a key technology in home and business environments, especially when access points need to serve more users. This provides powerful support for personal computers, tablets, smartphones, streaming media devices, televisions, printers and other devices that require large amounts of data to connect to the network as shown in Fig. 1.1. One of the major scenarios for the URLLC service with Wi-Fi is the gaming scenario. During gameplay, accessories undergo rapid motion as users actively engage in the game, e.g. due to the hand/head motion of a gamer amidst an exciting game. As indicated by our measurement study, these sudden movements result in “significant” changes in the wireless channel between the console and accessories within the order of 50-100 milliseconds – much faster than traditional Wi-Fi channels. The inability to adapt Wi-Fi data rates to these rapid changes results in packet losses and undermines three key aspects of the gaming experience – responsiveness to user-generated inputs from controllers, quality of streamed gaming audio/video, and battery longevity (due to packet re-transmissions). To address these challenges, we proposed a new algorithm to fast adapt and act prospectively for the Wi-Fi rate adaptation.

Moreover, the IEEE 802.11be Task Group (TGbe) working on Extremely High Throughput (EHT) [6] networks (Wi-Fi 7) is proposing new physical (PHY) and medium access control (MAC) layer enhancements. These enhancements aim to achieve low latency and high-reliability transmissions. A crucial feature of these enhancements is multi-link operation (MLO) that allows simultaneous transmission on multiple channels, typically in different bands, as depicted in Fig. 1.1, via the utilization of existing radio interfaces on Wi-Fi devices [7]. Opening the 6 GHz bands for greenfield Wi-Fi 7 operation enables time-sensitive network (TSN) operations [8]; for example, delay-sensitive traffic may be prioritized in MLO to ensure timely and reliable reception.MLO may be viewed as link or carrier aggregation that provides additional degrees of freedom for network optimization by allocating *link resources* so as to achieve desired latency and reliability targets. Consider Fig. 1.1 for XR gaming scenario, where an edge computing node renders multimedia frames for streaming to the headset over Wi-Fi links. Depending on the application data and refresh rates sent simultaneously over MLO. The end-2-end delay experienced thus depends on the worst-case delay among the multiple links. The wireless channel(s) may suffer from differential link congestion and PHY packet losses that negatively impact the end-2-end delay. Therefore, we propose an intelligent scheduling and resource allocation method based on the dynamic channel conditions to fully utilize the potential of MLO.

1.1 Summary of Research Contributions

This dissertation makes the following contributions to serve as evidence for this thesis and push the capabilities of URLLC in both 5G and Wi-Fi:

1. **Proposing new schedule algorithms for multiplexing URLLC traffic within eMBB Services in 5G NR.**
    - We build upon the existing eMBB scheduler and introduce a joint scheduling model that optimizes eMBB and URLLC traffic. It focuses on maximizing the overall

utility for eMBB users while adhering to latency constraints for URLLC users.

   - We present a model to evaluate the delay and reliability of URLLC traffic from a media access control (MAC) layer perspective.
   - We introduce two novel resource allocation algorithms to align with practical implementation for downlink scheduling in the 5G system.

2. **Designing new cross-Layer algorithm for resource allocation and routing in IAB Multi-Hop networks**
   - We formulate a routing optimization problem for the IAB multi-hop network that minimizes transmission latency while ensuring the reliability requirement.
   - We propose a deep reinforcement learning (DRL) framework to solve the routing and resource allocation problem based on local information in the multi-hop network.
   - We further utilize an innovative entropy-based reinforcement learning (RL) algorithm with federated learning (FL) mechanism to enhance the convergence speed. The computation complexity of the proposed algorithm is explored in terms of convergence speed and runtime.
3. **Designing a new ADR-X for link adaptation in Wi-Fi to reduce packet losses in gaming wireless channels**
   - We study the packet losses in gaming that occur in clusters due to the high dynamism of gaming wireless channels, challenging the traditionally held notion target of 0.5% loss rates.
   - We propose ADR-X, a novel ANN-based ADR scheme, to predictively adjust data rates based on the history of channel wireless measurements and packet losses.
   - ADR-X significantly reduces packet losses, even under dynamic gaming wireless channels while can operate efficiently on power, compute, and time-constrained embedded devices like Xbox controllers.

4. **Proposing the resource allocation scheme for 802.11be MLO**
    - We formulate a new resource allocation problem for MLO, aiming to minimize the worst delay over multiple channels while satisfying a cross-layer reliability constraint.
    - We design a new resource allocation scheme for 802.11be MLO to achieve more stringent latency and reliability targets for applications like VR/AR.
    - The results indicate that the proposed algorithm significantly reduces delay and satisfies the loss requirements, presenting a clear advancement to the state-of-art.

1.2 Thesis Outline

The structure of this thesis is as follows: In Chapter 2, we delve into the background and related work surrounding URLLC services and corresponding challenges in 5G and Wi-Fi. Chapter 3 is dedicated to discussing the implementation and proposed optimization of URLLC services in 5G networks. In Chapter 4, we shift our focus to the application and improvements of URLLC services in Wi-Fi networks. Finally, Chapter 5 concludes the thesis by summarizing our key findings and their significance. This thesis includes the material in the author's previous papers published on IEEE [8–11][1].

---

[1] In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University of Washington's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

Chapter 2

BACKGROUND AND RELATED WORK

In the rapidly evolving landscape of wireless communication, two key technologies, 5G New Radio (NR) and IEEE 802.11 ax/be (Wi-Fi 6/7), present promising advancements for ensuring Ultra-Reliable and Low Latency Communications (URLLC). As our digital world becomes more complex and demands grow, these technologies will need to serve a heterogeneous mix of users, offering consistent connectivity for a plethora of applications, such as AR/VR and other low-latency flows, whilst still providing for legacy data flows. Notably, Small Cells, especially in indoor environments, form a critical area where 5G NR and IEEE 802.11ax/be technologies may be both competing and complementary. Their applications are trending towards an overlapping goal - achieving URLLC for a diverse set of users and applications. Nevertheless, their journey towards this shared goal originates from considerably different starting points.

5G NR, primarily designed for wide area networks, is now focusing on indoor and dense urban scenarios using small cells. Its design principles and advancements in the Radio Access Network (RAN) architecture are grounded in mobile broadband applications, but are gradually shifting to accommodate URLLC applications. Conversely, Wi-Fi, traditionally purposed for local area networks (indoor, hotspots), has progressively matured to address wider coverage and better QoS through new iterations, such as 802.11ax and beyond, converging towards URLLC requirements. These distinctions in starting points underscore significant differences in their RAN architecture and Medium Access Control (MAC) protocols. 5G NR relies on a centralized, scheduled type of communication, ensuring QoS, whereas Wi-Fi predominantly uses a decentralized, contention-based approach. While this grants WiFi flexibility and adaptability, it might pose challenges for guaranteeing low latency and reliability in highly congested environments, which is a cornerstone for URLLC.

As such, the complexities and inherent differences between 5G NR and Wi-Fi 6/7 802.11ax/be instigate a need for thorough investigation and innovative solutions for both technologies. This thesis intends to address this need, aiming to provide an exhaustive understanding of how these technologies can be utilized and optimized to meet the ambitious URLLC targets. Through comparative and contrasting analysis, the study will shed light on possible cooperative mechanisms that could lead to an optimized and harmonized heterogeneous network, ensuring reliable, low-latency connectivity for the ever-increasing demands of our digital world.

2.1 Overview of URLLC in 5G network

The QoS requirements for downlink URLLC traffic in 5G are modeled as follows: a UE requests the BS to transmit a packet of size p successfully within an end-to-end delay of τ and the probability of failure at most σ [12][13]. The target reliability σ depends on the users' QoS requirements, while δ is the BLER for each transmission. The δ for single downlink transmission is affected by the channel condition and choice of the Modulation Coding Scheme (MCS). If BLER δ is higher than σ, it may take several retransmissions to achieve the reliability requirement.

2.1.1 Scheduling and Resource Allocation for Single Transmission

To ensure the probability of successful delivery, the delay caused by the retransmission scheme like HARQ in LTE/5G NR must be considered. In this part, we analyze the URLLC for one-shot transmission as a function of selected MCS, so the δ is known and configured according to the URLLC requirements. This method is extended to include retransmissions if we consider the retransmission packets as more constrained URLLC users, i.e., the retransmission latency is a reduction of the original latency by the passing time. Further, the model could also be extended to optimize the choice of MCS and HARQ processing.

3GPP standards introduce different use cases for URLLC [2], depending on traffic type and requirements. For example, information exchange in intelligent transportation between a UE supporting V2X application and a V2X application server requires a max end-to-end latency of 5ms and reliability of 99.999% with a 1 Mbps downlink data rate. In such cases, while bandwidth requirements for each user are modest, increasing user numbers and different traffic types will combine (as in a downtown rush-hour scenario) to render the joint scheduling problem considered in this work of increasing significance.

A key degree of freedom in 5G for resource allocation is the flexible numerology (u) [14] that allows sub-carrier spacing to scale as $2^u \times 15$ kHz to provide a balance between different service requirements. LTE system latency in the user plane is typically measured as a multiple of Transmission Time Interval (TTI). The analysis of NR can reuse the same approach but with different system parameters due to enhanced hardware capability and numerology, summarised by [9, 15]. The NR TTI length is equal to the slot length[1].

2.1.2 User Selection Based on URLLC Constraint

URLLC Reliability Constraint

In 5G downlink scheduling, we expect the channel condition between each user and base station, i.e. BLER to be below acceptable limits - 10^{-5} in NR, 10^{-1} in LTE. The UE estimates the downlink channel and reports back CQI for BS to choose the MCS to meet the BLER requirement δ for its transmission. Thus, during each scheduling procedure, the BS knows the BLER for given MCS according to the configuration. Note that most URLLC traffic has strict latency constraint target (less than 1 ms) with error rate less than 10^{-5}. In practice, the BLER target is set to 10^{-3} to achieve higher throughput but also allow retransmissions for lower BLER, thereby allowing adaptation for different service requirements. The URLLC scheduler assumes that the inputs from the PHY layer are accurate and estimates the BLER probability for the MCS chosen. Thereafter, it resorts to blind retransmissions if the ACK/NACK from HARQ is not received within a time-out interval. Each transmission/retransmission is

---

[1] For example, the TTI is $\frac{1}{2^3}$ ms when using numerology 3.

independent, and the probability of failure is given by

$$\mathbb{P}_{\text{fail}}(m) = (\delta_m)^{k_{\text{trans}}(m)} \leq \sigma_m, \tag{2.1}$$

where σ_m is the reliability constraint, $k_{\text{trans}}(m) = k_{\text{retrans}}(m) + 1$, denoting the total number of transmission and retransmissions for user m for BLER target δ, respectively. k_{trans} can be obtained from Eq. (2.7) to indicate the # of blind retransmissions:

$$k_{\text{trans}}(m) = \left\lceil \frac{\log(\sigma_m)}{\log(\delta_m)} \right\rceil . \tag{2.2}$$

URLLC Latency Constraint

LTE system latency in the user plane is typically measured as a multiple of Transmission Time Interval (TTI). The analysis for NR can reuse the same approach but with different system parameters due to enhanced hardware capability, summarised by Samsung [15] and [16]. The NR TTI length is equal to the slot (eMBB) or mini-slot (URLLC) length[2]. The total latency can be split into several parts shown in Fig. 4.14.

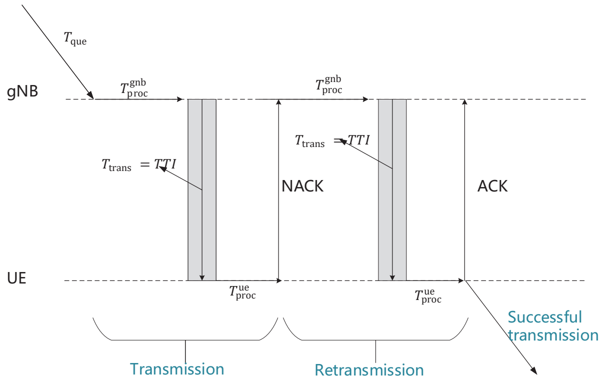


Figure 2.1: Illustration of latency components for DL transmissions in NR.

One-shot transmission: The total delay depends upon 4 components - i) queuing time before allocation T_{que}, ii) processing time at gNB $T^{\text{gnb}}_{\text{proc}}$, iii) transmission time T_{trans}, and iv)

---

[2] For example, for a 2-symbol mini-slot, the TTI is 1/7 ms when using numerology 0 whose slot length is 1 ms.

processing time at UE $T_{\text{proc}}^{\text{ue}}$. The transmission time $T_{\text{trans}} = TTI$ and we assume that all these delay components are a multiple of URLLC TTI (same length as mini-slot) as suggested above. Then the total delay of URLLC user m is calculated in Eq. (2.3).

$$\begin{aligned} T_{\text{delay}}^{\text{once}}(m) &= T_{\text{que}}(m) + T_{\text{proc}}^{\text{gnb}}(m) + T_{\text{proc}}^{\text{ue}}(m) + T_{\text{trans}}(m) \\ &= T_{\text{que}}(m) + T_{\text{proc}}(m) + T_{\text{trans}}(m) \\ &= (\alpha_m + \beta_m + 1) \times TTI, \end{aligned} \tag{2.3}$$

where $\alpha_m \in \mathcal{Z}^{++}$ is a positive integer and represents the multiplier of queuing delay which depends on the scheduler policy, and $\beta_m \in \mathcal{Z}^{++}$ is multiplier of total processing delay, normally $\beta = 4$ [15, 17]. Since retransmission have the highest priority and are immediately scheduled, there is no queuing delay, thus $T_{\text{retrans}}^{\text{once}}(m) = T_{\text{proc}}(m) + T_{\text{trans}}(m) = (\beta_m + 1) \times TTI$. So the total delay for the transmission and k_{retrans} retransmissions must satisfy

$$\begin{aligned} T_{\text{delay}} &= T_{\text{delay}}^{\text{once}}(m) + k_{\text{retrans}}(m) \times T_{\text{retrans}}^{\text{once}}(m) \\ &= (\alpha_m + \beta_m + 1 + k_{\text{retrans}}(m) \times (\beta_m + 1)) \times TTI \\ &= (\alpha_m + k_{\text{trans}}(m) \times (\beta_m + 1)) \times TTI \\ &\leq \tau_m, \end{aligned} \tag{2.4}$$

From the above, we conclude that if the system BLER is lower than the acceptable probability of failure, then only one transmission without any retransmission is sufficient. For VR/AR applications with 10 ms latency and 99.9% reliability requirements, the value of α_m can be increased such the URLLC packets can be transmitted in the *following* eMBB slot instead of the current mini-slot.

Let the actual delay T_{delay} equal the latency constraint τ in Eq. (2.4), then we obtain the upper bound of queuing delay α_m for user m, i.e.,

$$\alpha_m = \left\lfloor \frac{\tau}{TTI} \right\rfloor - k_{\text{trans}}(m) \times (\beta_m + 1). \tag{2.5}$$

Thus, the latency requirement is met by allowing user m to transmit within α_m mini-slots starting from the URLLC user arrival mini-slot d[3]. However, if $\alpha_m \leq 0$, the URLLC scheduler has no choice but to transmit instantly even though the latency requirement may not be satisfied. The value of queuing delay also indicates the priority of each URLLC user, i.e., larger the α_m, lower the priority.

2.1.3 Integrated Access and Backhaul Architectures in 5G NR

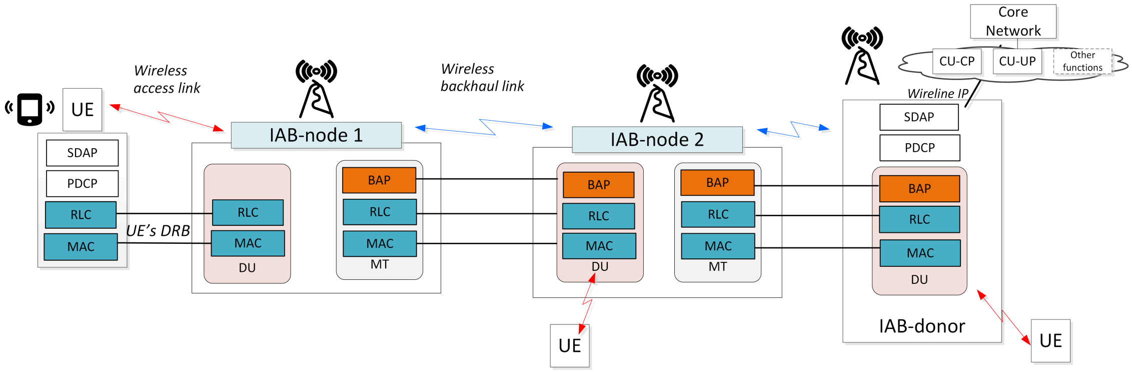


Figure 2.2: Protocol stack for UE-access using IAB-relaying with the new introduced adaptation layer. The backhaul adaptation protocol (BAP) layer is added within the layer 2 for the routing of the wireless backhoul link in 3GPP IAB architecture 1a.

To cope with the need for appropriate backhaul rates for small cell networks, 3GPP first proposed a study item on IAB in [18]. The physical-layer specification of IAB were completed in 2019, and higher-layer protocols and architecture were completed in 3GPP Rel-16 [19]. Further enhancements (e.g., mobile IAB) have been carried out in 3GPP Rel-17. However, despite the consensus about IAB's ability to reduce costs, designing a high-performance IAB network is still an open research challenge [20].

---

[3] We split the eMBB slot into several mini-slots, for example, seven mini-slots and the order is from 1-7. If $d = 3$, it means the URLLC user comes in at the third mini-slot. Note that α_m is an integer and TTI length equals to the mini-slot length, so if $d + \alpha_m > 7$, it means that the URLLC could wait for the next eMBB slot to be scheduled.

As Fig. 2.2 shows, two types of wireless links constitute the IAB network: access and backhaul links. An access link connects UE and an IAB node or IAB donor, while a backhaul link exists between IAB parent and IAB child node. The IAB functionality requires two network entities: IAB-donor and IAB-node(s). An IAB-donor is a gNB that provides network access to UEs via a network of backhaul and access links. The IAB donor is split into a centralized unit (CU), which terminates the Packet Data Convergence Protocol (PDCP) and the Radio Resource Control (RRC) protocol as well as a distributed unit (DU) that terminates the lower protocols, i.e., Radio Link Control (RLC), Medium Access Control (MAC) and the physical (PHY) [19]. The motivation of the CU/DU functional split in the IAB donor is that all time-sensitive functionalities, e.g., scheduling, fast re-transmission, segmentation, etc., can be realized in the DU, i.e., close to the radio and the antenna. At the same time, it is possible to centralize the more minor time-sensitive radio functionalities in the CU [21]. The IAB-donor connects to the IAB-node(s) using the 5G NR access interface and is connected to the Core Network (CN). A Backhaul Adaptation Protocol (BAP) layer is added above the RLC layer in order to include routing information and allow for hop-by-hop forwarding. The IAB node comprises mobile termination (MT) and DU functionalities. The IAB node connects to an upstream IAB node or an IAB donor's DU via MT function, while it also provides wireless backhaul for the downstream IAB nodes and UEs via the DU function. Note that IAB nodes can be cascaded without a technical limit to the number of IAB nodes. Therefore, it is important to consider the latency and reliability of multi-hop transmission.

The 3GPP standard proposed five different configurations for IAB architecture, with various levels of decentralization of the network and backhauling functionalities [20]. In this work, we consider Architecture 1a, which has more chance to be selected for future standardization according to the 3GPP nomenclature [20]. It should be noted that this architecture not only leverages CU/DU-split architecture but also adds an adaptation layer that replaces the IP functionality to hold wireless routing information enabling hop-by-hop forwarding [18]. Fig. 2.2 shows Architecture 1a, where multiple IAB-nodes use wireless backhaul, while IAB-donors have fiber connectivity toward the core network. In this architecture, the IAB donor is the node that serves the IAB nodes and other UEs that are directly connected to it. Each IAB node has a mobile termination (MT) function which connects to a parent DU (IAB donor DU or another IAB node DU) and a DU function that serves UEs or the MT functions of child IAB nodes. Such a configuration yields the most limited impact on the core network and signaling overhead and the lowest relay complexity and processing requirements [21]. Compared with other architectures, Architecture 1a implements a functional split of the radio protocol stack (the split happens at the RLC layer), with the control and upper layers in the IAB-donor CU and the lower layers in the DUs of the IAB-nodes. Therefore, the RRC, SDAP, and PDCP layers reside in the CU, while RLC, MAC, and PHY are in the DUs. An additional adaptation layer manages the routing on top of RLC, enabling the end-to-end connection between DUs and CU.

2.1.4 Latency for Multi-hop transmission

With regard to the multi-hop transmission, the packet in the relay nodes, which is forwarded to the next node, does not need to go through all the stack procedures as discussed above. By introducing the CU/DU split architecture, the processing time is reduced since no need to traverse the whole L2 and L3 stacks compared with a non-split option. We assume that each relay node would immediately forward the packet to the next node, the latency caused by each relay is thereby half of the processing time plus the transmission time, i.e., $T_{\text{relay}} = \frac{1}{2}T_{\text{proc}} + T_{\text{trans}}^{\text{relay}}$. Thus, the total delay from the source to destination through n relay nodes is calculated in Eq. (2.6):

$$T_{\text{delay}} = T_{\text{delay}}^{\text{dir}} + n \cdot T_{\text{relay}} = T_{\text{que}} + \frac{n+2}{2}T_{\text{proc}} + \sum_{i=1}^{n+1} T_{\text{trans}}(i), \tag{2.6}$$

where $T_{\text{trans}}(i)$ is the transmission time at the link i.

2.1.5 Resource allocation and Reliability

In 5G network, the receiver (RX) reports channel status information (CSI) to help the transmitter (TX) to determine the MCS and which RB to transmit. This method is also applied for the multi-hop networks. The feedback from the RX sides contains the channel quality indicator (CQI) that estimates at the RX and instructs the TX to select a corresponding MCS for a certain block error rate (BLER). The CQI feedback is based on the CSI reference signal of each subband (contains several contiguous RB). The TX then uses the CQI value to determine the MCS for each transmission. In a multi-hop network, the nodes are not all scheduled by a center controller, thus the collision and interference may happen when two close nodes transmit at the same time. On the basis of the procedures analysis above, the reliability is composed of two parts, the collision probability p_c [4] and the BLER p_b on each link. Regarding a multi-hop transmission with n relays (total $n+2$ nodes and $n+1$ transmissions), we assume the transmission between two nodes is independent, the probability of a successful multi-hop transmission $\mathbb{P}$ is thus given by

$$\mathbb{P} = \prod_{i=1}^{n+1} (1 - p_b(i))(1 - p_c(i)) = \prod_{i=1}^{n+1} p_s(i)(1 - p_c(i)), \tag{2.7}$$

where $p_b(i)$ is pre-configured for the i^{th} link and $p_s(i) = 1 - p_b(i)$ denotes the probability of a successful transmission expected by the current configuration. In this work, we consider the 3GPP channel model [22]. It supports the modeling of wireless channels between 0.5 and 100 GHz by means of a stochastic Spatial Channel Model (SCM), in which a single instance of the channel matrix $\mathbf{H}$ is computed according to random distributions for large scale fading parameters (i.e., the delay profile, the angles of arrival and departure, and the shadowing) and for the small scale fading (i.e., for small variations in the channel, for example, as given by the Doppler spread).

2.1.6 Related Work

Reducing end-to-end delay while achieving ultra-high reliability for URLLC continues to be a challenge over lossy networks, as it typically implies the need for retransmissions (or other similar time-consuming methods) that increase latency. To meet URLLC latency constraints,

---

[4] The collision happens when two close nodes select the same subband at the same time for packet transmission.

adaptive frame structures (that provide for mini-slots) and methods to reduce the processing times have been proposed by 3GPP [14]. Shorter user equipment (UE) and next-generation NodeB (gNB) processing times are achieved by introducing a new structure for the data channel and new channel codes [23, 24]. In [25], the authors provide an overview of URLLC and analyze the latency components in 5G NR downlink. Further, to reduce the block error rate (BLER) over a link, more accurate channel estimation by UE is studied in [26, 27], while configurable BLER targets for one-shot transmission given the channel condition is proposed in [28]. What's more, Hybrid Automatic Repeat reQuest (HARQ) is a known effective mechanism within LTE/NR to achieve efficient transmission and low residual BLER. However, opportunities for HARQ retransmissions in URLLC will be limited due to the latency constraints. Hence, to reduce the need to wait for round trip times for HARQ operation, *blind repetitions* whereby multiple packets are sent without waiting for HARQ acknowledgment (ACK) are adopted as a solution in 3GPP [28]. A null-space based preemptive scheduler for joint URLLC and eMBB traffic is proposed for dense 5G networks in [29]; non-orthogonal multiple access (NOMA) based multiplexing of URLLC and eMBB at the edge nodes is studied in [30, 31]. While these techniques could support simultaneous transmission for eMBB and URLLC users (rather than puncturing eMBB users), these methods are effective when the interference among users is modest. In several use cases, the above is infeasible[5] and puncturing is a practical alternative. Besides [30], the successive interference cancellation algorithm required as a result of NOMA at the receiver could introduce significant additional latency, making it difficult to meet strict latency constraints.

Prior work that focuses solely on URLLC performance includes [12, 32] that proposes a resource allocation and HARQ optimization scheme to minimize the necessary bandwidth for URLLC traffic. Similarly, [33] considers blind retransmission and proposed a model to reduce resource consumption. However, the need is to consider such allocation to URLLC *in conjunction with other existing 5G traffic* and not alone - thus, joint scheduling approaches

---

[5] For example, the passengers are playing video (eMBB) on an auto-driving car (URLLC).

are needed. In [34, 35], a RAN resource allocation problem is formulated as a sum-rate maximization problem subject to the orthogonality constraint (i.e., service isolation), latency-related constraint and minimum rate constraint while maintaining the reliability constraint with the incorporation of adaptive modulation and coding. Common physical network infrastructure is sliced into multiple end-to-end logical networks, each acting as a dedicated network for a specific service. In [36], an optimization problem is formulated that maximizes the minimum expected rate of eMBB users over the long run while meeting the URLLC requirements, but without considering fairness among the eMBB users.

There have been very few studies in this regard; models for eMBB rate loss associated with simultaneous URLLC is first proposed in [37, 38] and is the closest precursor to our effort. However, in that work, the authors propose new schedulers based on the assumption of *fixed set of backlogged eMBB users that all receive resources without using slot aggregation*. First, eMBB traffic is dynamic with users entering and leaving the system (hence not a fixed set). Second, our preferred solution is to come up with a URLLC RB allocation policy in NR given the already existing eMBB allocation, as a convenient upgrade path for systems operation. The proposed resource allocation for URLLC traffic works on a mini-slot scale and maintains current eMBB schedulers on the slot scale. Methodologically, our approach is similar to [39–43] that also proposed a constraint-based formulation that maximizes the eMBB aggregate data rate while satisfying the URLLC constraints. We differ from the above in two significant ways: the aggregate rate maximization often results in loss of fairness among eMBB users. We thus incorporate proportional fairness concepts within the aggregate utility maximization. Further, we also incorporate HARQ retransmission, which is not considered in the above. In this way, we push the state-of-art for a joint scheduler design while balancing multiple objectives (eMBB rate and latency constraints) while also preserving fairness.

As far as cellular networks, 3GPP defined a version of LTE relays [44] limited to two-hop communications, that did not achieve significant commercial success. Conversely, 5G NR is a beam-based air interface relying on dedicated reference symbols and channels, alleviating many of the constraints LTE suffered. Because mmWave transmissions are highly directional, interference is naturally mitigated with appropriately elevated BS locations. Thus, mmWave multi-hop based in IAB can be designed more like a wired multi-hop network with switches and hubs than a conventional wireless multi-hop system. The backhaul traffic is routed to the donor node, with scheduling at each hop, effectively managing the network interference.

In this work, we focus on *joint* routing and resource allocation for IAB multi-hop networking in 5G NR. The performance analysis of such mmWave multi-hop network is relatively recent, beginning with initial work in [**?**, 45–49]. [50] presented an analytical framework for IAB-enabled cellular networks on the coverage and performance. In addition, the study [51] proposed a global traffic allocation scheme to achieve the low latency requirements in multi-hop transmission. In [52], a novel joint incentive and resource allocation design were proposed for the IAB problem. However, the primary shortcoming of all above works is that they do not address the URLLC requirements in 5G networks. Due to a lack of global network status information and imperfect sensing, achieving URLLC latency bounds in multi-hop networks is a largely unsolved problem.

Towards intelligent operations and scheduling, there has been a growing interest in the application of artificial intelligence (AI) strategies in 5G NR. In tandem with the new broad and complex features offered by the new IAB protocols, data-driven approaches would enable their optimal usage for real-world applications to achieve the quality, reliability, latency, and efficiency requirements like URLLC. Among all the AI strategies, reinforcement learning (RL) is designed to learn from the environment by exploring the underlining connection of different parameters, which has shown a good fit and performance gain in the scheduling and resource allocation problem [53–55]. An optimization-aided DRL-based framework was developed in [56] to aim at maximizing the eMBB data rate subject to a URLLC reliability constraint in resource slicing problem. [57] proposed an Advantage Actor-Critic (A2C) based RL approach on the IAB resource allocation algorithm, which was able to cope with the dynamics of the link status in mmWave 5G IAB networks. The aforementioned works mainly apply the DRL algorithms to obtain a deterministic policy in the wireless communication optimization problem. However, convergence speed and model generalization are significant concerns regarding the development in the real world. This paper introduces a model-free off-policy DRL algorithm based on maximum entropy reinforcement learning, Soft Actor-Critic (SAC) [58], to accelerate the convergence speed and also align with the current 3GPP standard.

