NNT : 20XXSACLXXXX

Thèse de doctorat

Thesis title

Thèse de doctorat de l'Institut polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 IPP
Spécialité de doctorat: Information, Communications, Electronique

Thèse présentée et soutenue à Télécom Paris, le 4 juin 2020, par

APOSTOLOS AVRANAS

Composition du Jury :

Jean-Marie Gorce Professeur, INSA de Lyon	Rapporteur
Petar Popovski Professeur, Aarlborg University, Danemark	Rapporteur
Guiseppe Durisi Professeur, Chalmers University, Suède	Examinateur
Rémi Munoz Ingénieur de Recherche (Dr.), DeepMind	Examinateur
Michèle Wigger Professeure, Télécom Paris	Examinatrice
Philippe Ciblat Professeur, Télécom Paris	Directeur de thèse
Marios Kountouris Professeur, EURECOM	Co-directeur de thèse
Maxime Guillaud Ingénieur de Recherche (Dr.), Huawei	Invité

Contents

List of Figures

List of Tables

General Introduction

The work presented in this Ph.D. thesis has been produced thanks to the collaboration between the “Communications et Électronique” (COMELEC) department at Télécom Paris (France) and the “Huawei Technologies France” (Paris, France), within the framework of “Convention Industrielle de Formation par la Recherche” (CIFRE). It has been carried out since January 2017 under the supervision of Prof. Philippe Ciblat and Prof. Marios Kountouris.

Problem statement

The new generation of wireless systems 5G is not designed in order to just be a faster version of the predecessor LTE. It is designed to expand the utilities of the wireless networks and enable new applications. Under this scope, contrary to LTE which support only a single use case, namely Mobile broadband access (MTC), 5G expands the number of use cases and it includes the “enhanced Mobile BroadBand” (eMBB), the “massive Machine Type Communication” (mMTC) and the “Ultra-Reliable Low Latency Communications” (URLLC). The first type of connectivity, i.e. eMBB, aims to support high data rates, the second to facilitate the connection of a large number of devices to a single base station (BS) and the third to enable tasks that must be reliably performed within very short time constraints.

Even for eMBB, the new generation 5G is set to surpass the latency provided by the LTE’s single use case. From more than 10ms that MTC of LTE provides, the eMBB is restricted to around 4ms[1][1]. As the name suggests, URLLC aims to go beyond and support communication of under one millisecond [1]. **This thesis main objective is to propose schemes to facilitate the achievement of such low delays or optimize performance under stringent latency constraints.** We will first take a look at the lowest level of telecommunications which is the physical layer (PHY) and then go up one level to the Medium Access Control (MAC).

Before going further into how to accomplish such impressive latency constraints, a step back needs to be taken to pose the critical question: Is there any benefit for the communication delays to be so low? After all, 5G is a new product whose solid success depends on the existence of applications actually needing it and using it. So it is worth mentioning the deployment scenarios where acquiring so low latency is vital. In Figure 1 we see the Radiocommunication Sector of International Telecommunication Union (ITU-R) envision for the International Mobile Telecommunications of 2020 (IMT 2020).

Figure 1 encapsulates the fundamental trade-off between latency, data rate and number of

[1]The latency is measured counting only the communication between the device and the Base station (i.e. not considering delays concerning the generation or fetching the data) and assuming no time restriction due to Discontinuous reception (DRX)(i.e. both Base station and device are assumed to be fully active at any time).

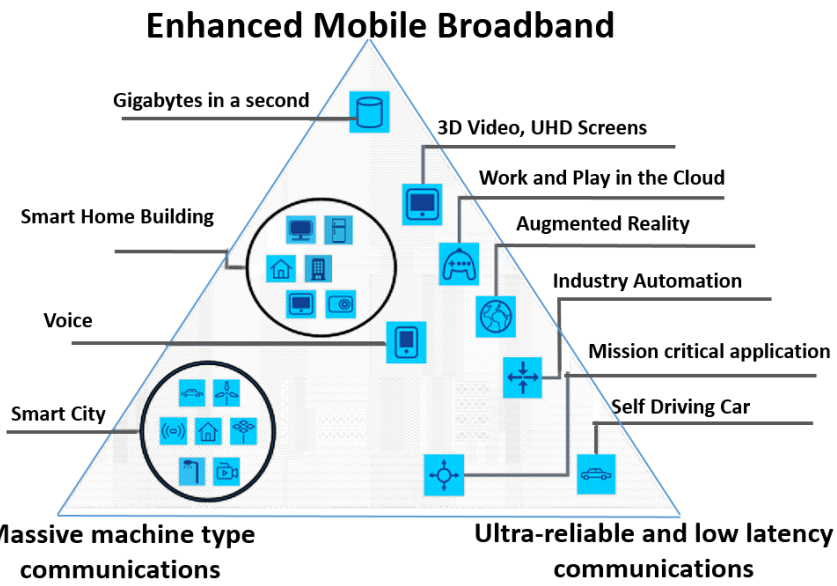

Figure 1: Designing the International Mobile Telecommunications of 2020 (Source: ITU-R IMT-2020).

devices that can be simultaneously served. The closer to the low right corner the lower latency and higher reliability but compromising on the delivered data rate and spectral efficiency is unavoidable. Concentrating a bit more on the most latency demanding application, Table 1 shows some typical values of latency that mission critical applications will require.

After establishing the necessity of low latency systems we turn to the main goal of the thesis which is **estimating the amount of resources necessary to succeed under these strict latency constraints and optimize possible schemes.** The scheme we focused on was the retransmission protocol. According to the Hybrid Automatic ReQuest (HARQ) protocol, the transmitter sends the requested packet to the receiver and then waits for. If the receiver successfully decodes it then,it sends an ACKnowledgment(ACK) back to the transmitter, otherwise a Negative ACKnowledgment (NACK). The NACK forces the transmitter to send additional information to help the receiver's decoding procedure and improving the reliability of the communication. Obviously this introduces additional delays and when operating in very limited time intervals, it is quite questionable that breaking them into smaller sub-intervals to allow a retransmission protocols remains a valid approach.

We first take a look on the lowest level of telecommunications, i.e. the physical layer (PHY). We inspect when time delays have to be very low, whether it is still beneficial the HARQ scheme and if yes how to manage the physical layer resources, i.e. the number of symbols and power each transmission should have. This problem is at first investigated under a simple setting of point to point communication where the signal is distorted only by Gaussian noise. Then we extend to the more realistic scenario of the signal experiencing fading since it can travel simultaneously through different paths so what finally arrives at the receiver is a mixture of many versions of the same signal interfering with one another. Under these multi-path phenomena is the exchange of additional information to learn the behavior of the channel, i.e. using Channel State Information (CSI), useful? It is an interesting question posed within the strict latency constraints. Finally going up to the MAC layer we move from point to point scenario to a multi user one. Many user come

Use Case	Latency	Data Rate	Remarks
Factory Automation	0.25-10 ms [2]	1 Mbps [3]	• Generally factory automation applications require small data rates for motion and remote control. • Applications such as machine tools operation may require latency as low as 0.25ms.
Intelligent Transport Systems (ITS)	10-100 ms [2]	10-700 Mbps [4]	• Road safety of ITS requires latency on the order of 10 ms. • Applicatinos such as virtual mirrors require data rates on the order of 700 Mbps
Robotics and Telepresence	1 ms [5]	100 Mbps [6]	• Touching an object by a palm may require latency down to 1 ms. • VR haptic feedback requires data rates on the order of 100 Mbps.
Virtual Reality (VR)	1 ms [7]	1 Gbps [6]	• Hi-resolution 360^oVR requires high rates on the order of 1Gbps while allowing latency of 1 ms.
Health care	1-10 ms [8]	100 Mbps [6]	• Tele-diagnosis, tele-surgery and tele-rehabilitation may require latency on the order of 1 ms with data rate of 100 Mbps
Gaming	1 ms [7]	1 Gbps [6]	• Immersive entertainment and human's interaction with the high-quality visualization may require latency of 1 ms and data rates of 1 Gbps for high performance
Smart Grid	1-20 ms [2],[7]	10-1500 Kbps [9]	• Dynamic activation and deactivation in smart grid requires latency on the order of 1 ms. • Cases such as wide area situational awareness require date rates on the order of 1500 Kbps.
Education and Culture	5-10ms [7]	1 Gbps [6]	• Tactile Interent enabled multi modal human-machine interface may require latency as low as 5 ms. • Hi-resolution 360^o and haptic VR may require data rates as high as 1 Gbps.

Table 1: Typical latency and data rate for different mission critical services (Source: [10])

and go and each wants to be satisfied within its own delay tolerance. We search for efficient resource allocation schemes that will satisfy the maximum number of users with the minimum amount of resources.

Outline and contributions

This thesis is composed of three chapters. In the first chapter we first set the system model describing a point to point communication where a fixed number of information bits has to be transmitted under requirements for low latency and high reliability. We allow the possibility the HARQ protocol with M transmissions (rounds) to be used, i.e. $M-1$ possible retransmissions, but always within the latency constraints. The goal is to assess if HARQ is useful, i.e. $M > 1$. The small time interval does not allow a large number of symbols to be transmitted, compelling to avoid standard Shannon, theory which assumes infinite number of symbols, and to resort to finite

blocklength theory. After describing the implications using this theory we cast the optimization problem to enable answering if $M > 1$ is beneficial. First we check the potential benefits under the scope of energy consumption and later of throughput. To be able to fully exploit HARQ we allow both the blocklength and power of each round of HARQ to be optimized. Since the problem does not exhibit any convenient property like convexity, we have to first mathematically analyze it to find some simplifications. Then we manage to find an algorithm to tackle using a dynamic programming based solution.

The first chapter uses the simple channel model where the signal is only distorted by Gaussian noise. Even though it provides intuition, we want proceed in the second chapter to more realistic but more complicated channel model. So under the same setup where high reliability is required even though the acceptable time delay is very low, the signal additionally to Gaussian noise it experiences fading due to multi-path interference. We separate two cases: in the first one, defined full CSI, the transmitter knows exactly the value of the channel and the second one, defined statistical CSI, only statistical properties of the channel are known. We succeed given a specific amount of resources and some statistical properties of the channel to mathematically analyze the feasibility region where it is possible to send a fixed number of information bits under a given low latency constraint. Interestingly it turns out that finding a scheme optimally exploiting the full CSI is very tough and a sub-optimal scheme can easily have smaller feasibility region than the one with statistical CSI, and without even accounting for the need of sending pilots to get a full CSI.

Finally on chapter 3 we introduce the possibility of many users simultaneously asking service from a provider. We keep the setup where a user to be satisfied, must get the demanded packet within a strict time constraint (whose value is not necessary low). Within that time interval again retransmissions are allowed. Also since 5G also categorize the users into classes (eMBB, mMTC,URLLC) which define different requirements, we introduce also in our traffic model set of classes and every user belongs to one of those. We investigate again the two separate cases where either full knowledge of the channel is provided to the transmitter for each user and every time slot or only statistical properties. To find good scheduling algorithms under those conditions we rely on tools from various fields, combinatorics (transforming the problem to a Knapsack one), Integer Linear Programming (ILP) and Optimization (by approaching as an optimization problem and either use ILP solving algorithm or Frank-Wolfe). We compare those traditional approaches against to a more recent one using the combination of Reinforcement learning and Neural Networks.

Publications

The work conducted during the years of the Ph.D. has led to the following publications. The main work of this thesis is linked to the publications [C3, C4, C5, J1] and therefore are the ones developed in this manuscript.

Peer-reviewed Journal

[J1] A. Avranas, M. Kountouris, and P. Ciblat, “Energy-latency tradeoff in ultra-reliable low-latency communication with retransmissions,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2475–2485, Nov. 2018.

International Conference

[C1] M. Kountouris, N. Pappas, and A. Avranas, “QoS provisioning in large wireless networks,” in *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2018.

[C2] M. Kountouris and A. Avranas, “Delay Performance of Multi-Antenna Multicasting in Wireless Networks,” in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Jun. 2018.

[C3] A. Avranas, M. Kountouris, and P. Ciblat, “Energy-Latency Tradeoff in Ultra-Reliable Low-Latency Communication with Short Packets,” in *2018 IEEE Global Communications Conference (Globecom)*, Dec. 2018.

[C4] ——, “Throughput Maximization and IR-HARQ Optimization for URLLC Traffic in 5G Systems,” in *2019 IEEE International Conference on Communications (ICC)*, May 2019.

[C5] ——, “The Influence of CSI in Ultra-Reliable Low-Latency Communications with IR-HARQ,” in *2019 IEEE Global Communications Conference (Globecom)*, Dec. 2019.

Chapter 1

IR-HARQ Optimization for Ultra Reliable, Low Latency Communications

1.1 Introduction

Wireless networks have traditionally been designed for increasing throughput and for improving coverage, focusing mainly on human-centric communication and delay-tolerant content. The emergence of the Internet of Things (IoT) we are experiencing nowadays, enables a transition towards device-centric communication and real-time interactive systems. Various socially useful applications and new uses of wireless communication are currently envisioned in areas such as industrial control, smart cities, augmented/virtual reality (AR/VR), automated transportation, and tactile Internet. Emerging communication systems and architectures will enable real-time connections between people and objects/machines and will be instrumental for supporting low-latency, high-fidelity, control-type applications, such as telesurgery, remote driving, and industrial remote monitoring [11], [12].

A key feature for realizing such delay-sensitive and/or mission critical applications is a network supporting very low end-to-end latency and extreme reliability. Ultra reliable, low latency communications (URLLC) is the key technology pillar in emerging mobile networks. Next-generation wireless systems (3GPP 5G) envision to support URLLC scenarios with strict requirements in terms of tolerable latency (1 millisecond) and reliability (higher than 99.999%)[13], [14].

Guaranteeing the URLLC requirements is a challenging task since the performance is constrained by fundamental tradeoffs between delay, throughput, energy, and error probability. Reducing drastically the latency imposes the use of very short messages and time-slots (mini slots). The reduced packet duration and number of channel uses lead to small blocklength channel codes, which yield faster decoding times. On the other hand, short channel codes result in a rate penalty term and transmission rates with non-zero error probability, revisiting key insights obtained via asymptotic information theoretic results. Recent progress has quantified the effect of finite blocklength, providing tight bounds and accurate normal approximation for the maximum coding rate to sustain the desired packet error probability for a given packet size [15], [16].

In order to compensate for the reliability loss introduced by short packets, reliable communica-

tion mechanisms creating diversity have to be carried. A standard technique to improve transmission reliability, which has been adopted in various wireless standards, is incremental redundancy (IR) hybrid automatic repeat request (HARQ). However, the benefits of time diversity could be rather limited under stringent latency constraints as the number of transmission rounds and channel uses is rather limited. Moreover, the benefit of feedback-based retransmissions, even with error-free but delayed feedback, is questionable since each transmit packet is much smaller due to the strict total latency constraints, thus more prone to errors.

At the expense of additional power or permitting substantial throughput degradation, it is relatively easy to shorten the delay without any compromise to meet the reliability and latency constraints. High power usage and reducing the information bits to be delivered can render even a very short transmission reliable. In the short-packet regime, this interplay is more pronounced as latency is minimized when all packets are jointly encoded, whereas power is minimized when each packet is encoded separately.

In this chapter, we investigate the interplay between latency, reliability, throughput, and energy, and investigate whether IR-HARQ can be optimized in order to be beneficial for URLLC systems. First, we fix a lower desirable throughput and analyze the fundamental tradeoff between latency (in terms of feedback and retransmission delay) and average consumed energy in the finite blocklength regime for URLLC systems with IR-HARQ. Considering that packets have to be decoded with a certain error probability and latency, we provide an answer whether it is beneficial to do one-shot transmission (no HARQ) or split the packet into sub-codewords and use IR-HARQ. Second, we propose a dynamic programming algorithm for energy efficient IR-HARQ optimization in terms of number of retransmissions, blocklength, and power per round. Furthermore, the impact of feedback delay on the energy consumption and IR-HARQ performance is also investigated. Finally, we reformulate the problem for maximizing the throughput and after some mathematical manipulations and using the same dynamic programming scheme, we derive the optimal IR-HARQ parameters that maximize throughput. The material presented in this chapter has been published in [C3, C4, J1].

1.2 Related work

Prior work has considered the problem of throughput maximization in [17] by adjusting solely the blocklength of each IR-HARQ round with only one retransmission and the optimization is done through an exhaustive search. Throughput maximization in [18] is performed via rate refinement over possibly infinite number of retransmissions of equal-sized and constant energy packets. Imposing as well a reliability constraint [19] performs rate maximization. In [20], sphere packing is used for optimizing the blocklength of every transmission with equal power. In [21], power and blocklength, as in our work but for only one packet transmission (without HARQ), are jointly tuned to minimize the energy consumed by packets scheduled in a FIFO manner. Also with rate and power adjustments [22] maximizes energy efficiency in SIMO systems. Throughput maximization for IR-HARQ problem is considered in [23] assuming infinitely large blocklength and performing blocklength adaptation.

In [24] for a variable-length stop feedback coding scheme, delay violation and peak-age violation probabilities are analyzed. Under quality of service and energy efficiency requirements the authors of [25] use full CSI to optimize the power that maximizes the effective capacity. In [26], the authors optimize the blocklength in order to maximize the rate. However, the optimization

problem considered therein is not subject to reliability and latency constraints and can easily be solved using sequential differential optimization. Finally, [27] proposes a new family of protocols and compares its throughput with a dynamically optimized IR-HARQ.