2.2 Overview of URLLC in IEEE 802.11ax/be Networks

In this section, we first provide readers with the necessary background information and the study about the operation of gaming consoles and accessories to motivate the need for new technology for higher reliability. Then we provide a more detailed analysis of the wireless channels and new multi-link operation features in Wi-Fi networks.

2.2.1 Gaming and XR over Wi-Fi

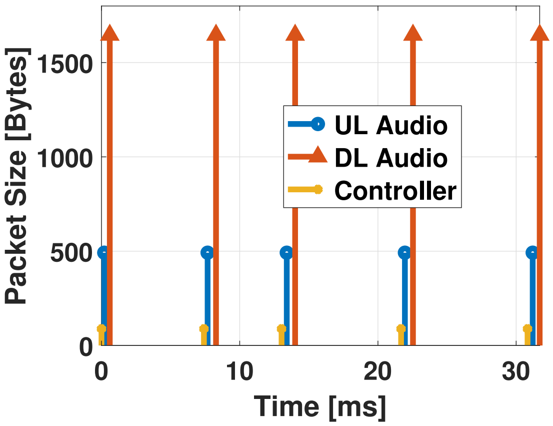


Figure 2.3: Xbox traffic.

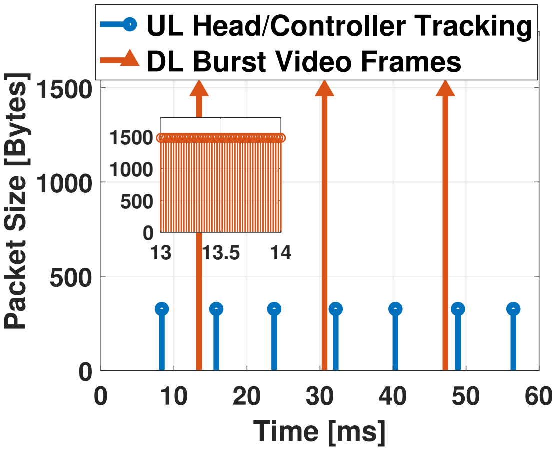


Figure 2.4: VR traffic.

Traffic patterns To understand how packets flow in gaming traffic, we collect packet traces for Xbox [59] and Oculus [60] Quest 2 using a sniffer during two gaming sessions – CrossFireX (Xbox) and Robo-Recall (Oculus Quest 2), both First Person Shooter (FPS) games. As seen in Figure 2.3, Xbox console transmits one PCM game audio (1646 bytes), receives one chat audio packet (492 bytes) and one game input packet (88 bytes) every 8ms. In VR traffic (Figure 2.4), a burst of roughly 50 packets (total of about 500Kb) comprising game video and

Figure 2.5: The setup for console gaming.

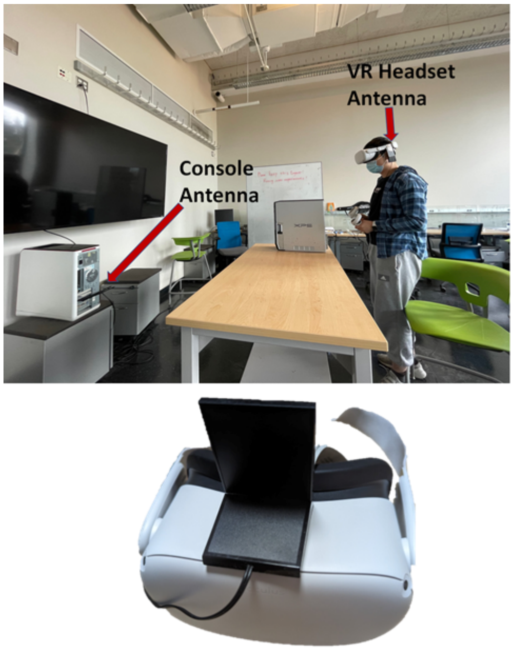


Figure 2.6: The setup for VR gaming.

audio are transmitted once every 16.6ms (corresponding to 60Hz video frame refresh rate). A headset tracking packet (326 bytes) is transmitted every 8ms.

Measurement Methodology. We do not have access to Xbox and Oculus Quest 2 radio firmware for controlled experiments, hence we replay collected game traces between two PCs. We affix antennas on the console/desktop and the controller (since players connect to the headset via an audio jack from the controller) and VR headset (Figure 4.2(a),4.2(b)). We ask the gamers to play a game using the controller/headset with the affixed antenna. This allows the antenna to experience the same head/hand motions as the controller/headset. When the actual gaming traffic is exchanged between the controller/headset and the console/desktop during a gaming session, in parallel, we transmit gaming traffic traces in an interference-free DFS channel [61] between the two PCs. We use PicoScenes toolbox [62, 63] to capture the transmitted packets and measure their CSI and SNR. We use these measurements to compute individual SNRs for each of the 52 Wi-Fi sub-carriers. We consider three scenarios.

- *static* – the controller/headset is static, on a table about 5 feet from the transmitter with

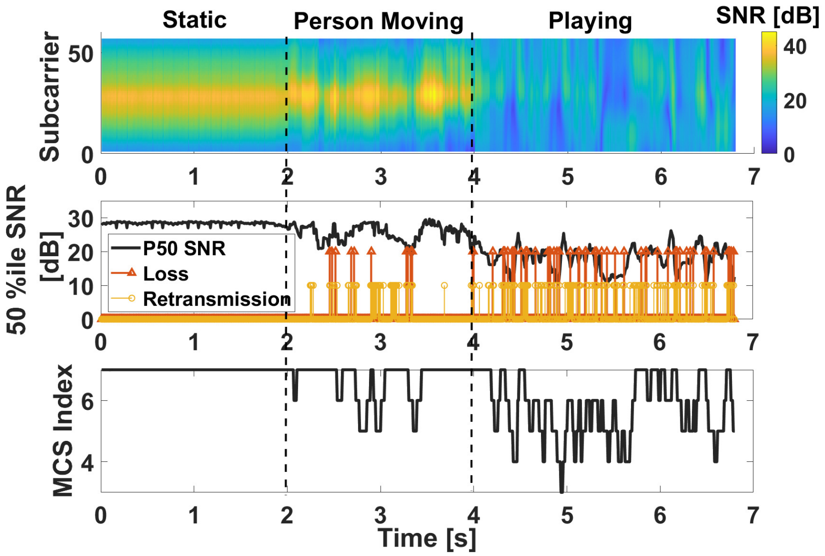


Figure 2.7: Channel, 50%ile SNR, re-transmissions, Losses and MCS for Xbox Traffic.

no one moving.

- *people movement* – the controller/headset is static on a table 5 feet from the transmitter but with a person walking around.
- *game play* – a gamer actively plays using the controller/ headset 5 feet from the transmitter (Figure 4.2(a), 4.2(b)).

Observations. Figure 2.7 depicts the SNR heatmap for each of the 52 Wi-Fi sub-carriers spanning a 20MHz Wi-Fi channel as a function of time for each of the three scenarios. Figure 2.7 also depicts the instantaneous 50%ile SNR across all the sub-carriers. This is based on the intuition that a packet loss in Wi-Fi will occur when a "significant" fraction of sub-carriers experience fading (low SNR) so that error correction is unable to recover the correct bits. We also plot packet re-transmission events and loss events (when the re-transmission fails) and the MCS index (data rate) chosen by the native ADR scheme in an off-the-shelf device.

In the static scenario, the wireless channel is excellent, with all the sub-carriers having an

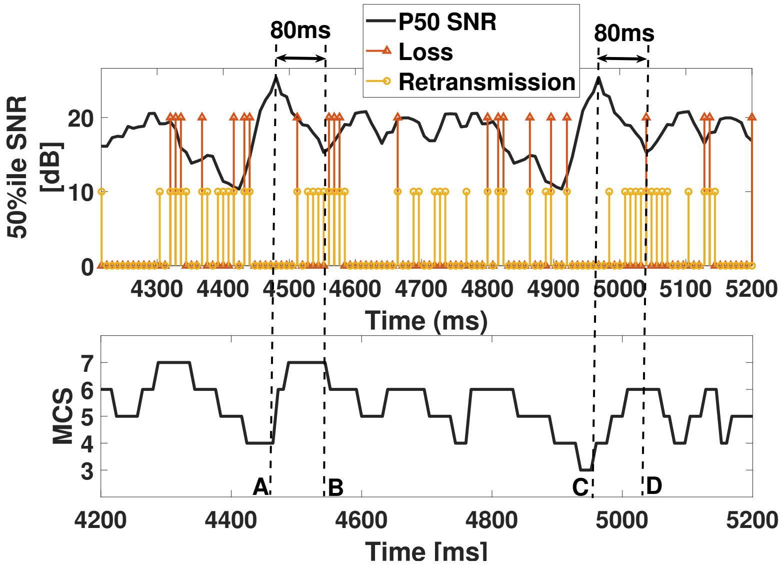


Figure 2.8: Rate adaptation is unable to keep up with rapid channel changes causing a cluster of losses.

SNR of 30`dB` or higher with no losses or re-transmissions. In the people movement scenario, the channel sees variations with time due to changes in multi-path reflections, and occasional packet losses. When the gamer holds the controller or wears the headset and starts playing the game, the channel changes rapidly, leading to a large number of re-transmissions and packet losses. The ADR schemes show significant and frequent variations in MCS Index.

Data rate adaptation is unable to keep-up. Figure 2.8 shows a zoomed 1s section of the wireless channel for the playing scenario between 4.2 to 5.2 seconds. The 50%ile SNR across subcarriers shows a variation of over 15`dB` with a rapid decline of up to 10`dB` within 80ms in sections AB and CD. *The interesting observation in Figure 2.8 is that while the channel is degenerating, as seen by the decreasing trend in the 50%ile SNR in AB and CD, the ADR scheme actually increases data rates (MCS Index) and then starts reducing it when the channel SNR is improving – the exact opposite of the desired behavior.* This is because of the native ADR scheme, which is *reactive*, i.e relies on recent packet re-transmission and loss statistics (e.g. running average) to adapt data rate. Before A, the channel 50%ile SNR

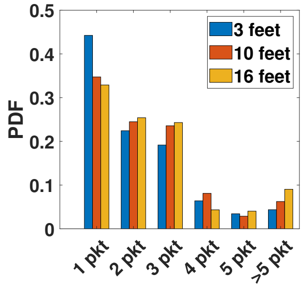


Figure 2.9: Consecutive packet losses due to slow rate adaptation.

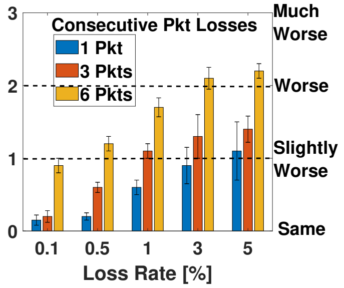


Figure 2.10: Effect of packet losses on game audio.

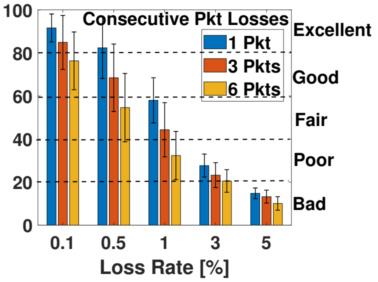


Figure 2.11: Effect of packet losses on game video.

improves by almost 15dB within a matter of 70ms. Based on the running average, after A, as the channel deteriorates, the data rate is increased causing a cluster of packet re-transmissions and losses. After B, even though the channel has an improving trend, data is decreased due to the history of failures.

Consecutive packet losses. As seen from Figure 2.8, due to ADR's reaction lag, packet losses occur in clusters and consecutive packets are lost when channel conditions change rapidly. Figure 2.9 shows the distribution of the number of consecutive losses for different distances between the controller and console. As seen from the figure, about 55-65% of packet losses occur with two or more packets lost consecutively. Our experiments with the VR headset (elided for space constraints) also show a similar cluster of losses.

Effect of Consecutive Packet Losses Prior studies [64–68] have shown that packet loss rate beyond 0.5% deteriorates video quality for online gaming. However, to the best of our knowledge, there has not been a study quantifying the effect of **consecutive** packet losses on video quality. Thus, we conduct a study to measure the effect of consecutive packet losses on gaming audio and video.

Test Data. For our study, we choose 30 Xbox game audio clips each 15s long, and 60 VR game video clips each 5s long as our original data set. These clips were drawn from 15 popular Xbox games and 5 VR games spanning a spectrum of genres such as racing, first-person shooter, strategy, etc. Audio data is streamed as PCM audio packets (similar to Xbox) with 8ms worth of data in each packet. The video data stream is encoded using H.264 video compression. Losses are injected for each stream randomly, parameterized by different loss rates $\rho = \{0.1\%, 0.5\%, 1\%, 5\%, 10\%\}$ and consecutive loss length $b = \{1, 3, 6\}$ packets. This gives us 15 different combinations of $< \rho, b >$. Once a packet is randomly dropped ρ, $b - 1$ following packets are also dropped to capture consecutive losses during rate adaptation.

Measuring Effect on Game Audio We conduct a user study adhering to the Comparative Mean Opinion Score (CMOS) test methodology as dictated by the ITU-T P.808 standard [69]. This work does not raise any ethical issues. Ten unique participants were recruited to listen and compare two audio clips – a lossy and an original clip, without knowing which clip was the original. On a scale from -3 (much worse) to +3 (much better), participants scored the audio quality. During the study, care is taken to sanitize the results for faulty data points by eliminating incoherent outputs, and a calibration step is used to ensure proper audio setup on the participants' computers before the experiment (as per ITU-T P.808). Each combination of $< \rho, b >$ received 300 votes in all (30 clips $\times$10 people). As seen from Figure 2.10, our study reveals that listeners are more sensitive to consecutive packet losses – deterioration becomes perceptible at even for 0.5% loss rates if $b \geq 6$.

Measuring Effect on Game Video To quantify the effect of packet losses on VR video, we use the Video Multimethod Assessment Fusion (VMAF) [70]. VMAF is a popular full-reference objective video quality assessment model developed by Netflix that uses human-vision modeling. VMAF predicts a quality score that ranges from 0 to 100 for each video. Figure 2.11 shows that $\rho \leq 0.5$ and $b \leq 3$ are required to ensure highest perception quality.

Conclusions We summarize the observations in this section as,

- During gaming, the wireless channel varies by 10-15`dB` within a matter of 70-100ms due to gamers' head/hand motions.
- Rate adaptation is unable to keep up with these rapid changes resulting in consecutive packet losses.

- User studies indicate more than 2 consecutive packet losses causes “significant” visual artifacts even at the widely accepted standard of 0.5% packet loss rate.

2.2.2 Channel Measurements

Signal to Noise Ratio (SNR), represented by η in this work, fundamentally determines the data rate at which bits can be successfully sent over any communication channel. A data rate too high causes bits to be lost, and a data rate too low is wasteful in terms of time and energy. Since wireless channel conditions vary significantly over time; Adaptive Data Rate (ADR) techniques aim to adapt data rate to these changes by choosing the most appropriate data rate. The Modulation Coding Scheme (MCS) index, a number between 0 to 9, in Wi-Fi determines the data rate of transmission – the higher the MCS index, the higher the data rate. For each MCS index m, communication theory allows the bit error rate to be calculated analytically using a function $ber_m(\eta)$ [71].

In OFDM [72] modulation, used by Wi-Fi, a 20MHz channel is split into 52 sub-carriers (sub-channels) in 802.11n and 242 sub-carriers in 802.11ax. Bits in a packet are spread out and transmitted over these sub-carriers.

CSI. A radio wave transmitted over the i^{th} sub-carrier undergoes changes in amplitude and phase represented by a complex number c_i. Channel State Information (CSI) collects all these values into a vector $\mathbf{c} =< c_1, \cdots, c_C >$. While all Wi-Fi radios have to necessarily measure CSI for each received packet to decode, not all radio hardware/firmware provides access to CSI information through an API.

RSS. Almost all radios provide Received Signal Strength (RSS) measured in `dBm`, for each received packet – this measures the power level (strength) of a received signal.

SNR. Some radio APIs provide SNR, η (in `dB`) for each packet. SNR and RSS in `dB` are related to each other as [73]

$$\mathbf{SNR} = \mathbf{RSS} - \mathbf{NF} - \mathbf{AGC} \tag{2.8}$$

In Eqn 2.8 NF is the noise floor of the radio and AGC is Automatic Gain Control of the radio which is a dynamic gain introduced by the radio circuit. While some radio APIs provide Noise Floor (in `dBm`) of the radio [74, 75], they do not provide AGC. In practice computing SNR using RSS and NF without accounting for AGC can lead up to a few `dB` of error.

SNR per subcarrier. As prior work [73] demonstrates, SNR is an extremely poor predictor for packet loss due to frequency selective fading, as each sub-carrier experiences a different SNR and hence experiences a different bit-error rate. The success or failure of a packet depends on the aggregate success of all the bits transmitted across all the sub-carriers. The SNR of the k^{th} sub-carrier, η_k can be computed by scaling η by the k^{th} component of unit vector of c,

$$\begin{aligned} \zeta_k &= \frac{C}{\sum_i \|c_i\|^2} \|c_k\| \\ \eta_k &= \eta - 10 \log_{10} \zeta_k \end{aligned} \tag{2.9}$$

Effective SNR (ESNR). Given the ineffectiveness of SNR in channels undergoing frequency selective fading, [73] introduces ESNR η_{esnr}. ESNR summarizes all per-subcarrier values $< \eta_1, \cdots, \eta_C >$ into a single value to represent an equivalent SNR value corresponding to a flat-fading channel (e.g. a conducting wire).

For each MCS index m, [73] suggests computing η_{esnr} analytically by calculating the average bit rate across all the sub-carriers,

$$\eta_{esnr} = ber_d^{-1} \left(\sum_{i=1}^{i=C} ber_d(\eta_i) \right) \tag{2.10}$$

ESNR Computation is Sensitive and Error Prone. Calibration errors and measurement errors of a few `dB` in CSI, RSS, noise floor, or SNR are common in all radios. Further, ber_m functions are exponential (based on the Q function) with very sharp transitions from 0 to 1 within 2-3 `dB`. This makes ESNR computation extremely sensitive to even small errors – 2-3 `dB` error can result over 10`dB` error in effective SNR. Consequently, subsequent work [76] uses an exponentially weighted mean of η_i, which is estimated during an initial calibration phase, rather than relying on Eqn 2.10.

2.2.3 Existing ADR Techniques

In this work, we broadly classify ADR techniques into *reactive* and *proactive* ADR techniques.

Reactive Techniques. These techniques rely on recent packet loss and re-transmission statistics (e.g. using running average estimates). ARF [77], AARF [78], and CARA [79] gradually increase or decrease the rate based on the success or failure of consecutive transmission results. Several rate control algorithms are designed to optimize specific metrics instead of losses only *e.g.* average transmission time [80, 81], frame loss ratio [82–85], bit error rate [86, 87] and throughput [**?**, 88–94]. Recently, a few reactive schemes have used AI-based techniques. For example, NeuRA [95] and MLRA [96] attempt to reduce sampling overhead by using a neural network model to predict the throughput of unsampled data rate. Thompson Sampling [97] uses a multi-armed bandit approach to improve the sampling of different data rates.

Proactive or channel measurement-based techniques. Channel measurements such as RSS, SNR *etc.* can be used to make a more timely choice based on the current state of the channel and offer the promise of faster and more precise rate adaptation. For example, [98] uses RSS while [99–106] uses SNR measurements from received packets to dynamically adjust transmission rates. As discussed in Section 2.2.2 the performance of these schemes is limited by the fact that RSS and SNR do not take into account the effects of frequency selective fading. While ESNR-based techniques [73] account for frequency selective fading, as discussed in Section 2.2.2 their performance is limited due to the sensitivity of ESNR computation to hardware calibration errors and the requirement of offline calibration for each device.

Recently, researchers have demonstrated the potential use of AI in proactive schemes through simulation studies [96, 107–109]. To the best of our knowledge, EDRA [110] is the only existing implemented proactive AI-based ADR scheme. EDRA uses reinforcement learning, aiming to maximize throughput through a joint rate and bandwidth adaptation by using Deep Q-Learning [111] with SNR, loss rates, and service times as inputs. However, EDRA imposes severe computational requirements – even a single inference takes 1.3-3.7 ms on high-performance CPUs like i7-8700 and i5-6200U – failing to meet the requirements of our target embedded gaming devices with low resource constraints.

2.2.4 802.11 be Multi-link Operation

IEEE 802.11be Extremely High Throughput(EHT) [6] is targeted for future home and enterprise hot-spot environments, expected to serve high-density traffic such as high-definition video traffic. Further, emerging applications such as Artificial or Virtual Reality (AR/VR), real-time gaming and edge cloud computing services are well known to require both EHT and end-to-end low latency/high link reliability [112]. Hence, beyond increasing PHY data rates via a combination of higher modulation orders, large channel bandwidths and increasing spatial streams, new MAC features such as multi-link operation (MLO) and multi access point coordination (MAPC)[11] have been proposed.

MLO promotes the simultaneous use of multiple wireless interfaces to allow concurrent data transmission and reception in access points (APs) and stations (STAs), enabled due to dual- or tri-band capabilities. In the context of optimizing hotspots or dense, overlapping cells operation, interference-induced channel loss and channel access collisions occur more frequently, especially when each AP (or cell) has no prior knowledge of other APs (or cell) current network status. Thus, enabling some degree of collaboration among neighboring APs is proposed via new control frames for inter-AP information exchange among a coordination set [11, 113]. Although MLO has been well known to provide higher reliability and latency in the next generation Wi-Fi [114, 115], the sole utilization of MLO is not enough without AP coordination; therefore, a coordinated AP architecture that enables the full potential (EHT, reduced worst-case latency and link stability) of MLO is a must.

A dynamic strategy such as multi-link congestion-aware load balancing (MCAB) has been proposed to periodically adjust the traffic-to-link allocation in order to follow channel occupancy changes. The traffic distribution policy over multiple links investigated in [116–118] has been demonstrated experimentally to improve worst-case latency and network throughput. Even though MCAB is proposed to improve on any existing MLO topology, these works haven not study the link association and selection problem, i.e., intelligent AP-STA pairing and radio link selection. Moreover, the global information of overlapping APs and STAs such as historical throughput, channel condition and the number of STAs each AP can support using MLO. In our paper, we propose an AP-STA pairing and radio link selection for optimized MLO in enterprise-level network architecture. It is noteworthy that the optimization problem is formulated assume the coexistence [118] of Single Link (SL), Multi-link Single Radio (MLSR), and Multi-link Multiple Radio (MLMR).

Performance analysis of synchronous MLO is investigated in [119]. This work has concluded that the longest backoff window and shortest backoff window should be closer to each other when the number of link grows in the network. In our work, on the other hand, we focus on STR MLO and the optimization of the MLO network topology considering multiple APs instead of the backoff window design for the TXOP contention in synchronous MLO with single AP. Experiments for MLO latency have also been implemented in [120] to show the importance of symmetric occupancy of all links can significantly reduce the network latency, which justifies the consideration of fairness in the MLO radio link allocation problem. Experiments for MLO latency have been implemented in [118, 121] that show improvements to worst-case latency and network throughput with simultaneous transmission and reception (STR) mode[6]. Performance analysis of non-STR MLO was investigated in [122]; it concluded that the longest and shortest backoff window should be closer to each other as the number of links (MLO) grows. In [123], the authors propose a dynamic traffic allocation scheme for MLO by considering channel occupancy/loading. However, the PHY state also impacts the latency and reliability, and a better cross-layer formulation is needed to fulfill more stringent latency and reliability requirements.

---

[6] STR is feasible if the multi-link channels chosen have sufficient spectral separation to support in-device coexistence (IDC).

Chapter 3

URLLC IN 5G NETWORKS

3.1 Joint scheduling and resource allocation for URLLC

This section introduces a novel downlink scheduler designed specifically for ultra-reliable and low-latency communications (URLLC). While developing this scheduler, we have meticulously maintained the integrity of the existing enhanced Mobile Broadband (eMBB) proportional fairness (PF) scheduling, ensuring compatibility with current 5G standard-oriented practices. The design of the URLLC scheduler is initiated by transforming the reliability and latency requirements of URLLC into an optimal policy formulation for user selection. This represents the first phase in our scheduler design. The subsequent stage involves formulating the resource allocation process as an integer programming problem. To address this, we propose two solution methodologies - the Constrained Concave Convex Procedure (CCCP) and a greedy algorithm. Both algorithms are adept at achieving proportional fairness among multiple eMBB users while simultaneously maximizing the aggregate data rate of eMBB users post-puncturing. A comprehensive comparative analysis of these two algorithms is conducted, focusing not only on their performance but also their running time. This enables an in-depth understanding of the trade-offs between the two, thereby assisting in the evaluation of their suitability for different scenarios.

3.1.1 System Model

In LTE and NR, frequency-time resources are split into several blocks in the MAC layer - Virtual Resource Block (VRB) - that are subsequently mapped into physical resource blocks

(PRBs)[1]. In this section, we first describe the joint scheduling architecture we aim to construct. Then, the latency budget for PRBs allocated to URLLC users is used to formulate a latency constrained resource allocation problem for each mini-slot, whereby the aggregate utility for eMBB users is maximized.

Review: PF Schedulers for Maximizing eMBB Aggregate Utility in LTE/NR

In 3GPP cellular networks, the MAC scheduler controls the slot-based resource allocation for users based on their respective traffic demands. Three different algorithms are widely used in 4G LTE - round robin, max rate and proportional fair, which have been naturally extended to 5G NR networks. Among those scheduling algorithms, PF scheduler is widely used in cellular networks for the following reason. In any communication network, different users normally experience different channel conditions, so the scheduler shall attempt to exploit the temporal variations by *allocating resources to users with the best conditions for the upcoming time period.* The PF scheduler not only aims to increase the system throughput considering the channel condition but also maintains the long-term allocation proportional fairness between users [124], which is superior to RR and Max Rate scheduler. In this work, we analyze and use the well-known PF scheduler algorithm that is widely adopted by NR and LTE for the downlink scheduling of eMBB users. Denote $x(n)$ the data rate of the eMBB user n and $U_n(x(n))$ the utility of the eMBB user n. From [125], the PF schedule's solution $x(n)$ satisfies

$$\sum_{n \in N} \frac{x(n) - x^*(n)}{x^*(n)} \leq 0, \tag{3.1}$$

where $x^*(n)$ denotes the optimal PF solution and for any other feasible $x(n)$, the aggregate proportional fairness gain is non-positive.

We first present the PF scheduler algorithm used in the downlink of several cellular network standards, and discuss its connection to the network utility maximization formulation.

---

[1] While the mapping details depends on the implementation according to the 3GPP standard [28], different methods have no impact on the scheduling performance as long as we adhere to resource allocation schemes. Therefore we only focus on PRB allocations in this work.

Consider a cellular downlink with N users, $R_t(n)$ denotes the average data rate user n has received up to time t in a window of T time slots according to

$$R_t\left(n\right) = \left(1 - \frac{1}{T}\right) R_{t-1}\left(n\right) + \frac{1}{T} x_{t-1}\left(n\right), \tag{3.2}$$

where $x_{t-1}(n)$ denoting the actual data rate of user n at time $t-1$.

The PF scheduler allocates resources to user n^* who maximizes the ratio of achievable instantaneous data rate over average received data rate, as shown below:

$$n^* = \arg\max \left\{P_t\left(n\right)\right\}, \tag{3.3}$$

where

$$P_t\left(n\right) = \frac{DRC_t\left(n\right)}{R_t\left(n\right)}, \tag{3.4}$$

where $\mathrm{DRC}_t(n)$ denotes the achievable *instantaneous* data rate for user n at time slot t. The scheduler finds the user with the highest $P_t(n)$ to allocate all RBs in the current time slot t. Then, the user corresponding to the second largest $P_t(n)$ is allocated and so on until there are no RBs left in current slot. As example in Fig. 1 shows ongoing eMBB allocation pattern, with orange representing allocation to user n^* (top two rows in the resource block table), blue representing RBs allocated to the user with the second largest $P_t(n)$ etc. The PF algorithm [126] has been shown (see App. 3.3.1) to maximize the aggregate eMBB utility while guaranteeing the asymptotic proportional fairness.