Many other work exists for optimizing HARQ mechanism but the vast majority of them consider infinite packet length, see for instance [28]–[30]. In [28], they consider a type-I HARQ with capacity-achieving codes and the blocklength is adapted for improving the throughput without any constraint on the packet reliability or latency. In [30], the authors assume infinite blocklength and consider length adaptation in order to maximize the throughput; the energy efficiency of the optimal solution is checked afterwards.

1.3 System Model

We consider a point-to-point communication link, where the transmitter sends B information bits within a certain predefined latency, which can be expressed by a certain predefined maximum number of channel uses, denoted by N. If no ARQ/HARQ mechanism is utilized, the packet of B bits is transmitted only once (one-shot transmission) and its maximum length is N. When a retransmission strategy is employed, we consider hereafter IR-HARQ with M transmissions (rounds), i.e., $M-1$ retransmissions. Setting $M=1$, we recover the no-HARQ case as a special case of the retransmission scheme. We denote n_m with $m \in \{1,2,...,M\}$ the number of channel uses for the m-th transmission.

The IR-HARQ mechanism operates as follows: B information bits are encoded into a parent codeword of length $\sum_{m=1}^{M} n_m$ symbols. Then, the parent codeword is split into M fragments of codeword (sub-codewords), each of length n_m. The receiver requests transmission of the m-th sub-codeword only if it is unable to correctly decode the message using the previous 1 to $(m-1)$ fragments of the codeword. In that case, the receiver concatenates the first until m-th fragments and attempts to jointly decode it. We assume that the receiver knows perfectly whether or not the message is correctly decoded (through CRC) and ACK/NACK is received error free but with delay. The effect of feedback error is discussed in Section 1.10. Every channel use (equivalently the symbol) requires a certain amount of time, therefore we measure time by the number of symbols contained in a time interval. The latency constraint is accounted for by translating it into a number of channel uses as follows: we have $\sum_{m=1}^{M} (n_m + D(\vec{n}_m)) \leq N$ where $\vec{n}_m$ is the tuple $(n_1, n_2, ..., n_m)$ and $D(\cdot)$ is a penalty term introduced at the m-th transmission due to delay for the receiver to process/decode the m-th packet and send back acknowledgment (ACK/NACK). The penalty $D(\cdot)$ on the m-th round may depend on the previous transmissions, i.e., $\vec{n}_{m-1}$, since IR-HARQ is employed and the receiver applies a decoding processing over the entire $\vec{n}_m$.

The channel is considered to be static within the whole HARQ mechanism, i.e., there is only one channel coefficient value for all the retransmissions associated with the same bytes. This is a relevant model for short-length packet communication and IoT applications. Indeed, for a system operating at carrier frequency $f_c = 2.5$ GHz, for a channel coherence time $T_c = 1$ ms (so equal to the URLLC latency constraint, i.e., the maximum duration of all the retransmissions associated with the same bytes), the maximal receiver speed to satisfy the static assumption is $v = cB_d/f_c \approx 180$ km/h, where $B_d = 0.423/T_c$ [31, (8.20)] is the Doppler spread and c is the speed of light. So for any device whose speed is smaller than 180 km/h, the channel is static during the HARQ process. This is a relatively high speed for most mission-critical IoT or tactile Internet applications. Therefore, our communication scenario consists of a point-to-point link with

additive white Gaussian noise (AWGN). Specifically, in m-th round, the fragment (sub-codeword) $c_m \in \mathbb{C}^{n_m}$ is received with power $P_m = \frac{||c_m||^2}{n_m}$ and distorted by an additive white circularly-symmetric complex Gaussian random process with zero mean and unit variance. As the channel is static along with the transmission, the channel gains are constant and the noise variance is assumed equal to one without loss of generality. The power allocation applied during the first m rounds is denoted by $\vec{P}_m = (P_1, ...P_m)$.

1.4 Problem Statement and Preliminaries

We first fix the number of information bits B to be delivered with a given packet error probability and under a certain latency constraint (URLLC requirements) and try to derive the best HARQ mechanism that minimizes the average consumed energy by optimally tuning both $\vec{n}_M$ and $\vec{P}_m$ for a prefixed M (number of transmissions per HARQ mechanism). Then we consider M to be variable and aim to find the optimal number of transmission rounds M for different feedback delay models. Finally, we alleviate even the assumption of the fixed number of information bits B, viewing it as an optimizable parameter so as to maximize the throughput.

The first step for reaching the above objectives is to characterize the probability of error in the m-th round of the HARQ mechanism as a function of $\vec{n}_m$ and $\vec{P}_m$. To derive the packet error probability in short-packet communication, we resort to results for the non-asymptotic (finite-blocklength) regime [15].

In IR-HARQ with $(m-1)$ retransmissions, the packet error probability or equivalently the outage probability, denoted by ϵ_m, can be expressed as $\epsilon_m = \mathbb{P}\left(\bigcap_{i=1}^{m} \Omega_i\right)$ where Ω_m is the event "the concatenation of the first m fragments of the parent codeword, with length $\vec{n}_m$ and energy per symbol $\vec{P}_m$, is not correctly decoded assuming optimal coding".

When an *infinitely* large blocklength is assumed, an error occurs if the mutual information is below a threshold and for IR-HARQ, it can easily be seen that for $k < m$ we have $\Omega_m \subseteq \Omega_k$ [32], [33], which leads to $\epsilon_m = \mathbb{P}(\Omega_m)$. In contrast, when a real coding scheme (and so *finite* blocklength) is used, the above statement does not hold anymore and an exact expression for ϵ_m seems intractable. Therefore, in the majority of prior work on HARQ (see [33] and references therein), the exact outage probability ϵ_m is replaced with the simplified ε_m defined as $\varepsilon_m = \mathbb{P}(\Omega_m)$, since ε_m and ϵ_m perform quite closely when evaluated numerically. Note that for $m = 1$ the definitions coincide and $\varepsilon_1 = \epsilon_1 = \mathbb{P}(\Omega_1)$. In the remainder of this chapter, we assume that this approximation is also valid in the finite blocklength regime as in [17], [26]. Then, ε_m can be upper bounded [15, Lemma 14 and Theorem 29] and also lower bounded as in [34] by employing the $\kappa\beta$-bounds proposed in [15]. Both bounds have the same first two dominant terms and the error probability is approximately given by

$$\varepsilon_m \approx Q\left(\frac{\sum_{i=1}^{m} n_i \ln(1+P_i) - B\ln 2}{\sqrt{\sum_{i=1}^{m} \frac{n_i P_i (P_i+2)}{(P_i+1)^2}}}\right) \tag{1.1}$$

where $Q(x)$ is the complementary Gaussian cumulative distribution function. For the sake of clarity, we may show the dependency on the variables, i.e., $\varepsilon_m(\vec{n}_m, \vec{P}_m)$ or $\varepsilon_m(n_1, n_2, ..., P_1, P_2, ...)$

instead of ε_m, whenever needed.

Notice that some works have tried to approximate more accurately the term ϵ_m or ε_m [35]–[37]. For instance, in [35], the authors provide more involved expressions for ϵ_m, but the feedback scheme considered is different from ours; the feedback time index in [35] is not predefined (it is a random variable) and is adapted online. In [26], [36], justifications for the approximation $\epsilon_m \approx \varepsilon_m$ when using non-binary LDPC codes or tail-biting convolutinoal code can be found. In [37], the authors used saddlepoint approximation to find a tight approximation of ε_m, especially for binary erasure channels (BEC). Unfortunately, no closed-form expressions are provided for AWGN channel and significant effort is required in order to adapt the saddlepoint approximation of [37] to AWGN channels and even more (if not intractable) to adopt to our problem. Therefore, we consider that using the Gaussian approximation of [15] provides a very good tradeoff between analytical tractability and tightness of the approximations.

1.5 Energy Minimization Problem

We employ an IR-HARQ with $M-1$ retransmissions with variable blocklengths and powers over rounds. We first address the problem of minimizing the average energy consumed to achieve a target reliability T_{rel} (e.g. $T_{\mathrm{rel}} = 99.999\%$ in 3GPP URLLC or equivalently an outage probability $P_{\mathrm{out}} = 1 - T_{\mathrm{rel}} = 10^{-5}$) without violating the latency constraint $\sum_{m=1}^{M} (n_m + D(\vec{n}_m)) \leq N$ by properly setting $\vec{n}_M$ and $\vec{P}_M$.

1.5.1 Optimization Problem

Letting $\varepsilon_0 = 1$, the problem of minimization of the average energy consumed by a HARQ mechanism is mathematically formulated as follows:

Problem 1.

$$\min_{\vec{n}_M, \vec{P}_M} \quad \sum_{m=1}^{M} n_m P_m \varepsilon_{m-1} \tag{1.2}$$

$$\textit{s.t.} \quad \sum_{m=1}^{M} (n_m + D(\vec{n}_m)) \leq N \tag{1.3}$$

$$\varepsilon_M \leq 1 - T_{\mathrm{rel}} \tag{1.4}$$

$$\vec{n}_M \in \mathbb{N}_+^M \tag{1.5}$$

$$\vec{P}_M \in \mathbb{R}_+^M \tag{1.6}$$

where $\mathbb{N}_{+,*}$ is the set of positive integers, and $\mathbb{R}_+$ corresponds to the set of non-negative real-valued variables.

To illustrate how feedback delay can impact the performance, we consider two different models:

- The first model assumes a constant delay per retransmission, i.e., $D(\vec{n}_m) = d$. This simple model corresponds to the current real communication systems (e.g., 3GPP LTE) where the feedback is sent back through frames that are regularly spaced in time.

- The second model assumes a non-constant delay per retransmission and that feedback is sent right after the decoding outcome at the receiver side. In that case, the limiting factor to send back the feedback is the processing time required by the receiver to decode the message. We consider this time to be proportional to the size of the set of sub-codewords the receiver has already received. Therefore, after the m-th transmission, we have $D(\vec{n}_m) = r\sum_{i=1}^{m} n_i$ with r a predefined constant.

We stress that forcing the same number of symbols per transmission, i.e. $n_m = n, \forall m$, is just a sub case of the general problem where we chose to only optimize the power per transmission. But, except otherwise stated, we hereafter we mainly focus on the general case of variable blocklength per transmission ($n_m \neq n_{m'}, \forall m, m'$) as a means to study the maximum capability of IR-HARQ to improve the performance.

Problem 1 is a Mixed Integer Nonlinear Programming (MINLP) problem and a first approach to overcome its hardness is to relax the integer constraint by looking for $\vec{n}_M \in \mathbb{R}_{+,*}^M$ instead of $\vec{n}_M \in \mathbb{N}_{+,*}^M$. Even with that relaxation, the problem remains hard in the sense that the non-linearity cannot be managed through convexity properties of the relaxed problem. Indeed, in Figure 1.1 we plot the objective function of Problem 1 for $M = 2$, $D(\vec{n}_m) = 0$ and equality in the latency and reliability constraints, i.e., (1.3) and (1.4) in order to have only a 2D search on variables (n_1, P_1). We observe that the objective function is neither convex nor quasi-convex nor biconvex, consequently standard convex optimization methods cannot be used.

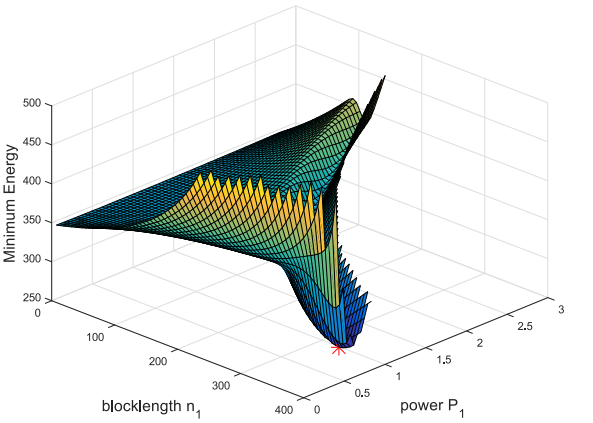

Figure 1.1: Average consumed energy versus (n_1, P_1) for $N = 400$, $B = 32$ bytes, and $T_{\text{rel}} = 99.999\%$. The red asterisk marks the minimum.

Therefore, our objective is not providing a closed-form optimal solution for Problem 1 but deriving a low complexity algorithm finding the optimal solution. But let's first start simplifying by showing that Problem 1 can be written with equality in its constraints.

1.5.2 Equality Constraints

We first start with the simple case where no delay penalty is considered $D(\vec{n}_m) = 0, \forall m$ and afterwards it will be incorporated. For this case we obtain the following result.

Result 1. *When $D(\vec{n}_m) = 0, \forall m$, the optimal solution of Problem 1 satisfies the latency constraint given by* (1.3) *and the reliability constraint given by* (1.4) *with equality.*

This result has two consequences:

(i) Equality in (1.3) and (1.4) enables us to reduce the number of variables since one n_m and one P_m can be removed from the unknown variables, i.e., we search over $2(M-1)$ instead of $2M$ variables.

(ii) Equality in (1.3) has a conceptual meaning. It implies that it is advantageous to send as many symbols as possible during transmission but with less energy used for each symbol.

In other words, *given an energy budget, it is preferable to spread this budget into many symbols with low power rather than to few ones with high power*.

Proving the above result requires the following lemmas:

Lemma 1. *The optimal solution of Problem 1, denoted by* $(\vec{n}_M^\star, \vec{P}_M^\star)$*, satisfies* $\varepsilon_{M-1} > \varepsilon_M$*.*

Proof. See Appendix of A.1 □

Lemma 2. *If* $(\vec{n}_M^\dagger, \vec{P}_M^\dagger)$ *satisfies* $\varepsilon_{M-1} > \varepsilon_M$*, then the function* $P \mapsto \varepsilon_M(\vec{n}_M^\dagger, \vec{P}_{M-1}^\dagger, P)$ *is decreasing in the neighborhood of* $P_M^\dagger$*.*

Proof. See Appendix A.2. □

Lemma 2 enables us to force the constraint (1.4) to be satisfied in equality, and so proves the second part of Result 1. To prove that, we assume that the optimal point $(\vec{n}_M^\star, \vec{P}_M^\star)$ satisfies $\varepsilon_M < 1 - T_{\text{rel}}$. According to Lemma 1, we know that $\varepsilon_{M-1} > \varepsilon_M$. Consequently, according to Lemma 2, $P_M^\star$ can be decreased to P_M' such that $\varepsilon_M < 1 - T_{\text{rel}}$ is still true (due to continuity of the function). This implies that $(\vec{n}_M^\star, \vec{P}_{M-1}^\star, P_M')$ is a better solution than the optimal one, which leads to contradiction preventing $\varepsilon_M < 1 - T_{\text{rel}}$ at the optimal point.

Passing to the second constraint (1.3), to prove that the optimal point satisfies it with equality we firstly need to establish the following result.

Lemma 3. *Let* $\mathcal{B} = \{(n_1, \cdots, n_M, P_1, \cdots, P_M) \in \mathbb{R}_{+,*}^{2M} : 0.5 > \varepsilon_1(n_1, P_1) > \varepsilon_M(n_1, \cdots, n_M, P_1, \cdots, P_M) > Q(\sqrt{2B\ln 2}/3)\}$*. As long as* $(an_1, n_2, \cdots, n_M, P_1/a, P_2, \cdots, P_M) \in \mathcal{B}$*, it is true that both* $\varepsilon_1(an_1, P_1/a)$ *and* $\varepsilon_M(an_1, n_2, \cdots, n_M, P_1/a, P_2, \cdots, P_M)$ *are decreasing with respect to* a*.*

Proof. See Appendix A.3. □

Lemma 3 enables us to force the constraint (1.3) to be satisfied in equality, and so proves the first part of Result 1 as soon as the optimal point belongs to $\mathcal{B}$, i.e., satisfies $0.5 > \varepsilon_1 > \varepsilon_M = 1 - T_{\text{rel}} > Q(\sqrt{2B\ln 2}/3)$. To prove that, we assume that the optimal point $(\vec{n}_M^\star, \vec{P}_M^\star)$ satisfies $\sum_{m=1}^M n_m^\star < N$. For any $a > 1$ such that $(an_1^\star, n_2^\star, .., n_M^\star, P_1^\star/a, P_2^\star, ..., P_M^\star) \in \mathcal{B}$ and $an_1^\star + \sum_{m=2}^M n_m^\star \leq N$ yields a better solution. And there exists at least one $a > 1$ in $\mathcal{B}$ by continuity of ε_1 and ε_M with respect to a. Actually $an_1^\star$ may belong to $\mathbb{R}_{+,*}$ instead of $\mathbb{N}_{+,*}$. To overcome this issue, we assume that the scheme with $a = (n_1^\star + 1)/n_1^\star$ is still in $\mathcal{B}$, i.e., increasing the blocklength of the first fragment by one symbol does not bring us out of $\mathcal{B}$.