Joint Scheduling Architecture

Two different approaches to joint scheduler designers for eMBB and URLLC users have been conceptualized. The first is to leverage the superposition/puncturing framework and schedule based on different time scales, i.e., NR slot and mini-slot, as shown in Fig. 1.1. The eMBB scheduler decides at each time slot the PRB resources allocated, and the URLLC scheduler allocates the mini-slots resources within the ongoing eMBB slot, based on the eMBB allocation results. The URLLC scheduler receives the URLLC user demand information within the current slot. Then based on the results from eMBB allocation, the URLLC scheduler determines the allocation policy for each mini-slot. Since each mini-slot has a duration of less than 0.125 ms, it implies that the URLLC scheduler algorithm complexity must be for real-time operation. This approach does not impact the current schedule for eMBB traffic but leads to a reduction of eMBB data rate after puncturing. The second approach is to place both user types in the same queue and jointly schedule on the same time scale without puncturing. This will require more scheduling slots and higher numerology to fulfill the URLLC requirements. While this approach could maintain the data rate of eMBB users, it will require a more complex scheduler to fulfill the low latency requirements of URLLC. Further, in the sub-6 GHz bands, the numerology index is restricted to $\{0, 1\}$, implying that scheduling decisions are based on 1 or 0.5 ms slots, limiting the latency constraints achieved for URLLC traffic. In light of the above considerations, we choose to investigate the first approach in this work. Since already allocated eMBB users' PRBs will be punctured to multiplex URLLC users, thus inevitably reducing eMBB users' data rate, the objective of our scheduler is to maximize the aggregate utility of eMBB users while meeting the latency and reliability requirements of URLLC users. Thus we split the URLLC scheduler into two parts - user selection and resource allocation. We first decide which mini-slots can meet the latency and reliability constraints for an arriving URLLC user and then decide how to allocate the URLLC users within these mini-slots so as to maximize the aggregate utility of the eMBB users. The joint scheduler architecture is shown in Fig. 3.1. The Radio Link Control (RLC) layer reports the buffer status at the beginning of the slot/mini-slot to the scheduler. The Physical (PHY) layer will report the channel condition for different users and the selected MCS from the channel quality indicator (CQI)[2] reported from UE. The eMBB scheduler selects users for transmission and allocates resources for them on a slot basis and reports to the URLLC scheduler. Then the URLLC scheduler seeks to send out its

---

[2] There are two types of CQI transmission - periodic and aperiodic - for both eMBB and URLLC. Aperiodic mode causes more delay since BS must request the UE to send their report, so we consider the only periodic transmission. CQI is transmitted periodically with a minimum periodicity of 2 ms - the value in the simulation - specified by a higher layer message. The scheduler uses the latest CQI value for choosing the MCS as per the standard.

scheduled packets in each mini-slot based on the eMBB result and return it to the eMBB users after its allocation. The URLLC scheduler will notify the PHY layer to set the eMBB users' transmission power to zero on the resource allocated to URLLC users as puncturing and notify the eMBB scheduler of the allocation results. If the URLLC scheduler receives the NACK or ACK timeout, it will start retransmission or blind retransmission by immediately scheduling at the current mini-slot.

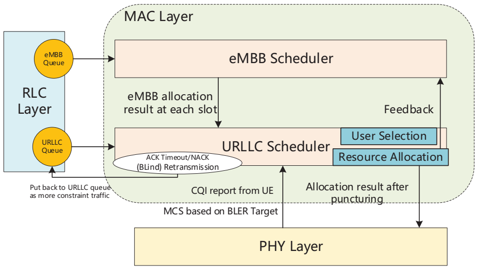


Figure 3.1: Joint scheduler of eMBB and URLLC users

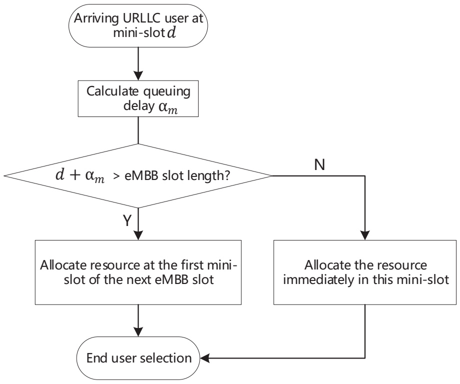


Figure 3.2: The flow chart of the user selection procedure.

Resource Allocation: Maximize eMBB Aggregate Utility Consider PRB level resource allocation for both eMBB and URLLC users. Denote $r(i)$ as the number of PRBs user i needs defined in Eq. (3.5).

$$r(i) = \left\lceil \frac{p(i)}{\psi(i)} \right\rceil, \tag{3.5}$$

where $p(i)$ denotes the packet size of user i, and $\psi(i)$ denotes the data amount carried by this PRB for chosen MCS. Hence $\psi(i) = TTI \times b \times \rho(i)$ where b denotes the bandwidth of the PRB, TTI is the TTI (mini-slot/slot) length of the PRB. $\rho(i)$ denotes the spectral efficiency of chosen MCS of user m with in unit bit/s/Hz. For a selected numerology, TTI and b are the same for all users, so the $\rho(i)$ determines the capacity of the PRB. For each eMBB user, the data rate loss results from the amount of PRBs replaced by the URLLC users in each slot. The eventual data rate of eMBB user n, $\phi(n)$ after the allocation of the URLLC users in a slot is

$$\phi(n) = \left(1 - \frac{\sum\limits_{m \in \mathcal{S}(n,m)} \tilde{r}_n(m)}{r(n)}\right) x(n), \tag{3.6}$$

where $x(n)$ denotes the original data rate of eMBB user n. $x(n)$ will be updated by $\phi(n)$ once the optimal $\phi(n)$ is obtained so that the optimization for the next mini-slot includes the information from the previous mini-slot, $\tilde{r}_n(m)$ denotes that the number of PRBs of URLLC user m overlapping with PRBs of eMBB user n, and $\mathcal{S}(n,m)$ denotes the index set of URLLC users m whose PRBs overlap with PRBs of the eMBB user n. Note that $\phi(n)$ only represents the data rate of the harmed eMBB user n within a given mini-slot. To achieve the proportional fairness for long term, we need to consider the data rate reduction of eMBB user over sufficiently large number of time slots, i.e. average eMBB user data throughput over T time slots is

$$\Phi(n) = \frac{1}{T}\phi(n) + (1 - \frac{1}{T})R_{t-1}(n), \tag{3.7}$$

where the current mini-slot is in time slot t, $R_{t-1}(n)$ denotes the average data rate of eMBB user n in the previous $T-1$ time slots.

Denote the allocation result of each URLLC user m with a row vector $\tilde{\mathbf{x}}_m^T \in \{0,1\}^{1\times L}$ where $\tilde{\mathbf{x}}_{m,l}^T = 1$ means that URLLC user m takes up l^{th} resource block and 0 otherwise. L denotes the total number of PRBs available for scheduling. Then we get the final allocation result in a mini-slot for all M URLLC users as a new matrix $\mathbf{X} \in \{0,1\}^{M\times L}$, expressed as

$$\mathbf{X} = \begin{bmatrix} \tilde{\mathbf{x}}_1^T \\ \vdots \\ \tilde{\mathbf{x}}_m^T \\ \vdots \\ \tilde{\mathbf{x}}_M^T \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \cdots \mathbf{x}_l \cdots \mathbf{x}_L \end{bmatrix}. \tag{3.8}$$

Since in each mini-slot, one URLLC user can be allocated, implies that any column of the matrix $\mathbf{X}$ has at most one "1", i.e., $\mathbf{card}(\mathbf{x}) <= 1$. Since $\mathbf{x}$ is binary $0-1$ vector, it follows that

$$\mathbf{1}^T\mathbf{x}_l \leq 1, \;\; \mathbf{card}(\mathbf{x}) \;\leq\; 1, \; \forall l \in \{1, \ldots, L\}. \tag{3.9}$$

For the PRB allocation for N eMBB users, $\mathbf{Y}$ is defined as follows,

$$\mathbf{Y} = \left[\, \mathbf{y}_1 \cdots \mathbf{y}_n \cdots \mathbf{y}_N \,\right], \tag{3.10}$$

where $\mathbf{Y} \in \{1\}^{1\times L}$. Denote $[0 \ldots \mathbf{y}_n \ldots 0]$ the matrix with all other $\mathbf{y}_i = \mathbf{0}$, $i \in \{1, \ldots, N\} - \{n\}$, i.e., only $\mathbf{y}_n$ remains unchanged and all the other matrices are set to zero. For example, for $\mathbf{Y} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$ where $\mathbf{y}_1 = \begin{bmatrix} 1 \end{bmatrix}$, $\mathbf{y}_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}$, and $\mathbf{y}_3 = \begin{bmatrix} 1 \end{bmatrix}$. $[0\ \mathbf{y}_2\ 0] = [0\ 1\ 1\ 0]$ denotes that 2^{nd} and 3^{rd} resource blocks are taken by eMBB user 2 corresponding to $\mathbf{y}_2 = [1\ 1]$. To quantify the amount eMBB users is harmed by URLLC, we calculate the number of the overlapping part between $\mathbf{y}_n$ and matrix $\mathbf{X}$ for eMBB user n, i.e.,

$$\sum_{m\in\mathcal{S}(n,m)} \tilde{r}_n(m) = \mathrm{Tr}(\mathbf{X}(:, \mathrm{col}([0 \ldots \mathbf{y}_n \ldots 0]))^T(\mathbf{1}_{M\times 1}\mathbf{1}^T_{L\times 1})), \tag{3.11}$$

where $\mathrm{col}(\bullet)$ returns non-zero column indexes of matrix, and $\mathbf{X}(:, \mathrm{col}(\mathbf{Y}))$ returns the matrix corresponding to $\mathrm{col}(\mathbf{Y})$ indices. The data rate after puncturing each eMBB user n is expressed as follows:

$$\phi(n) = (1 - \frac{\mathrm{Tr}(\mathbf{X}(:, \mathrm{col}([0 \ldots \mathbf{y}_n \ldots 0]))^T(\mathbf{1}_{M\times 1}\mathbf{1}^T_{L\times 1}))}{r(n)})x(n). \tag{3.12}$$

Eq. (3.12) is a convex function in $\mathbf{X}$, since $\mathbf{X}(:, \mathrm{col}([0 \ldots \mathbf{y}_n \ldots 0]))$ and $\mathrm{Tr}(\bullet)$ are both linear operation. Then we express the problem as an Integer Program (IP) as shown below.

Problem 1 (Original Resource Allocation Problem)**.**

$$\max \sum_{n=1}^{N} U(\Phi(n)) \tag{3.13}$$

$$s.t.\ (3.9), (3.12),$$

$$\mathbf{X}_{m,l} = \{0, 1\} \;\; \forall m \in \{1, \ldots, M\} \; and \; l \in \{1, \ldots, L\}. \tag{3.14}$$

Note that **Problem** 1 is an IP problem because each entry of $\mathbf{X}$ can only be 0 or 1. The utility function $U(\Phi(n)) = \log \Phi(n)$, for example, when we aim to achieve proportional fairness.

3.1.2 URLLC Resource Allocation in Mini-slot

In this section, we first outline a combinatorial brute-force solution for Eq. (3.13) for URLLC resource allocation in each mini-slot to maximize the eMBB utility functions and quickly move towards a pragmatic solution by reformulating Eq. (3.13) into a difference of convex problem, which can be solved using the concave-convex procedure. Finally, an online greedy algorithm with lower time complexity is proposed based on proportional fairness (PF) scheduler.

Combinatorial Approach

Assume that the number of PRBs for all URLLC in the current mini-slot is less than the number of total PRBs L in the current mini-slot, so we have the following:

$$L - \sum_{m=1}^{M} r(m) > 0. \tag{3.15}$$

Now we analyze the complexity of the brute-force algorithm. If we fix the URLLC PRB allocation area with only the freedom to permute the order of each URLLC PRB, then there are $\{\sum_{m=1}^{M} r(m)\}\boldsymbol{!}$ different permutations of URLLC PRB allocations. Hence, we know the time complexity of such brute-force Algorithm is obviously much larger than $\mathcal{O}\left(\{\sum_{m=1}^{M} r(m)\}\boldsymbol{!}\right)$. Therefore, the brute-force ordering algorithm is an NP-hard problem.

Convex Relaxation

In this section, we reformulate the natural Integer Programming (IP) problem as equivalent to difference of convex (DC) programming that can be solved with an exact solution exploiting concave-convex procedure (CCCP) efficiently. First, note that an equivalent constraint to Eq.(3.14) is

$$\begin{aligned} &\mathbf{X}_{m,l}(1 - \mathbf{X}_{m,l}) \leq 0, \mathbf{X}_{m,l} \in [0,1], \\ &\forall\, m \in \{1, \dots, M\}, \forall\, l \in \{1, \dots, L\}. \end{aligned} \tag{3.16}$$

Problem 2 (Equivalent Resource allocation Problem)**.**

$$\max \sum_{n=1}^{N} U(\Phi(n))$$
$$s.t.\ (3.9), (3.12), (3.16).$$

Note that while it is now cast as a problem over continuous variables, Problem 2 is still not a convex optimization problem since constraint (3.16) represents a disconnected set. To be more specific, $\mathbf{X}_{m,l}(1-\mathbf{X}_{m,l})$ is a concave function, so any sub-set is obviously a disconnected set.

Next in the interests of tractability, we further transform the above by penalizing the concave constraint into the objective function.

Problem 3 (Penalized Resource Allocation Optimization)**.**

$$\begin{aligned} &\min -\sum_{n=1}^{N} U(\Phi(n)) + \zeta \sum_{m=1}^{M}\sum_{l=1}^{L} \mathbf{X}_{m,l}(1-\mathbf{X}_{m,l}) \\ &s.t.\ (3.9), (3.12) \end{aligned} \tag{3.17}$$

where we introduced a penalty parameter $\zeta > 0$. The above function can be visualized as a linearly constrained indefinite quadratic problem (IQP) which falls into the category of different of convex (DC) problem. The following lemma shows the equivalency of the DC problem to the original problem [127].

Lemma 1. *If* $u > \zeta_0$ *where*

$$\zeta_0 = \frac{-\sum_{n=1}^{N} U(\Phi(n)) - \gamma}{max_{\mathbf{X}}\{\sum_{m=1}^{M}\sum_{l=1}^{L} \mathbf{X}_{m,l}(\mathbf{X}_{m,l}-1) : (3.9), (3.12)\}}, \tag{3.18}$$

where

$$\gamma = min_{\mathbf{X}}\{-\sum_{n=1}^{N} U(\Phi(n)) : \mathbf{X}_{m,l}(1-\mathbf{X}_{m,l}) \geq 0\}, \tag{3.19}$$

then Problem 2 is equivalent to Problem 3.

Algorithm 1 CCCP Algorithm

```
**Initialize** Find an initial feasible matrix $\mathbf{X}^{(0)}$ of Problem 3 and set $i = 0$
**repeat**
    Solve for $\mathbf{X}^{(i+1)}$ as the optimal solution in the following convex problem:
        $$\begin{aligned} &\min\{-\sum_{n=1}^{N} U(\Phi(n)) - \zeta \mathbf{F}(\mathbf{X}; \mathbf{X}^{(i)}))\} \\ &s.t. (3.9), (3.12) \end{aligned} \quad (3.20)$$
    Where $\mathbf{F}(\mathbf{X}; \mathbf{X}^{(i)}) \triangleq \sum_{l=1}^{L} \sum_{m=1}^{M} \mathbf{X}_{m,l}^{(i)}(1 - \mathbf{X}_{m,l}^{(i)}) + \sum_{l=1}^{L} \sum_{m=1}^{M} (\mathbf{X}_{m,l} - \mathbf{X}_{m,l}^{(i)})(1 - 2\mathbf{X}_{m,l}^{(i)})$.
    $i \leftarrow i + 1$
**until** converge to acceptable error $\epsilon$
```

It is noteworthy that $\mathbf{X}_{m,l}(1 - \mathbf{X}_{m,l}) \geq 0$ is a convex constraint. The local optima of Problem 3 can be obtained in finite steps via DC algorithms (DCA). Moreover, the objective function in Problem 3 is differentiable, so DCA is naturally reduced to CCCP [128], as shown in Algorithm 1. CCCP iteratively solves convex problems obtained by linearizing the penalty term of the objective function in IQP. Taking derivative of the penalty term gives us $1 - 2\mathbf{X}_{m,l}$, then we apply first-order linear approximation and we obtain $\mathbf{X}_{m,l}^{(t)}(1 - \mathbf{X}_{m,l}^{(t)}) + (1 - 2\mathbf{X}_{m,l})(\mathbf{X}_{m,l} - \mathbf{X}_{m,l}^{(t)})$, where $\mathbf{X}_{m,l}^{(t)}$ denotes the solution of t^{th} CCCP iteration. The time complexity of each iteration of the CCCP algorithm is lower bounded by $\mathcal{O}((ML)^{2+\frac{1}{6}} \log(\frac{ML}{e}))$ [129], where e denotes the relative accuracy.

Greedy Algorithm Although the above CCCP indeed yields a near-optimal solution, the computation efficiency may not suffice to output the solution within each mini-slot. In order to solve the resource allocation more efficiently, we propose a greedy algorithm 2, which maximizes the aggregate utility for each arriving users inspired by the PF scheduler shown in Algorithm. 2. The complexity of this algorithm is $\mathcal{O}(MLN)$. We use the same formula to

Algorithm 2 Greedy Algorithm using PF scheduler in each mini-slot. $R_t(n)$ denotes the low-pass filtered average data-rate user n has received up to time t.

```
$\phi(n) \leftarrow x(n)$
if packets of URLLC user $m$ arrival then
    for each PRB of URLLC user $m$ do
        $R_t(n) \leftarrow \left(1 - \frac{1}{T}\right) R_{t-1}(n) + \frac{1}{T}\phi(n),$    (3.21)
        where $T$ is the number of time slots we do average for the data rate, normally $T = 100$.
        $n^* \leftarrow \arg\max \{P_t(n)\},$    (3.22)
        where
        $P_t(n) = \frac{\phi(n)}{R_t(n)},$    (3.23)
        Puncture eMBB $n^*$ with the current PRB of URLLC user and the data rate of the eMBB user is updated as follows:
        $\phi(n^*) \leftarrow \phi(n^*) - \frac{1}{r(n^*)}\phi(n^*).$    (3.24)
Set the $\phi(n)$ as the original data rate $x(n)$ in the next mini-slot, i.e., $x(n) \leftarrow \phi(n)$
```

calculate the PF weight $P(n)$ in Eq. (3.4) for eMBB users. The main difference between the traditional PF algorithm and our proposed greedy algorithm is that the traditional PF will sort all eMBB users according to the $P(n)$ and then sequentially allocate one by one. In our proposed greedy algorithm based on PF scheduler, we allocate the RB requested by URLLC one-by-one by puncturing the RBs which has already been allocated to the existing eMBB users, by choosing those that have the least impact on original $P(n)$. It is noteworthy that, after each allocation, the $P(n)$ varies in response to the changes of eMBB users' data rates, which in turn will lead to different allocation decisions for the future RBs.

3.1.3 Simulations

In this section, we analyze the performance of the basic RR, CCCP and dynamic greedy algorithm for downlink URLLC scheduling [3]. Basic RR, as a baseline algorithm, only allocates PRBs from top to bottom within each mini-slot, so it does not consider the impact upon the fairness among eMBB users. RR algorithm also does not help maximize the eMBB users' data rate since it does not consider which eMBB is more susceptible to data rate reduction from URLLC users' PRBs. The trade-off between the CCCP algorithm and the dynamic greedy algorithm is analyzed in terms of the performance and time efficiency.

Simulation Setup

The simulation is implemented on a computer with a single CPU (Intel(R) Core(TM) i7-7700K) and implemented using MATLAB R2019b with the CVX toolbox. To accelerate overall runtime, we implement the greedy algorithm in C on the same machine. We consider a system with a total of 120 PRBs (60 RBGs) per eMBB slot, with seven mini-slots per eMBB slot and numerology 1. The system has five eMBB users, and the RBG numbers needed to achieve their maximum DRC are randomly selected as {10, 12, 12, 16, 18}. We assume these remain the same during each simulation such that, for example, eMBB user one always request 10 RBGs to achieve its maximum DRC in each slot. Note that the total number of RBGs requested by the eMBB users is larger than the system can afford. Hence, in order to achieve the long-time fairness among eMBB users, we apply the PF principle that forces the eMBB scheduler to choose (out of 60 RBGs), which chunk of RBGs is going to be occupied by which eMBB user.

Cases: Different channel conditions for the eMBB users are used to demarcate cases. In Case 1, all the eMBB users have same channel condition, $\psi = 2$ Kbits, i.e., each RBG carries 2 Kbits data. In case 2, the channel condition varies - User 1 has the best channel condition - $\psi(1) = 3$ Kbits - while users 2 & 3 remains the same as case 1, i.e., $\psi(2) = \psi(3) = 2$ Kbits. User 4 and user 5 have worse channel condition such that $\psi(4) = 1.75$ Kbits and $\psi(5) = 1.5$

---

[3] Note that the basic RR algorithm is not applied to eMBB but URLLC scheduling.

Table 3.1: Simulation Parameters

Parameters	**Value**		
eMBB Users	Numbers		5
	$\psi(n)$ (Kbits)	Case1	{2,2,2,2,2}
		Case2	{3,2,2,1.75,1.5}
	RBG Numbers Request Each slot		{10, 12, 12, 16, 18}
	Scheduler		Proportional Fair
URLLC users	RBG Numbers Request Each Mini-Slot		uniformly distribution from {1,2,3}
System	Total PRB Numbers		120 (60 RBGs)
	Numerology		1
	Mini-Slot Symbol Number		2

Table 3.2: Simulation Parameters in ns-3

Parameter	**Value**
Power	20 dBm
Numerology	1
Bandwidth	100MHz
Center Frequency	28 GHz
Environment	3GPP Urban Macro(UMa)
Moving Module	2D Random Walk
Speed	Uniform distributed from 4 m/s to 8 m/s
Initial UE distance to BS	{100, 200, 300, 400 ,500} m

Kbits. DRC in unit of Mbps for each user can be expressed as follows:

$$DRC(n) = \frac{r(n) \times \psi(n)}{2 \times 0.5 \times 10^{-3}\ \mathrm{s}}, \tag{3.25}$$

where $r(n)$ is the PRB numbers (thus $\frac{r(n)}{2}$ is the RBG numbers) requested by eMBB user n and the slot length in our simulation is 0.5×10^{-3} s.

The RBG numbers required by URLLC users are uniformly distributed with the mean value of 2 RBGs from the set {1, 2, 3} RBGs. In this setup, the simulation explores the extent to which the number of URLLC users impacts the performance of three algorithms for simulation parameters in Table 3.1.

URLLC User Selection: Latency and Reliability

For the current setting with Numerology 1, slot length 0.5 ms and mini-slot length equals to 2 OFDMA symbols, the TTI quals to 0.0714 ms for URLLC users and processing time coefficient β equals to 4. Each simulation run duration is 200s and is repeated a total of 100 times to acquire statistics. The QoS requirements for downlink URLLC traffic is modeled as follows: a UE requests the BS to transmit successfully within an end-to-end delay of τ and the probability of failure at most σ. In the simulation, we consider the cases with different τ and calculate the corresponding failure probability. We conduct the simulation of 10 URLLC users arriving in each mini-slot versus different BLER target configuration from 10^{-5} to 10^{-1}. The percentage of the transmission failure within 1ms delay versus different BLER settings is plotted to verify how well the user selection performs to meet the URLLC QoS

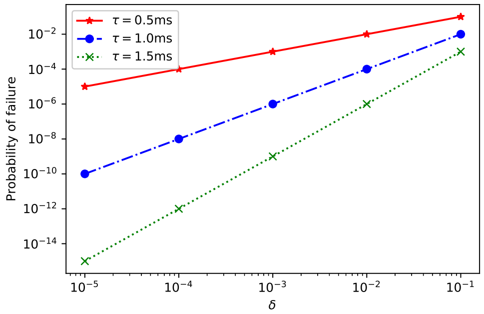


Figure 3.3: URLLC Users' probability of failure with latency less than τ ms vs. BLER target δ.

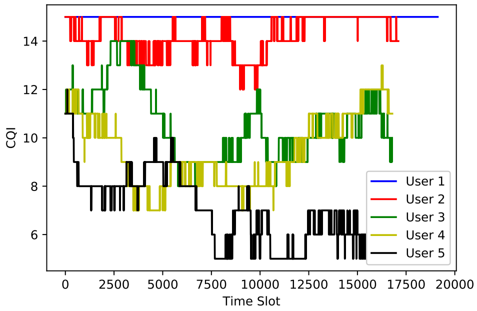


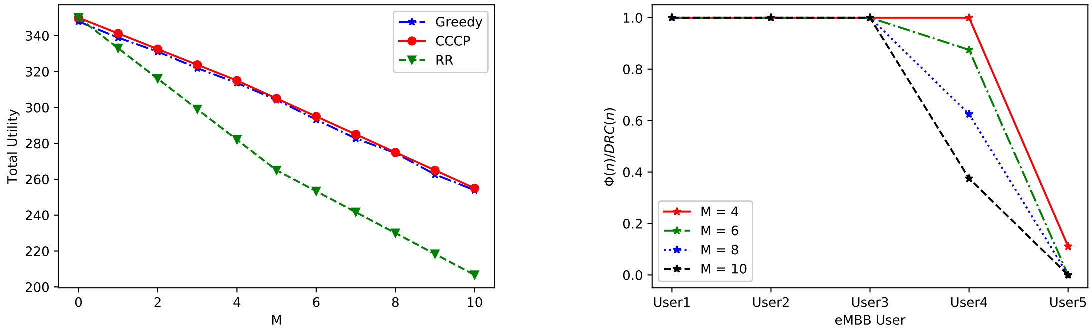
Figure 3.4: Illustration of the CQI value for different eMBB users during the simulation

requirements. The results for the URLLC constraint is shown in Fig. 3.3; it is observed that the BLER target impacts the final result of the reliability within τ ms latency constraint. For the current numerology setting (0.5 ms slot), the maximum transmission and retransmission times k_{trans} are $\{1, 2, 3\}$ for $\tau = \{0.5, 1, 1.5\}$ ms respectively. As the BLER target σ increases, the reliability drops, and more retransmission is required, increasing the latency. It cannot fulfill the requirements ($\sigma \leq 10^{-5}$) after $\delta \geq 10^{-1}$ for all different τ, even if all the packets are immediately scheduled in the current setting.

URLLC Resource Allocation

In this section, we first present a simple case with the objective of maximizing the aggregated throughput of eMBB users, i.e., $U(\Phi(n)) = \Phi(n)$ with the same setting as in case 2 above. Fig. 3.5 shows that using greedy algorithm and CCCP algorithm yield (nearly) the same result, because with this objective function, the scheduler only needs to select the user with the worst channel condition (user with the smallest $\psi(n)$) with the least harm. As the number of URLLC users per mini-slot M increases, the last two users see throughput drop (even to 0), hence there is no fairness between users.

Proportional Fairness: Next we simulate the 2 different cases mentioned above with the



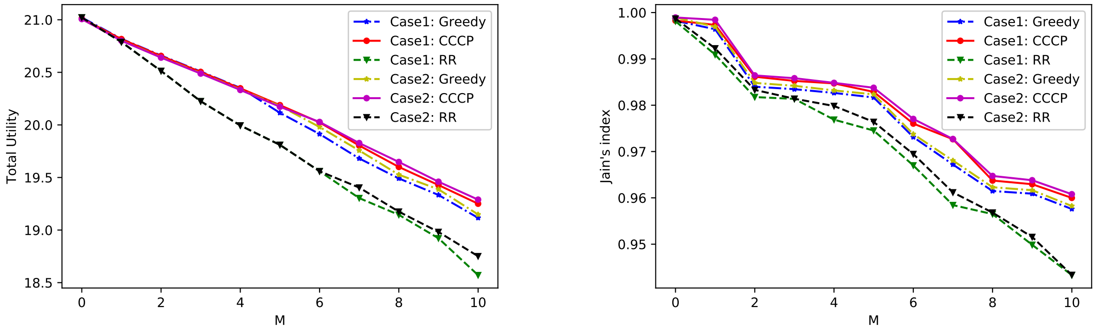
(a) Total utility vs URLLC users M per mini-slot.

(b) The $\frac{\Phi(n)}{DRC(n)}$ of each eMBB user.

Figure 3.5: Simulation result with the objective of maximizing total rate, i.e., $U(\Phi(n)) = \Phi(n)$.

objective to achieve the proportional fairness, i.e., $U(\Phi(n)) = \log\big(\Phi(n)\big)$ with 200s simulation run durations and compare the results of different algorithms proposed in Section 3.1.2. In Fig. 3.6(a), eventual aggregate eMBB utility affected by URLLC allocation versus the number of URLLC users are plotted. As the number of URLLC users increases, the utility drops for all RR, CCCP, and greedy algorithms. The performance of the CCCP algorithm is the best, especially when the density of the URLLC network increases. For the RR algorithm, the same eMBB users are constantly punctured because RR always punctures eMBB users from top to bottom (from orange to green PRBs) in Fig. 1.1. In other words, no matter how high the priority, the first user is punctured by the RR algorithm. Hence, there is no fairness, and the total utility drops, as expected in Fig. 3.6(a). We conclude that the RR algorithm is not suitable for the URLLC scheduler, although it is useful in eMBB scheduling.

The CCCP and greedy algorithms consider fairness among the eMBB users and have much better performance than the RR algorithm in general. However, greedy algorithms decide on a single PRB allocation at a time, given all URLLC demands for PRBs. Hence, its performance is compromised compared to CCCP, which makes decisions based on all information of requested URLLC PRBs. In Fig. 3.6(a), we observe that the slope of CCCP does not change as the number of URLLC users increases while the slope for the greedy algorithm drops slightly, supporting the previous analysis. The fairness among eMBB users is calculated based on the Jain's fairness index and shown in Fig. 3.6(b); increasing the URLLC numbers per mini-slot leads to greater unfairness among the eMBB users and hence reduces Jain's index.

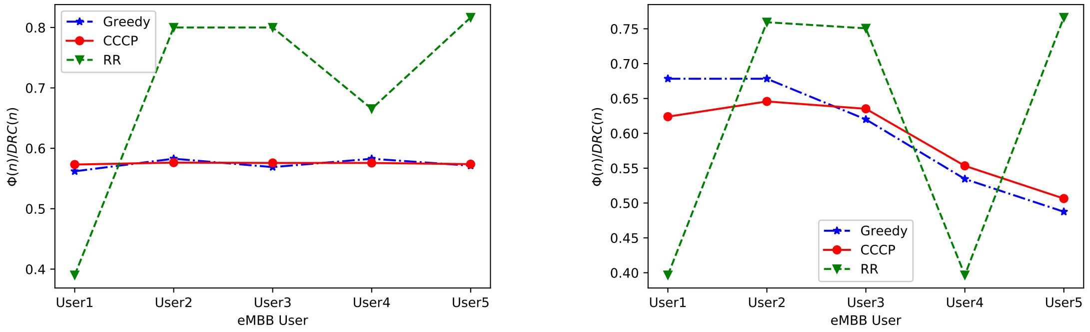


(a) Total utility $\sum_{n=1}^{N} U(\Phi(n))$

(b) Jain's fairness index

Figure 3.6: Total utility and Jain's fairness index vs the number of URLLC user numbers per mini-slot. The difference between the two cases is in case 2, $\psi(n)$ varies among eMBB users while $\psi(n)$ is equal for all eMBB users. M denotes the total URLLC numbers in each mini-slot.

Another approach to proportional fairness for eMBB users is the observation that the average throughput of the eMBB users is proportional to its DRC, as in Eq. (3.44). Thus $\frac{\Phi(n)}{DRC(n)}$ is computed after each 200s simulation run when the number of URLLC users arriving at each mini-slot is 8. The results are shown in Fig. 3.7(a) and Fig. 3.7(b). In case 1, $\psi(n)$ of each eMBB user n is the same according to Tab. 3.1, i.e., the channel condition is the same for all users, and the resulting loss is only related to how many PRBs of the eMBB user is punctured. After the allocation of URLLC users, the proportion remains constant between eMBB users using the CCCP algorithm. The greedy algorithm achieves nearly as a good PF metric as CCCP even though the greedy algorithm only allocates one PRB at a time, while RR achieves no fairness.



(a) Case1

(b) Case2

Figure 3.7: The $\frac{\Phi(n)}{DRC(n)}$ of each eMBB, $M = 8$ per mini-slot.

Case 2 in Fig. 3.7(b) is more complex as the $\psi(n)$ varies for each eMBB user according to Tab. 3.1. As we can observe, the CCCP outperforms the greedy algorithm but is unable to achieve ideal PF (constant). This is understandable for the following reason: each eMBB user takes a *different* number of RBGs per mini-slot. If there is no URLLC user, then the throughput of each eMBB user is always proportional to DRC. However, in the presence of URLLC user, to obtain the perfect proportional fairness, i.e., $\frac{\Phi(1)}{DRC(1)} = \cdots = \frac{\Phi(n)}{DRC(n)} = \cdots = \frac{\Phi(N)}{DRC(N)} = c$, we must have $\Phi(n) = c \times DRC(n)$. Note that in this case, ideal PF (constant value for eMBB users) is not achieved, although the Fairness index is considerably better than RR. Since the original problem is naturally an integer programming, i.e. the punctured RBs are integers, we cannot achieve the desired ideal PF solution obtained over the continuous real domain, and is the fundamental reason for the approximate PF values achieved.

The previous two cases illustrate that the average throughput is proportional to DRC, given a fixed ψ during the simulation. For more realistic simulation, we use the ns-3 NR model [130] to generate the various channel condition and corresponding CQI value for eMBB users. We then evaluate our proposed scheduler via simulation using this approach; the major parameters for the channel model in ns-3 are summarized in Table 3.3. The 3GPP recommended millimeter wave (mmWave) propagation loss model [131] is used where eMBB users are attached to BS at different distances and indulge in a random walk[4] from their initial location during the simulation. We use the periodical CQI feedback (2 ms) and then choose the MCS based on the CQI value. The CQI values for different eMBB users are illustrated in Fig. 3.4. The URLLC traffic and other settings remain the same as case 1 and case 2.

The results in Fig. 3.8 demonstrate that as we increase the URLLC number to the limit the system can afford (up to 30 in current setting[5]), all the resources are occupied by the URLLC. When the URLLC traffic load exceeds the maximum resources the system can provide, the utility of the eMBB users drop to 0. While the eMBB users still get scheduled, their actual throughput is 0, so the priority among the eMBB users and the Jain's fairness index remains the same. The average latency for 99.999% reliability is shown in Fig. 3.8(c). As the URLLC number increases, the complexity of RR and greedy algorithms increase only marginally and the resulting latency is mainly dependent on the mini-slot that is scheduled. However, the complexity of the CCCP algorithm increases rapidly, which results in longer execution times and increases the total latency. For URLLC numbers beyond 30, all URLLC users could not be scheduled to meet the latency requirement of 1 ms; the latency is caused mainly by the queuing rather than the URLLC scheduling delay, hence the change of the latency after reaching the maximum of 30 in Fig. 3.8(c) behaves the same for different schedulers.

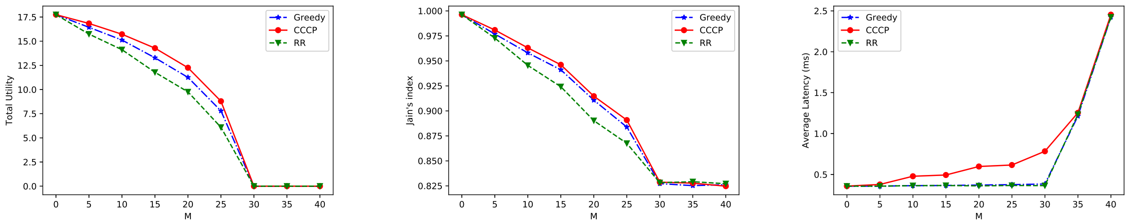
Algorithm Run times

We implemented the greedy algorithm using C and measured runtime for a single execution thread, for 10000 Monte Carlo simulations with different URLLC users in each mini-slot.

---

[4] We use the ns-3 Random Walk 2d-Mobility Model. Each UE moves with a speed (uniformly distributed from 2 m/s to 4 m/s) and a direction (the radians are uniformly distributed from 0.0 to 6.283184). When the UE hits the boundary (15m from the start point), it will change the direction and speed again.

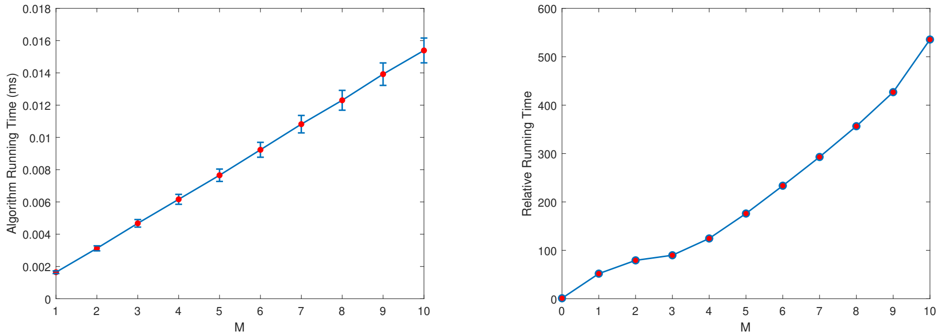
[5] The total number of RBG in the simulation is 60, and each URLLC user attempts to take up 2 RBGs each mini-slot. So when the URLLC numbers beyond 30, all the RBGs are taken by URLLC users.



(a) Total utility (b) Jain's fairness index (c) Average latency

Figure 3.8: Total utility, Jain's fairness index and average latency vs the number of URLLC user numbers per mini-slot for various CQI.

The average run time of the greedy algorithm is shown in Fig. 3.9(a). Because the CCCP algorithm invokes the MATLAB CVX toolkit, we also implemented the greedy algorithm in MATLAB to provide a logical comparative baseline for the run times between the greedy and CCCP algorithm. As shown in Sec. IV, the time complexity of greedy algorithm is $\mathcal{O}(MLN)$ and the CCCP algorithm $\mathcal{O}((ML)^{2+\frac{1}{6}}\log(\frac{ML}{e}))$. We can conclude that the CCCP algorithm has a higher order than the greedy algorithm in terms of the time complexity, which is also corroborated by Fig. 3.9(a). Hence, we summarize the following trade-off: when the number of URLLC users is large, CCCP has better performance but longer run times. On the other hand, the greedy algorithm can time efficiently allocate URLLC users' PRBs with some performance sacrifices. The choice of CCCP vs. greedy algorithm also depends on the performance of the embedded system employed and use case latency requirements. For less strict latency constrained scenarios implemented on embedded systems with a fast CPU, the CCCP algorithm may be uniformly preferred for its performance. If the latency constraint is stringent, and the embedded system has a constrained CPU, the greedy algorithm is likely to be preferred.



(a) Running Time of greedy Algorithm implemented in C for case 2.

(b) Relative Running time for CCCP and greedy Algorithm for case 2.

Figure 3.9: Running time comparison of the 2 algorithms

3.2 Routing and Resource Allocation for IAB Multi-Hop Network

In this section, we delve into the intricate process of cross-layer modeling of routing and resource allocation within a multi-hop Integrated Access and Backhaul (IAB) network under the contemporary 5G New Radio (NR) standard, all in the pursuit of achieving the stringent requirements of URLLC. We commence by formulating and analyzing an optimal routing problem aimed at minimizing transmission latency while concurrently satisfying the transmission reliability constraint. Subsequently, we introduce a Deep Reinforcement Learning (DRL) framework designed to address the proposed routing and resource allocation challenges within the IAB network, all based on local information. Advancing further, we propose a cutting-edge entropy-based reinforcement learning algorithm, integrated with a federated learning mechanism. This unique approach is designed to both expedite the convergence speed and decrease the algorithmic complexity, striking a balance between efficiency and simplicity. Lastly, we substantiate our methodology by presenting numerical results. The evidence demonstrates that our proposed algorithm not only stands up to, but also surpasses existing algorithms, outshining them in terms of both latency and reliability from diverse perspectives. This empirical validation strengthens the credibility of our approach and highlights its potential for application in real-world scenarios.

3.2.1 System Model

In this section, we introduce the overall system model for the multi-hop resource allocation problem.

Multi-hop forwarding and routing with BAP layer

The multi-hop forwarding is newly enabled via the IAB-specific BAP, inserted as a specific header in the RAN layer 2 stack. Consider a general IAB network. Several IAB nodes are transmitting backhaul traffic to the IAB-donor over the wireless link and then forward to the core network. The IAB-donor assigns a unique L2 address (BAP address) to each IAB node that it controls. After the initialization, the IAB donor will know the existing IAB nodes inside its network. For the transmission between the IAB nodes and IAB donor, the BAP header will include the source and destination ID as well as an optional path ID. Each IAB node has its routing table (configured by the IAB donor) containing the next hop identifier for each BAP ID. The routing tables for the downlink (DL) and uplink (UL) directions can be different, used by DU or MT parts separately.

Features for IAB networks on PHY, MAC and RLC layer

The physical layer of IAB is intended to support in-band backhauling with the same carrier frequencies for both the NR backhaul links and the access links. The in-band operation comes with a half-duplex constraint, implying that the IAB-MT part of an IAB node cannot receive while its collocated DU is transmitting and vice versa to avoid intra-site interference. Therefore, a strict time-domain separation is required between transmission and reception phases within each IAB node. At the MAC layer, the IAB-nodes support flexible resource allocation for both DL and UL, which is thus similar to the normal UE allocation. An IAB network attempts to schedule the wireless resources to meet each UE bearer's requirement regardless of the number of hops a given UE is away from the Donor DU. The scheduler on the wireless backhaul link can distinguish the Quality of Service (QoS) profiles associated with different RLC channels. It may also apply information regarding the number of hops a packet needs to traverse, in addition to the QoS profile of the bearers, in order to provide hop-agnostic performance. Backhaul (BH) channel is a logistic mapping for transporting packets between IAB nodes/donor. Different packets from UEs will map to a single BH RLC channel that is established only between two IAB entities.

With the features provided by the IAB network, we can design our own routing and resource allocation algorithms. However, wireless backhaul links are vulnerable to blockage, e.g., due to moving objects such as vehicles, seasonal changes (foliage), or infrastructure changes (new buildings). Such vulnerability also applies to physically stationary IAB-nodes. In addition, traffic variations can create uneven load distribution on wireless backhaul links, leading to local link or node congestion [21]. Therefore, topology adaptation for physically fixed relays shall be supported to enable the robust operation and dynamic routing, which is still a challenge for the IAB networks.

Node management and Routing

In the IAB routing mechanism, each IAB node is assigned a unique address (BAP address), and the IAB donor CU configures a routing table at each IAB node to direct the flow of traffic based on these node addresses. A mechanism is established within the IAB network to help forward it via multiple intermediate IAB nodes between the IAB donor and a specific UE from a packet perspective. It includes the route selection and the next-hop destination at each IAB node once a route is selected. However, due to the multi-hop nature of IAB networks, the backhaul link failure may occur on intermediate IAB nodes along a transmission path which is caused by the differences in their effective link capacities (i.e., different SINRs). In addition, high latency will also be incurred because of the different congestion conditions on intermediate IAB nodes. Therefore, an optimal route needs to be selected between the IAB donor and the specific UE in the IAB networks based on the reliability and latency requirements. Note that each IAB is only able to acquire the local information regarding all its nearby IAB nodes; by contrast, IAB donor obtains the information from all IAB nodes in an IAB network. When an IAB node obtains new local information, the relevant routing decisions (routing tables) will be globally renewed/reconfigured by the IAB donor. Meanwhile, in order to avoid congestion-related packet drops among the IAB-nodes, the routing within IAB networks is also supported for both UL and DL directions which can happen on different nodes and links between the IAB donor and a specific UE. Although the link failure or the congestion problem can be handled by higher-layer protocols, e.g., Transmission Control Protocol (TCP), the scope of the impacted nodes will extend well beyond the RAN/IAB network. Furthermore, if packets are dropped due to congestion in the IAB network, the TCP congestion avoidance and slow start mechanisms may be triggered, and the end-to-end performances can be significantly impaired.

Deep Reinforcement Learning (DRL) is quite an appropriate approach to solve the multi-hop transmission problem to cope with the possible issues above. The main reason is that each agent (node) in the IAB network can be constructed to find a route that maximizes the expected discounted reward through interaction with the real-time environment. Under the DRL algorithm, the selected route(s) in the UL/DL direction can consider both the latency and reliability requirements. Meanwhile, since such optimal route(s) can be achieved with a fast convergence speed, the DRL approach is still applicable when some changes happen abruptly in the IAB network, e.g., the leaving or coming UEs or the change of IAB network topology.

Routing and Graph Model Without loss of generality, the connection and links between different devices could be modeled as a graph. In this graph $G = (\mathcal{V}, E)$, the vertex $\mathcal{V}$ represents the network devices (nodes) and the edges E represent the communication links between the pairs of network nodes. Each node is aware of the connection thereof to neighbour nodes and associated channel condition, and then according to the configuration, it can obtain the p_b in the link. We first consider the routing problem, that the destination node (serving as a controller) have the perfect knowledge of the network topology and the channel condition. A further assumption is that there is no hidden terminal problem and the edge nodes are far enough not to interfere with each other thus there is no collision due to the RB selection, i.e., $p_c = 0$. Then this could be formulated as a problem of finding the path q of minimum latency for a given reliability constraint σ. Notably, the retransmission procedure can improve the reliability, thus, the formulation is conducted to minimum latency as the objective and find a path from the set $\mathcal{Q}$ of all paths connecting Source (S) to Destination (D) for a given MCS.

Problem 4 (Optimal Routing Problem)**.**

$$\begin{aligned} \min_{q\in\mathcal{Q}} \quad & T_{delay}(q) \\ s.t.\ & \mathbb{P}(q) \geq \sigma, \end{aligned} \tag{3.26}$$

The constrained optimization problem in (3.26) can be transformed into an unconstrained problem by applying the Lagrange multiplier method and expressions as follows with immediately scheduling (no queuing delay):

$$\begin{aligned} \min_{q\in\mathcal{Q}} \quad & T_{\text{delay}}(q) - \mu \log(\mathbb{P}(q)) \\ & = T_{\text{proc}} + \sum_{i\in q}(\frac{1}{2}T_{\text{proc}} + T_{\text{trans}}(i) + \mu \cdot \log(\frac{1}{p_s(i)}) = C + \sum_{i\in q} c(i,\mu), \end{aligned} \tag{3.27}$$

where $\mu \geq 0$ is the Lagrange multiplier, C is a constant and $c(i,\mu) \triangleq \frac{1}{2}T_{\text{proc}} + T_{\text{trans}}(i) + \mu \cdot \log(\frac{1}{p_s(i)})$.

In terms of a particular Lagrange multiplier μ, the optimal path q^* can be obtained by the Dijkstra's algorithm [126] by setting $c(i,\mu)$ as the weight of each link. Then the problem is reduced to finding a solution of μ^* by the following lemma and algorithm.

Lemma 2. *The optimal $q^*(\mu)$ solves the problem 4, when there exists a μ that achieves $q^*(\mu) = \sigma$.*

Proof. Please see the proof in App. 3.3.3. □

Lemma 2 introduces a way to obtain the optimal μ^* by finding the one that satisfies $\mathbb{P}\left(q^*(\mu)\right) = \sigma$. A bisection approach is employed with exponential convergence rate to find the optimal μ^* as shown in Algorithm 3 in App. 3.3.2.

For each link i, the $T_{\text{trans}}(i)$ is also related to the MCS choice from the possible set M_{cs}. The weight now used in the unconstrained optimization (3.27) become

$$\tilde{c}(i,\mu) \triangleq \min_{M_{cs}\in\{1,2,\dots\}} \frac{1}{2}T_{\text{proc}} + T_{\text{trans}}(i) + \mu \cdot \log(\frac{1}{p_s(i)}). \tag{3.28}$$

The optimal MCS $M^*_{cs}(i,\mu^*)$ then can be obtained according to Lemma 2 by

$$M^*_{cs}(i,\mu^*) = \arg\min_{M_{cs}\in\{1,2,\dots\}} \frac{1}{2}T_{\text{proc}} + T_{\text{trans}}(i) + \mu\cdot\log(\frac{1}{p_s(i)}). \tag{3.29}$$

Recalling the BAP layer that contains the information of the path from the source node to the destination, the nodes can build up the network topology from the history information. Notice that increasing the relay nodes in the path will increase delay and decrease the reliability, so we can use Dijkstra's algorithm with the same weight $c(i)$ to avoid the backwards path and the neighbours with more hops. In this way, we narrow down the original routing and resource allocation problem so that IAB node selects the next hop with corresponding MCS and slots.

3.2.2 Deep Reinforcement Learning

We show an optimal routing solution with the assumption of the perfect network knowledge and no interference among nodes for the resource allocation in the uplink (UL) and downlink (DL), respectively. Though the algorithm requires global information, we can apply it to reduce the complex routing problem based on the graph to the neighbor selection and resource allocation problem. In this section, we propose using the soft actor-critic algorithm on each node aided by federated learning to determine the next-hop and select the slots for transmission, which can be implemented in the future 5G and beyond construction. We choose the SAC algorithm since it combines entropy regularization, off-policy learning, value function learning, and an actor-critic architecture to tackle complex DRL problems [132–134]. These characteristics make SAC an attractive algorithm for a wide range of applications and have contributed to its popularity in the field of deep reinforcement learning.

Deep Reinforcement Learning As illustrated in Fig. 3.10, the framework of reinforcement learning consists of agents and environments where each agent interacts with each other. In the multi-hop network, each IAB node is considered as an agent, and the wireless channel and transmission results are regarded as the environment that presents a collection of the channel states and the feedback for each transmission. Meanwhile, each IAB node is able to know the total number of nodes in the current network based on the global configuration of the IAB donor. Each node is assumed to allocate the resources based on the Time Division Multiple Access (TDMA) method.

At each time slot t, each IAB node can acquire the related information of its neighboring IAB nodes in the UL/DL direction respectively. The obtained information from each neighboring IAB node includes the channel information (i.e., CSI), total latency T_{delay}, and reliability (the probability of success $\mathbb{P}$). If we denote the total number of nodes in the multi-hop network is M, the state $\mathbf{s}_t$ of the IAB node $m(1 \leq m \leq M)$ at slot t is given by

$$\mathbf{s}_t^m = \begin{bmatrix} CSI_1^{UL}, & T_{1,delay}^{UL}, & \mathbb{P}_1^{UL} \\ CSI_1^{DL}, & T_{1,delay}^{DL}, & \mathbb{P}_1^{DL} \\ \vdots & \vdots & \vdots \\ CSI_M^{UL}, & T_{M,delay}^{UL}, & \mathbb{P}_M^{UL} \\ CSI_M^{DL}, & T_{M,delay}^{DL}, & \mathbb{P}_M^{DL} \end{bmatrix} \tag{3.30}$$

where $T_{n,delay}$ and $\mathbb{P}_n$ of node $n(1 \leq n \leq M)$ in the UL/DL direction are expressed in Eq. (2.6) and Eq.(2.7). Note that if a node does not belong to the neighboring node set of node m (including node m), all parameters related to this node in $\mathbf{s}_t^m$ will be set to a special value to indicate such a relation. As a result, each IAB node has the state with the same dimension, however, only the information from the neighboring nodes are truly effective in its state matrix.

The node m then takes action, i.e., choosing the best neighboring nodes to transmit its packets. Since the channel condition, latency and reliability vary at each neighboring node., the best chosen neighboring nodes may be different each time. To solve the problem above, we thereby implement the DRL algorithm where the state $\mathbf{s}_t^m$ of node m is regarded as the input while the output is an action score list. The action score list of node m at time slot t includes the scores of all nodes in the IAB network, which is expressed in Eq. (3.31).

$$\mathbf{a}_t^m = \begin{bmatrix} a_1^{UL}, & \cdots, & a_n^{UL}, & \cdots, & a_M^{UL} \\ a_1^{DL}, & \cdots, & a_n^{DL}, & \cdots, & a_M^{DL} \end{bmatrix}, \tag{3.31}$$

where a_n denotes the score of the node n in the UL/DL direction. This also benefits the model transfer and relay selection in the following discussion. Once the action score list is updated, node m chooses the node(s) with the highest UL/DL transmissions scores. For the UL resource allocation, the node m chooses some nodes with the highest scores and schedules the corresponding time slots for the chosen nodes to transmit their packets. Meanwhile, node m chooses the nodes with the highest scores for the DL resource allocation and then forwards their own packets to the chosen nodes immediately. Note that each node buffers a different number of packets to be transmitted in the UL direction, the chosen neighboring nodes need to transmit the packets in their buffers in order of node priority [6]. Therefore, when the node m chooses l nodes with the highest scores in the UL direction, l will depend on the number of packets as well as the packet size in the buffer of the higher priority node.

The decision function taken by each node is determined by the policy π_θ, where θ is the parameter of the policy π. There are many different RL algorithms to find and improve the policy π, while the objective of the standard RL is to maximize the expected sum of rewards from time t:

$$R_t(\pi) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\pi_\theta}\left[\sum_{k=t}^{\infty}\gamma^{(k-t)}r_k\right], \tag{3.32}$$

where $\gamma \in [0,1]$ is the discount factor used to avoid the accumulated reward to be infinity, and $r(s_t, a_t)$ is the reward by taking action a_t at state s_t. In reinforcement learning, the transition of the state $\mathbf{s}_t$ and reward r_t are stochastic and modelled as a Markov decision process (MDP), where the transition probability of state $\mathbf{s}_{t+1}$ depends only on the last state $\mathbf{s}_t$ and the action a_t taken by the agent. Therefore, each transition from $\mathbf{s}_t$ to $\mathbf{s}_{t+1}$ can be characterized by a conditional probability $p\left(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t\right)$. The reward r_t is used to guide the training and improve the policy. In the IAB multi-hop network, each node can calculate the reward based on the feedback (ACK/NACK) from the environment each time.

---

[6] After the node with the highest score transmits all the packets in its buffer, the node with the second highest score is the allowed to transit packets. The same procedure is applied for the remaining chosen nodes in the UL direction.

The node selection's objective is to meet the latency and reliability constraints in the UL/DL direction. After each action in a direction, the environment returns a reward to the agent to evaluate such an action. Therefore, the reward function that guides learning should be consistent with the objective. In our framework, the reward function consists of two components: the latency component and the reliability component. Let o_t denotes the feedback at time slot t from the environment: $o_t = 0$ when ACK is received, otherwise $o_t = 1$ when NACK is received. For the latency component, if the ACK is received within the latency constraint τ (i.e., $T_{\text{delay}} < \tau$), a successful transmission happens with a positive reward which is expressed as $\tau - T_{\text{delay}}$. This indicates the smaller T_{delay} is, the higher the reward can be returned from the environment. However, if the NACK (or timeout) happens (i.e., $T_{\text{delay}} \geq \tau$ [7]), $\tau - T_{\text{delay}}$ is also used to quantify the impact on the reward. If the remaining time within the latency constraint is long enough for accommodating a re-transmission, i.e., $\frac{\tau - T_{\text{delay}}}{T_{\text{delay}}} > 1$, due to the higher latency, a positive but lower reward will be returned for a successful re-transmission. Otherwise, a negative reward will be returned. For the reliability component, the more re-transmissions a packet needs, the lower the returned reward of this packet. Considering both components, the reward r_t in the UL/DL direction is thereby expressed as:

$$r_t = \psi_d \left(\frac{(\tau - T_{\text{delay}})}{(T_{\text{delay}})^{o_t}} + (-1)^{o_t} \right) - \psi_r (K_{\text{trans}} - 1), \tag{3.33}$$

where τ is the latency constraint. K_{trans} is the total number of transmissions for the same packet. ψ_d, and ψ_r are the coefficients which determine the weight of the latency and the reliability component, respectively. In order to obtain a long term performance which successfully achieves the URLLC requirements, both the immediate rewards and future rewards should be taken into consideration as the RL objective in Eq. (3.32). Note that Eq.(3.33) is applied for both UL and DL direction. However, the reward r_t in the UL/DL direction are different even though the link used for UL and DL between the agent and its neighbouring node is the same, this is because the channel conditions and latency/reliability

---

[7] The reason $\tau \geq T_{\text{delay}}$ is that the timer will timeout and set NACK before T_{delay} reaches τ.

requirements are different in two directions.

Soft Actor-Critic The brittle convergence properties and the requirements for meticulous hyperparameter tuning at different RL algorithms environments limit such methods' applicability to a complex, real-world domain like the routing and resource allocation problem for IAB multi-hop network. Most RL algorithms applied in current wireless network problems, like Deep Q Network (DQN) and Deep Deterministic Policy Gradient (DDPG), always obtain a deterministic policy, i.e., the policy only considers one optimal action for a given state. However, it is hard to generalize the property to other similar environments. Besides, the policy for routing and resource allocation in the IAB network is not always unique. Thus it is natural to consider a more robust algorithm with a stochastic policy for the model generalization in our resource allocation and routing problem.

In this section, we introduce Soft Actor-Critic (SAC), a model-free off-policy deep reinforcement learning algorithm based on maximum entropy reinforcement learning [58]. Instead of maximizing the expected sum of rewards in Eq. (3.32), the SAC algorithm introduces the entropy component into the objective at time t with the discount factor:

$$J_t(\pi) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi}\left[\sum_{k=t}^{\infty}\gamma^{(k-t)}\mathbb{E}\left[r_k + \alpha\mathcal{H}\Big(\pi\left(\cdot \mid \mathbf{s}_k\right)\Big) \mid \mathbf{s}_k, \mathbf{a}_k\right]\right], \tag{3.34}$$

where the temperature parameter α controls the degree of randomness of the optimal strategy and the importance of entropy relative to the reward, and $\mathcal{H}\Big(\pi\left(\cdot \mid \mathbf{s}_t\right)\Big)$ is the entropy of each action obtained by the policy. We use $\rho_\pi\left(\mathbf{s}_t, \mathbf{a}_t\right)$ and $\rho_\pi\left(\mathbf{s}_t\right)$ to denote the state and state-action marginals of the trajectory distribution induced by a policy $\pi(\mathbf{a}_t \mid \mathbf{s}_t)$.

The SAC algorithm consists of an actor-critic architecture with separate policy and value function networks as illustrated in Fig. 3.10. The actor updates the policies based on the policy gradient method, and the objective of the critic part is to evaluate the policy that the learning algorithm searches. More specifically, the SAC algorithms aim to use deep neural networks to learn the basic two functions - the policy function π_θ with parameter θ and the soft Q-function Q_ω with parameter ω.