We consider now the case of $D \neq 0$. The nonzero feedback delay does not modify Result 1 for the reliability constraint (1.4). For the latency constraint (1.3), the extension of Result 1 is less obvious, and the reasoning depends on the type of delay feedback model:

- For $D(\vec{n}_m) = d, \forall m$, we can simply consider Problem 1 with blocklength $N' = N - \lceil Md \rceil$, where $\lceil \cdot \rceil$ stands for the ceiling operator, and no delay penalty. Therefore the latency constraint is equivalent to the following equality:

$$\sum_{m=1}^{M} n_m = N - \lceil Md \rceil. \tag{1.7}$$

- For $D(\vec{n}_m) = r\sum_{i=1}^m n_i, \forall m$, lemma 3 should be cautiously employed. Indeed, increasing the blocklength of the first fragment by one leads to an increase in the feedback delay at

each fragment by $\lceil r \rceil$. After M transmissions, the additional delay is at most $M\lceil r \rceil$. We know that the optimal solution lies in the following interval

$$N - M\lceil r \rceil \leq \sum_{m=1}^{M} (n_m + D(\vec{n}_m)) \leq N, \tag{1.8}$$

since the right-hand side (RHS) inequality in (1.8) ensures the latency constraint, and the left-hand side inequality in (1.8) is necessary for the optimal solution. Indeed, without this inequality, it is still possible to expand the first round by one without violating the latency constraint, hence obtaining a better solution than the optimal one, which leads to a contradiction.

1.6 Simple case of $M = 2$ transmissions

We first will concentrate on the simpliest case of only one possible retransmission, i.e. $M = 2$, to get a basic sense of the behavior of the problem. We will address (i) fully optimized IR-HARQ, (ii) partially optimized IR-HARQ when $n_1 = n_2 = N/2$, and (iii) no HARQ ($n_1 = N$ and $n_2 = 0$).

1.6.1 Optimized IR-HARQ

The problem is stated as follows:

Problem 2.

$$\min_{n_1, P_1, n_2, P_2} \quad n_1 P_1 + n_2 P_2 \varepsilon_1 \tag{1.9}$$

$$\textit{s.t.} \quad n_1 + n_2 = N \tag{1.10}$$

$$\varepsilon_2 = 1 - T_{\text{rel}} \tag{1.11}$$

The constraints (1.10) and (1.11) are presented as equalities since the result 1 holds and the original form with inequalities can be replaced with one of equalities. The only concern that will be resolved later is that if it is restrictive or reasonable the optimal solution belonging to the set $\mathcal{B}$, i.e. the optimal point $(n_1^\star, P_1^\star, n_2^\star, P_2^\star)$ satisfies $0.5 > \varepsilon_1 > \varepsilon_2 > Q(\sqrt{2B \ln 2}/3)$, but this indeed applies in practice. Therefore the four optimization variables in Problem 2 are reduced to only two since indeed given (n_1, P_1), we can induce the values for (n_2, P_2). This is exactly how we created the previous Fig. 1.1, where we plot the objective function given by (1.9) with respect to (n_1, P_1) in the feasible domain defined by constraints (1.10) and (1.11).

As no convexity or quasi-convexity properties are observed and the optimization problem is reduced to simply finding a two-dimensional (2D) bounded parameter, an exhaustive search can easily do the job. More precisely, we need to quantize the power P_1 with an approximation error θ and afterwards perform a 2D search over (n_1, P_1) with $n_2 = N - n_1$ and a bisection method to retrieve P_2 (with the same approximation error θ). The bisection method is efficient since the outage probability ε is a decreasing function with respect to P_2. The complexity is $\mathcal{O}(N\theta^{-1} \log(1/\theta))$, where θ is the approximation error.

1.6.2 Partially optimized IR-HARQ

We consider here the case where the retransmission packet has the same blocklength as the first packet ($n_1 = n_2$). That case is referred to as partially optimized IR-HARQ, where the sole parameters to optimize are the power P_1 and P_2.

Problem 3.

$$\begin{aligned} \min_{P_1,P_2} \quad & \frac{N}{2}(P_1 + P_2\varepsilon_1) \\ \textit{s.t.} \quad & \varepsilon_2 = 1 - T_{\mathrm{rel}} \end{aligned}$$

Only one-dimensional exhaustive search over P_1 is needed since, once again, P_2 can be found through a bisection method solving $\varepsilon = 1 - T_{\mathrm{rel}}$. The complexity is $\mathcal{O}(\theta^{-1}\log(1/\theta))$.

1.6.3 No HARQ

Finally, if we assume that $n_1 = N$ and $n_2 = 0$ to get the one-shot transmission. As the outage is a decreasing function with P_1 (see lemma 2)), we just have to find the root in P of the equation

$$\varepsilon_1(N, P) = 1 - T_{\mathrm{rel}}. \tag{1.12}$$

A bisection method can be used, whose complexity is $\mathcal{O}(\log(1/\theta))$.

1.6.4 Asymptotic Regime for $M = 2$

As discussed previously the latency constraint (1.10) is in equality because it is preferable to exploit the complete available blocklength N. That means if the resource N was to increase to $N + 1$ then the optimal solution will surely change to include the additional spare channel use. Hence, the consumed energy for sending a fixed number of B information bits, as seen in Problem 2, is a non-increasing function with respect to the latency N. The question raised is whether this decreasing behavior has an asymptotic floor, which as seen in Fig.1.2 there is one as $N \to \infty$. Below we characterize it.

Proposition 1. *Let $(n_1^\star, P_1^\star, n_2^\star, P_2^\star)$ be the optimal point of Problem 2, $E_i = n_i^\star P_i^\star$ the energy spent on the i-th fragment and $\beta = n_1^\star/N \in (0, 1)$. The minimum average consumed energy under the constraints given by Problem 1 is independent of β when $N \to \infty$ and equals to the solution of the following optimization problem:*

$$\begin{aligned} \min_{E_1,E_2} \quad & E_1 + Q\left(\frac{E_1 - B\ln 2}{\sqrt{2E_1}}\right)E_2 \\ \textit{s.t.} \quad & E_1 + E_2 = E^{\infty}_{\mathrm{No-HARQ}} \end{aligned}$$

with $E^{\infty}_{\mathrm{No-HARQ}} = \frac{(Q^{-1}(1-T_{\mathrm{rel}}))^2}{2}\left(1+\sqrt{1+\frac{2B\ln 2}{(Q^{-1}(1-T_{\mathrm{rel}}))^2}}\right)^2$.

Proof. In result 3 the general case of $M \in \mathbb{N}_{+,*}$ is provided and a simple substitution $M = 2$ yields this proposition. □

Notice that $E^{\infty}_{\mathrm{No-HARQ}}$ corresponds to the required average energy when $N \to \infty$ for the case of no HARQ. We will later extend the result to the general case of $M > 2$.

1.6.5 Numerical Results and Discussion for $M = 2$

For the simplified case of $M = 2$, we provide here numerical results based on our analysis as a means to shed light to whether or not it is beneficial from an energy point of view to split in two the packet transmission in URLLC systems. Except otherwise stated, we set $B = 32$ bytes and $T_{\text{rel}} = 99.999\%$. According to these values, we have $1 - T_{\text{rel}} \gg Q(\frac{\sqrt{2B \ln 2}}{3}) \approx 1.7 \cdot 10^{-10}$ and it is reasonable to consider design parameters n_1 and P_1 such that $\varepsilon_1 < 0.5$. Thus forcing the parameters (n_1, P_1, n_2, P_2) to be in $\mathcal{B}$, defined by 3, is not restrictive at all; hence we consider the constraints of the optimization problems as equalities.

In Fig. 1.2, we plot the minimum average consumed energy versus N (with $D = 0$) for the two HARQ and no HARQ schemes. As stated in Proposition 3, the consumed energy for sending a

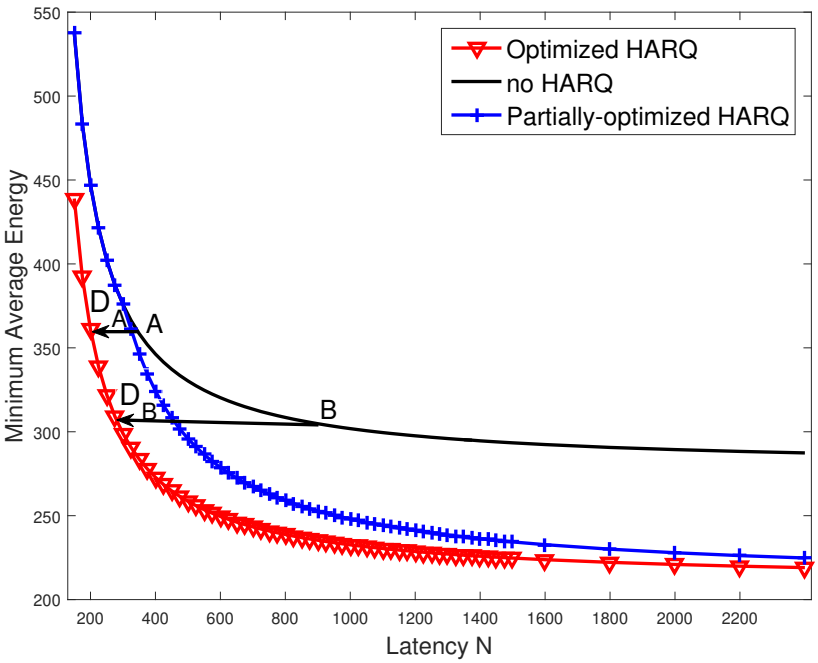

Figure 1.2: Minimum average energy versus latency N (with $D = 0$).

packet of B information bits decreases for any configuration when N increases. Nevertheless, the gain is less substantial when N is large enough since an asymptotic floor occurs. In the asymptotic regime, we have $E^{\infty}_{\text{noHARQ}} = 278$ for no HARQ and as anticipated both other configurations converge to the same smaller value $E^{\infty}_{\text{HARQ}} = 210$. Clearly, for $D = 0$, IR-HARQ always performs not worse than no HARQ. The reason is the feedback of the IR-HARQ mechanism enables, at times, to only need a portion of the total available blocklength N, i.e. n_1 channel uses, and thus saving energy by not using the remaining n_2. Of course all these due to the free of charge use of feedback as $D = 0$.

The effect of feedback delay D on the performance it is possible to be observed in Fig. 1.2 by taking two example points $A(N_A, E_A)$ and $B(N_B, E_B)$ where N_A (resp. N_B) is the minimum satisfied latency for a given consumed energy E_A, (resp. E_B) when no HARQ is used. Using optimized IR-HARQ can lead to lower the necessary latency $N'_A = N_A - D_A$ (resp. $N'_B = N_B - D_B$) for the same amount of energy. Consequently, D_A (resp. D_B) is the latency gain of optimized IR-HARQ against no HARQ. In other words, optimized IR-HARQ can support a feedback delay $D < D_A$ (resp. $D < D_B$) while offering gain in terms of energy consumption when

this energy is upper bounded by E_A (resp. E_B). It can also be seen that as the initial available increases from N_A to N_B the acceptable maximum delay grows much faster going from D_A to D_B. This will be reconfirmed from Fig. 1.4.

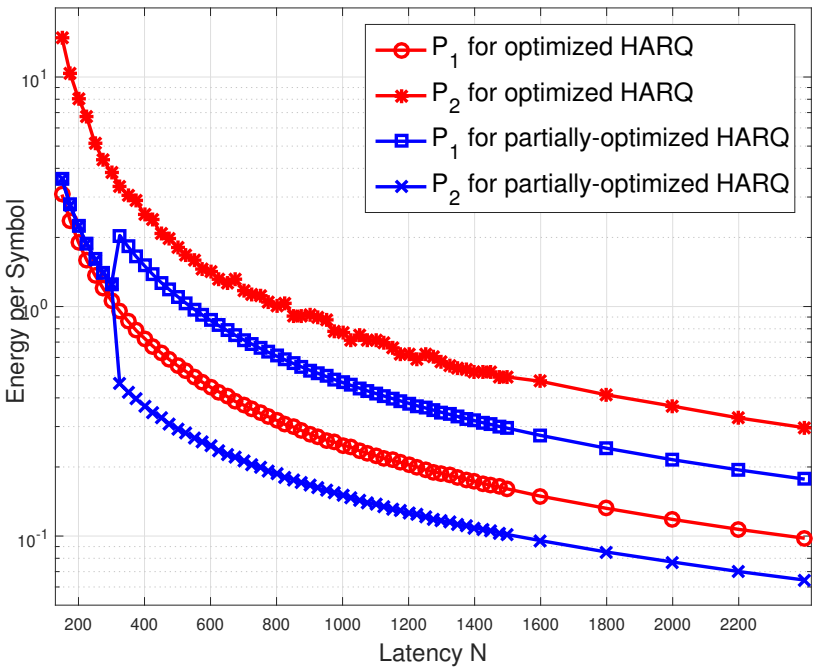

Figure 1.3: Power allocation for minimum average energy in both optimized and partially optimized IR-HARQ.

In Fig. 1.3, we show the optimal power allocation $(P_1^\star, P_2^\star)$ versus the latency N for both IR-HARQ configurations. We see that in optimized IR-HARQ we always have $P_1^\star < P_2^\star$, and on the contrary in partially optimized IR-HARQ, we have $P_1^\star > P_2^\star$ for large N but $P_1^\star = P_2^\star$ for small N. But $P_1^\star = P_2^\star$ is the equivalent case to no HARQ and this explains the blue and black curves of Fig. 1.2 to coincide for small N. The reason of this behaviour is based on the fact that HARQ has the benefit of *early termination* offering the possibility of no retransmission, thus saving power-blocklength resources. The advantage of HARQ, as compared to no HARQ, is more pronounced when early termination occurs very frequently, i.e. low ε_1, but without sacrificing a large amount of energy for doing so, i.e. when both ε_1 and $n_1 P_1$ are small. In the no HARQ case, a smaller error can be achieved even by decreasing the energy (increasing the available blocklength leads to even less needed energy - see the curve for no HARQ in Fig. 1.2 as N grows). Therefore, both $n_1 P_1$ and ε_1 can be kept small by increasing n_1 and decreasing P_1 in the optimized IR-HARQ. That's why we get that the optimal $(n_1^\star, n_2^\star)$ leads to $n_1^\star > n_2^\star$ (specifically $n_1^\star \approx 0.89N$ for almost any value of N), and that P_1^* is small compared to P_2^*. In contrast, in the partially optimized IR-HARQ, one cannot adapt n_1. Therefore, decreasing ε_1 depends on the available N. If N is inadequate, then decreasing ε_1 requires excessively high P_1, which yields to an inefficient solution. That is why for small N, ε_1 is almost 1 (i.e. retransmission should always be employed) and the behavior of IR-HARQ is similar to no HARQ. When N becomes sufficiently large, then the only solution for decreasing ε_1 is increasing P_1. That is why $P_1^\star > P_2^\star$ in the partially optimized IR-HARQ case.

In Fig. 1.4, we plot the difference (in percentage) between the energy consumed in no HARQ and the optimal average energy consumed in IR-HARQ versus D. Positive gains mean that an

IR-HARQ mechanism performs better than no HARQ. We observe that the splitting approach

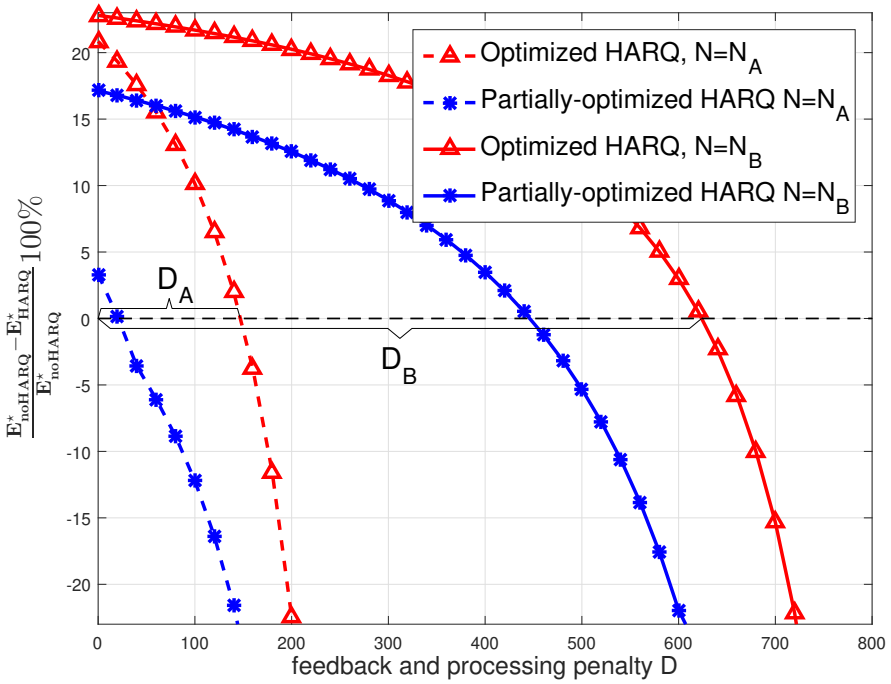

Figure 1.4: $(E^{\star}_{\text{no HARQ}} - E^{\star}_{\text{HARQ}})/E^{\star}_{\text{no HARQ}}$ in $\%$ versus D for $N_A = 350$ and $N_B = 900$.

(optimized IR-HARQ) is better than one-shot transmission (no splitting) for a large amount of feedback delay D (around $N/2$ or even more). As N increases, the amount of feedback delay that optimized IR-HARQ can support while being more energy efficient also increases. For example, when $N = N_A = 330$, we have $D_A = 0.42N_A$, whereas when we increase to $N = N_B = 900$ we have $D_B = 0.69N$. Therefore, as N grows, IR-HARQ becomes a more robust solution with respect to feedback delay. We note also that an unoptimized or partially optimized IR-HARQ does not necessarily provide better performance than no HARQ, even with almost zero feedback delay especially as the available N shrinks which is the case of URLLC systems.