The Q-function in typical RL algorithms is defined as a cumulative discounted reward by taking action $\mathbf{a}_t$ at state $\mathbf{s}_t$, and can be calculated using the Bellman equation [135]. In the maximum entropy reinforcement learning framework, we then regard the entropy as part of the reward to calculate the soft Q-function.

$$Q^{\pi}\left(\mathbf{s}_t, \mathbf{a}_t\right) = r\left(\mathbf{s}_t, \mathbf{a}_t\right) + \gamma \mathbb{E}_{\left(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}\right) \sim \rho_{\pi}}[V\left(\mathbf{s}_{t+1}\right)], \tag{3.35}$$

where $V(\mathbf{s}_t)$ is the value function defined as

$$V^{\pi}(\mathbf{s}_t) = \mathbb{E}_{(\mathbf{a}_t) \sim \pi}[Q^{\pi}\left(\mathbf{s}_t, \mathbf{a}_t\right) - \alpha \log \pi\left(\mathbf{s}_t, \mathbf{a}_t\right)]. \tag{3.36}$$

The soft Q-function parameters can be trained to minimize the soft Bellman residual

$$J_Q(\omega) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}}\left[\frac{1}{2}\left(Q_{\omega} - \left(r_t + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim \pi}\left[V_{\bar{\omega}}\left(\mathbf{s}_{t+1}\right)\right]\right)\right)^2\right], \tag{3.37}$$

where the value function is implicitly parameterized through the soft Q-function parameters via Eq. (3.36), and it can be optimized with stochastic gradient

$$\hat{\nabla}_{\omega} J_Q(\omega) = \nabla_{\omega} Q_{\omega}\left(Q_{\omega} - \left(r_t + \gamma\left(Q_{\bar{\omega}} - \alpha \log\left(\pi_{\theta}\left(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}\right)\right)\right)\right)\right) \tag{3.38}$$

The update makes use of a target soft Q-function with parameters $\bar{\omega}$ that are obtained as an exponentially moving average of the soft Q-function weights, which has been shown to stabilize training [136].

Federated Learning

Each IAB node in the coverage of the IAB donor first uses the described SAC algorithm for the routing and resource allocation with random initialization of the neural network (NN) weight θ. However, some nodes are likely initialized with worse NN weights. There may arise an issue where the weights of NN in these nodes may never converge due to the faster change in the environment, such as channel condition. As a result, the applied SAC algorithm does not function in some nodes. To cope with such an issue, we consider adding one more mechanism to this network, Federated Learning (FL) [137, 138], which refers to learning a high-quality global model based on decentralized data storage for many nodes. Note we initialize the states of the IAB-nodes inside the same networks with the same size, thus we can use the FL algorithms to average the NN weights. FL has been shown to be a fast convergent method in distributed networks. We thereby propose our FL mechanism at the IAB donor side based on FedAvg [139].

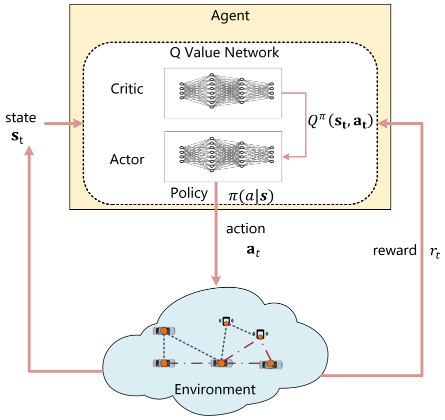


Figure 3.10: Illustration of DRL Architecture.

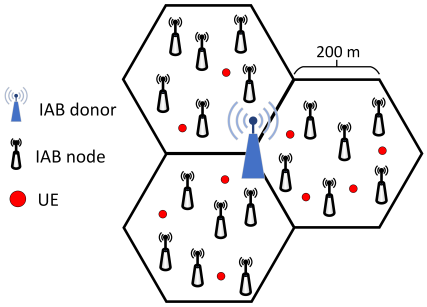
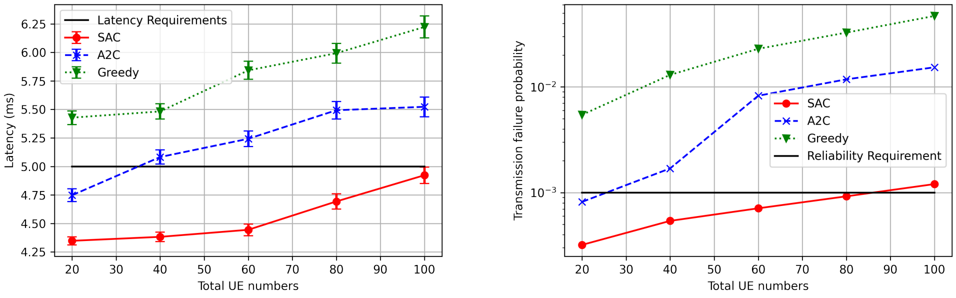


Figure 3.11: Simulation Scenario.

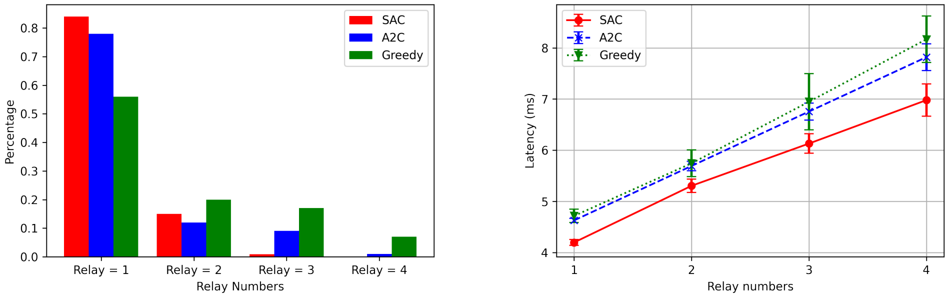
3.2.3 Performance Evaluation

In this section, we analyze the performance of our proposed methods. We conduct the simulation based on the homogeneous scenario (urban micro) based on 3GPP standard [18]. As shown in Fig. 3.11, we consider three hexagonal grids with the IAB-donor located in the center, and six IAB nodes are located inside each grid. We also use the 3GPP channel model in mmWave for the links among IAB nodes and UEs. The major parameters for the channel model are summarized in Table 3.3. The UEs are dropped independently with uniform distribution and connected to the closest IAB nodes. The UEs randomly walk within its IAB nodes' coverage and move with a speed of 80% indoor (3km/h), 20% outdoor (30km/h) as suggested in the standard. We change the number of UEs to generate different traffic loads



(a) Average delay.

(b) Average transmission failure probability.



(c) Percentage of different relay nodes (40 UEs).

(d) Delay for different relay numbers (40 UEs).

Figure 3.12: Simulation results of different UE numbers.

in the simulation. We adapt the FTP model 3 as the traffic model where the packet size is 0.1 Mbytes while the packet's arrival follows a Poisson distribution with a mean of 100/3 per second. In addition, the ratio of access DL/UL traffic is 4:1. We set the traffic type as the VR/AR traffic with the expected latency less than 5 ms and reliability of 0.999 successful rates as defined in the 3GPP standard [140].

For comparison, we implement a greedy algorithm characterized by Eq.(3.26)-Eq.(3.29) as the baseline. Note that the greedy algorithm is an extension of Semi-Persistent Scheduling algorithm defined in 3GPP standard [141, 142], because greedy algorithm always tends to stick with the current transmission policy and adopt a new transmission policy only if some

Table 3.3: Simulation Parameters in ns-3

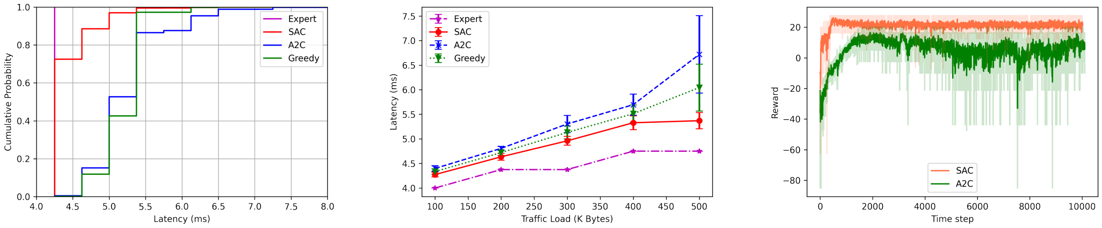
Parameter	**Value**	**Parameter**	**Value**
Power	23 dBm	Numerology	3
Bandwidth	100MHz	Center Frequency	28 GHz
Channel Model	3GPP mmWave channel model	Pathloss Model	3GPP MmWave propagation loss model
Environment	3GPP Urban Micro (UMi)	BS receiver noise figure	7 dB
UE receiver noise figure	10 dB	UE traffic model	FTP model 3

conditions fulfill. In the greedy algorithm, each IAB node selects the next-hop with the best channel condition, i.e., choosing the maximum MCS in the current transmission, and if one transmission fulfills the URLLC requirement, it keeps the transmission on the same nodes until the URLLC requirement will not be satisfied. Besides, we choose the Advantage Actor-Critic (A2C) method proposed in [57] to compare the enhancement of our proposed SAC algorithm. A2C approach can leverage merits of both value based approach and policy gradient and it empirically performs better than other similar RL approaches on coping with dynamic link blockages in a complicated IAB scenario, where each node selects a pattern (a set of links activated in parallel) according to the current policy, and then the links in this pattern are activated and enabled to transmit data.

Fig. 3.12 shows the simulation results of different numbers of UEs. The proposed SAC algorithm outperforms the other two algorithms from latency and reliability perspectives. As Fig. 3.12(a) shows, the average latency in the three algorithms increases with the increasing number of UEs. This is because the more the number of UEs is, the higher the queuing delay will be induced in T_{delay}^{dir} which is expressed in Eq. (2.3), leading to higher total latency. Note that both greedy and A2C methods do not meet the target latency requirement under the cases of a large number of UEs. However, our proposed SAC algorithm can always fulfill such a defined requirement, whatever the number of UEs is. Meanwhile, as is shown in Fig. 3.12(b), the average transmission failure probability with SAC is significantly lower than the other two algorithms, especially when the number of UEs is large. Notice that the proposed SAC algorithm is able to fulfill the target reliability requirement in most cases while the other two algorithms always fail. Besides, the failure probability in the three algorithms increases with the increasing number of UEs. With more UEs, IAB nodes are more likely to be scheduled to transmit at the same slot, causing a higher collision probability p_c. Thus the probability of a successful multi-hop transmission, $\mathbb{P}$ expressed in Eq. (2.7), will decrease accordingly. As a result, as the number of UEs increases, the average delay in Fig. 3.12(a) will be impacted by the average transmission failure probability since a successful multi-hop transmission is likely to need more retransmissions.

We further explore the relevant performances under a fixed topology with a fixed number of UEs. Fig. 3.12(c) shows the distribution of the number of relay nodes used for multi-hop transmissions among three algorithms. Compared with the other two algorithms, the proposed SAC always utilizes fewer relay nodes to complete a multi-hop transmission. Particularly, the percentage of using one relay node with SAC is around 10% higher than that with A2C while around 30% higher than that with greedy. Meanwhile, the number of relay nodes with SAC is up to 3 while it reaches 4 in both greedy algorithm and A2C. From the latency perspective, the fewer the number of relay nodes is utilized, the lower aggregate transmission time and processing time that will be induced in the total delay expressed in Eq.(2.6), i.e., the smaller the expectation of n will be in Eq.(2.6). Therefore, with the same number of UEs, the average delay with SAC is always lower than that with the other two algorithms, which has been validated in Fig. 3.12(a). From the reliability perspective, fewer relay nodes also reduce the number of links in a multi-hop transmission, as a result, one packet transmission is less likely to be impacted by the collision probability or the BLER, which thereby increases $\mathbb{P}$ expressed in Eq.(2.7). The simulation results can also validate this regarding the average transmission failure probability in Fig. 3.12(b).

Fig. 3.12(d) illustrates how the number of relay nodes impacts the average delay under the same topology with the same number of UEs. As we can see, the average delay shows a linear increase with the increasing number of relay nodes among the three algorithms. This can be explained by the fact that the processing time and transmission time included in the total delay, as Eq.(2.6) suggests, is proportional to the number of relay nodes. Notice that although the increase of relay nodes also increases the average delay due to a higher failure probability, the increment of failure probability is not so sensitive, indicating that the increment of average delay is dominated by the processing time and transmission time when a node is added. Besides, SAC always outperforms the other two algorithms on the average delay, regardless of the number of relay nodes. Therefore, the SAC algorithm is more capable of learning from the environment and making stochastic decisions by regarding the entropy as part of the reward in the SAC algorithm. This adapts to different environments and handles the decision-making procedure.



(a) Delay CDF for traffic load 300K Bytes.

(b) Traffic load.

(c) Reward vs time slot for traffic load 300K Bytes.

Figure 3.13: Simulation results under a fixed topology with 40 UEs and different traffic loads.

More details under a fixed topology with 40 UEs and different traffic loads are shown in Fig. 3.13. With the fixed topology, we can then use the graph model proposed in Sec 3.2.1 to give the expert solution. We allocate the traffic among the users proportional to the traffic load it requires among the IAB nodes and UEs for both DL and UL directions. Fig. 3.13(a) shows the cumulative distribution function (CDF) of delay for traffic load 300 Kbytes among four algorithms. The CDF with SAC grows faster than the other two algorithms, indicating that SAC outperforms the other two algorithms due to the higher ratio of low-latency packets. Particularly, the probability of the delay ≤ 5 ms reaches around 95% with SAC, while the corresponding probability with the other two algorithms only reaches around 50%. Note that the CDF with A2C has the longest tail; its variance is thereby the largest among them. As a result, compared with the greedy algorithm, a multi-hop transmission with A2C may suffer more due to the higher ratio of high-latency packets, even though it grows faster when the delay $\leq$ 5 ms. Subsequently, we investigate the latency regarding the traffic load, which is shown in Fig. 3.13(b). Due to the linear increment of traffic load, the latency with each algorithm shows a linear growth in the region of low traffic loads, where the latency with SAC is always the lowest. However, when traffic load is 500 Kbytes, the latency with both A2C and greedy algorithms grows faster because a large traffic load also incurs the increase of collision probability that impacts the total delay. Fig. 3.13(c) shows the comparison between SAC and A2C on the reward convergence for traffic load 10 Kbytes. We conducted such a simulation with 10000-time steps to ensure that the returned reward could converge in both algorithms. As we can see, SAC outperforms A2C on the reward convergence in three aspects: 1) Convergence speed: SAC converges quite faster than A2C. More precisely, SAC takes around 700-time steps while A2C takes around 2100 time steps to reach the convergence; 2) Steady reward value: the steady reward value with SAC is at least 30% larger than that with A2C; 3) Stability: the returned reward trend with SAC is more stable than that with A2C after 2100 time steps. Therefore, the stochastic policy generated by the SAC algorithm enhances the ability to transfer knowledge compared with the A2C algorithm. Besides, the SAC algorithm's objective also encourages exploring more possible actions that contribute to faster convergence.

Afterward, we study the impact of FL on the convergence speed of SAC and A2C. As Fig. 3.14 shows, the reward in the four algorithms converges to the same value. The reward in both algorithms with FL converges faster than that without FL. In particular, while the convergence speed of two algorithms without FL are quite close, SAC with FL takes around 2500 time steps fewer than that with SAC without FL, and A2C with FL take around 2000 time steps fewer than that with A2C without FL, indicating that SAC achieves more improvement on A2C after adding FL. Hence the reward with SAC with FL still outperforms A2C with FL. This indicates that the structure of the entropy-based reinforcement learning with federated learning has the potential to be implemented in the radio intelligence controller

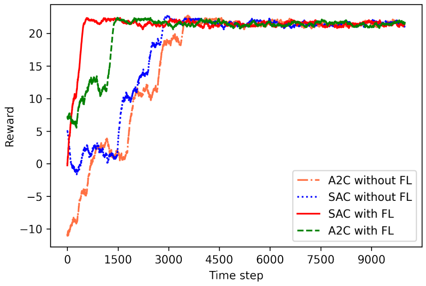


Figure 3.14: The impact of FL on the convergence speed.

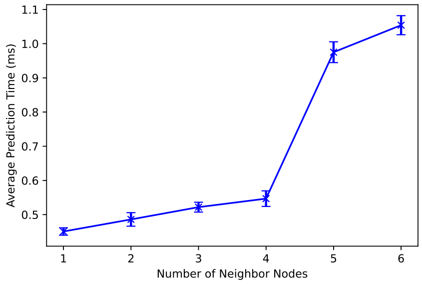


Figure 3.15: The average running time for each prediction.

in 5G and beyond networks.

Furthermore, we implement the SAC algorithm through PyTorch to measure the runtime of selecting actions for a given state. We run such a simulation with the mean packet size is 30K Bytes. The simulation results are shown in Fig. 3.15. As the number of neighboring nodes increases, the average runtime increases slowly at first. Since the policies are running parallel, the increased time is not so significant based on the comparison of the action score. After the number exceeds 4, however, the running time doubles due to the limitation of the computation resources; thus the policies are no longer running parallel. Note that the algorithm runtime is approximately around 0.5 ms for the case of 1 neighboring node. This performance may be impacted by the efficiency of PyTorch and the computing ability of the computer. The efficiency and computing ability can be further improved in a real system where the proposed algorithm can be implemented on the hardware and software.

3.3 Proof of Equations

3.3.1 Asymptotic Proportional Fairness

Denote $\pi_t(n)$ the probability of the transmission of the user n at time slot t. $DRC(n)$ denotes the achievable instantaneous data rate for user n. Let $R(n)$ denote the average rate allocated to user n, so

$$R(n) = \sum_t \pi_t(n) DRC(n). \tag{3.39}$$

Therefore, the network utility maximization problem [126] is

$$\max \sum_n \log(\sum_t \pi_t(n) DRC(n)) \tag{3.40}$$

$$\text{s.t. } \sum_n \pi_t(n) \leq 1, \quad \forall t, \pi_t(n) \geq 0, \quad \forall n, t. \tag{3.41}$$

Using Lagrange multipliers, we obtain

$$\sum_n \log(\sum_t \pi_t(n) DRC(n)) - \sum_t \lambda_t(\sum_n \pi_t(n) - 1). \tag{3.42}$$

So the optimal solution satisfies

$$\frac{DRC(n)}{R^*(n)} - \lambda^* = 0 \quad \text{if} \quad \pi_t^*(m) > 0, \tag{3.43}$$

Asymptotically, the PF preserves the proportional fairness between the eMBB users, i.e.,

$$\lim_{t \longrightarrow \infty} \frac{R_t(1)}{DRC(1)} = \cdots = \lim_{t \longrightarrow \infty} \frac{R_t(n)}{DRC(n)}. \tag{3.44}$$

So, our joint allocation approach preserves the proportional fairness between the eMBB users.

3.3.2 Bisection Approach

The Bisection approach to find the optimal μ^* is shown in Algorithm 3.

3.3.3 Proof of Lemma 1

Proof. When $q^*(\mu) = \sigma$, we have $\log \mathbb{P}\left(q^*(\mu)\right) = \log \sigma$. For any other $q^{'} \in \mathcal{Q}$ that satisfies $\mathbb{P}\left(q^{'}(\mu)\right) \geq \sigma$, we have by applying Eq. (3.27),

$$T_{\text{delay}}\left(q^*(\mu)\right) - \mu \log \mathbb{P}\left(q^*(\mu)\right) \leq T_{\text{delay}}\left(q^{'}\right) - \mu \log \mathbb{P}\left(q^{'}\right) \leq T_{\text{delay}}\left(q^{'}\right) - \mu \log \mathbb{P}\left(\sigma\right). \tag{3.45}$$

From Eq. (3.45), we then obtain that $T_{\text{delay}}\left(q^*(\mu)\right) \leq T_{\text{delay}}\left(q^{'}\right)$. Therefore $q^*(\mu)$ is the optimal solution for the original Problem 4. $\square$

Algorithm 3 Bisection approach to find μ^*.

```
$\mu_{\text{low}} = \mu = 0,\ \mu_{\text{up}} = \text{MAX\_MU}$
while $(\mathbb{P}\left(q^*(\mu)\right) < \sigma)$ or $(\mathbb{P}\left(q^*(\mu)\right) > \sigma + \eta_0)$ do
    if $(\mathbb{P}\left(q^*(\mu)\right) < \sigma)$ then    $\mu_{\text{low}} \leftarrow \mu$
    else    $\mu_{\text{up}} \leftarrow \mu$
    $\mu \leftarrow \frac{\mu_{\text{low}} + \mu_{\text{up}}}{2}$
return $\mu^* \leftarrow \mu$
```

Chapter 4

URLLC IN IEEE 802.11AX/BE NETWORKS

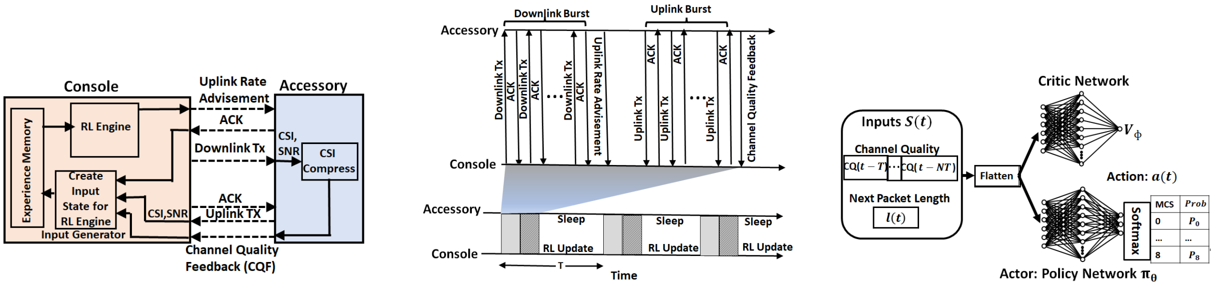
4.1 Reinforcement Learning Based Link Rate Adaptation for Gaming

In this section, we present and assess ADR-X, a novel Reinforcement Learning (RL)-based wireless rate adaptation technique. This system harnesses the power of online learning to deftly navigate the intricate variables of host radio hardware, background interference, and rapid player movements, all with the aim of delivering ultra-reliable and low-latency communications (URLLC) performance over Wi-Fi networks. ADR-X employs a predictive model to select the most suitable data rates based on channel measurements, thereby achieving an impressive 10-fold reduction in packet losses when compared to contemporary adaptive data rate schemes. It exhibits a quick response to sudden changes, converging swiftly within a span of 10-20 seconds. Furthermore, ADR-X demonstrates a pioneering ability to utilize federated models, enabling it to learn from the experiences of other devices. This innovative approach assists in the seamless integration of new devices into the network, enhancing overall network adaptability and performance. Through the evaluation of ADR-X, we explore the transformative potential of leveraging advanced learning techniques in wireless rate adaptation.

4.1.1 The Design of ADR-X

ADR-X is an online reinforcement learning (RL) based adaptive data rate scheme designed to combat the fast channel changes experienced by the wireless channel between console and its accessories due to hand/head/body motions of the player during active gaming (Section 2.2.2). In this section we describe the design of ADR-X.

Overview of ADR-X The overall architecture of ADR-X is depicted in Figure 4.1(a). The



(a) Architecture of ADR-X (b) Communication Cycle in ADR-X operation (c) The PPO Architecture for Reinforcement Learning

Figure 4.1: The Design of ADR-X

design of ADR-X is based on a console-centric architecture where the RL engine resides at the console. This is because, gaming accessories (controllers, headsets *etc.*) are typically low power embedded devices with limited computation capabilities while the console is powered and typically has more computational resources.

ADR-X Console Side. ADR-X on the console side comprises three main components. The *RL Engine*, *RL Experience Memory* and an *RL Input Generator*.

RL Input Generator : This component is responsible for converting all observations and measurements such as CSI and SNR measurements, received and missed acknowledgements etc. into features that can be used by the RL engine for training as well as inference as described later in this section.

RL Experience Memory : This component stores recent history of feature inputs generated by the RL Input Generator. The history is then used by the RL Engine to adapt by learning. When there are multiple accessories connected to the console, history must be maintained on a per wireless link basis – two per each accessory, uplink and downlink.

RL Engine : The RL engine uses the Experience Memory to update itself and learn as well as to generate inferences *i.e.* predict the appropriate transmission data rate for each packet.

ADR-X Accessory Side. ADR-X on the accessory side is responsible only for generating the channel quality feedback from channel measurements and transmitting to the console.

ADR-X Communication Cycle. As described in Section 2.2.2, gaming traffic is periodic with periods ranging between 4-20 ms. ADR-X's operation is designed to confirm to this underlying traffic periodicity as depicted in Figure 4.1(b) with period Tms. The accessories sleep periodically to conserve power by switching off their radios and placing their processor in a low power mode and then wakeup for uplink and downlink transmission bursts once every T ms. The wakeup period comprises the following steps and transmissions.

Downlink Transmission Burst : The console transmits multiple downlink packets in a burst and receives acknowledgements (ACKs) from the accessory for each successful reception. The lack of receipt of an ACK triggers a packet retransmission. The accessory obtains downlink channel quality information (CSI, SNR *etc.*) during these transmissions.

Uplink Rate Advisement : At the end of each downlink burst, the console must advise the accessory as to the uplink data rates it must use. The uplink data rates however depend on the size of data in the packets. This is because, smaller packets experience less packet loss rates. Thus, the uplink rate advisement packet comprises a lookup table mapping packet lengths to uplink data rates. This table is computed by the RL engine inference engine by using the experience memory corresponding to that accessory and a set of lengths. The uplink rate advisement is transmitted at a low transmission rate for reliability and only when there are changes.

Uplink Transmission Burst : The accessory uses the received rate advisement packet to determine the rates for each of its packets based on their lengths. It then transmits all the packets to the console, retransmitting each time that an acknowledgement is not received from the console.

Channel Quality Feedback : At the end of the uplink transmissions, the accessory computes the channel quality feedback from the channel measurements obtained during the downlink transmission burst. It then transmits this channel quality feedback to the console. The channel quality feedback is transmitted at a very low transmission rate for reliability.

Online Reinforcement Learning ADR-X uses a reinforcement algorithm (RL) based on the Proximal Policy Optimization (PPO) [143] (Figure 4.1(c)). In this section we describe the RL scheme used in ADR-X.

A General Overview of RL. At time t, in order to determine the required transmission rate for a packet, the RL algorithm takes as input the state vector $S(t)$ comprising experience vector from the experience memory and generates the appropriate action state $a(t)$ comprising the appropriate data rate (MCS) to be used. Internally the RL engine maintains mapping from state to action $S(t) \rightarrow a(t)$ deemed the state-action policy map π_θ (θ represents the parameters of the policy map function that are updated while learning). After each transmission, a reward $r(t)$ is calculated to evaluate the performance by choosing the action $a(t)$ at the current state $S(t)$. These rewards are then used to train the RL Engine to update its state-action policy function by adjusting θ so as to maximize the expected future rewards.

Overview of PPO Algorithm. PPO algorithms belong to class of RL algorithms known an Actor-Critic algorithms comprising two separate neural network models – an Actor and a Critic (Fig. 4.1(c)). An actor model is a state-action policy map, π_θ to learn what action to take under a particular observed state; θ represent the weights of the neural network. The critic model V_ϕ evaluates the effectiveness of π_θ by predicting the expected future reward based on past history of action-reward pairs;its weights are represented by ϕ. The critic learns by trying to minimize the discrepancy between its past estimates of future rewards and those that it actually observes. The actor learns by attempting to maximize the expected future rewards as predicted by the critic network. Both actor and critic networks learn in conjunction taking turns based on the evolution of state, actions and rewards by employing gradient descent optimization. As time progresses, the critic learns to predict future rewards more accurately and the actor learns to take more optimal actions for each state. While updating the policy-map, PPO algorithms constrain the amount of change allowed in the policy to limit sudden/drastic changes and hence are stable to sudden changes.

In our implementation, we chose a Multilayer Perceptron (MLP) with two hidden layers to represent both actor and critic models. While we believe that larger models can provide better performance, our choice was based on a limited time budget (<5ms) to update the neural network weights and compute inferences coupled with unavailability of a GPU in the

Wi-Fi device to speed up computation.

State, Action and Rewards in ADR-X. Central to ADR-X's performance is the choice and definitions of state, action and rewards that are used to train ADR-X.

State $s(t)$***:*** The state must ideally comprise all information that is relevant for the actor network and enable it to predict an appropriate action (data rate). In our implementation these comprise channel quality CQ and packet length of the current packet to be transmitted l_t. Since, the most recent channel measurement is usually taken in the past (typically Tms in the past), the channel may have changed within these Tms. The actor must be able to implicitly predict the current state of the channel from history. To enable this, we provide as state, a vector comprising a history of n channel states and lengths. The state is thus computed as $S(t) =< \mathbf{CQ}(t-T), \mathbf{CQ}(t-2T), \cdots, \mathbf{CQ}(t-kT), l_t >$. We use $k = 3$ in our implementation since higher values did not provide any "significant" benefit. The contents of **CQ** are computed by the Input Generator and depend on the type of channel measurement available in the device. We have designed ADR-X for three different kinds of channel measurements, each of which uses a different **CQ**.

1) CSI and SNR : When the NIC card provides both CSI and SNR, it is possible to compute the per-subcarrier SNR by scaling CSI by the SNR. The i^{th} subcarrier SNR Γ_i of the N subcarriers is calculated as $\Gamma_i = SNR\frac{|CSI_i|^2}{N}$ [144]. Alternatively, if the NIC provide RSS and Noise Floor (NF) instead of SNR directly, SNR maybe calculated using $\frac{RSS}{NF}$.

In the 802.11ax standard, we have total $N = 242$ sub-carriers for 20 MHz bandwidth and $N = 48$ for 802.11n. If all the subcarriers are used as CQ then the size of **CQ** will be 726 numbers. This in turn places a significant computational burden for training and inference using ANNs. To reduce the overhead, we compress Γ_i further to use the SNR per resource unit (RU) as the channel quality :

$$\Gamma_{RU} = \sum_{k=id}^{id+K} \frac{\Gamma_k}{K}, \tag{4.1}$$

where id is the start subcarrier for a certain RU and K is the total number of subcarriers that one RU has. In this way we reduce the size of **CQ** from 242 values to 8 values for 802.11ax and 7 values for 802.11n. This per-RU average SNR is available as HE CQI report field as standardized in [145]. Therefore, this low overhead feedback will be available on commercial Wi-Fi 6 devices.

2) CSI, RSSI : When either the Noise floor and SNR are not provided by the NIC card, it is not possible to compute the exact noise floor. In this case we may choose a value from the data sheet of the radio for the noise floor, with the caveat that the actual value on the radio may be different and need calibration. Fortunately, since ADR-X uses a neural network it will probably be able to implicitly learn this calibration. Using this NF, we compute **CQ** as in equation 4.1.

3) SNR Only : Some devices can provide only SNR. In this case **CQ** comprises only one value, that of the SNR.

Action $a(t)$***:*** The action space is all the possible MCSs determined by the IEEE standards for each packet. The policy network outputs the probability estimated by the neural network for actions to achieve the highest reward. Since the action space is discrete, we choose the action with the largest possibility with 95% of the time and 5% of the time we choose the action randomly to explore other rates.

Reward $r(t)$***:*** We design the reward in ADR-X to specifically target packet loss and power consumption as they are the primary determining factors for user experience in a gaming scenario. In our design we prioritize packet loss more than power consumption since packet loss results in immediate loss of user experience. Further, consecutive packet losses are discouraged to greater degree as they have more significant impact on user experience. The reward function as follows:

$$r(t) = -\tau(MCS) - Q * \rho(t), \tag{4.2}$$

where $\tau(MCS)$ is the total transmission time by choosing the current MCS, including the retransmission time if the first packet is lost and $\rho(t)$. The term $-\tau(MCS)$ is a penalty on long packet transmission times and hence discourages ADR-X from choosing low data rates. The term $Q * \rho(t)$ penalizes packet losses with the weight Q determining the trade-off between the power consumption and the packet losses. To prioritize avoiding packet loss $\rho(t)$

is computed as a running average of the packet loss given by :

$$\rho(t) = \rho(t-1) * \alpha + b(t),$$
$$b(t) = \begin{cases} 0 & \text{if no packet loss} \\ 1 & \text{if packet loss} \end{cases} \tag{4.3}$$

α denotes the reliability requirements, for instance, $\alpha = 0.99$ for the audio packets. In this manner, each packet loss has a long lasting negative impact on the reward, and consecutive packet losses have a greater impact on the reward calculation. This choice discourages ADR-X to quickly eliminate losses.

Neural Network Architectures Used Out input size for the actor network was 25 $(8 \times 3 + 1)$ for 802.11ax CSI measurements, 22 $(7 \times 3 + 1)$ for 802.11n and 2 when we used SNR measurements only. We used two hidden layers with 32 hidden nodes each. The number of outputs were 9 corresponding to data rates MCS0-MCS8. For the critic network has a single output we used the same number of inputs and outputs except each of the two hidden layers had 64 nodes each.

Training Methodology. The objective of the PPO algorithm is to maximize the expected accumulative reward from current time t: $R^{\theta}(t) = \mathbf{E}_{(S(t),a(t))\sim\pi_\theta}\left[\sum_{j=t}^{\infty} \gamma^{(j-t)} r(j)\right]$, where $\gamma \in [0,1]$ is the discount factor (usually 0.99) used to avoid the accumulated reward to be infinity, and $r(t)$ is the reward by taking action $a(t)$ at state $S(t)$.

The Actor-Critic structure first obtains a finite mini-batch of sequential samples from the trajectory memory. The PPO algorithm randomly chooses a start point within each batch and uses the sub-sequential data to train the network. A new objective function is proposed in PPO to achieve mini-batch updates and update the policy smoothly. PPO introduces importance sampling to obtain the expectation of samples gathered from an old policy π_{old} under the new policy π_{new} we want to refine with the probability ratio $R^{\theta}(t) = \frac{\pi_\theta\left(a_t|s_t\right)}{\pi_{\theta_{\text{old}}}\left(a_t|s_t\right)}$. They maximize the following surrogate objective function: $L(\theta) = \widehat{E}\left[\min\left(R^{\theta}(t), \text{clip}\left(R_t(\theta), 1-\epsilon, 1+\epsilon\right)\right)\widehat{A}_t\right]$, where ϵ is the clipping parameter. $\widehat{A}_t$ is an estimator of the advantage function at time step t. We use the generalized advantage estimator (GAE) [146] to calculate $\widehat{A}_t$. By introducing the clipped objective function, the PPO algorithm won't stick to the favoring actions with positive advantage, and make quicker update to avoid actions with a negative advantage function from a mini-batch of samples. $\theta = \theta - \eta_\theta \nabla L^\theta$, where η_θ is the learning rate for the actor model optimization.

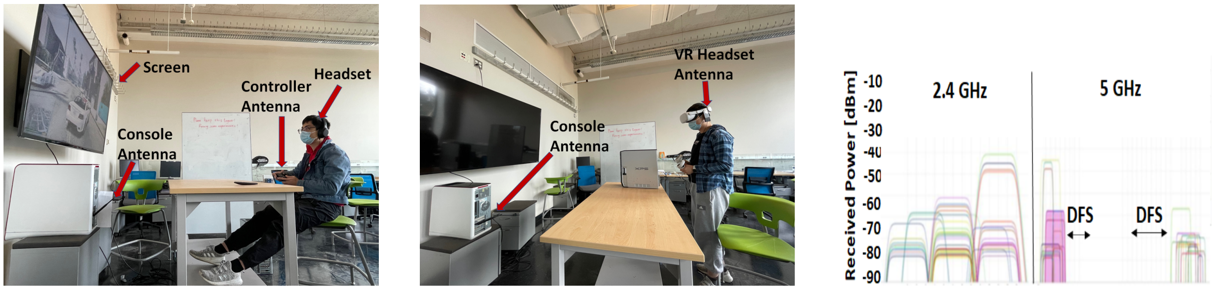
Avoiding Losses During Convergence. When ADR-X is training for the first time or when the environment changes suddenly, ADR-X may experience higher packet losses during the time it takes to learn, adapt and converge to a steady state. In order to avoid the losses during these vulnerable times, ADR-X chooses a relies on a conservative retransmission strategy. While RL in ADR-X provides only the data rate of the initial transmission, the retransmission is provided greater reliability by reducing the data rate to an MCS three steps below that suggested by ADR-X. Note that the corresponding longer transmission time also acts a discouraging penalty to the reward function.

Federated Learning for Initialization When the device is turned on first time, ADR-X has to start learning from no prior experience. In practice this means initializing weights of the neural networks θ and ϕ randomly. Starting from no experience, ADR-X may take a long time to converge (1-4 minutes as shown in our evaluations). It is possible to reduce this initial convergence time by leveraging the experiences learned from other similar consoles using Federated Learning (FL) [147].

The FL algorithm conducts the *weighted model aggregation* over all the users. Specifically, suppose we have a total number of M users with the same dimension of the neural networks. For each user, we then obtain the weight of the neural networks as a matrix $\mathbf{W_m}$. To fuse the individual models, FL performs a weighted averaging operation:

$$\widetilde{\mathbf{W}} = \sum_{m=1}^{M} \Omega_m \mathbf{W}_m, \tag{4.4}$$

where Ω_m is the weight matrix of user m for each neuron in the neural networks. For a newly joined user (i.e., a user without any prior experience), we generate and apply a model with average experience from all users. Since games played on different consoles can be in different environments, the federated model represents a generic model for initialization.



(a) The setup for console gaming experiments

(b) The setup for VR gaming experiments

(c) The Wi-Fi environment in 2.4 GHz (High Interference), 5 GHz (Medium Interference) and DFS channels (No Interference)

Figure 4.2: Experimental Setup

Implementation of ADR-X We implemented the ADR-X on the stationary desktop or portable laptop equipped with commercial NIC cards. We conduct our experiments under Ubuntu 20.04.3 since getting the PHY layer measurements and setting up the NIC is easier on Linux. We use the PicoScenes toolbox [63] to collect the CSI measurements for each packet from the NIC. To customize the packet sizes and feedback, we use the packet injection mode provided by the PicoScenes Plugins [148]. PicoScenes allows us to modify packet size, traffic pattern, and MCS directly from the toolbox. We implemented the ADR-X with more than 1.5 K lines of C++ code for the whole ADR-X algorithm, including the basic operation for PPO like gradient descent and updates of the neural network within the PicoScenes Plugins. The training time for PPO is 3-5 ms tested on the AMD Ryzen Threadripper 3970X CPU.

4.1.2 Evaluation of ADR-X

In this section we evaluate the efficacy of using ADR-X for gaming by answering the following questions.

- How does ADR-X's performance compare to state of the art ADR techniques?
- How quickly does ADR-X learn during game play and improve with experience?

- How can ADR-X benefit from experiences of other gaming consoles using federated learning?
- How does ADR-X adapt upon experiencing sudden changes in the ambient interference?
- ADR-X compresses the channel quality (CSI) measurements by a factor of 100× - how much performance loss does it incur compared to an uncompressed version?

Experimental Setup We conduct our experiments for two different gaming scenarios – Console Gaming and VR, given the differences in their traffic characteristics. In order to be able to compare multiple schemes we replayed recorded 15 minutes long packet traces for each of the gaming scenarios.

Console Gaming : A gamer sitting about 3 feet in front of a gaming console (a desktop) and playing with a wireless controller. An audio headset is connected to the controller through an audio jack (Figure 4.2(a)). The receiving antenna was attached to the controller so that it moved with the device as the player played actual games like GTA-V and experienced rapid changes as an actual device would. The gaming video is not transmitted wireless as it is displayed on a screen that is connected via an HDMI cable to a T.V.

VR Gaming : In this setup A gamer standing and playing a VR game wearing a VR headset (Figure 4.2(b)). The receiving antenna of the headset is attached to the headset and moves with the player as he moves his head.

External Interference Scenarios. In a practical environment packet losses occur not only due to the wireless channel but also due to interference and collisions with transmissions from other Wi-Fi networks devices. Packet losses due to other Wi-Fi devices can make learning harder for the RL engine. Wi-Fi operates in three distinct bands – 2.4 GHz band, 5 GHz band (corresponding to U-NII-1 and U-NII-3 bands which does not require DFS for operation) and the DFS band. Typically, 2.4 GHz experiences the most severe congestion and interference, followed by 5 GHz while DFS is typically free. Figure 4.2(c) depicts the power level and channel width for each of the interfering Wi-Fi devices in the three different Wi-Fi bands within the apartment complex where the experiments were conducted. Our measurements indicated that there were 161 active interfering Wi-Fi devices in 2.4 GHz band, 85 in 5 GHz band and none in DFS band. Further, there was a 50% split between 20 MHz and 80 MHz channels in each of these bands. To test ADR-X's performance under different congestion conditions we conduct experiments in each of these bands –*No Interference* (DFS Bands), *Medium Interference* (5 GHz Bands) and *High Interference* (2.4 GHz bands).

Different Wi-Fi Radios and Channel Measurements. Different wireless NIC cards differ in terms of the type and quality of channel measurements they provide. Many NIC cards do provide sufficient information that cannot be used to compute the exact per sub-carrier SNR. We use the PicoScenes toolbox [63] to inject traffic as well obtains the CSI data from the Linux driver.

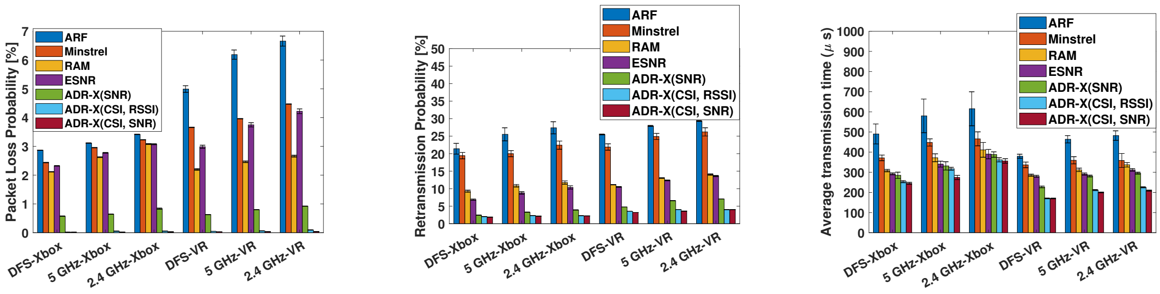
In order to demonstrate how ADR-X performs even for cards that provide incomplete information we make use of two different wireless NIC cards for our experiments – NIC-I : Qualcomm Atheros 9300 (802.11a/b/g/n) and NIC-II : Intel AX200 (802.11a/b/g/n/ac/ax). NIC-I provides CSI, RSSI, NoiseFloor and SNR information for each received packet while NIC-2 provides only CSI and RSSI. We consider four different scenarios, in order of decreasing detail regarding the wireless channel.

Complete Channel Information - We use SNR and CSI information from NIC-I to compute per-subcarrier-SNR as described in Eqn 4.1 and train ADR-X using it. We refer to this flavor as ADR-X(CSI,SNR).

Partial Channel Information - NIC-II provides only RSSI and CSI information but no SNR or receiver noise floor. This does not allow us to compute the exact per-subcarrier-SNR. To simplify the modification on the algorithm, we use a constant value for the noise floor and perform similar calculation as above. We refer to this flavor as ADR-X(CSI,RSSI).

SNR Information Only - Several Wi-Fi devices do not provide CSI but only SNR. To test this scenario we use only the SNR information provided by NIC-I to train ADR-X. We refer to this flavor as ADR-X(SNR).

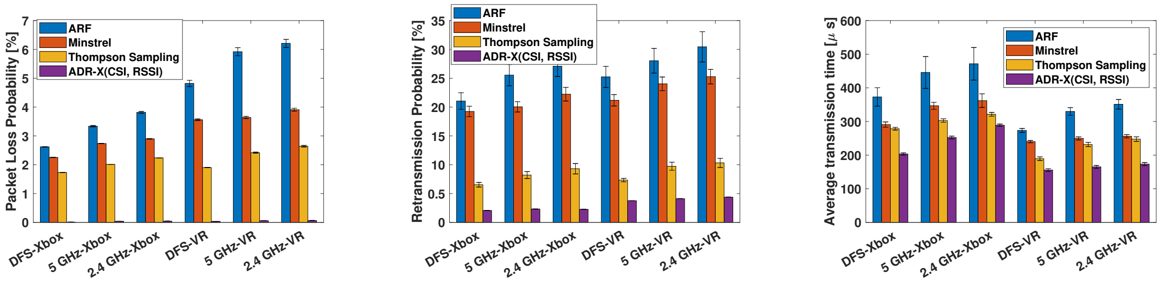
Performance of ADR-X In this section we seek to demonstrate that ADR-X performs significantly better than state of the art. While we quantify the convergence of ADR-X



(a) Comparison of avg. packet loss rates using NIC-I

(b) Comparison of avg. packet retransmission rates using NIC-I

(c) Comparison of avg. packet transmission times using NIC-I.



(d) Comparison of avg. packet loss rates using NIC-II

(e) Comparison of avg. packet retransmission rates using NIC-II

(f) Comparison of avg. packet transmission times using NIC-II

Figure 4.3: Performance of ADR-X

during learning in later sections, in this section we only compare against the steady state performance of ADR-X. We consider five state of the art adaptive data rate schemes for comparison.

ARF [77] - This is a commonly found ADR scheme in off-the-shelf Wi-Fi devices. In this scheme packet loss rate is continuously measured as a moving average. Data rate is decreased quickly in reaction of packet loss rate falling below a certain threshold and increased when the loss rate drops below another threshold. In our implementation we chose the default setup that ARF increases the rate after 10 successful transmission and decreases after two fails.

Minstrel [88] - Minstrel algorithm is typically implemented in the Linux Driver as the default ADR algorithm. It probes the maximum achievable throughput by trying each rate periodically, and chooses the rate with the maximum throughput. Minstrel dedicates 10% of its traffic to probe the performance statistics of other rates by randomly selecting a rate that is not currently in use. For retransmissions Minstrel uses the Multi-Rate Retry (MRR) chain, which proposes four candidate rates to attempt in case retransmissions are necessary.

RAM [105] - In Rate Adaptation in Mobile environments (RAM) the receiver uses the transmission rate of ACK to notify the sender about the selected rate for the next frame. RAM maintains a throughput-vs-(rate, SNR) table. It uses measured SNR, RAM looks up the table and selects the rate that can maximise throughput. To deal with high SNR fluctuation, RAM adopts a conservative SNR prediction algorithm to avoid overestimating future SNR values which may cause unnecessary frame losses and retransmissions.

ESNR [73] - Since SNR does not capture the effect of frequency-selective fading, ESNR computes the effective SNR to address this issue. Then ESNR uses the inverse mapping from BER to get the effective SNR from the SNR for each subcarrier in the OFDM system. The sender estimates the CSI and subcarrier SNR from the reverse path to calculate the effective SNR assuming channel symmetry. Given the most recent CSI measurement, the highest rate that is predicted to have a frame delivery ratio above 90% is selected.

Thompson Sampling Based Multi-Armed Bandit [97] - A multi-armed bandit [149] based explore vs exploit approach to selecting the right data rate. Thompson sampling provides for an efficient way to explore and exploit.

Packet Loss Rates. Figure 4.3(a) depicts the average packet loss rate (despite one retransmission) over a game for various ADR schemes for each of the three different interference conditions (2.4 GHz, 5 GHz and DFS channels) for NIC-I. As seen from Figure 4.3(a), all three flavors of ADR-X(CSI,SNR) and ADR-X(CSI,RSS) perform significantly better than all existing schemes. Packet loss rates of Xbox are almost close to zero ($0.02\% - 0.06\%$) while those for VR are about 0.2% in 2.4 GHz channels, indicating that these losses are due to high ambient interference from other networks. ADR-X(SNR) even while using only SNR information significantly outperforms RAM as well as all other schemes including ESNR. For

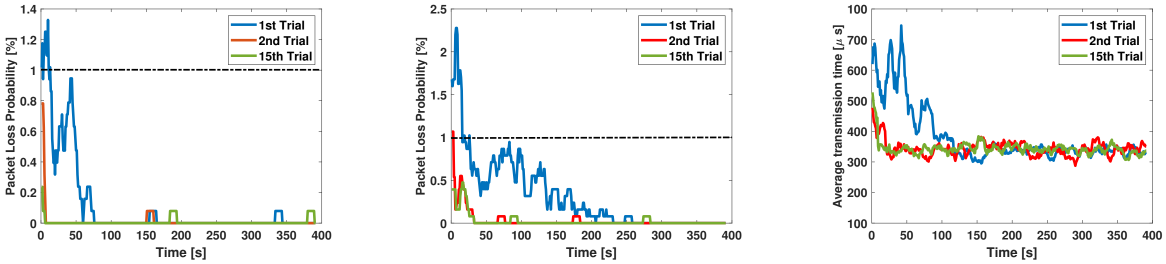
NIC-II we could not implement ESNR and RAM since the card does not provide SNR. As depicted in Figure 4.3(d), without having SNR information, ADR-X(CSI,RSSI) significantly outperforms the existing schemes.

Re-transmission Rates.ADR-X uses a conservative retry mechanism (decreasing MCS rate by 3 steps) to avoid packet losses during training time (Section 4.1.1). To demonstrate that ADR-X learns to predict rate accurately the first time without necessarily relying on the re-transmission scheme, in Figures 4.3(b) and 4.3(e), we compare the retransmission probabilities for each of the schemes on each of the two NIC cards for various interference conditions as well as VR and Xbox. As seen from Figures 4.3(b) and 4.3(e), all flavors of ADR-X provide low re-transmission rates (below 5%) in all cases, while other schemes have re-transmission rates between 5-20%.

Average Transmission Times. ADR-X must learn to choose data rates that are not so high to cause packet losses but not too low to be wasteful in terms of power consumption and bandwidth. Figures 4.3(b) and 4.3(f) compare the avg per packet transmission times for audio packets (1536 bytes) for Xbox and those for video packets for VR respectively. In case of a packet re-transmissions, the packet transmit time is measured as the sum of original and re-transmitted packet times. As seen from these figures, all three flavors of ADR-X offer the least average transmission times compared to existing schemes. ADR-X offers about ≈40% reduction compared to ARF and ≈25% compared to Minstrel. An interesting observation is that even ADR-X(SNR) using RL with only SNR offers a lower SNR than all existing schemes. The other flavors of ADR-X provide about 10% lower average transmission times compared to ESNR and RAM.

Summary of Results. In summary, ADR-X outperforms state of the art significantly with almost zero packet loss in steady state. At the same time ADR-X also has the lowest average packet transmission time compared to all existing schemes – 20-40% lower for reactive schemes and 10-20% lower for proactive schemes.

Convergence of ADR-X In this section we ask the question, *“How quickly does ADR-X learn during game play?"*. For this section, the weights of ADR-X's neural networks are



(a) Packet loss as a function game play time for ADR-X(CSI,SNR) in DFS Channels for 1st, 2nd and 15th game plays

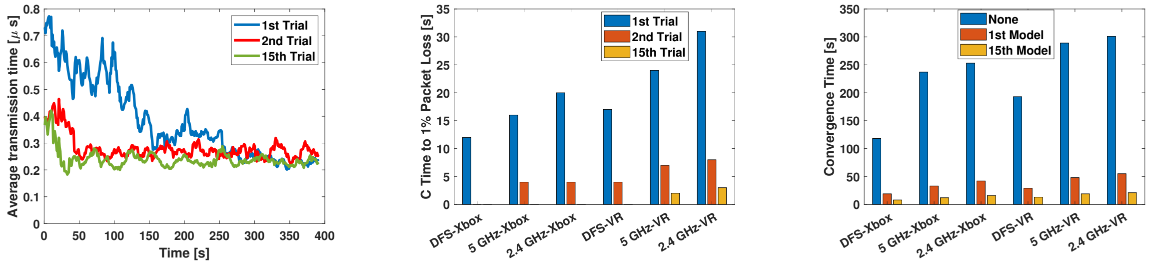
(b) Packet loss as a function game play time for ADR-X(CSI,SNR) in 2.4 GHz Channels for 1st, 2nd and 15th game plays.

(c) Avg. packet transmission time as a function game play time for ADR-X(CSI,SNR) in DFS Channels for 1st, 2nd and 15th game plays.

Figure 4.4: Convergence of ADR-X

initialized randomly and adjusted each time a channel quality feedback is received (≈8ms for Xbox and ≈16ms for VR). As the player continues, ADR-X's performance will improve and eventually converge. When the player leaves and comes back to play a game again for the second time or the n^{th} time, ADR-X remembers the latest weights from the last game play. However, the player will not have "exactly " the same conditions and position as the first time and ADR-X. Thus, each time the player start playing afresh, ADR-X will have to learn and adapt. With more experience however, ADR-X will converge faster.

In order to measure ADR-X's convergence, we conduct a series of experiments where a gamer plays 15 instances of the game, each time taking a short break (ten minutes) between each of the games by leaving the controller and walking away. The duration of each game play lasts about 20 minutes. During each instance of the game we measure packet loss rates and average packet transmission times over one second intervals. We then maintain a running average of these measurements using an exponentially weighted moving average (EWMA) with the memory parameter 0.9. We conduct the experiments for VR and Xbox scenarios for all three interference scenarios.



(a) Avg. packet transmission time as a function game play time for ADR-X(CSI,SNR) in 2.4GHz Channels for 1st, 2nd and 15th game plays.

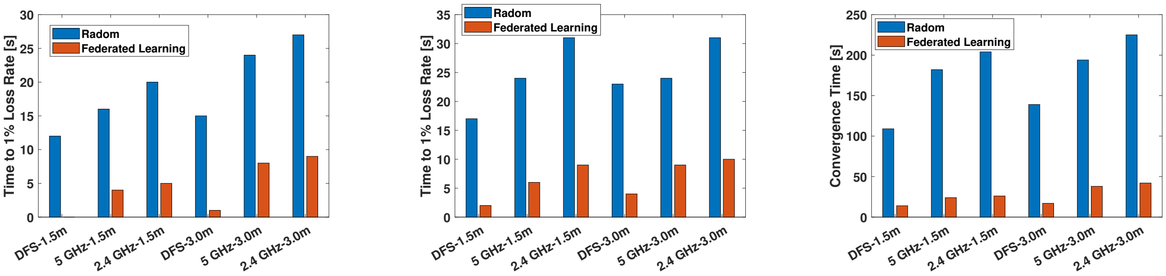
(b) Time for Packet Loss to fall below 1% for 1st, 2nd and 15th game plays

(c) Time for avg. packet transmission time to converge for 1st, 2nd and 15th game plays

Figure 4.5: Convergence of ADR-X

Packet Loss for Xbox in Interference Free Environment. Figures 4.4(a) depicts the evolution of packet loss rates with time as ADR-X learns for the 1st, 2nd and 15th instance of game play in an interference free environment (DFS channels) for audio packets. As seen from Figure 4.4(a), the first time ADR-X learns, its packet loss rate decreases to below 1% (the acceptable rate for audio packets) quickly (within about 10 seconds - 1000 transmitted packets) and continues to drop to almost zero packet losses within 75 seconds. In the second trial, ADR-X, due to its prior experience takes only several 100s of milliseconds to achieve close to 0 packet loss rates. In the 15th instance of game play, ADR-X's loss rate is always below 0.3% and is quicker to converge.

Packet Loss for Xbox in High Interference Environment. In the face of high interference from other networks, ADR-X experiences packet losses due to collisions as well as lower SINR. Since such packet losses are random and do not necessarily depend on channel state information, ADR-X takes much longer to learn the correct policy. Figure 4.4(a) depicts the the evolution of packet loss rates with time as ADR-X learns for the 1st, 2nd and 15th instance of game play in a high interference environment (2.4GHz channels) for audio packets.



(a) Times to packet loss reduction to under 1% with and without federated learning model initialization for Xbox scenarios.

(b) Times to packet loss reduction to under 1% with and without federated learning model initialization for VR scenarios.

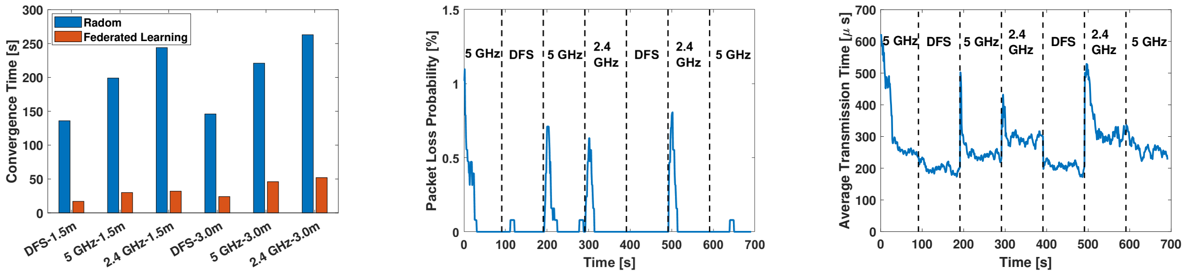
(c) Convergence with and without federated learning model initialization for Xbox scenarios

Figure 4.6: Federated Learning

As depicted in Figure 4.4(b), ADR-X takes about 20s to achieve packet loss rates below 1% and about 230s to converge to zero almost loss rate. However, the second instance of game play ADR-X is must faster to converge. It takes less than 3 seconds for the loss rate to fall below 1% about 40s to converge to zero loss rate. At the 15th instance of game play, the packet loss rate is always below 0.5%.

Avg. Packet Transmission Times for Xbox in No and High Interference Environments. Figures 4.4(c) and 4.5(a) depict the time evolution of average packet transmission times for no interference and high interference environments respectively. As seen from the figures, it takes about 150s-250s before ADR-X converges during the first instance of game play. However, the second time, it takes only 20s-50s and about 10s-20s in the 15th instance.

Summary of Results in all Situations. As seen from Figure 4.5(c) and 4.5(b), starting from no experience (random ANN weights), ADR-X takes 10s-20s to achieve low packet loss (below 1%) and converges to almost zero packet loss and low packet transmission times within 100s-250s. However, after the player has played he game several times, ADR-X is able to maintain below 1% packet loss at all times and converges to zero packet loss and low packet transmission times within 20s even in high interference conditions.