1.7 Is increasing M further a good idea?

Before proceeding with tackling the general case of $M > 2$, we need some indication that this could be beneficial. For that we prove the following result, but which only holds when $D(\vec{n}_m) = 0, \forall m$.

Result 2. *When $D(\vec{n}_m) = 0, \forall m$, and given T_{rel} and N, increasing the number of retransmissions M always yields a lower optimal average energy.*

Proof. See Appendix A.4. □

Result 2 implies that when ideal feedback and no delay are guaranteed, a HARQ mechanism is always advantageous, i.e., it is always preferable to split the sub-codewords into even smaller sub-codewords.

1.8 Low Complexity Algorithm with Dynamic Programming

Until now, we have managed to reduce the set of feasible points without losing optimality (as established from Result 1) and as a consequence, the search for the optimal solution of Problem 1 has been simplified. Nevertheless, due to the lack of convexity or other favorable properties for the objective function, an exhaustive search seems to be required as has be done in the simple case of $M = 2$. That involves the need for power quantization, which introduces an approximation error (denoted by θ). The procedure is as follows: first, $\vec{n}_{M-1}$ and $\vec{P}_{M-1}$ are fixed; then, n_M is obtained through (1.3) with equality, and P_M is subsequence obtained through a bisection method for solving (1.4) with equality. The bisection method is possible since Lemma 2 establishes the monotonicity of ε_M. Finally, it remains to perform a $2(M-1)$-D exhaustive search to solve Problem 1. The described brute force algorithm yields a complexity in $\mathcal{O}(N^{M-1}(1/\theta)^{M-1}\log(1/\theta))$. If M is small enough (typically less than 3), the algorithm can be implemented. However, when M is large, performing exhaustive search is prohibitively costly and an alternative approach is required. For that, we propose an algorithm based on dynamic programming (DP). We start from the case of zero delay feedback.

We assume the optimal solution to belong in $\mathcal{B}$ (as stated in Lemma 3) so (1.3) and (1.4) become equalities. Let the state at the end of the round m

$$S_m = (N_m, V_m, c_m)$$

with $N_m = \sum_{i=1}^{m} n_i$, $V_m = \sum_{i=1}^{m} n_i P_i(P_i+2)/(P_i+1)^2$, and $c_m = Q^{-1}(\varepsilon_m)$. The state sequence forms a Markov chain, i.e., $p(S_m|S_{m-1}, \cdots S_1) = p(S_m|S_{m-1})$ since we have

$$N_m = N_{m-1} + n_m \tag{1.13}$$

$$V_m = V_{m-1} + n_m\left(1 - \frac{1}{(P_m+1)^2}\right) \tag{1.14}$$

$$c_m = \frac{c_{m-1}\sqrt{V_{m-1}} + n_m \ln(1+P_m)}{\sqrt{V_m}} \tag{1.15}$$

and the way to go from S_{m-1} to S_m depends only on the current round m through n_m and P_m. Notice that the assumption in Lemma 3 ensures $c_M = Q^{-1}(1-T_{\text{rel}})$ and $0 \le c_1 \le c_M \le \sqrt{2B\ln 2}/3$, while Result 1 ensures $N_M = N$.

The idea comes from the fact that the m first components of the objective function can be written as follows

$$\sum_{i=1}^{m} n_i P_i \varepsilon_{i-1} = \sum_{i=1}^{m-1} n_i P_i \varepsilon_{i-1} + \Delta E(S_{m-1}, S_m) \tag{1.16}$$

where $\Delta E(S_{m-1}, S_m) = n_m P_m \varepsilon_{m-1}$. Let $E^\star(S_m)$ be the minimum average energy going to the state S_m. According to (1.16), it is easy to prove that

$$E^\star(S_m) = \min_{\forall \text{ possible } S_{m-1}} \{\Delta E(S_{m-1}, S_m) + E^\star(S_{m-1})\} \tag{1.17}$$

since our problem boils down to the dynamic programming framework, and so Viterbi's algorithm can be used.

Compared to the exhaustive search, the complexity is significantly reduced, but can be still

very large depending on the number of states S_{m-1} and S_m that has to be tested in (1.17). First, we see that the set of states S_m for $m \in \{1, \cdots, M\}$ is not $\mathbb{R}^3$ but a much smaller set. Indeed:

$$N_m \in \mathcal{N}_d = \{1, 2, ..., N\} \tag{1.18}$$

$$V_m \in \mathcal{V}_d = (0, \min(N_m, c_m + \sqrt{c_m^2 + 2B \ln 2})) \tag{1.19}$$

$$c_m \in \mathcal{C}_d = [0, Q^{-1}(1 - T_{\text{rel}})] \tag{1.20}$$

The (1.19) holds as a combination of $\sum_{i=1}^{m} n_i \ln(1+P_i) - B \ln 2 = c_m \sqrt{V_m}$ and $\sum_{i=1}^{m} n_i \ln(1+P_i) \geq V_m/2$ (which stems from the inequality $P(P+2)/(1+P)^2 < 2\ln(1+P)$). Now we can assert $V_m/2 - B \ln 2 \leq c_m \sqrt{V_m}$ and so $V_m \leq c_m + \sqrt{c_m^2 + 2B \ln 2}$. For (1.20), we need the next Lemma:

Lemma 4. *If $D(\vec{n}_m) = 0$ then the optimal solution $(\vec{n}_M^\star, \vec{P}_M^\star)$ satisfies $\varepsilon_1 > \varepsilon_2 > ... > \varepsilon_M$, and so $c_1 < c_2 < ... < c_M$.*

Proof. See Appendix A.5. □

Now focusing on the S_{m-1} case, we straightforwardly have

$$(m-1)n_{min} \leq N_{m-1} \leq N_m - n_{min} \tag{1.21}$$

$$V_m - n_m \leq V_{m-1} \leq \min\{V_m, N_{m-1}\} \tag{1.22}$$

where n_{min} is the minimum blocklength of the transmitted packets. Finally, given the target S_m and (N_{m-1}, V_{m-1}) there is at most one feasible c_{m-1} which emerges from (1.14)-(1.15)

$$c_{m-1} = \frac{c_m \sqrt{V_m} + 2(N_m - N_{m-1}) \ln\left(1 - \frac{V_m - V_{m-1}}{N_m - N_{m-1}}\right)}{\sqrt{V_{m-1}}}. \tag{1.23}$$

Let us now focus on the initialization. When $M = 1$, the states S_1 are 2D since given (N_1, c_1) there can be only one feasible P_1 (and so V_1) which satisfies the equation $\varepsilon_1(N_1, P_1) = Q(c_1)$. Therefore we start from $M = 2$. To find $E^\star(S_2)$, we need to minimize over only one variable (N_1), which renders this case computationally easier. Formally,

$$\begin{aligned} E^\star(N_2, V_2, c_2) = \min_{N_1} \quad & N_1 P_1 + n_2 P_2 \varepsilon_1(N_1, P_1) \\ \text{s.t.} \quad & n_2 = N_2 - N_1 \\ & V_2 = \frac{N_1 P_1 (P_1 + 2)}{(P_1 + 1)^2} + \frac{N_2 P_2 (P_2 + 2)}{(P_2 + 1)^2} \\ & \varepsilon_2(N_1, P_1, n_2, P_2) = Q(c_2). \end{aligned} \tag{1.24}$$

The DP-based algorithm is summarized at algorithm 1 where some technicalities, such as the dependence of $\mathcal{V}_d$ from N_m and c_m are neglected for simplicity and exposition clarity.

Letting the approximation error due to quantization of V and c be θ_V and θ_c, respectively, then the complexity is of order $\mathcal{O}(MN^2(\frac{1}{\theta_V})^2 \frac{1}{\theta_c})$. In other words, the complexity of the dynamic programming algorithm is linear with respect to M, whereas the complexity of exhaustive search is exponential in M.

Extension of the above algorithm to the case of non-zero delay is easy when $D(\vec{n}_m) = d$ since we can simply reconsider the problem as having available blocklength $N' = N - \lceil Md \rceil$

Algorithm 1 DP-based Avg. Energy Minimization ($N, T_{\rm rel}, M$)

quantize $\mathcal{N}_d, \mathcal{C}_d, \mathcal{V}_d$ **into** N_q, C_q, V_q
create structure E **of dimensions** $|N_q|\times|C_q|\times|V_q|\times(M-1)$
for all $S_2 \in N_q \times C_q \times V_q$ **do**
  $\mathrm{E}(S_2, 1) \leftarrow \infty$
  if (1.24) is feasible **then**
    $\mathrm{E}(S_2, 1) \leftarrow E^\star(N_2, c_2, V_2)$
  end if
end for
for $m := 3$ **to** M **do**
  for all $S_m \in N_q \times C_q \times V_q$ **do**
    $\mathrm{E}(S_m, m-1) \leftarrow \infty$
    for all $(N_{m-1}, V_{m-1}) \in N_q \times V_q$ satisfying (1.21) and (1.22) **do**
      use (1.23) **to find** c_{m-1} **and after** P_m, n_m
      let $\tilde{c}_{m-1} \in C_q$ **be the closest to** c_{m-1}
      $E_{temp} \leftarrow n_m P_m Q(c_{m-1}) + \mathrm{E}((N_{m-1}, V_{m-1}, \tilde{c}_{m-1}), m-2)$
      if $E_{temp} < \mathrm{E}(S_m, m-1)$ **then**
        $\mathrm{E}(S_m, m-1) \leftarrow E_{temp}$
      end if
    end for
  end for
end for
Output $\min_V \{\mathrm{E}((\mathrm{N}, \mathrm{Q}^{-1}(1-\mathrm{T}_{\rm rel}), \mathrm{V}), \mathrm{M}-1)\}$

and no delay penalty. When $D(\vec{n}_m) = r\sum_{i=1}^{m} n_i$ more changes are required: first, N_m now represents the available latency at the m-th round, second, an additional data structure $\mathrm{N}_{\rm net}$ is needed which stores the number of symbols sent disregarding the delays, and third to find every $E^\star((N_m, V_m, c_m))$ an additional search within the states $(N, V_m, c_m), \forall N \in [N_m - m\lceil r \rceil, N_m - 1]$ is employed.

1.9 Asymptotic Regime ($M \in \mathbb{N}_{+,*}$)

In the section 1.6.4 we argued that the minimum average energy for sending a fixed number of B information bits is a decreasing function with respect to the latency N in the case of $M = 2$. The same reasoning stands also to the general case of $M \in \mathbb{N}_{+,*}$). A different way to confirm it is that, as seen in Problem 1, the optimal solution for a given N is a feasible solution of $(N+1)$ and so equal or worse than the optimal solution for the latency $(N+1)$. In following result, we extend the lemma 1 regarding the asymptotic regime of $N \to \infty$.

Result 3. *When $N \to \infty$, the minimum average energy of Problem 1 for fixed M is given by*

$$\begin{aligned} E^\star_{as}(M, B, T_{rel}) \quad &= \quad \min_{(E_1, \cdots, E_M)} \quad r(E_1, \cdots, E_M) \\ &\textit{s.t.} \quad \sum_{m=1}^{M} E_m = E^{\infty}_{\mathrm{No-HARQ}} \end{aligned} \tag{1.25}$$

with

$$r(E_1,\cdots,E_M)=E_1+\sum_{m=2}^{M} Q\left(\frac{\sum_{i=1}^{m-1}E_i-B\ln 2}{\sqrt{2\sum_{i=1}^{m-1}E_i}}\right)E_m \tag{1.26}$$

and $E^{\infty}_{\mathrm{No-HARQ}}=\frac{(Q^{-1}(1-T_{\mathrm{rel}}))^2}{2}\left(1+\sqrt{1+\frac{2B\ln 2}{(Q^{-1}(1-T_{\mathrm{rel}}))^2}}\right)^2$.

Proof. See Appendix A.6. □

Again $E^{\infty}_{\mathrm{No-HARQ}}$ corresponds to the required average energy when $N\to\infty$ for the case of no HARQ and can also be obtained from [15, eq.(4.309)]. It is worth mentioning when $N\to\infty$, a non-zero delay feedback $D>0$ - irrespectively of the model considered - does not impact the asymptote value since the latency constraint (1.3) vanishes and with it the terms $D(\vec{n}_m)$, which makes Result 3 still hold.

Allowing M to also grows to infinity, yields an additional result.

Result 4. *When $M\to\infty$, the asymptotic minimum average energy stated in Result 3 behaves as follows:*

$$\lim_{M\to\infty} E^{\star}_{as}(M)=\int_0^{E^{\infty}_{\mathrm{No-HARQ}}} Q\left(\frac{E-B\ln 2}{\sqrt{2E}}\right)\mathrm{d}E. \tag{1.27}$$

Proof. See Appendix A.7. □

As an illustration, in Figure 1.5 we plot $E^{\star}_{as}(M,B,T_{\mathrm{rel}})$ versus M for different B and T_{rel}. We also plot two curves corresponding to the minimum energy, one given in [35, Theorem 3] for no feedback ("no-fb" in the figure) and the other ("stop-fb" in the figure) given in [35, Theorem 10] where ACK/NACK feedback is sent after the transmission of each symbol. Actually, the "no-fb" line corresponds to our case $M=1$ when removing its third-order term. The "stop-fb" line is close to our eq.(1.27) since its adaptive feedback can be mimicked in our case if infinite available number of transmissions are considered.

Given B, increasing T_{rel} to a new value $\overline{T}_{\mathrm{rel}}$ also increases $E^{\infty}_{\mathrm{No-HARQ}}$ to $\overline{E}^{\infty}_{\mathrm{No-HARQ}}$. This in turn implies $\lim_{M\to\infty}E^{\star}_{as}(M)<\lim_{M\to\infty}\overline{E}^{\star}_{as}(M)$. In Figure 1.5, these limit values cannot be distinguished and seem to coincide since they are very close to each other. This happens because, as it easily can be shown, $\lim_{M\to\infty}\overline{E}^{\star}_{as}(M)-\lim_{M\to\infty}E^{\star}_{as}(M)<(1-T_{\mathrm{rel}})(\overline{E}^{\infty}_{\mathrm{No-HARQ}}-E^{\infty}_{\mathrm{No-HARQ}})$ and T_{rel} is close to zero.

1.10 Numerical Results and Discussion

We now provide numerical results to validate our analysis in the general case of $M\in\mathbb{N}_{+,*}$. As in section 1.6.5 for $M=2$, we consider $B\geq 32$ bytes and $T_{\mathrm{rel}}>99.99999\%$, i.e., $1-T_{\mathrm{rel}}\gg Q(\sqrt{2B\ln 2}/3)\geq 1.7\cdot 10^{-10}$ always holds, and (n_1,P_1) such that $\varepsilon_1<0.5$ so as the assumption on $\mathcal{B}$ in Lemma 3 is again not restrictive. The latency constraint (1.3) is expressed either according to (1.7) for fixed delay feedback model (including $D=0$) or according to (1.8) for the linear delay feedback model.

First, we assume $D=0$. In Figure 1.6, we plot the minimum average energy versus N and reconfirm the energy for sending B information bits decreasing as N increases. Additionally, the energy attains the asymptotic value predicted by Result 3. Moreover, we confirm Result 2, since the minimum average energy decreases when M increases for the case of zero delay feedback;

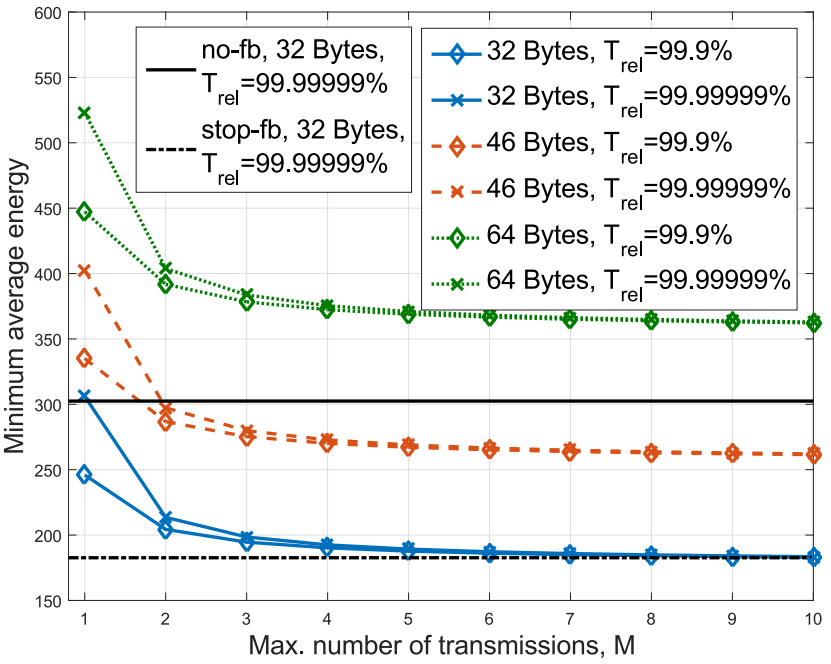

Figure 1.5: Minimum average energy (when $N \to \infty$) versus M.

however, the gain becomes negligible when M is large enough. In Figure 1.7 we reformulate Figure 1.6. For the same B and T_{rel}, we plot the energy gain by using HARQ with M rounds over $M = 1$ (denoted as $E_{\text{No-HARQ}}$). We observe that the energy gain monotonically increases when N grows. As the latency constraint becomes more stringent, the benefit from employing HARQ diminishes.

In Figure 1.8, we plot the energy gain for different values of M versus B when $N \to \infty$. The energies and the corresponding gains are derived using Result 3. The higher the reliability or the lower B, the higher the gain. This remark also holds for non-zero delay feedback since we are in the asymptotic regime.