(a) Convergence with and without federated learning model initialization for VR scenarios

(b) Effect of sudden changes in interference levels on packet loss

(c) Effect of sudden changes in interference levels on average transmission times

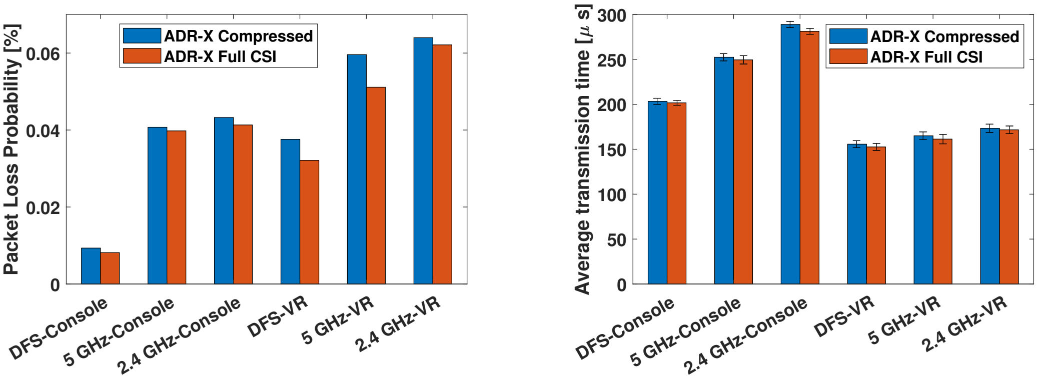
Figure 4.7: Robustness

Federated Learning with ADR-X One way to avoid a long convergence time during the first time a user plays is to use a pre-trained ANN model instead of training it from scratch (random weights) as discussed in Section 4.1.1. In this section, we try and quantify how well ADR-X performs the first time, when a federated model is used for the ANN instead of learning from scratch. To this end, we train ADR-X on six different scenarios including two different distances at which the player plays (1.5m and 3m) and three different interference conditions (No, Medium and High interference). We then generate a federated ANN model using the weights obtained from each of these six scenarios. We then use this federated to bootstrap ADR-X's ANN weights rather than using random weights.

Figures 4.6(a) depicts the time taken for the packet loss rates to fall below 1% for audio packets with and without federated learning bootstrap for the ANN weights in ADR-X for Xbox traffic and Figure 4.6(b) depicts the same for VR traffic. As seen from Figure 4.6(a) and 4.6(b), using federated learning, packet loss rate falls to below 1% within 5 seconds instead of 10s-20s with a random initialization. Figures 4.6(c) and 4.7(a) depict the convergence times for the average transmit times to converge. The convergence times with federated learning bootstrap are below 50s instead of 250s with a random initialization.

Sudden Changes in Interference Levels In this section we ask the question "How does

ADR-X adapt upon experiencing sudden changes in the ambient interference?". This can occur when the radio changes its operating channel into a highly congested channel. In case of such a sudden change, ADR-X must identify and adapt to the new conditions. To test these conditions, we change the operating channel between DFS, 5GHz and 2.4 GHz channels once every 100seconds. The radio starts at a 5GHz channel with a federated model in Section 4.1.2. Figures 4.7(b) and 4.7(c) depict the packet loss rates and average transmit power as a function of time for Xbox. As seen from these figures, each time the channel changes from to a channel with higher congestion levels, ADR-X needs to adapt once again. This is because, reinforcement learning engine in ADR-X does not explicitly use interference levels measurements as input and has to infer the strategy from rewards. However, this is more complex since it fundamentally depends on the distribution and frequency of different interferers and their traffic characteristics. We believe that this indicates that there is room for improving ADR-X in the future, where the console can measure ambient interference and use it as one of the RL engine's inputs.



(a) Comparison of packet losses of ADR-X(CSI,SNR) with and without compression of channel quality

(b) Comparison of average packet transmission times of ADR-X(CSI,SNR) with and without compression of channel quality

Figure 4.8: Channel Feedback Compression

Channel Feedback Compression In ADR-X we compress the 242 CSI values in channel quality feedback into 9 values in order to reduce overhead as well as the computational burden of the ANN used in ADR-X. In this section we quantify the loss in performance of ADR-X due to this reduction. To this end, we implemented a flavor of ADR-X that uses all the 242 CSI values with an ANN and transmits all the values in its channel feedback. Figures 4.8(a) and 4.8(b) compare the performance of ADR-X(CSI,SNR) on NIC2 with and without using compression. As seen from these figures, the loss in performance is almost negligible.

4.2 Scheduling in Multi-Link Operation for URLLC

In this section, we examine the specific traffic requirements for VR/AR and propose an innovative scheme for resource allocation utilizing the Multi-Link Operation STR features present in the next-generation Wi-Fi technology. This novel scheme significantly outperforms the baseline Proportional Fairness (PF) algorithm, offering an enhanced solution for resource allocation. We begin by providing a detailed depiction of the system model, elucidating the fundamental elements that drive the proposed solution. Following this, we present a performance analysis showing how this advanced resource allocation scheme for 802.11be MLO fulfills more stringent latency and reliability targets. This scheme particularly excels in the context of applications such as VR/AR, offering a compelling solution that enhances the performance and potential of these immersive technologies.

4.2.1 System Model

Seamless connectivity for emerging real-time applications like XR and Metaverse implies the need for provisioning Quality of Support (QoS) for data bursts, sent wirelessly from edge computing devices for rendering on a user headset as shown in Fig 4.9 while meeting strict latency and reliability requirements. In addition, the dynamic channel changes (due to even subtle user position and orientation variations) and channel congestion can significantly impact the XR experience. Thus MLO proposed in Wi-Fi 7 (802.11be) draft standard to support channel or bandwidth aggregation will result in opportunities to exploit PHY layer channel diversity for enhanced link robustness. However, how to allocate the traffic under

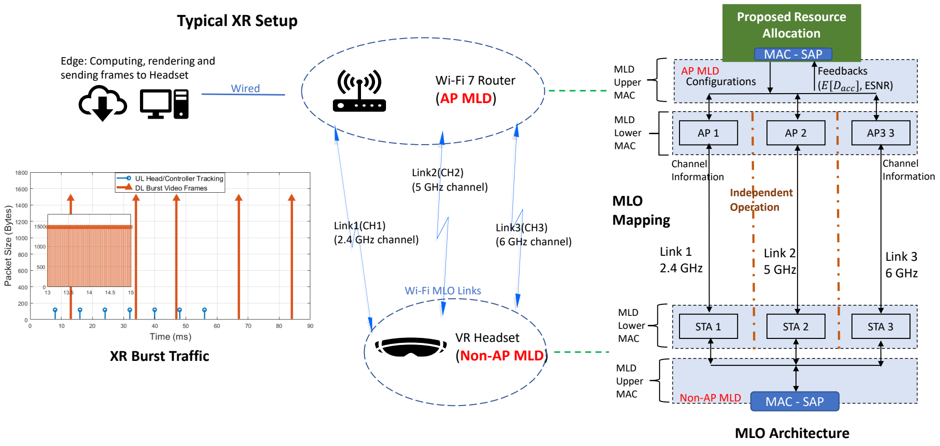


Figure 4.9: XR setup and Wi-Fi Multi-link Operation.

MLO to fulfill the requirements is largely unsolved.

Multi-link Operation 802.11be allows channel aggregation over 2.4, 5, and 6 GHz spectrum bands for simultaneous operation within a multi-link device (MLD). The XR gaming system in Fig.4.9 shows the protocol stack implementation in support of such carrier aggregation at the MAC layer. For convenience, the MAC layer functions can be divided into upper MAC (UMAC) and lower MAC (LMAC). The UMAC coordinates the distribution and management parameters across the lower MAC sublayers of the links. Hence, UMAC selects the MLD lower MAC sublayer parameters for transmission, defined as TID-to-link mapping, buffers the traffic from the upper layer before it is assigned to a specific PHY interface for transmission. The LMAC controls link-specific actions such as channel access based on distributed coordination function (DCF), rate adaptation per frame, and block acknowledgment (BACK), which operates *independently* on links in MLD via simultaneous transmit-receive (STR) mode. However, the LMAC may experience different levels of interference and channel access contention due to overlapping BSSs operating over different PHY channels; thus, UMAC should coordinate the traffic allocation among the LMAC links based on the current network and channel state information to reduce delays and improve reliability. In this work, we focus on the UMAC resource allocation function that splits the burst traffic onto the different links based on the channel information and feedback from each LMAC.

Traffic Model XR traffic present a blend of different uplink and downlink packet streams, e.g., control input (e.g., joystick position, headset orientation, etc.), haptic feedback, VR video stream, etc., which are (small) 200 Byte frames generated by the application layer periodically. Figure 4.9 depicts examples of traces collected for XR headsets where dominant traffic is video frames on the downlink (DL) comprising bursts of several consecutive video frames. The average Inter-Frame Inter-arrival (IFI) duration is $\frac{1}{F}$, where F is the refresh rate. Each Layer-2 video frame contains approximately a MAC Service Protocol Unit (MPDU). 1500 Bytes and 20 - 140 frames arrive at the MPDU queue in a burst based on application data rates [150]. In this work, we mainly consider DL burst video traffic, which can leverage simultaneous transmission via MLO.

Channel Model The end-to-end delay D_{e2e} of each frame can be split into two components: the channel access delay D_{acc} and the transmission delay T, i.e., $D_{e2e} = D_{acc} + T$. The competition for access between nodes over the same channel impacts the former, while the transmission delay depends on the modulation coding scheme (MCS), i.e., $T = \frac{Fr_{size}}{R}$, where R is the PHY rate corresponding to MCS and channel bandwidth and Fr_{size} is the frame size of the aggregated MPDUs. The burst traffic latency is the maximum latency among all the channels, determined by the worst link.

Channel Access Delay Wi-Fi channel access is based on Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA), which defines the long-standing DCF protocol. The nodes do not decrease their back-off counter until the channel is idle for a slot duration (the energy detected is less than a threshold). At each idle slot, nodes compete for channel access and may transmit when the backoff window reaches 0. If two or more nodes count down to 0 simultaneously and transmit, a collision occurs, and all packets are considered lost. In dense networks, increased competition for channel access impacts the channel access delay.

For MLO links, we assume independent channel access operating on each link by participating nodes based on DCF. We consider estimating the channel access delay as a function of channel utilization via moving average of the access delay for each frame transmitted, i.e. the difference between the timestamp instants at which the frame is sent on channel, minus the timestamp at which it arrived at head-of-line (HOL). The channel utilization is defined as the percentage of channel busy time, as determined by outcome of the physical Clear Channel Assessment (CCA)[1]. We split the utilization percentage into 5% bins to obtain estimate of the channel access delay $E[D_{acc}(i)]$ on each link i as a function of utilization.

Transmission Delay and Reliability IEEE 802.11 PHY is based on orthogonal frequency-division multiplexing (OFDM) whereby a single data stream is divided into subcarriers for simultaneous transmission, using independent coding and modulation for subcarrier bits followed by interleaving to spread coded bits in frequency [2]. Assuming that each subcarrier has the same modulation and code rate (without any prior channel knowledge at the transmitter), frequency-selective fading (as well as network interference) impacts each sub-carrier differently, leading to per-channel (per-subcarrier) Packet Error Rates (and corresponding, per-subcarrier Signal-to-Noise ratio). To estimate the Packet Error Rate (PER) over the full 20 Mhz channel bandwidth, we introduce the notion of effective signal-to-noise ratio (ESNR) based on the channel state information (CSI), which has been shown to accurately predict the PHY rate for the channel[151, 152] in our previous work.

The effective SNR maps the subcarrier SNR to a single overall ESNR value for the channel, which allows direct estimation of the full-channel PER and, in turn, selecting the MCS based on a suitable look-up table [151]. In this work, we apply exponential effective SNR mapping (EESM) for ESNR as described in [153]. For any target PER (p_e), the MCS to achieve data rate R is obtained via table lookup.

---

[1] The channel utilization is normally measured by the AP using QoS Basic Service Set (QBSS) defined in the 802.11e amendment. The QBSS information is then broadcast periodically via the beacon to all associated network devices.

[2] For example, 20 MHz channels in 802.11ax support 256 sub-carriers, while noting that *not* all are used for data transmission.

4.2.2 Dynamic Resource Allocation Scheme

We now consider the resource allocation problem with MLO to achieve low latency and high reliability for XR applications. We first introduce a cross-layer formulation and then propose a solution by relaxing the original problem to an equivalent linear program.

Problem Formulation The architecture of our proposed resource allocation scheme is shown in Fig 4.10. Considering a single Wi-Fi network with MLO-AP and one MLO-STA with M links, as in Fig. 4.9. For each link $i \in M$, we introduce the metrics discussed in the previous section as the input of our proposed scheme.

- $E[D_{acc}(i)]$: Expectation of channel access delay that is measured by the channel utilization percentage. $E[D_{acc}(i)]$ is measured by collecting the statistics from the packet transmission based on different channel utilization as discussed in Sec 4.2.1.

- $R(i)$: Data rate for certain MCS of link i based on the ESNR with target PER p_e. The ESNR can be calculated through the CSI and SNR from the receiving packets in Sec 4.2.1.

- $p_e(i)$: Packet error rate on link i based on MCS and ESNR.

Our objective is to allocate the traffic on each link so as to minimize the maximum delay over the traffic burst, as shown in Fig 4.10. Assume at the current time, a total of Φ packets (equal size MPDUs) await transmission opportunity. The output of our proposed scheme is to determine how many packets $S(i)$ (integer) to allocate to each link i and then aggregate the packets as A-MPDU to send out. Thus the total delay on each link can be expressed as $D_{e2e}(i) = E[D_{acc}(i)] + \frac{Fr_{size}(i)}{R(i)}$, where $Fr_{size}(i) = S(i) * P_{size}$, and P_{size} is UDP Packet size (constant). The delay for this burst then depends on the largest delay over all the links. Hence to minimize the maximum delay within the reliability target σ, we have:

Problem 5 (Resource Allocation Problem)**.**

$$\min_{\mathbf{S}}\{\max_{i\in M} D_i = E[D(i)] + \frac{S_i}{R_i} * P_{size}\} \tag{4.5}$$

$$s.t. \quad \sum_{i=0}^{M} \frac{p_i * S_i}{S} \leq \sigma \tag{4.6}$$

$$\sum_{i=0}^{M} S_i = S \tag{4.7}$$

$$S_i \in \{0, 1, \ldots, S\} \tag{4.8}$$

In this problem, $f_i(S_i) = E[D(i)] + \frac{S_i}{R_i} * P_{size}$ is a linear function; hence, it is trivially convex function. Function $\max\{f_1(S_1), \ldots, f_M(S_M)\}$ is a point-wise maximum. Hence the objective in (4.5) is preserves convexity. As for constraints (4.6) and (4.7) of the optimization problem, they are both linear constraints. However, constraint (4.8) is an integer constraint because packet numbers can only take integer values. Therefore, we convert this problem to linear programming (LP) by relaxing the constraint (4.8) to be a continuous constraint as follows:

$$0 \leq S_i \leq S, \;\; S_i \in \mathcal{R} \tag{4.9}$$

Then, we will have the following relaxed problem:

Problem 6 (Relaxed Resource Allocation: LP)**.**

$$\min_{\mathbf{S}}\{\max_{i\in M} f_i(S_i)\} \tag{4.10}$$

$$s.t. \quad (4.6), (4.7), (4.9) \tag{4.11}$$

The solution from LP is solved using algorithms such as the interior point method, ellipsoid method, etc. It is noteworthy that the solution must be rounded to integers for feasibility, which can deliver the near-optimal solution.

4.2.3 Performance Evaluation

We evaluate the performance of the proposed scheme based on packet-level simulation of a single network with MLO-AP and one MLO-STA. The Python-based simulator implements the basic MAC DCF operation validated against the Bianchi model and the PHY abstraction based on ESNR for IEEE channel model B, as discussed earlier. Each MLO-enabled MLD is equivalent to multiple single link devices (SLD) operating independently in parallel channels. Multiple non-MLO stations transmit (variable) Constant Bit Rate (CBR) traffic as background on each channel that alters channel utilization.

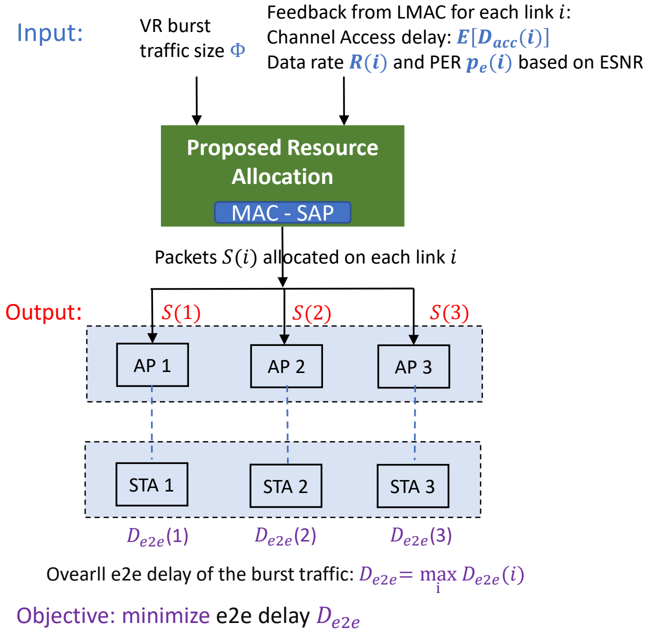


Figure 4.10: Architecture of the proposed scheme.

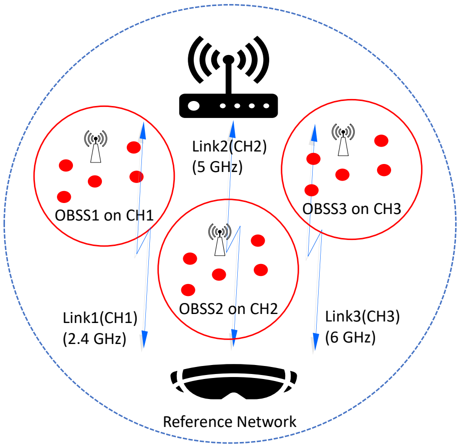


Figure 4.11: Simulation Scenario.

Simulation Setup Consider an overlapping network with N nodes from other Basic Service Sets (BSSs) on each channel, as shown in Fig. 4.11. The nodes in the overlapping BSSs (OBSSs) are transmitting uplink traffic that compete for channel access with the reference network. The reference network has one AP and one XR headset with MLO capacity. The MLO-AP and MLO-STA pair transmit burst XR traffic over $M = 3$ channels in STR mode on downlink; uplink traffic can typically be sent within one PDU. The objective is to explore the *incremental* impact of one MLD AP-STA pair traffic on the network metrics, in the

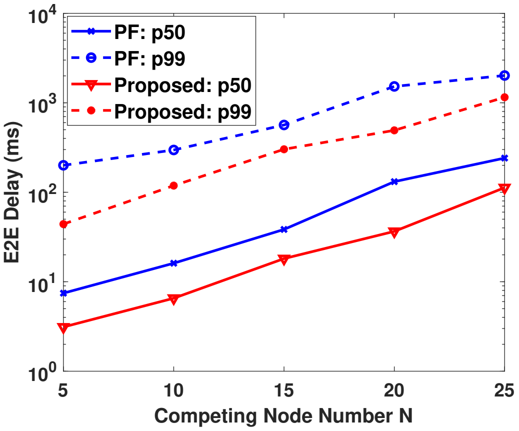


Figure 4.12: E2E delay with different competing nodes N: $\sigma \leq$1% and VR Rate 60 Mbps.

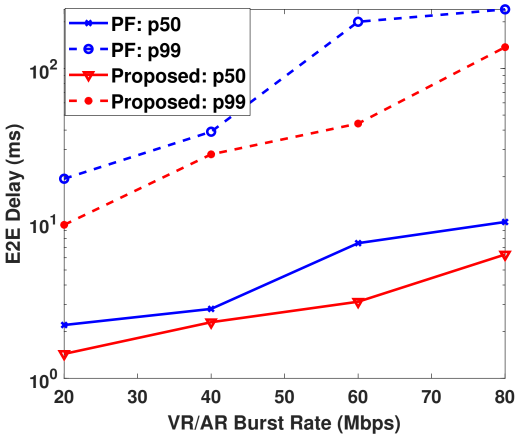


Figure 4.13: E2E Delay with different XR traffic rates: $\sigma \leq$1% and $N = 5$.

presence of background variable SLDs loading generated by the overlapping network[3]. We utilize a 2-D random walk model to track the ESNR changes during active game playing and choose the MCS based on measured ESNR to achieve a PHY packet loss rate $\leq$ 0.01. The simulation parameters are summarized in Table 4.1. For XR gaming, delay and packet loss primarily impact the user experience. Thus we consider: a typical loss target under 1%, the 50th latency percentile (p50), i.e., the user experience on average, and the 99th latency percentile (p99), i.e., the worst case user experience [154]. To compare the performance of our proposed algorithm, we implement the proportional fairness (PF) algorithm as a baseline, which allocates the packet burst size proportional to the channel bandwidth. We compare performance by considering the channel access delay $E[D_{acc}]$ for a given channel (PHY) transmission rate R.

Performance Evaluation

We analyze the Expected Access Delay for average channel utilization of 60%, 40%, and 20% on 2.4 GHz, 5 GHz, and 6 GHz channels ($M = 3$). Collisions may happen due to

---

[3] The impact of multi-user MLD scheduling is left for future work.

Table 4.1: Simulation Parameters

Parameter	**Value**
Carrier frequency	2.412 GHz, 5.190 GHz, 6.515 GHz
Channel number / bandwidth	1/20 MHz, 36/20MHz, 133/40MHz
Average channel utilization	60%, 40%, 20%
Channel coherence time	20-120 ms
MSDU payload size	1500 bytes
XR traffic rate	Various
XR refresh rate	120 Hz

multiple nodes competing for channel access with DCF, and retransmissions after collision contribute to increasing in delay. In a typical Wi-Fi deployment scenario, higher interference can be expected on the congested 2.4 GHz bands, while newly allocated 6 GHz bands are relatively free. We enabled STR for the MLD pair since the chosen channels have sufficient spectral separation to avoid IDC interference. The results are for N SLDs competing for channel access on each channel (link) with one MLD pair, and the MLDs use an aggregate bandwidth of 80 MHz over the three channels.

Scaling of Network Access Delay We first explore how XR traffic delay with MLO scales with the number of competing (overlapping BSS) nodes N and burst rates. As per Fig 4.12, as the number N of co-channel nodes increases, so does the collision probability and channel access delay. In Fig 4.13, as the XR traffic rates increase, we observe the increase of both p50 and p99 delay. In addition, as the rate increases from 40 Mbps to 60 Mbps, the p50 and p99 of the PF algorithm increase significantly because the 2.4 GHz channel may be saturated with burst traffic. Our proposed optimized resource allocation algorithm outperforms the baseline algorithm because it can allocate resources based on the channel congestion level. The optimized algorithm also results in a more smooth increase since it can allocate the application traffic to the different queues based on utilization, and better utilizes the potential benefits of MLO.

Table 4.2: Performance comparison

	p50 (ms)	p99 (ms)	loss (σ) (%)
XR Requirements [154]	5	50	1
MLO-Proposed	2.4260	21.6238	0.537
Baseline 1 (PF for MLO)	3.0934	63.4274	1.487
Baseline 2 (Single Link)	4.3463	90.1472	1.24

Comparison with SLO and utilizing channel information partially We now consider the performance with $N = 5$ and a traffic rate of 60 Mbps to evaluate the different schemes:

- Alg1: This method uses the channel access delay and ESNR.
- Alg2: Baseline 1 for MLO: PF algorithm that allocates the packets only based on the bandwidth of each link.
- Alg3: $E[D]$ only - assumes all the channels have the same link quality (ESNR).
- Alg4: R only - use the CSI to indicate ESNR and assume the same channel access delay on each link.
- Alg5: Baseline 2 for Single link operation (SLO): Replace the AP-STA in the reference network with SLD on an 80 MHz channel and 40% utilization on the 5 GHz bands with 5 nodes in the overlapping network.

The parameters in Table I results in 42 packets per burst, each with a size of 1500 Bytes. The corresponding expected channel access delays are 2.1, 1.2 and 0.3 ms for 2.4 GHz (60% utilization), 5 GHz (40%) and 6 GHz (20%) channels, respectively, in the MLO setup. The expected channel access delay for SLO is 1.7 ms on the 5 GHz (40%), because the 42 packets needs two A-MPDU frames to transmit, increasing the net access delay due to the second frame. The PF algorithm will split the burst into 10, 10 and 22 packets on 2.4 GHz, 5 GHz and 6 GHz channels, while other algorithms allocate the packets dynamically based on the channel access delay and transmission rate. The XR requirements and performance of the schemes are summarized in Table 4.2. In dense overlapping environments, SLO operation cannot fill the requirements of XR applications, especially on the p99 delay and loss rate. Though we observe the improvements on the p99 delay in MLO operation with the basic PF algorithm, it still can not meet the strict requirements. Only our proposed advanced algorithm can provide a satisfying user experience by utilizing the channel conditions and allocating the traffic dynamically. The supporting simulation results are illustrated in Fig 4.14 and Fig 4.15. The packet loss rate in Fig 4.15 for the PF algorithm with single-channel device operation, could not satisfy the loss target $\sigma \leq 1\%$. Our proposed algorithms use the loss target as the constraint, and from the resulting delay distribution in Fig 4.14 for the XR traffic, we see significant improvements over PF. Moreover, we observe the performance gains are retained when only part of the channel information (e.g., either of channel access delay and channel condition but not both) is available in Fig 4.14, suggesting inherent robustness.

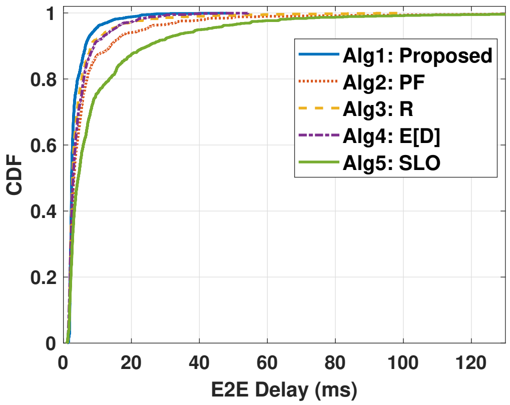


Figure 4.14: E2E delay distribution for different traffic allocation schemes.

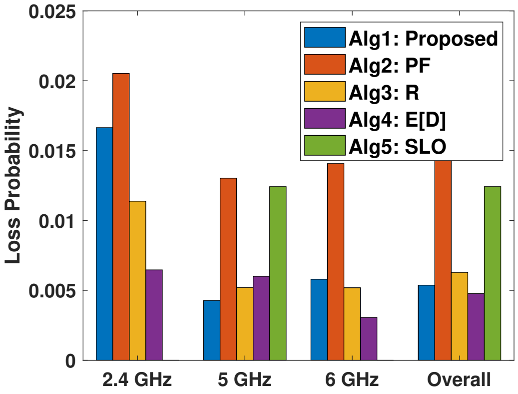


Figure 4.15: Packet losses over each channel for different traffic allocation schemes.

Chapter 5

CONCLUSION

In conclusion, this dissertation pioneers innovative solutions to some of the key challenges facing ultra-reliable and low-latency communications (URLLC) in both 5G and Wi-Fi networks. This research delves into the complex issues surrounding resource allocation and scheduling strategies, significantly advancing us towards optimized, ultra-reliable, and low-latency wireless communication services.

In the 5G New Radio domain, the dissertation provides novel methodologies for resource allocation, which are geared towards the simultaneous maximization of eMBB utility and satisfaction of URLLC constraints. The study has resulted in the development and validation of two distinct approaches—convex relaxation and a greedy algorithm. Their efficacy in balancing the needs of eMBB and URLLC users more effectively than traditional methods was demonstrated. Moreover, the integration of an entropy-based reinforcement learning algorithm with a federated learning mechanism was proven to accelerate algorithm convergence, resulting in a effective, state-of-art solution to Integrated Access and Backhaul requirements for meeting URLLC service requirements.

In Wi-Fi networks, with a focus on gaming and Virtual/Augmented Reality applications, this thesis has achieved considerable advancements in adaptive rate control and resource allocation. A novel AI-based wireless rate adaptation technique, ADR-X, is presented, which exhibits excellent potential in managing dynamic channel conditions and delivering optimal user experiences. The investigation of the Multi-Link Operation feature in Wi-Fi 7 and the proposed cross-layer MLO resource allocation algorithm, together present an effective method for achieving lower latency and higher throughput. This, in turn, creates a robust environment for Wi-Fi networks to accommodate applications requiring URLLC services.

Pivoting towards the broader impact of this thesis, AI-based methodologies which have already proven successful in areas such as Large Language Model (LLM) and Computer Vision (CV), are potentially applicable to challenging, high-dimensional problems in wireless networking as well. However, the use of AI in wireless must conform to the network protocols in place, as the computational algorithms will be closely coupled with the stack architecture for the system under study. This thesis, through its usage of AI algorithms as applied to current .11 network protocols, demonstrates initial promise via ability to solve complex network optimization problems. Practical applications such as ADR-X confirm the effectiveness of AI in real-world settings. Both 5G Open Radio Access Networks (ORAN) and Wi-Fi 8 have begun to incorporate intelligent controllers and Artificial Intelligence and Machine Learning (AI/ML), respectively. This sets the stage for continued research in this promising area, via embedding wireless network stack with powerful AI/ML algorithms to further optimize end-to-end performance, particularly for seriously latency-constrained applications.

As research directions for the future, we will further explore the ever-expanding set of new protocol features that characterize upcoming Wi-Fi 8 and 5G Advanced/6G technologies - by focusing on refining resource allocation strategies, exploring new methodologies that incorporate sophisticated AI-based learning methods to meet diverse service requirements. Similarly, for current Wi-Fi 7, we will deepen our understanding of its features to better optimize resource allocation, enhance traffic management, and reduce latency. The insights and foundations established in this thesis are expected to provide a solid platform for addressing these new challenges and further advancing the future of wireless communication.