We pass now to the more realistic case where feedback delays are taken into account, i.e. $D(\vec{n}_M) \neq 0$. In Figure 1.9, we plot the minimum average energy versus M for different delay feedback models (solid lines for fixed delay and dashed line for the linear delay model). When $D > 0$, splitting the packet/transmission in many rounds is not always advantageous. Of course if now M grows too much then the negative impact of feedback delays will be overwhelming as it will squeeze significantly the available blocklength resources for the transmitted packets. On the other hand, for small values of M, the delay penalty is small and so we can exploit further the gains of splitting. Hence, we observe that an optimal bounded value of M, denoted by $M^\star$, exists. The same statement holds when the linear delay feedback model is applied.

In Figure 1.10, we plot $M^\star$ versus N restricting $M \leq 8$. The delay penalties become more significant when N decreases when eventually prevents from using an HARQ mechanism. Therefore, $M^\star$ increases with respect to N. In the case of linear delay feedback model, $M^\star$ increases much slower than in the fixed delay feedback model since the effect of delay in the energy consumption is higher when M increases.

The effect of feedback error is investigated assuming that the feedback error is modeled by a binary symmetric channel (BSC) with error probability p as in [38]. $\overline{E}_f(p)$ denotes the average

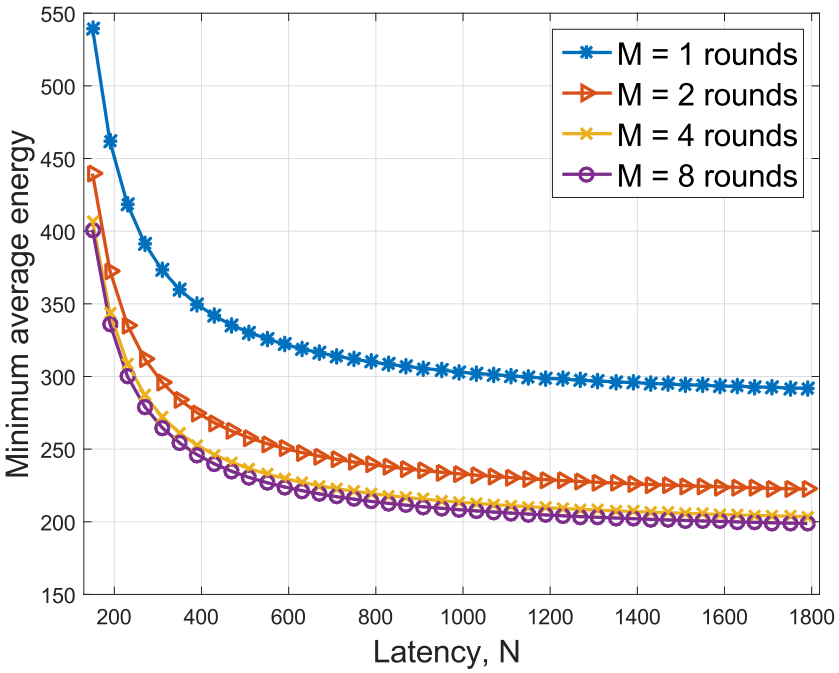

Figure 1.6: Minimum average energy vs. N for $B = 32$ Bytes, $T_{\mathrm{rel}} = 99.999\%$ and $D(\vec{n}_m) = 0$.

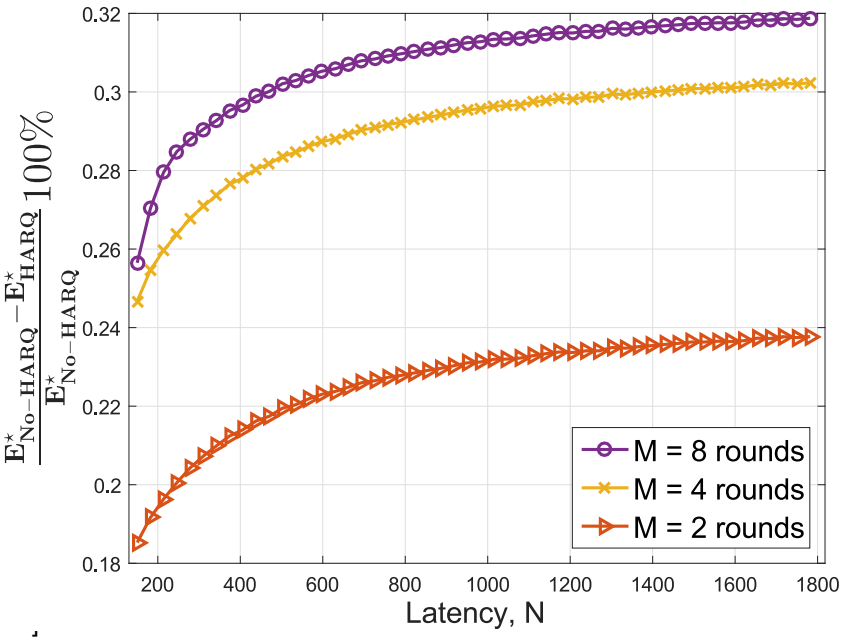

Figure 1.7: Energy gain of M rounds over no HARQ (M=1) vs. N for $B = 32$ Bytes, T_{rel}=99.999% and $D(\vec{n}_m) = 0$.

consumed energy and ε_f denotes the overall error probability when feedback error p is considered. Closed-form expressions with respect to p can be obtained for $\overline{E}_f$ and ε_f (not reported here due to space limitation) using results from [38]. In Figure 1.11 for some optimal configuration $(\vec{n}_M^\star, \vec{P}_M^\star)$ we plot (i) the relative loss in energy, i.e., $(\overline{E}_f(p) - \overline{E}_f(0))/\overline{E}_f(0)$ and (ii) ε_f versus p. We observe that there is only a slight increase of the consumed energy, even for bad feedback

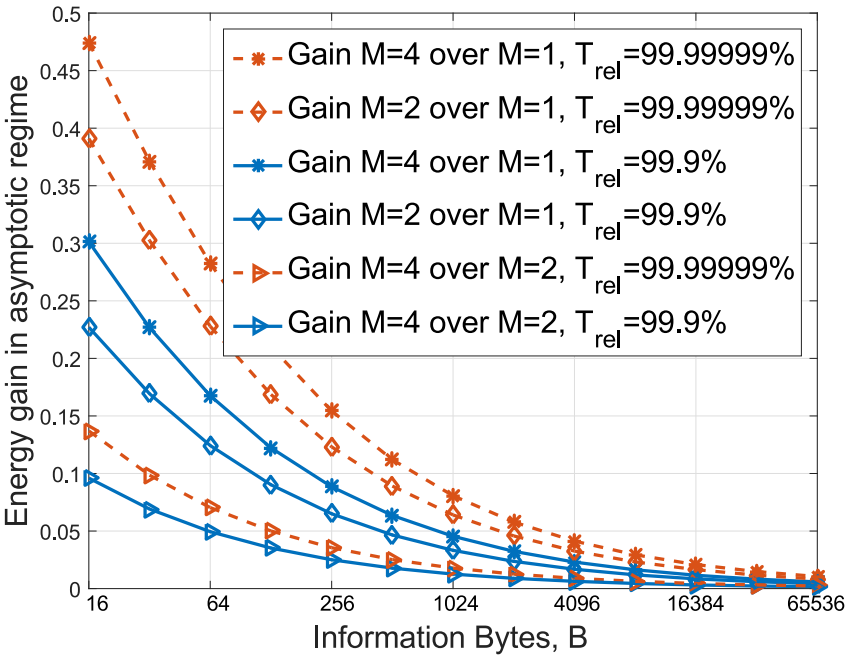

Figure 1.8: Energy gain vs. B in the asymptotic regime ($N \to \infty$) and $D(\vec{n}_m) = 0$.

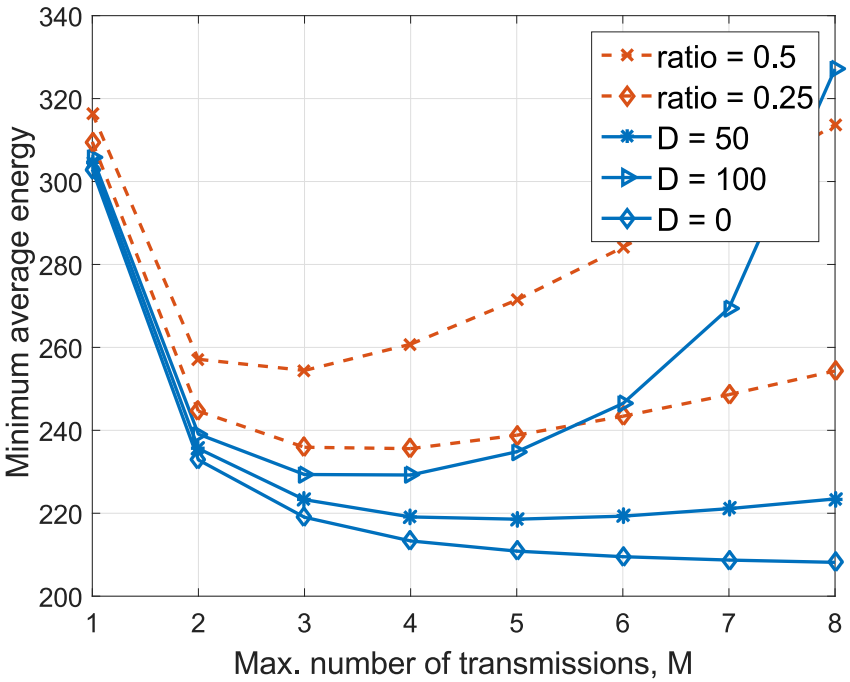

Figure 1.9: Minimum average energy vs. N for $B = 32$ Bytes, $T_{\mathrm{rel}} = 99.999\%$ and $D(\vec{n}_m) = 0$.

channels. In contrast, the reliability is significantly affected by feedback errors except when p is small enough compared to $(1 - T_{rel})$. Indeed, if approximately $p < 0.1(1 - T_{rel})$, then the URLLC requirements are still satisfied. Hence, the feedback has to be protected on the control channel according to this error probability constraint; this is relatively easy to achieve without consuming a lot of resources since it is just one bit.

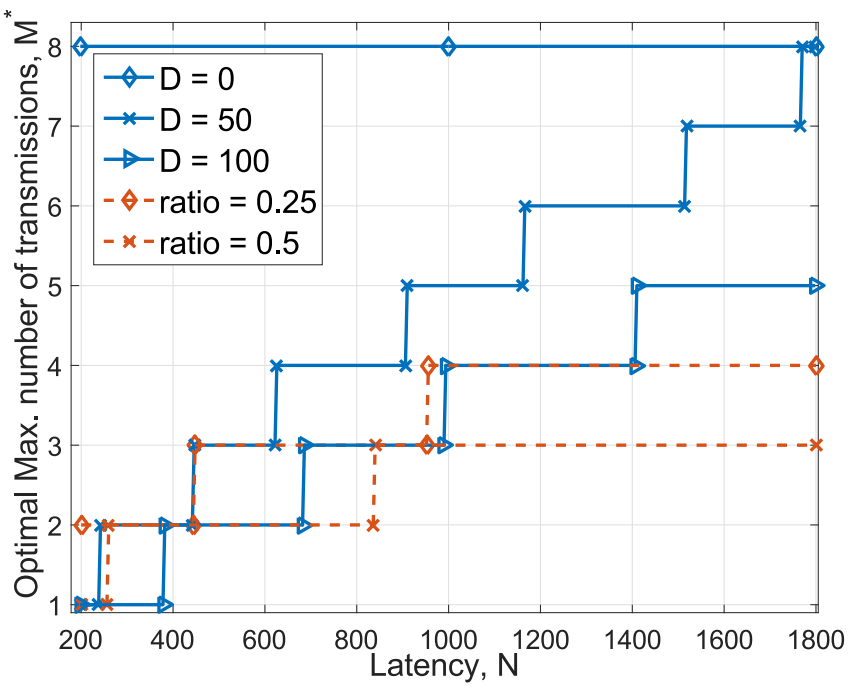

Figure 1.10: $M^{\star}$ (assuming $M \leq 8$) vs. N for B=32 Bytes and T_{rel}=99.999%.

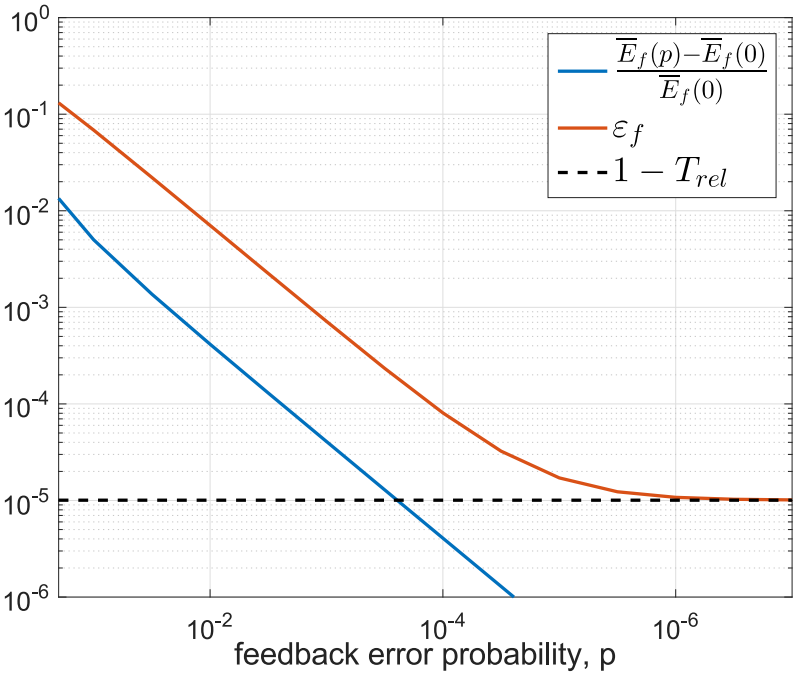

Figure 1.11: $\frac{\overline{E}_f(p)-\overline{E}_f(0)}{\overline{E}_f(0)}$ and ε_f vs. p when B=32 Bytes, T_{rel}=99.999%, $N = 450$c.u. and M=4 rounds.

1.11 Throughput Optimization

We investigated the energy minimization for a specific value of the information bits B and under a specific latency constraint. Therefore we set a minimum value on the information bits per time, i.e.

throughput, and tried to minimize the energy. This section is dedicated on reversing the roles. We aim to maximize the throughput but constraining the acceptable average energy consumption.

Our main goal is to optimize again an IR-HARQ scheme by determining the blocklength $\vec{n}_M$ and the power $\vec{P}_M$ of the packet sent in every round and also to quantify the number of information bits B maximizing the throughput without the average consumed energy exceeding an energy budget $\mathrm{E_t}$. Obviously since we remain on the URLLC regime the latency and reliability constraints are still imposed. Throughput is defined as the average ratio of successfully decoded bits divided by the number of symbols used. The throughput can be derived using the renewal theory [39] where the expected value of delay is $\sum_{m=1}^{M} n_m \varepsilon_{m-1}$ and the expected reward is $B(1-\varepsilon_M)$. Consequently, our goal can be translated into the following optimization problem.

Problem 4.

$$\max_{B,\vec{n}_M,\vec{P}_M} \quad \frac{B(1-\varepsilon_M)}{\sum_{m=1}^{M} n_m \varepsilon_{m-1}} \tag{1.28}$$

$$\textit{s.t.} \quad \sum_{m=1}^{M} n_m \leq N \tag{1.29}$$

$$\varepsilon_M \leq 1 - T_{\mathrm{rel}} \tag{1.30}$$

$$\sum_{m=1}^{M} n_m P_m \varepsilon_{m-1} \leq \mathrm{E_t} \tag{1.31}$$

$$\vec{P}_M \in \mathbb{R}_+^M \,,\, \vec{n}_M \in \mathbb{N}_+^M$$

Solving the general problem 4 is intractable. The objective (1.28) is given as a fraction of Q functions having involved expressions as arguments. Our goal is to reformulate the problem so as to get rid of the fraction and remain with a simpler expression. In the first step we modify slightly the objective function by forcing the numerator to be equal to $B(1-\varepsilon_{\mathrm{rel}})$ which means we force the constraint given in (1.30) to be active. This leads to the following modified optimization problem.

Problem 5.

$$\max_{B,\vec{n}_M,\vec{P}_M} \quad \frac{B(1-\varepsilon_{\mathrm{rel}})}{\sum_{m=1}^{M} n_m \varepsilon_{m-1}} \tag{1.32}$$

$$\textit{s.t.} \quad \sum_{m=1}^{M} n_m \leq N \tag{1.33}$$

$$\varepsilon_M \leq 1 - T_{\mathrm{rel}} \tag{1.34}$$

$$\sum_{m=1}^{M} n_m P_m \varepsilon_{m-1} \leq \mathrm{E_t} \tag{1.35}$$

$$\vec{P}_M \in \mathbb{R}_+^M \,,\, \vec{n}_M \in \mathbb{N}_+^M$$

The following result proves that the solution of Problem 5 achieves almost the same performance as the one of the original Problem 4.

Proposition 2. *Let $(B^{mod}, \vec{n}_M^{mod}, \vec{P}_M^{mod})$ be the solution of Problem 5, which result in a value Th for the throughput according to* (1.28)*. Let $Th^\star$ be the highest (optimal) value for the throughput given by the solution of Problem 4. Then $(B^{mod}, \vec{n}_M^{mod}, \vec{P}_M^{mod})$ is a feasible point of Problem 4 and it holds that $Th \leq Th^\star \leq \frac{Th}{T_{\mathrm{rel}}}$.*

Proof. See Appendix A.8 □

We propose to perform the optimization over B via one-dimensional grid-search. Consequently, Problem 5 can be further simplified and leads to the following Problem 6.

Problem 6.

$$\min_{\vec{n}_M, \vec{P}_M} \quad \sum_{m=1}^{M} n_m \varepsilon_{m-1} \tag{1.36}$$

$$\textit{s.t.} \quad \sum_{m=1}^{M} n_m \leq N \tag{1.37}$$

$$\varepsilon_M \leq 1 - T_{\mathrm{rel}} \tag{1.38}$$

$$\sum_{m=1}^{M} n_m P_m \varepsilon_{m-1} \leq \mathrm{E_t} \tag{1.39}$$

$$\vec{P}_M \in \mathbb{R}_+^M \, , \, \vec{n}_M \in \mathbb{N}_+^M$$

1.11.1 Using M rounds

As in the energy minimization problem 1, we show with the following proposition the same behavior on the problem 6, i.e. the greater number of rounds M is available the better.

Proposition 3. *Given T_{rel} and resources $N, \mathrm{E_t}$, increasing the number of retransmissions M always yields solutions of Problem 6 with better values of the objective function* (1.36)*.*

Proof. See Appendix A.9 □

We should cautiously interpret the above results because we have not included feedback delays in this setup. Of course in practice, as M grows the feedback delays are aggregating prohibiting any further splitting. Nonetheless, after showing the dynamic programming algorithm solving the problem 6 and having already discussed on the energy minimization problem on how to incorporate feedback delays and their effects, it is straightforward how to proceed accordingly for the throughput maximization problem and incorporate those delays, so we choose to neglect them.

1.11.2 Equality constraints

Let's try again to further simplify the problem by turning some inequality constraint into equalities.

Proposition 4. *Let $(\vec{n}^\star_{M^\star}, \vec{P}^\star_{M^\star})$ be the optimal point of Problem 6 and $\varepsilon^\star_m = \varepsilon(\vec{n}^\star_m, \vec{P}^\star_m)$, where $\vec{n}^\star_m$ (resp. $\vec{P}^\star_m$) is an extracting vector from the m-th first components of $\vec{n}^\star_M$ (resp. $\vec{P}^\star_M$), be the error probability at every round $m < M$. We have $\varepsilon^\star_m > 1 - T_{\mathrm{rel}}$ and finally at round M we have $\varepsilon^\star_M \leq 1 - T_{\mathrm{rel}} < \varepsilon_M(\vec{n}^\star_{M-1}, n^\star_M - 1, \vec{P}^\star_M)$.*

Proof. See Appendix A.10 □

As $\varepsilon^\star_M \leq 1 - T_{\mathrm{rel}} < \varepsilon_M(\vec{n}^\star_{M-1}, n^\star_M - 1, \vec{P}^\star_M)$, we conjecture that $\varepsilon^\star_M \approx 1 - T_{\mathrm{rel}}$ since it makes sense the last symbol of the last round not being able to throw us too far away from the boundary of $1 - T_{\mathrm{rel}}$. An important remark is that equality $\varepsilon^\star_M \approx 1 - T_{\mathrm{rel}}$ is not necessarily true for the

initial problem 4. As can be confirmed by Fig. 1.12 there is an area where we have to consider operating on smaller error probabilities to get higher throughput. But this is relevant only for very low reliability values (in Fig. 1.12 is $1 - T_{\mathrm{rel}} > 0.1$) which also do not allow to concretely pass from problem 5 to problem 6.

The $\varepsilon_M^\star \approx 1 - T_{\mathrm{rel}}$ also leads to $\mathrm{E}_M^\star \approx \mathrm{E_t}$, where $\mathrm{E}_M^\star$ is the average energy consumed by the optimal solution of Problem 6. The reason is that if enough energy is allowed by the energy constraint (1.39) (i.e. $\mathrm{E_t} - \mathrm{E}_M^\star$) to be spent on P_M so as to compensate for a one symbol decrease of n_M (and still satisfy the reliability constraint $\varepsilon_M^\star \approx 1 - T_{\mathrm{rel}}$), then we can arrive to a better solution with smaller objective (1.36). Therefore, the average energy spent by the optimal solution $\mathrm{E}_M^\star$ should be close to the boundary $\mathrm{E_t}$. To conclude, the problem we aim to solve becomes:

Problem 7.

$$\min_{\vec{n}_M, \vec{P}_M} \quad \sum_{m=1}^{M} n_m \varepsilon_{m-1} \tag{1.40}$$

$$\textit{s.t.} \quad \sum_{m=1}^{M} n_m \leq N \tag{1.41}$$

$$\varepsilon_M = 1 - T_{\mathrm{rel}} \tag{1.42}$$

$$\sum_{m=1}^{M} n_m P_m \varepsilon_{m-1} = \mathrm{E_t} \tag{1.43}$$

$$\vec{P}_M \in \mathbb{R}_+^M \, , \, \vec{n}_M \in \mathbb{N}_+^M$$

1.11.3 Dynamic Programming approach

We can now solve Problem 7 iteratively in a similar way as before by using a dynamic programming approach but with some changes. First of all, we redefine the states at the end of m-th round:

$$S_1 = (N_1, c_1)$$

$$S_m = (N_m, c_m, \mathrm{E}_m, V_m), m \in \{2, 3, ...\}$$

where N_m, c_m and V_m are defined exactly like in (1.13-1.15). We remind that $\varepsilon_m = Q(c_m)$. This time we need to add extra component to the state which refers to average energy spent till m-round $\mathrm{E}_m = \sum_{i=1}^{m} n_i P_i \varepsilon_{i-1}$. We have $N_m \in \mathcal{N}_d$, $V_m \in \mathcal{V}_d$ and $c_m \in \mathcal{C}_d$ as in (1.18),(1.19) and (1.20). Let $\mathbb{S}_M$ be the set of all feasible final states. By feasibility, we mean that a state $S_M \in \mathbb{S}_M$ satisfies the constraints of Problem 7 and there is a path $(\vec{n}_M, \vec{P}_M)$ leading to S_M. Our objective is to find the sequences/paths of states minimizing (1.40) to every $S_M \in \mathbb{S}_M$ being a possible candidate to achieve optimality. Then, the optimal solution of Problem 7 is retrieved by choosing out of those S_M the one with the smallest minimum.

The justification for choosing the first three variables of the states S_m is to be able to check the constraints (1.37)-(1.39). The dispersion variable V_m is added so as the description of S_m to depend only on the previous state S_{m-1} and the variables n_m and P_m, which constitute the branch between S_{m-1} and S_m. The functions connecting these states can be easily found and let them be: $S_m = f_S(S_{m-1}, n_m, P_m)$, $S_{m-1} = f_S^{-1}(S_m, n_m, P_m)$.

For sake of simplicity, we introduce the following notation "$\min_{X|Y} f(X)$" which stands for "minimize

$f(\cdot)$ over the variables X given constraints Y". Now the Problem 7 can be seen as the solution of

$$\min_{\vec{n}_M,\vec{P}_M|S_M \in \mathbb{S}_M} \sum_{m=1}^{M} n_m \varepsilon_{m-1}.$$

This minimization can be solved dynamically since it can be written as

$$\min_{n_M,P_M|S_M} \Big\{ \min_{\vec{n}_{M-1},\vec{P}_{M-1}|S_M,n_M,P_M} \{ n_M \varepsilon_{M-1} + \sum_{m=1}^{M-1} n_m \varepsilon_{m-1} \} \Big\}.$$

The inner minimization is done under fixed (S_M, n_M, P_M), which allows the first term $n_M \varepsilon_{M-1}$ to get out as a constant since this term can be expressed as a function, let it be $K(\cdot)$, of only those fixed variables. Moreover, $S_{M-1} = f_S^{-1}(S_M, n_M, P_M)$ is fixed, which can be confirmed that it is an equivalent to (S_M, n_M, P_M) constraint when minimizing the second term. So, we have

$$\min_{\vec{n}_M,\vec{P}_M|S_M} \{ \sum_{m=1}^{M} n_m \varepsilon_{m-1} \} = \min_{n_M,P_M|S_M} \Big\{ K(S_M, n_M, P_M) + \min_{\vec{n}_{M-1},\vec{P}_{M-1}|S_{M-1}=f_S^{-1}(n_M,P_M,S_M)} \{ \sum_{m=1}^{M-1} n_m \varepsilon_{m-1} \} \Big\}.$$

The above formula can be proven for every $m \in \{1, ..., M\}$, which enables to use a dynamic programming approach. Specifically, in order to find the optimal solution for the state S_m, it is sufficient to know the optimal solution of every S_{m-1} connected to it through a branch (n_m, P_m). Therefore we can start by straightforwardly computing the values for all feasible S_1 and afterwards in every m iteration of the dynamic programming algorithm, we compute the optimal solution for S_m by using the corresponding S_{m-1}.

1.11.4 Algorithm Implementation

In practice, the dynamic programming algorithm requires the variables of the states to take discrete values. Specifically:

- $N_m \in \mathtt{N}_d$ has already a discrete form since it is an integer, but it can be quantized using bigger than one symbol step size for accelerating the simulation. Let $\mathtt{N}$ be the set of the discrete values that N_m can take.
- c_m is real and $c_m \in \mathcal{C}_d$. Let $\mathtt{C} \subset \mathcal{C}_d$ be the set of the discrete values that the dynamic algorithm allows c_m to take.
- E_m is real and $\mathrm{E}_m \in [0, \mathrm{E}_t]$. After quantization, let $\mathtt{E}$ be the set of the discrete values E_m can take.
- V_m is real and $V_m \in \mathcal{V}_d$. After quantization, let $\mathtt{V}$ be the set of the discrete values V_m can take.

The implementation of the dynamic algorithm has some differences compared to one of energy minimization problem. It consists of two stages: a first one for computing the performance of the feasible states and a second one for searching over those states to find the optimal solution. The

complexity is dominated by the first stage and is equal to the number of iterations of the dynamic algorithm multiplied by the number of states examined per iteration times the number of branches departing from every state. In this implementation, we compute the branch (n_{m+1}, P_{m+1}) departing from a state S_m through fixing the variables N_{m+1} and E_{m+1} of the arriving state S_{m+1} and subsequently we acquire the feasible $c_m = Q^{-1}(\varepsilon_{m+1})$ and V_{m+1}. Therefore the overall complexity is $\mathcal{O}(M \cdot |\mathtt{N}||\mathtt{E}||\mathtt{C}||\mathtt{V}| \cdot |\mathtt{N}||\mathtt{E}|)$.

The above complexity characterizes a rather slow algorithm, however in practice the algorithm can be faster by remarking that most of the times all paths ending up at states with the same (N_m, c_m, E_m) which the algorithm considers, present dispersion V_m within a small range of values. Therefore if a reasonable resolution of the discrete set $\mathtt{V}$ is considered so as no significant approximation errors are introduced, the number of feasible states with same (N_m, c_m, E_m) and different V_m turns out to be rather small (often just one value). Therefore, the variable $|\mathtt{V}|$ can be thought as constant. A final implementation remark for speeding up significantly, is that after a feasible solution appears, by keeping track of the best feasible solution found, we have an upper bound of the optimal solution. This upper bound can cut prematurely all paths that already exhibit higher than that upper bound objective function.

1.11.5 Numerical Results and Discussion

In this section, we carry our numerical evaluations to assess the system performance. In Fig. 1.12, we investigate the effect of the error probability the reliability constraint forces on throughput. The figure can be obtained by solving Problem 7 for $M = 5$ and $1 - T_{\mathrm{rel}} = 10^{-6}$ but memorizing for $m \in \{2, 3, 5\}$ and for every c_m all the feasible states and their maximum throughput performance. Therefore we are actually required to do only one run of the dynamic algorithm because after the computation of the performance of each state, we can restrict the search of the minimum only among the states with the given $\varepsilon_m = Q^{-1}(c_m), m \in \{2, 3, 5\}$.

As shown in Proposition 3 and confirmed by Fig. 1.12, more transmission rounds result in higher throughput. Moreover since we have short packets (finite blocklength regime), it is not possible to attain $\varepsilon_M \to 0$ with a finite energy budget. Therefore, there exists a certain value beyond which the reliability cannot go. This is the reason why the curve of $M = 2$ in Fig. 1.12 terminates at a certain error probability past of which there is no feasible point. Of course the same applies for the other two curves but it cannot be depicted in the figure due to the limits of the horizontal axis. Finally, we remark, as in [18], that there is a certain value of error probability that maximizes the throughput, which is relatively high though (close to 0.1). Accordingly, in our case, higher reliability can be achieved at the expense of throughput.

The impact of the number of symbols used on the throughput performance is shown in Fig. 1.13, which is obtained by imposing equality in the latency constraint (1.41) of Problem 7 or equivalently retrieving the optimal solutions by only searching within states with $N_M = N$.

When the available number of symbols are inadequate, no feasible solution exists and the throughput vanishes. Interestingly, as N grows beyond a certain threshold, only a slight increase in throughput is achieved, followed by a slow decrease. This means that it is not always beneficial from a throughput perspective to use the whole available blocklength since the denominator of (1.28) may increase. Asymptotically, if $N \to \infty$, then for some $m \in \{1, \cdots, M\}$ it should be $n_m \to \infty$, which in turn will result in vanishing throughput. Therefore, all curves in Fig. 1.13 will asymptotically converge to zero.

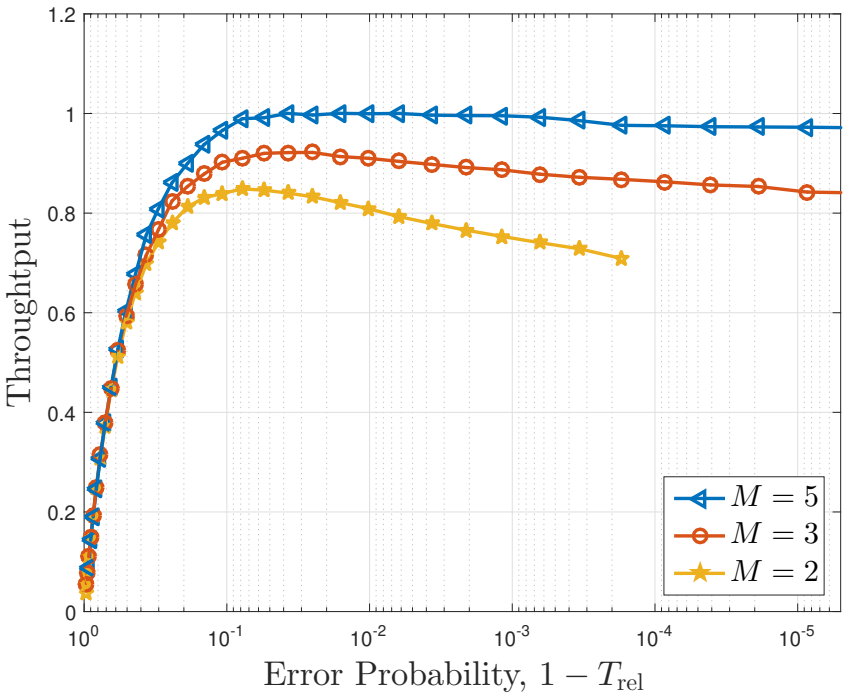

Figure 1.12: Throughput vs. error probability for $N = 400$, $\mathrm{E_t} = 265$, and $B = 32$ bytes.

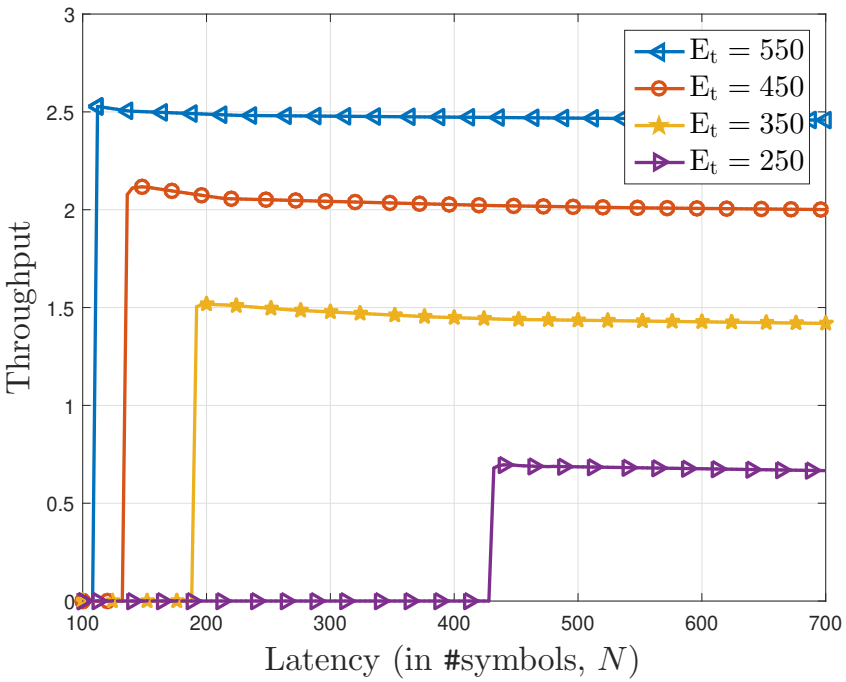

Figure 1.13: Throughput vs. number of symbols used for $1 - T_{\mathrm{rel}} = 10^{-5}$, $B = 32$ bytes, and $M = 3$.