BIBLIOGRAPHY

[1] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The next generation wireless access technology*. Academic Press, 2020.

[2] 3GPP, "Study on physical layer enhancements for NR ultra-reliable and low latency case (URLLC)," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.824, Mar 2019.

[3] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 166–174, 2017.

[4] M. Pagin, T. Zugno, M. Polese, and M. Zorzi, "Resource management for 5g nr integrated access and backhaul: a semi-centralized approach," *arXiv preprint arXiv:2102.09938*, 2021.

[5] S. M. A. Zaidi, M. Manalastas, H. Farooq, and A. Imran, "Mobility management in emerging ultra-dense cellular networks: A survey, outlook, and future research directions," *IEEE Access*, vol. 8, pp. 183 505–183 533, 2020.

[6] IEEE 802.11 Working Group, "IEEE draft standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. amendment 8: Enhancements for extremely high throughput (EHT)," *IEEE P802.11be/D2.2, October 2022*, 2022.

[7] I. P. (https://standards.ieee.org/project/802_11be.html), "Project authorization request," IEEE Standard Association, Tech. Rep., 2019.

[8] H. Yin, S. Roy, and S. Jin, "IEEE WLANs in 5 vs 6 GHz: A comparative study," in *Proceedings of the 2022 Workshop on Ns-3*, ser. WNS3 '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 25–32. [Online]. Available: https://doi.org/10.1145/3532577.3532580

[9] H. Yin, L. Zhang, and S. Roy, "Multiplexing URLLC traffic within eMBB services in 5G NR: Fair scheduling," *IEEE Transactions on Communications*, pp. 1–1, 2020.

[10] H. Yin, S. Roy, and L. Cao, “Routing and resource allocation for IAB Multi-Hop network in 5G advanced,” *Major revision to IEEE Transactions on Communications*, 2022.

[11] L. Zhang, H. Yin, S. Roy, and L. Cao, “Multiaccess point coordination for next-gen wi-fi networks aided by deep reinforcement learning,” *IEEE Systems Journal*, pp. 1–12, 2022.

[12] A. Anand and G. De Veciana, “Resource allocation and HARQ optimization for URLLC traffic in 5G wireless networks,” *IEEE IEEE J. Sel. Areas Commun.*, Nov Oct. 2018.

[13] M. Bennis, M. Debbah, and H. V. Poor, “Ultrareliable and low-latency wireless communication: Tail, risk, and scale,” *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[14] 3GPP, “Physical channels and modulation,” The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.211, Jan 2020.

[15] Samsung, “4G-5G Interworking White Paper,” Tech. Rep., Jul 2017.

[16] T. Fehrenbach, R. Datta, B. Göktepe, T. Wirth, and C. Hellge, “URLLC services in 5G low latency enhancements for LTE,” in *Proc. IEEE 88th Veh. Technol. Conf.(VTC)*, 2018, pp. 1–6.

[17] 3GPP, “Feasibility study for further advancements for E-UTRA (LTE-Advanced),” The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR36.912, Jul 2020.

[18] ——, “Study on Integrated Access and Backhaul,” The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.874, Dec 2018.

[19] ——, “NG-RAN; architecture description,” The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.401, Oct 2021.

[20] M. Polese, M. Giordani, T. Zugno, A. Roy, S. Goyal, D. Castor, and M. Zorzi, “Integrated access and backhaul in 5g mmwave networks: Potential and challenges,” *IEEE Communications Magazine*, vol. 58, no. 3, pp. 62–68, 2020.

[21] C. Madapatha, B. Makki, C. Fang, O. Teyeb, E. Dahlman, M. S. Alouini, and T. Svensson, “On integrated access and backhaul networks: Current status and potentials,” *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1374–1389, 2020.

[22] 3GPP, “Study on channel model for frequencies from 0.5 to 100 GHz,” The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.901, Jan 2020.

[23] M. Sybis, K. Wesolowski, K. Jayasinghe, V. Venkatasubramanian, and V. Vukadinovic, "Channel Coding for Ultra-Reliable Low-Latency Communication in 5G Systems," in *Proc. IEEE 84th Veh. Technol. Conf.(VTC)*, Sept. 2016.

[24] 3GPP, "Study on new radio (NR) access technology," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.912, Jun 2018.

[25] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 124–130, 2018.

[26] V. Kumar and N. B. Mehta, "Modeling and analysis of differential CQI feedback in 4G/5G OFDM cellular systems," *IEEE Trans. Wireless Commun.*, Apr. 2019.

[27] M. Angjelichinoski, K. F. Trillingsgaard, and P. Popovski, "A statistical learning approach to ultra-reliable low latency communication," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 5153–5166, 2019.

[28] 3GPP, "Physical layer procedures for data," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.214, Mar 2019.

[29] A. A. Esswie and K. I. Pedersen, "Opportunistic spatial preemptive scheduling for URLLC and eMBB coexistence in multi-user 5G networks," *IEEE Access*, vol. 6, pp. 38 451–38 463, 2018.

[30] R. Kassab, O. Simeone, P. Popovski, and T. Islam, "Non-orthogonal multiplexing of ultra-reliable and broadband services in fog-radio architectures," *IEEE Access*, vol. 7, pp. 13 035–13 049, 2019.

[31] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.

[32] C. She, C. Yang, and T. Q. S. Quek, "Joint uplink and downlink resource configuration for ultra-reliable and low-latency communications," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2266–2280, 2018.

[33] S. E. Elayoubi, P. Brown, M. Deghel, and A. Galindo-Serrano, "Radio resource allocation and retransmission schemes for URLLC over 5G networks," *IEEE J. Sel. Areas Commun.*, Apr. 2019.

[34] P. Korrai, E. Lagunas, S. K. Sharma, S. Chatzinotas, A. Bandi, and B. Ottersten, "A RAN resource slicing mechanism for multiplexing of eMBB and URLLC services in OFDMA based 5G wireless networks," *IEEE Access*, vol. 8, pp. 45 674–45 688, 2020.

[35] P. K. Korrai, E. Lagunas, S. K. Sharma, S. Chatzinotas, and B. Ottersten, "Slicing based resource allocation for multiplexing of eMBB and URLLC services in 5G wireless networks," in *Proc. IEEE 24th Int. Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2019, pp. 1–5.

[36] A. K. Bairagi, M. S. Munir, M. Alsenwi, N. H. Tran, and C. S. Hong, "A matching based coexistence mechanism between eMBB and URLLC in 5G wireless networks," in *Proc. 34th ACM/SIGAPP Symposium on Applied Computing*, ser. SAC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2377–2384.

[37] A. Anand, G. De Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," in *Proc. IEEE INFOCOM*, April 2018.

[38] A. Anand, G. de Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 477–490, 2020.

[39] S. R. Pandey, M. Alsenwi, Y. K. Tun, and C. S. Hong, "A downlink resource scheduling strategy for URLLC traffic," in *Proc. IEEE Int. Conf. on Big Data and Smart Computing (BigComp)*, Feb. 2019.

[40] M. Alsenwi, S. R. Pandey, Y. K. Tun, K. T. Kim, and C. S. Hong, "A chance constrained based formulation for dynamic multiplexing of eMBB-URLLC traffics in 5G new radio," in *Proc. Int. Conf. on Information Networking (ICOIN)*, Jan. 2019.

[41] X. Gao, G. Xing, S. Roy, and H. Liu, "Ramp-cnn: A novel neural network for enhanced automotive radar object recognition," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 5119–5132, 2020.

[42] ——, "Experiments with mmwave automotive radar test-bed," in *2019 53rd Asilomar conference on signals, systems, and computers*. IEEE, 2019, pp. 1–6.

[43] Y. Wang, Z. Jiang, X. Gao, J.-N. Hwang, G. Xing, and H. Liu, "Rodnet: Radar object detection using cross-modal supervision," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 504–513.

[44] C. Hoymann, W. Chen, J. Montojo, A. Golitschek, C. Koutsimanis, and X. Shen, "Relaying operation in 3GPP LTE: challenges and solutions," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 156–162, 2012.

[45] J. M. B. da Silva, G. Fodor, and T. F. Maciel, “Performance analysis of network-assisted two-hop d2d communications,” in *2014 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 12/7/2014 - 12/11/2014, pp. 1050–1056.

[46] J. Huang, Y. Liao, C.-C. Xing, and Z. Chang, “Multi-hop d2d communications with network coding: From a performance perspective,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2270–2282, 2019.

[47] X. Gao, S. Roy, and G. Xing, “Mimo-sar: A hierarchical high-resolution imaging algorithm for mmwave fmcw radar in autonomous driving,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7322–7334, 2021.

[48] X. Gao, S. Roy, G. Xing, and S. Jin, “Perception through 2d-mimo fmcw automotive radar under adverse weather,” in *2021 IEEE International Conference on Autonomous Systems (ICAS)*. IEEE, 2021, pp. 1–5.

[49] X. Gao, H. Liu, S. Roy, G. Xing, A. Alansari, and Y. Luo, “Learning to detect open carry and concealed object with 77 ghz radar,” *IEEE journal of selected topics in signal processing*, vol. 16, no. 4, pp. 791–803, 2022.

[50] C. Saha, M. Afshang, and H. S. Dhillon, “Bandwidth partitioning and downlink analysis in millimeter wave and backhaul for 5g,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8195–8210, 2018.

[51] G. Yang, M. Haenggi, and M. Xiao, “Traffic allocation for low-latency multi-hop networks with buffers,” *IEEE Transactions on Communications*, vol. 66, no. 9, pp. 3999–4013, 2018.

[52] Y. Liu, A. Tang, and X. Wang, “Joint incentive and resource allocation design for user provided network under 5g and backhaul networks,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 673–685, 2020.

[53] C. Zhong, Z. Lu, M. C. Gursoy, and S. Velipasalar, “A Deep Actor-Critic Reinforcement Learning Framework for Dynamic Multichannel Access,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1125–1139, 2019.

[54] H. Yang and X. Xie, “An actor-critic deep reinforcement learning approach for transmission scheduling in cognitive internet of things systems,” *IEEE Systems Journal*, vol. 14, no. 1, pp. 51–60, 2019.

[55] H. Yang, X. Xie, and M. Kadoch, “Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency IoV communication networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4157–4169, 2019.

[56] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong, "Intelligent resource slicing for embb and urllc coexistence in 5g and beyond: A deep reinforcement learning based approach," *arXiv preprint arXiv:2003.07651*, 2020.

[57] B. Zhang, F. Devoti, and I. Filippini, "Rl-based resource allocation in mmwave 5g iab networks," in *2020 Mediterranean Communication and Computer Networking Conference (MedComNet)*, 2020, pp. 1–8.

[58] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *International Conference on Machine Learning (ICML)*, 2018.

[59] "Xbox Series X," https://www.xbox.com/en-US/consoles/xbox-series-x.

[60] "Quest 2," https://www.oculus.com/quest-2/.

[61] National Telecommunications and Information Administration. (2003) Agreement reached regarding u.s. position. [Online]. Available: https://www.ntia.doc.gov/legacy/ntiahome/press/2003/5gHzAgreement.htm

[62] Z. Jiang, T. H. Luan, X. Ren, D. Lv, H. Hao, J. Wang, K. Zhao, W. Xi, Y. Xu, and R. Li, "Eliminating the barriers: Demystifying wi-fi baseband design and introducing the picoscenes wi-fi sensing platform," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4476–4496, 2022.

[63] "PicoScenes," https://ps.zpj.io/.

[64] K.-T. Chen, C.-J. Chang, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "Quadrant of euphoria: a crowdsourcing platform for qoe assessment," *IEEE Network*, vol. 24, no. 2, pp. 28–35, 2010.

[65] S. Schmidt, B. Naderi, S. S. Sabet, S. Zadtootaghaj, and S. Möller, "Assessing interactive gaming quality of experience using a crowdsourcing approach," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.

[66] G. Bingol, L. Serreli, S. Porcu, A. Floris, and L. Atzori, "The impact of network impairments on the qoe of webrtc applications: A subjective study," in *2022 14th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2022, pp. 1–6.

[67] S. Schmidt, S. Zadtootaghaj, S. Wang, and S. Möller, "Towards the influence of audio quality on gaming quality of experience," in *2021 13th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2021, pp. 169–174.

[68] A. Wahab, N. Ahmad, and J. Schormans, "Variation in qoe of passive gaming video streaming for different packet loss ratios," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–4.

[69] B. Naderi and R. Cutler, "An open source implementation of itu-t recommendation p.808 with validation," in *Proc. Interspeech 2020*, 2020, pp. 1166–1170.

[70] Netflix, "Vmaf - video multi-method assessment fusion," https://github.com/Netflix/vmaf, 2021.

[71] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005. [Online]. Available: https://www.cambridge.org/core/books/wireless-communications/800BA8A8211FBECB133A7BB77CD2E2BD

[72] R. Prasad, *OFDM for Wireless Communications Systems*. Artech House, 2004. [Online]. Available: https://books.google.com/books/about/OFDM_for_Wireless_Communications_Systems.html?id=gVE9vkreKWMC

[73] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," ser. SIGCOMM '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 159–170. [Online]. Available: https://doi.org/10.1145/1851182.1851203

[74] Qualcomm, "Qualcomm csrc9300 bluetooth & wi-fi combo chipset," 2014. [Online]. Available: https://www.qualcomm.com/products/technology/bluetooth/csrc9300

[75] Intel, "Intel ultimate n wifi link 5300," 2009. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/ultimate-n-wifi-link-5300-brief.pdf

[76] R. Crepaldi, J. Lee, R. Etkin, S.-J. Lee, and R. Kravets, "Csi-sf: Estimating wireless channel state using csi sampling & fusion," in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 154–162.

[77] A. Kamerman and L. Monteban, "Wavelan®-ii: A high-performance wireless lan for the unlicensed band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, 1997.

[78] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: A practical approach," in *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 126–134. [Online]. Available: https://doi.org/10.1145/1023663.1023687

[79] J. Kim, S. Kim, S. Choi, and D. Qiao, "Cara: Collision-aware rate adaptation for ieee 802.11 wlans," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, 2006, pp. 1–11.

[80] J. C. Bicket, "Bit-rate selection in wireless networks," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

[81] S.-C. Wang and A. Helmy, "Beware: Background traffic-aware rate adaptation for ieee 802.11," in *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2008, pp. 1–12.

[82] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 146–157. [Online]. Available: https://doi.org/10.1145/1161089.1161107

[83] I. Pefkianakis, S. H. Wong, H. Yang, S.-B. Lee, and S. Lu, "Toward history-aware robust 802.11 rate adaptation," *IEEE Transactions on Mobile Computing*, vol. 12, no. 3, pp. 502–515, 2013.

[84] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: A practical approach," in *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 126–134. [Online]. Available: https://doi.org/10.1145/1023663.1023687

[85] "PID," http://linuxwireless.sipsolutions.net/en/developers/Documentation/mac80211/RateControl/PID/.

[86] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, ser. SIGCOMM '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 3–14. [Online]. Available: https://doi.org/10.1145/1592568.1592571

[87] L. Song and S. Wu, "Aarc: Cross-layer wireless rate control driven by fine-grained channel assessment," in *2015 IEEE International Conference on Communications (ICC)*, 2015, pp. 3311–3316.

[88] "Minstrel rate control algorithm," https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel.

[89] W. Yin, P. Hu, and J. Indulska, "Rate control in the mac80211 framework: Overview, evaluation and improvements," *Computer Networks*, vol. 81, pp. 289–307, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128615000675

[90] R. Grünblatt, I. Guérin-Lassous, and O. Simonin, "Simulation and performance evaluation of the intel rate adaptation algorithm," in *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWIM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 27–34. [Online]. Available: https://doi.org/10.1145/3345768.3355921

[91] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu, "Mimo rate adaptation in 802.11n wireless networks," in *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 257–268. [Online]. Available: https://doi.org/10.1145/1859995.1860025

[92] S. Byeon, K. Yoon, C. Yang, and S. Choi, "Strale: Mobility-aware phy rate and frame aggregation length adaptation in wlans," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[93] G. Xiangyu, D. Sihao, V. Karl, D. H. Reddy, and S. Henrik, "Deformable radar polygon: A lightweight and predictable occupancy representation for short-range collision avoidance," *arXiv preprint arXiv:2203.01442*, 2022.

[94] D. Sihao, X. Gao, and H. R. Dasari, "Deformable radar polygon systems and methods for a virtual bumper," May 18 2023, uS Patent App. 17/980,618.

[95] S. Khastoo, T. Brecht, and A. Abedi, "Neura: Using neural networks to improve wifi rate adaptation," in *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 161–170. [Online]. Available: https://doi.org/10.1145/3416010.3423217

[96] C.-Y. Li, S.-C. Chen, C.-T. Kuo, and C.-H. Chiu, “Practical machine learning-based rate adaptation solution for wi-fi nics: Ieee 802.11ac as a case study,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10 264–10 277, 2020.

[97] A. Krotov, A. Kiryanov, and E. Khorov, “Rate control with spatial reuse for wi-fi 6 dense deployments,” *IEEE Access*, vol. 8, pp. 168 898–168 909, 2020.

[98] L. Kriara and M. K. Marina, “Samplelite: A hybrid approach to 802.11n link adaptation,” *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 2, p. 4–13, apr 2015. [Online]. Available: https://doi.org/10.1145/2766330.2766332

[99] G. Holland, N. Vaidya, and P. Bahl, “A rate-adaptive mac protocol for multi-hop wireless networks,” in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’01. New York, NY, USA: Association for Computing Machinery, 2001, p. 236–251. [Online]. Available: https://doi.org/10.1145/381677.381700

[100] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, “Opportunistic media access for multirate ad hoc networks,” in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’02. New York, NY, USA: Association for Computing Machinery, 2002, p. 24–35. [Online]. Available: https://doi.org/10.1145/570645.570650

[101] G. Judd, X. Wang, and P. Steenkiste, “Efficient channel-aware rate adaptation in dynamic environments,” in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 118–131. [Online]. Available: https://doi.org/10.1145/1378600.1378615

[102] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini, “Frequency-aware rate adaptation and mac protocols,” in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 193–204. [Online]. Available: https://doi.org/10.1145/1614320.1614342

[103] S. Sen, N. K. Madabhushi, and S. Banerjee, “Scalable WiFi media delivery through adaptive broadcasts,” in *7th USENIX Symposium on Networked Systems Design and Implementation (NSDI 10)*. San Jose, CA: USENIX Association, Apr. 2010. [Online]. Available: https://www.usenix.org/conference/nsdi10-0/scalable-wifi-media-delivery-through-adaptive-broadcasts

[104] X. Tie, A. Seetharam, A. Venkataramani, D. Ganesan, and D. L. Goeckel, "Anticipatory wireless bitrate control for blocks," in *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '11. New York, NY, USA: Association for Computing Machinery, 2011. [Online]. Available: https://doi.org/10.1145/2079296.2079305

[105] X. Chen, P. Gangwal, and D. Qiao, "Ram: Rate adaptation in mobile environments," *IEEE Transactions on Mobile Computing*, vol. 11, no. 3, pp. 464–477, 2012.

[106] O. Lee, J. Kim, J. Lim, and S. Choi, "Sira: Snr-aware intra-frame rate adaptation," *IEEE Communications Letters*, vol. 19, no. 1, pp. 90–93, 2015.

[107] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, "Smartla: Reinforcement learning-based link adaptation for high throughput wireless access networks," *Computer Communications*, vol. 110, pp. 1–25, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366417306370

[108] ——, "An online learning approach for auto link-configuration in ieee 802.11ac wireless networks," *Computer Networks*, vol. 181, p. 107426, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620311154

[109] R. Queiros, E. N. Almeida, H. Fontes, J. Ruela, and R. Campos, "Wi-fi rate adaptation using a simple deep reinforcement learning approach," 2022.

[110] S.-C. Chen, C.-Y. Li, and C.-H. Chiu, "An experience driven design for ieee 802.11ac rate adaptation based on reinforcement learning," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[111] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, p. 2094–2100.

[112] D. Lopez-Perez, A. Garcia-Rodriguez, L. Galati-Giordano, M. Kasslin, and K. Doppler, "IEEE 802.11be Extremely High Throughput: The Next Generation of Wi-Fi Technology Beyond 802.11ax," *IEEE Communications Magazine*, vol. 57, no. 9, pp. 113–119, Sept. 2019.

[113] W. Ahn, "Novel Multi-AP Coordinated Transmission Scheme for 7th Generation WLAN 802.11 be," *Entropy*, vol. 22, no. 12, p. 1426, 2020.

[114] M. Yang, B. Li, Z. Yan, and Y. Yan, "Ap coordination and full-duplex enabled multi-band operation for the next generation wlan: Ieee 802.11be (eht)," in *2019 11th*

International Conference on Wireless Communications and Signal Processing (WCSP), 2019, pp. 1–7.

[115] G. Lacalle, I. Val, O. Seijo, M. Mendicute, D. Cavalcanti, and J. Perez-Ramirez, “Analysis of latency and reliability improvement with multi-link operation over 802.11,” in *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, 2021, pp. 1–7.

[116] A. Lopez Raventos and B. Bellalta, “Multi-link operation in ieee 802.11be wlans,” *IEEE Wireless Communications*, vol. 29, no. 4, pp. 94–100, 2022.

[117] A. Lopez-Raventos and B. Bellalta, “Dynamic traffic allocation in ieee 802.11be multi-link wlans,” *IEEE Wireless Communications Letters*, vol. 11, no. 7, pp. 1404–1408, 2022.

[118] M. Carrascosa, G. Geraci, E. Knightly, and B. Bellalta, “An experimental study of latency for IEEE 802.11 be multi-link operation,” in *2022-IEEE International Conference on Communications*, 2022, pp. 2507–2512.

[119] J. Zhang, Y. Gao, X. Sun, W. Zhan, P. Liu, and Z. Guo, “Synchronous multi-link access in ieee 802.11be: Modeling and network sum rate optimization,” in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 2309–2314.

[120] M. Carrascosa, G. Geraci, E. Knightly, and B. Bellalta, “An experimental study of latency for ieee 802.11be multi-link operation,” in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 2507–2512.

[121] A. López-Raventós and B. Bellalta, “Multi-link operation in IEEE 802.11 be WLANs,” *IEEE Wireless Communications*, 2022.

[122] J. Zhang, Y. Gao, X. Sun, W. Zhan, P. Liu, and Z. Guo, “Synchronous multi-link access in IEEE 802.11be: Modeling and network sum rate optimization,” in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 2309–2314.

[123] A. López-Raventós and B. Bellalta, “Dynamic traffic allocation in IEEE 802.11be multi-link WLANs,” *IEEE Wireless Communications Letters*, vol. 11, no. 7, pp. 1404–1408, 2022.

[124] C. Wengerter, J. Ohlhorst, and A. G. E. von Elbwart, “Fairness and throughput analysis for generalized proportional fair frequency scheduling in OFDMA,” in *Proc. IEEE 61st Veh. Technol. Conf.(VTC)*, vol. 3, 2005, pp. 1903–1907 Vol. 3.

[125] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[126] R. Srikant and L. Ying, *Communication networks: an optimization, control, and stochastic networks perspective.* Cambridge University Press, 2013.

[127] H. A. Le Thi, T. P. Dinh, and H. Van Ngai, "Exact penalty and error bounds in dc programming," *Journal of Global Optimization*, vol. 52, no. 3, pp. 509–535, 2012.

[128] B. K. Sriperumbudur and G. R. Lanckriet, "On the convergence of the concave-convex procedure," in *Proc. 22nd Int. Conf. Neural Information Processing Systems.* Curran Associates Inc., 2009, pp. 1759–1767.

[129] M. B. Cohen, Y. T. Lee, and Z. Song, "Solving linear programs in the current matrix multiplication time," in *Proc. 51st annual ACM SIGACT symposium on theory of computing*, 2019, pp. 938–942.

[130] N. Patriciello, S. Lagen, B. Bojovic, and L. Giupponi, "An E2E simulator for 5G NR networks," *Simulation Modelling Practice and Theory*, vol. 96, p. 101933, 2019.

[131] 3GPP, "Study on channel model for frequency spectrum above 6 GHz," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.900, Jun 2018.

[132] X. Gao, S. Roy, and L. Zhang, "Static background removal in vehicular radar: Filtering in azimuth-elevation-doppler domain," *arXiv preprint arXiv:2307.01444*, 2023.

[133] X. Gao, Y. Sun, H. Chen, X. Xu, and S. Cui, "Soft actor-critic learning-based joint computing, pushing, and caching framework in mec networks," *arXiv preprint arXiv:2305.12099*, 2023.

[134] X. Gao, "Towards millimeter-wave radar signal processing and learning-based applications," 2021.

[135] S. Tosatto, C. D'Eramo, J. Pajarinen, M. Restelli, and J. Peters, "Exploration driven by an optimistic bellman equation," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[136] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[137] M. M. Wadu, S. Samarakoon, and M. Bennis, "Joint client scheduling and resource allocation under channel uncertainty in federated learning," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 5962–5974, 2021.

[138] S. Savazzi, S. Kianoush, V. Rampa, and M. Bennis, "A joint decentralized federated learning and communications framework for industrial networks," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020, pp. 1–7.

[139] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[140] 3GPP, "Study on scenarios and requirements for next generation access technologies," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR38.913, Jul 2020.

[141] ——, "Physical layer procedures," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR36.213, Dec 2020.

[142] ——, "Medium Access Control (MAC) protocol specification," The 3rd Generation Partnership Project (3GPP), Tech. Rep. TR36.321, Dec 2020.

[143] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[144] R. Patidar, S. Roy, T. R. Henderson, and A. Chandramohan, "Link-to-system mapping for ns-3 wi-fi ofdm error models," in *Proceedings of the Workshop on Ns-3*, ser. WNS3 '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 31–38. [Online]. Available: https://doi.org/10.1145/3067665.3067671

[145] "Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 1: Enhancements for high-efficiency wlan," *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020)*, pp. 1–767, 2021.

[146] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[147] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[148] "Developing Your PicoScenes Plugins," https://ps.zpj.io/plugin.html.

[149] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0196885885900028

[150] M. Lecci, M. Drago, A. Zanella, and M. Zorzi, "An open framework for analyzing and modeling XR network traffic," *IEEE Access*, vol. 9, pp. 129 782–129 795, 2021.

[151] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, p. 159–170, aug 2010.

[152] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Accurate: Constellation based rate estimation in wireless networks," in *NSDI*, vol. 4, 2010, pp. 1–15.

[153] S. Jin, S. Roy, and T. R. Henderson, "Efficient phy layer abstraction for fast simulations in complex system environments," *IEEE Transactions on Communications*, vol. 69, no. 8, pp. 5649–5660, 2021.

[154] ITU, "Functional requirements of E2E network platforms to enhance the delivery of cloud-VR services over integrated broadband cable networks," International Telecommunication Union (ITU), Tech. Rep. ITU-T Rec. J.1631, Nov 2021.

VITA

Journal articles:

H. Yin, L. Zhang and S. Roy, "Multiplexing URLLC Traffic Within eMBB Services in 5G NR: Fair Scheduling," in IEEE Transactions on Communications, vol. 69, no. 2, pp. 1080-1093, Feb. 2021, doi: 10.1109/TCOMM.2020.3035582.

H. Yin, S. Roy and L. Cao, "Routing and Resource Allocation for IAB Multi-Hop Network in 5G Advanced," in IEEE Transactions on Communications, vol. 70, no. 10, pp. 6704-6717, Oct. 2022, doi: 10.1109/TCOMM.2022.3200673.

L. Zhang, **H. Yin**, S. Roy and L. Cao, "Multiaccess Point Coordination for Next-Gen Wi-Fi Networks Aided by Deep Reinforcement Learning," in IEEE Systems Journal, vol. 17, no. 1, pp. 904-915, March 2023, doi: 10.1109/JSYST.2022.3183199.

L. Cao, S. Roy and **H. Yin**, "Resource Allocation in 5 G Platoon Communication: Modeling, Analysis and Optimization," in IEEE Transactions on Vehicular Technology, 2022, doi: 10.1109/TVT.2022.3223351.

Conference publications:

H. Yin, P. Liu, K. Liu, L. Cao, L. Zhang, Y. Gao, and X. Hei. "Ns3-ai: Fostering Artificial Intelligence Algorithms for Networking Research," in Proceedings of the 2020 Workshop on ns-3 (WNS3 '20). https://doi.org/10.1145/3389400.3389404

L. Zhang, **H. Yin**, Z. Zhou, S. Roy and Y. Sun, "Enhancing WiFi Multiple Access Performance with Federated Deep Reinforcement Learning," 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), Victoria, BC, Canada, 2020, pp. 1-6, doi: 10.1109/VTC2020-Fall49728.2020.9348485.

H. Yin, S. Roy, and S. Jin. "IEEE WLANs in 5 vs 6 GHz: A Comparative Study," in Proceedings of the 2022 Workshop on ns-3 (WNS3 '22). Association for Computing Machinery, New York, NY, USA, 25–32. https://doi.org/10.1145/3532577.3532580

Industry Experience:

1. Research Intern, Microsoft Research, Redmond, WA, Apr. 2021 - Jun. 2022, Oct.2022 - Mar. 2023.

2. Wireless research intern, Meta Platforms, Inc., Sunnyvale, CA, Jun. 2022 - Sept. 2022