Fig. 1.14 depicts the throughput versus the energy budget. In practice, we do not force equality in the energy constraint (1.35), since, as stated previously, the optimal solution consumes by default (almost) all the available energy. In our simulations, we set the minimum possible blocklength

for the first IR-HARQ round to be $N_{1,min} \geq 100$ (which is set likewise so that the approximation (1.1) remains accurate). Consequently, the throughput cannot exceed the value $\frac{B}{N_{1,min}}$, which represents the unrealistic case of only one packet sent with minimum blocklength and achieving perfect reliability. This upper bound is closely attained as the available energy grows up to a point where only one transmission may fulfill the constraints and thus, further increase of the energy is worthless. Moreover, Fig. 1.14 reconfirms (as in Fig. 1.13), since the curves coincide, that past a certain threshold, any further increase in blocklength is meaningless.

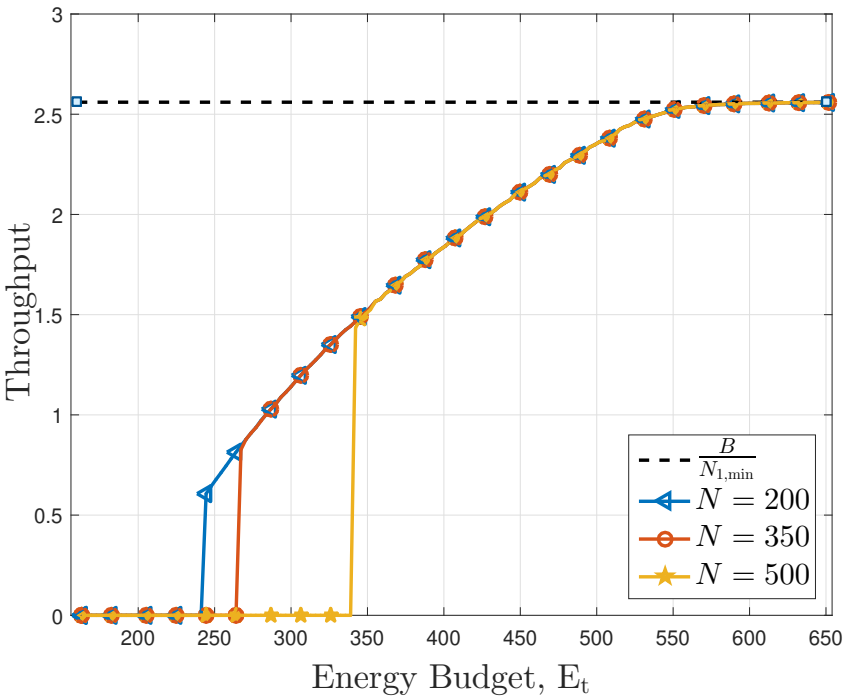

Figure 1.14: Throughput vs. energy spent for $1 - T_{\mathrm{rel}} = 10^{-5}$, $B = 32$ bytes, and $\mathrm{M_r} = 3$.

Finally, in Fig. 1.15 we depict the throughput (via a contour plot) versus the available average energy $\mathrm{E_t}$ and the information bits to transmit B. There is an upper left area with no feasible points. Keeping a constant $\mathrm{E_t}$ by moving vertically, we see that the throughput is a unimodal function over B and there is a specific value of B that achieves optimality. This also agrees with [18] where a simple ARQ scheme with no URLLC constraints was employed.

1.12 Conclusion

In this chapter we characterized the interplay between energy, throughput, latency, and reliability for point-to-point communication in AWGN channels. In URLLC systems where only a limited number of symbols can be transmitted, we formulate optimization problems that enable tuning the IR-HARQ parameters by estimating the number of rounds, the blocklength, and the transmit power for each transmitted packet. After analyzing mathematically the problems, it is possible to reach a simpler form where even though the problem remains non-convex, its solution accepts a dynamic programming based approach. First, we set the objective function to be the minimization of the average consumed energy; second, the objective is to maximize the throughput. It turns out that even when operating with strict latency constraints, a proper optimization of IR-HARQ

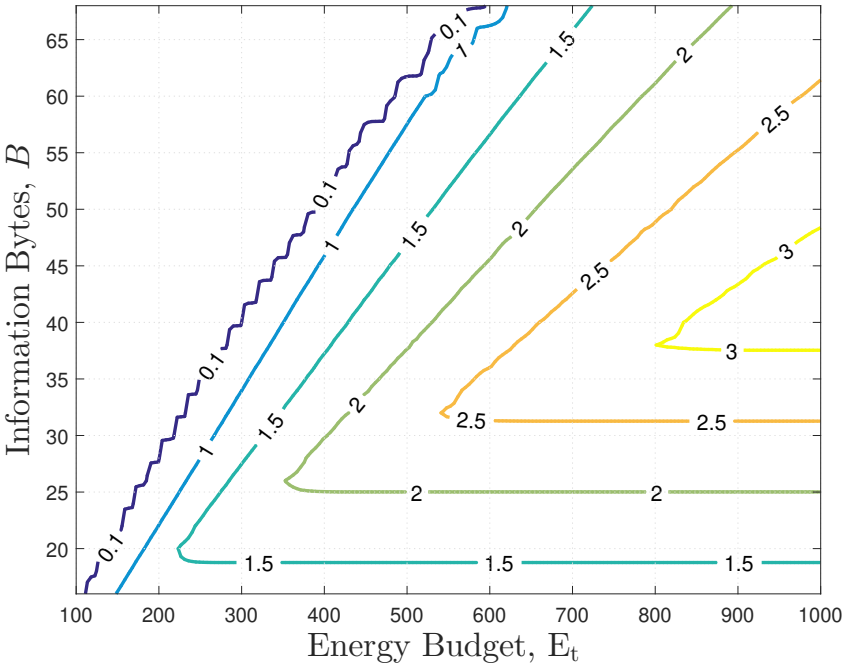

Figure 1.15: Throughput vs. energy and information bits for $T_{\mathrm{rel}} = 10^{-5}$, $\mathrm{N}_{\ell} = 600$, and $\mathrm{M}_{\mathrm{r}} = 3$.

can be beneficial as long as the feedback delay is reasonable compared to the packet duration. The main takeaway is that for either energy minimization or throughput maximization, one could expect gains of the magnitude of 20% compared to schemes with no-HARQ. Reasonably enough, the optimal solution to the throughput maximization problem turns out to use all available energy budget. Respectively, the solution that minimizes the energy consumption adversely affects the throughput. So a natural question arises whether there is a graceful point on this tradeoff relationship. This is the rationale behind the following chapter under a more general setup where fading is taken also into account.

Chapter 2

IR-HARQ Optimization for URLLC: Fading channels and the Effect of CSI

2.1 Introduction

In this chapter, we further explore the potential benefits of a properly optimized incremental redundancy hybrid automatic repeat request (IR-HARQ) retransmission protocol in ultra-reliable, low-latency communication (URLLC) systems. So far we have investigated an IR-HARQ mechanism in additive white Gaussian noise (AWGN) channels. We tuned the blocklength and the power during each IR-HARQ round so as not only the URLLC requirements are met, i.e. latency of few milliseconds and reliability higher than 99.999%, but also a certain performance metric, such as to be optimized according to some objective. First we set that objective to be the minimization of the average consumed energy and then we deal with throughput maximization.

In this chapter, we generalize the objective by taking into account simultaneously both dimensions. We set here a (linear) combination of energy consumption and throughput to be our optimizing goal. To further extend the work on top of the AWGN, we add small-scale fading. In general the transmitted signal arrives at the receiver by several paths, interfering with one another and causing the fading of the channel. To estimate the influence of the main path of the signal being interfered with reflected versions of itself on our system we used a Ricean fading model.

Unfortunately, channel fading can deteriorate significantly the channel quality and fulfilling of URLLC requirements becomes significantly challenging. Poor channel quality can definitely render communication impossible except if an excessive loss of throughput and energy is allowed. To estimate the price of higher energy and/or lower throughput that has to be paid, we try to analyze the feasibility and performance in block fading channels. Some leverage of the impact of the channel unpredictability so as mitigate the destructive effect of bad channel realizations can be achieved by sending pilots in order to assess the channel's quality. Acquiring Channel State Information (CSI) helps to better tailor the IR-HARQ and operate in a better point in terms of reliability-energy-throughput tradeoff.

We quantify here the impact of CSI on the tradeoff between energy and latency. We investigate how the two standard cases of CSI, i.e., (i) *statistical* where only the statistics of the channel

is known and (ii) *full* where the exact channel coefficient is available, influence the optimization of the IR-HARQ scheme. We demonstrate that performing purely energy minimization or throughput maximization leads to a bad trade-off point and a multi-objective optimization should be considered instead. Moreover, we analyze and compute the feasibility region of the IR-HARQ schemes. Surprisingly, it turns out that with statistical CSI we obtain a larger feasibility region than with a reasonable scheme exploiting full CSI and avoiding deep fading instances. In general, under URLLC conditions the difficulty of finding a scheme calibrating perfectly to the full CSI renders it less robust even without taking into account the shrinking of the latency constraint due to the training phase of learning the channel. Aside from robustness, the full CSI may yield better throughput with lower energy. The material presented in this chapter has been published in [C5].

2.2 Revisiting the System Model

A point-to-point communication link is again considered, where the transmitter has to convey B information bits within a certain predefined latency, expressed by a certain predefined maximum number of channel uses denoted by N. If no retransmission mechanism is utilized, the packet of B bits is transmitted only once (one-shot transmission) and its maximum possible blocklength is N. In case of retransmission, IR-HARQ protocol is used with M transmission rounds, i.e., $M-1$ retransmissions. By setting $M=1$, we recover the no-HARQ case. We denote n_m, $m \in \{1,2,...,\mathrm{M}\}$, the number of channel uses for the m-th transmission. Since the model is significantly more complicated (as we incorporate fading and CSI), here we will mainly concentrate on $M=2$. We keep the assumption that the receiver knows perfectly whether or not the message is correctly decoded (through CRC) and ACK/NACK is received error-free. The latency constraint is expressed again by translating it into a number of channel uses as follows: we have $\sum_{m=1}^{M} n_m \leq N$[1].

We consider a block flat fading channel, where the channel $h \in \mathbb{C}$ is an independent realization of an underlying random variable $\mathcal{H}$ following a specific distribution and remains constant in each block. The signal is also subject to additive white circularly-symmetric complex Gaussian random process with zero mean and unit variance. The IR-HARQ mechanism takes place within one block, i.e., there is only one channel coefficient value h for all retransmissions associated with the same bits. Consequently, we assume that the coherence block duration is around or greater than N. As explained in the previous chapter, this is a relevant model for short-length packet communication and IoT applications, where point to point communication is performed. Without loss of generality, in the m-th round the fragment (sub-codeword) $c_m \in \mathbb{C}^{n_m}$ is received with power $gP_m = \frac{\|h \cdot c_m\|^2}{n_m}$, where we defined the channel gain $g = |h|^2$, and distorted by an additive white circularly symmetric complex Gaussian random process with zero mean and unit variance.

2.3 Problem Statement

Similarly to the previous chapter, the problem we study here is that of optimizing the IR-HARQ mechanism by tuning the blocklengths and the powers but now with the more general aim of mini-

[1]Penalty terms $D(n_1,...n_m)$ can easily be introduced at each m-th transmission in order to take into account the delay for the receiver to process/decode the m-th packet and send back acknowledgment (ACK/NACK). We focus on the simplified version where $D(n_1,...n_m)=0$ since $D(n_1,...n_m)>0$ is effectively equivalent to $D=0$ but with more stringent latency constraint (smaller N).

mizing a multi-objective function, involving both average energy consumption and throughput. We require a low error probability $1 - T_{\mathrm{rel}}$ without consuming more than a total energy budget $\mathrm{E_t}$ and within a latency N. Since channel fading is included in the model the formula (1.1) characterizing the probability of error (or equivalently the outage probability) in the m-th round has to be modified.

Since the IR-HARQ mechanism takes place within one fading block, it is simple to adapt the equation (1.1) by just a scaling of the power according to the channel gain g the signal experiences during that block. Formally,

$$\varepsilon_m \approx Q\left(\frac{\sum_{i=1}^{m} n_i \log(1+gP_i) - B\log 2}{\sqrt{\sum_{i=1}^{m} n_i \left(1 - \frac{1}{(1+gP_i)^2}\right)}}\right) \tag{2.1}$$

where $Q(x)$ is the complementary Gaussian cumulative distribution function. For the sake of clarity, we may show the dependency on the variables, i.e., $\varepsilon_m(n_1, ...n_m, P_1, ...P_m, g)$ instead of ε_m.

2.4 Optimization

By "full CSI" case we mean that the transmitter knows exactly the channel coefficient h, which is independently varying block by block. On the other hand if only the channel distribution $\mathcal{H}$ is known to the transmitter we refer to "statistical CSI" case. In both configurations we optimize the weighted sum of the average throughput and energy consumption.

Throughput is defined as the average ratio of successfully decoded bits divided by the number of symbols used. Given a channel realization (and so its gain $g = |h|^2$), the expected throughput can be derived using the renewal theory [39] where the expected value of delay is $\sum_{m=1}^{M} n_m \varepsilon_{m-1}$ and the expected reward is $B(1 - \varepsilon_M)$ which leads to

$$\mathcal{T}_h(0) = \frac{B(1-\varepsilon_2)}{n_1 + n_2\varepsilon_1}.$$

The expected energy spent for transmitting B information bits (conditioned on the channel realization) is

$$\mathcal{E}(1) = n_1 P_1 + n_2 P_2 \varepsilon_1.$$

2.4.1 Full CSI

The optimization problem is cast as follows.

Problem 8. *Full CSI problem.*

$$\min_{n_1(g),n_2(g),P_1(g),P_2(g)} \quad \mathbb{E}_g\left[-\frac{\mathcal{T}_h(a)}{\mathcal{T}_{h,max}}+\frac{\mathcal{E}(a)}{\mathcal{E}_{min}}\right] \tag{2.2}$$

$$\textit{s.t.} \quad \mathbb{E}_g[\varepsilon_2(n_1(g),n_2(g),P_1(g),P_2(g),g)] \leq 1-T_{\mathrm{rel}} \tag{2.3}$$

$$n_1(g)+n_2(g) \leq N, \quad \forall g \tag{2.4}$$

$$n_1(g)P_1(g)+n_2(g)P_2(g) \leq \mathrm{E_t}, \quad \forall g \tag{2.5}$$

$$P_i(g) \leq \mathrm{P_{max}}, \quad i \in \{1,2\}, \quad \forall g \tag{2.6}$$

where

- $\mathcal{T}_h(a) = (1-a)\mathcal{T}_h(0)$,
- $\mathcal{E}(a) = a\mathcal{E}(1)$. *So the variable a is a weight balancing throughput maximization and energy minimization.*
- $\mathbb{E}_g[\cdot]$ *is the expectation over the channel gain realizations.*
- $\mathcal{T}_{h,max} = \max \mathbb{E}_g[T_h(0)]$ *s.t.* (2.4-2.6) *hold*
- $\mathcal{E}_{min} = \min \mathbb{E}_g[E(1)]$ *s.t.* (2.4-2.6) *hold.*

Since the values of energy and throughput are very different in scale, to facilitate the multi objective optimization we normalize each of those quantities using the optimal value that they can attain if they were the single quantity to be optimized, i.e. using $\mathcal{T}_{h,max}$ and $\mathcal{E}_{min}$. Furthermore, in order for the solutions of the Problem 8 to be possible to consume the maximum energy budget $\mathrm{E_t}$ we assume

$$\mathrm{P_{max}} \geq \frac{\mathrm{E_t}}{N}. \tag{2.7}$$

As the channel is known, we can adapt the blocklengths and powers accordingly. The solution of the optimization problem depends on the channel gain realization g and so what we really try to find are functions of g. To avoid the very complicated functional optimization we simplify the problem by enforcing the solution to have certain reasonable characteristics. Firstly, the simple yet intuitive that the transmissions are avoided over deep fading. Mathematically, the proposed solutions satisfy:

$$(n_i, P_i) = \begin{cases} (0,0), & g < g_{th} \\ (n_i(g), P_i(g)), & g \geq g_{th} \end{cases} \tag{2.8}$$

Secondly, we force each transmission (when done) to achieve the same error probability ε_{on}, i.e.:

$$\varepsilon_2(n_1(g),n_2(g),P_1(g),P_2(g)) = \begin{cases} 0, & g < g_{th} \\ \varepsilon_{on}, & \geq g_{th} \end{cases}. \tag{2.9}$$

Hence, the reliability constraint (2.3) becomes:

$$\mathbb{P}(g < g_{th}) + \mathbb{P}(g \geq g_{th})\varepsilon_{on} \leq 1 - T_{\mathrm{rel}}. \tag{2.10}$$

These simplifications enable decoupling the problem by treating every g with $g \geq g_{th}$ individually. An additional simplification can be applied (with a proof being a simple combination of the

proofs of lemma 2 and proposition 4 so it is omitted) that asserts $\varepsilon_2 \approx 1 - T_{\rm rel}$. This means that trying to achieve lower error probability than the already very low required ε_{on} (whenever $g \geq g_{th}$), results in much greater waste of energy and blocklength resources than benefit to the multi-objective. Summing up, the proposed scheme for optimizing IR-HARQ with full CSI comes as the solution of the following optimization problem:

Problem 9. *Simple Scheme with Full CSI problem.*

$$\min_{n_1(g),n_2(g),P_1(g),P_2(g),g_{th}} \quad \mathbb{E}_g\left[-\frac{\mathcal{T}_h(a)}{\mathcal{T}_{h,max}} + \frac{\mathcal{E}(a)}{\mathcal{E}_{min}}\right] \tag{2.11}$$

$$\textit{s.t.} \quad \varepsilon_2(n_1(g), n_2(g), P_1(g), P_2(g), g)] = \mathbb{1}\{g \geq g_{th}\}\varepsilon_{on} \tag{2.12}$$

$$n_1(g) + n_2(g) \leq N, \quad \forall g \tag{2.13}$$

$$n_1(g)P_1(g) + n_2(g)P_2(g) \leq \mathrm{E_t}, \quad \forall g \tag{2.14}$$

$$P_i(g) \leq \mathrm{P_{max}}, \quad i \in \{1,2\}, \quad \forall g \tag{2.15}$$

$$n_i(g) = P_i(g) = 0 \quad i \in \{1,2\}, \quad \forall g < g_{th} \tag{2.16}$$

where $\mathcal{T}_h(a)$, $\mathcal{E}(a)$, $\mathbb{E}_g[\cdot]$ are defined as in Problem 8 and

- $\mathcal{T}_{h,max} = \max \mathbb{E}_g[T_h(0)]$ *s.t.* (2.12-2.16) *hold*
- $\mathcal{E}_{min} = \min \mathbb{E}_g[E(1)]$ *s.t.* (2.12-2.16) *hold*
- $\varepsilon_{on} = \dfrac{1 - T_{\rm rel} - \mathbb{P}(g < g_{th})}{\mathbb{P}(g \geq g_{th})}$

2.4.2 Statistical CSI

If only the distribution of the channel $\mathcal{H}$ is known then the channel realization is not known in advance and changes independently in every coherence bloc. Therefore it is impossible adapting the blocklengths and powers at each time and so we aim to find an optimal blocklength-power configuration which is independent of the channel gain g.

Problem 10. *Statistical CSI problem.*

$$\min_{n_1,n_2,P_1,P_2} \quad \mathbb{E}_g\left[-\frac{\mathcal{T}_h(a)}{\mathcal{T}_{h,max}} + \frac{\mathcal{E}(a)}{\mathcal{E}_{min}}\right] \tag{2.17}$$

$$\textit{s.t.} \quad n_1 + n_2 \leq N \tag{2.18}$$

$$\mathbb{E}_g[\varepsilon_2(n_1, n_2, P_1, P_2, g)] \leq 1 - T_{\rm rel} \tag{2.19}$$

$$n_1P_1 + n_2P_2 \leq \mathrm{E_t}, \tag{2.20}$$

$$P_i \leq \mathrm{P_{max}}, \quad i \in \{1,2\} \tag{2.21}$$

where

- $\mathcal{T}_{h,max} = \max \mathbb{E}_g[T_h(0)]$ *s.t.* (2.18-2.21) *hold,*
- $\mathcal{E}_{min} = \min \mathbb{E}_g[E(1)]$ *s.t.* (2.18 - 2.21) *hold.*

We can again assert equality in the reliability constraint (2.19).

2.5 Feasibility region

Notice that it is not always possible to meet the constraints and to get a non-empty feasible set if the average channel gain average is very low or the available resources are scarce. The following lemma helps characterizing the feasibility set.

Lemma 5. *The solution of the problem:*

$$\min_{n_1,\ldots,n_M,P_1,\ldots,P_M,M} \quad \varepsilon_M(n_1,\ldots n_m,P_1,\ldots P_m,g) \tag{2.22}$$

$$\textit{s.t.} \quad \sum_{i=1}^{M} n_i \leq N \tag{2.23}$$

$$\sum_{i=1}^{M} n_i P_i \leq \mathrm{E_t} \tag{2.24}$$

is $M=1$ *with* $(n_1,P_1)=(N,\frac{\mathrm{E_t}}{N})$. *For meaningful/practical solutions, we restrict to*

$$n_i \geq Q^{-1}(10^{-9}) \approx 36, \textit{ and} \tag{2.25}$$

$$\max\{Q(0.45\sqrt{B\ln 2}),10^{-9}\} < \varepsilon_M < 0.5. \tag{2.26}$$

Proof. See Appendix B.1 □

Lemma 1 tells us that the best blocklength-power allocation of IR-HARQ within a coherence block for minimizing the outage probability given a maximum available amount of energy and channel uses is to employ one packet consuming all the available blocklength and energy. The nice property is that this is independently of the channel realization g. Otherwise stated the safest way to increase the reliability is to use directly all the resources in one shot, no matter the channel g.

Since the knowledge of the channel is not required it means that both statistical and full CSI can follow this strategy. This remark allows us to draw the borderlines of the feasibility region. Infeasibility for our Problems 8 and 10 occurs if with resources $(N,\mathrm{E_t})$, is impossible to reach reliability $1-T_{\mathrm{rel}}$ which according Lemma 5 it corresponds to $\min\{\mathbb{E}_g[\varepsilon_2(n_1,n_2,P_1,P_2,g)]\} > 1-T_{\mathrm{rel}} \Leftrightarrow \mathbb{E}_g[\varepsilon_2(N,0,\frac{\mathrm{E_t}}{N},0,g)] > 1-T_{\mathrm{rel}}$.

Passing to Problem 9 the simple scheme for full CSI which will be the one we apply provided full CSI, we cannot always apply the policy for maximizing the feasibility region. When $g \geq g_{th}$ it is permissible but specifically for deep fading cases, i.e. $g < g_{th}$ we avoid the transmitting. So to find now the feasibility region we must concentrate on the interval $[g_{th},\infty)$ if it is possible to reach error probability ε_{on}. It is easy to check that the minimum error probability is decreasing as the channel gain gets larger. So the infeasibility can be checked only by only looking at the worst channel $g=g_{th}$. Consequently, if $\varepsilon_2(N,0,\frac{\mathrm{E_t}}{N},0,g_{th}) \leq \varepsilon_{on}$, there are feasible solutions. Obviously the restriction to stay inactive during deep fading events leads the policy driven by Problem 9 to have smaller feasibility region than the one by Problems 8 and 10.

Lemma 5 brings also an interesting corollary:

Corollary 5.1. *The solution of the problem:*

$$\min_{n_1,...,n_M,P_1,...,P_M,M} \quad \sum_{i=1}^{M} n_i P_i \tag{2.27}$$

$$\textit{s.t.} \quad \sum_{i=1}^{M} n_i \leq N \tag{2.28}$$

$$\varepsilon_M(n_1,...n_m,P_1,...P_m,g) \leq 1 - T_{\rm rel} \tag{2.29}$$

is $M = 1$ *with* $n_1 = N$ *and power* $P_1 = P_o$ *such that* $\varepsilon_1(N, P_o, 1) = 1 - T_{\rm rel}$*, given the additional restrictions* (2.25) *and* (2.26) *hold.*

Proof. Assuming that the optimal solution of the problem is different than (N, P_o) and it is with $M = M^\star > 1$ and $(n_1^\star, ..., n_{M^\star}^\star, P_1, ..., P_{M^\star}^\star)$. The first solution consumes power $E_o = NP_o$ and the second $E^\star = \sum_{i=1}^{M^\star} n_i^\star P_i^\star$. We assume $E^\star < E_o$ and prove it leads to contradiction. Casting an error minimization problem like in lemma 5 with resources (N, E_o), we know that problem has the optimal solution $(n_1, P_1) = (N, P_o = \frac{E_o}{N})$ and so the minimum attainable value the objective function can have is $1 - T_{\rm rel}$. But this leads to a contradiction since we can use the point $(n_1^\star, ..., n_{M^\star}^\star, P_1, ..., P_{M^\star}^\star)$ which spends less energy $E^\star$ and attains the same or smaller value of the objective (due to the constraint (2.29)), i.e. $1-T_{\rm rel}$ and reach to a better solution. If it is smaller than $1 - T_{\rm rel}$ we already reached to a better than optimal solution. If it is equal to $1 - T_{\rm rel}$ then by simply increasing some power $P_i^\star$ with the available surplus $E_o - E^\star$ will do the job and this concludes the proof. □

Setting $g = 1$, we return to the AWGN case. Interestingly, the corollary describes the a problem almost the same as Problem 1 but it gives a contrasting solution. The difference of the Corollary's Problem is that the objective is to minimize the maximum available energy we need to have to attain the URLLC constraints but in Problem 1 we minimized the averaged consumed energy. At that problem HARQ could save even 25% of energy by using IR-HARQ and here the one-shot transmission is optimal.

2.6 Numerical Results and Discussion

We assume $B = 256$ information bits (32 bytes) have to be transmitted through a Ricean fading channel with K-factor and unit-variance, i.e. $|h| \sim Rice(K, 1)$. The K-factor represents the ratio between the direct path (Line Of Sight) and the other paths. $K = 0$ corresponds to the Rayleigh fading while $K \to \infty$ corresponds to the AWGN. We also assume that $n_1 \geq 100$ such that Polyanskiy's formula approximation (2.1) is accurate and also that $1 - T_{\rm rel}, \varepsilon_{on} \in [10^{-9}, 0.5]$ to satisfy Eq. (2.26). The solutions of the Problem 9 for the full CSI (named henceforth *Full CSI simple* in figures) case are found using a 4D grid search. 1D over g_{th} and for every $g > g_{th}$ a 3D over (n_1, n_2, P_1) because P_2 can then be found through (2.9). Luckily the channel gain only scales the powers (P_1, P_2) so finding for one g the optimal configuration determines the configurations corresponding to almost all the other g just by scaling the powers (hence we get away with approximately $4D$ search and not 5D). This is not true only for g close to g_{th} where the power needs to be scaled too high and the constraints (2.14-2.15) may not allow it so in that case a different configuration of power-blocklengths has to be found. For the Problem 10 with the

statistical CSI (named henceforth *Stat. CSI* in figures) to a 3D grid search over (n_1, n_2, P_1) finds the optimal point.

In Figure 2.1, we depict the feasibility regions in $(\mathrm{E_t}, K)$ for different CSI configurations and different N. For the same constraints in latency N and reliability T_{rel}. As discussed previously the feasibility region for full CSI when we use the scheme of avoiding the deep fading, is smaller than the one with only statistical CSI. We remind that for full CSI, the transmitter policy is to remain idle when $g < g_{th}$, so additional resources are needed when it is active to achieve a pre-fixed outage probability ε_{on} smaller than $1 - T_{\mathrm{rel}}$ to compensate for. The full CSI policy is more constrained. The threshold g_{th} cannot be tuned to zero since we force for every $g \geq g_{th}$ an error probability $\varepsilon_{on} \leq 1 - T_{\mathrm{rel}}$ to be achieved and this requires an infinite amount of resources when $g \to 0$. We also observe from Figure 2.1, that the reliability constraint T_{rel} strongly affects the feasibility region, while this is not the case for the latency constraint N. We emphasize that, as we will see later, when both CSI setups are feasible, the full CSI outperforms the statistical one.

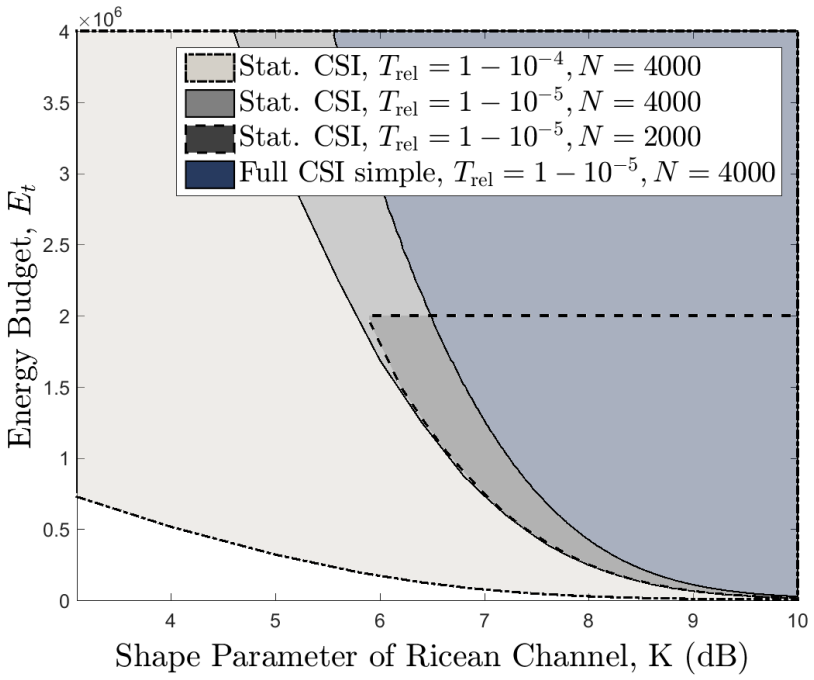

Figure 2.1: Feasibility region for different channel, $B = 32$Bytes, maximum energy budget $\mathrm{E_t} = P_{\max} N$ with $P_{\max} = 30dB$.

In Figure 2.2, we plot the relative throughput $\frac{\mathcal{T}_h(a)}{\mathcal{T}_{h,max}}$ (left scale) and relative energy $\frac{\mathcal{E}(a)}{\mathcal{E}_{min}}$ (right scale) versus a. Performing either throughput maximization ($a = 0$) or energy minimization ($a = 1$) is not a good strategy since by allowing a small decrease of throughput (in the first case) or a small increase of energy (in the second case), the other metric in the objective function significantly improves. A good tradeoff for full and statistical CSI is around $a = 0.3$ in the employed here.

By solving the optimization problems for every $\alpha \in [0, 1]$ we get the Pareto frontier for throughput and energy which is displayed in Figure 2.3 for various setups. We remark that the K factor as well as the target reliability play the important role. On the contrary, the constraints on latency N, energy $\mathrm{E_t}$ and power $\mathrm{P_{max}}$ seem to have a minor impact except when they are so stringent that we get close to the boundary of the feasibility area.

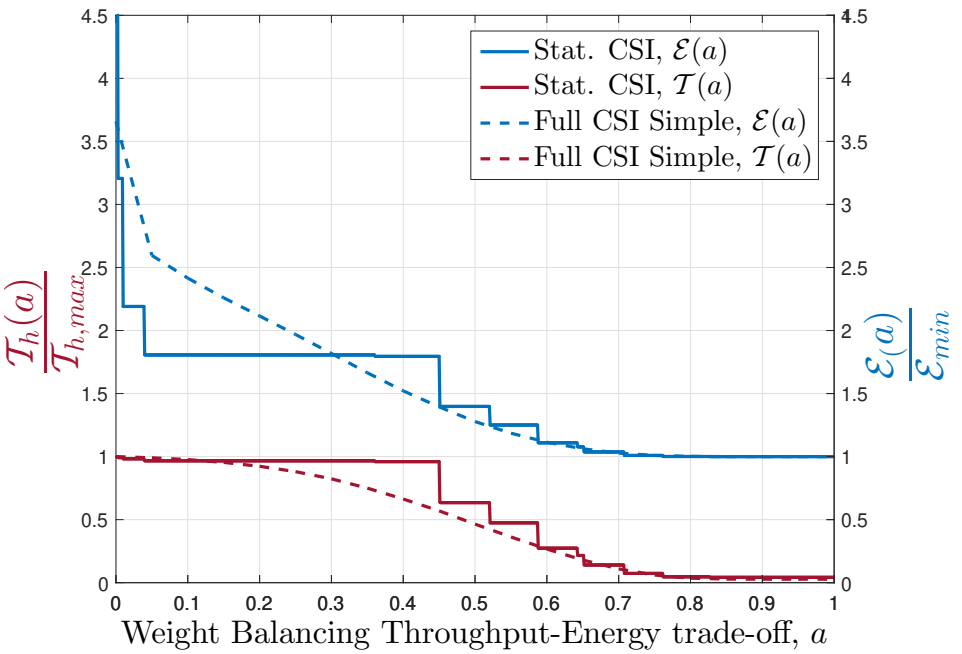

Figure 2.2: Throughput and energy relative to their optimal value for Ricean channel with $K = 7$dB, $B = 32$Bytes, $1 - T_{\rm rel} = 10^{-5}$ and maximum energy $\mathrm{E_t} = P_{\max}N$ with $P_{\max} = 30$dB and $N = 4000$.

Focusing on the full CSI case we can explain why resources (N, E_t) don't affect the performance as long as feasibility is assured. The reason is that we measure the *average* throughput and energy which are affected mostly by the most probable scenario where there is no deep fading and only a small portion of the resources (N, E_t) is needed. So the full (N, E_t) is mainly needed only for assuring reliable communication when deep fading occurs. But since deep fading is a rare instance does not change the *average* performance. Indeed if we gradually reduce (N, E_t), the performance remains almost the same until the point it is infeasible to attain the targeted reliability. This observation is substantial since it can help us assess when full CSI preferable to statistical. The answer is that assuming $(\beta N, \beta \mathrm{E_t})$ (with $\beta \in (0, 1)$) resources are needed to send pilots so as to learn the channel and enable full CSI scheme, then if with the rest $((1-\beta)N, (1-\beta)E_b)$ is still feasible to achieve with full CSI the targeted reliability then full CSI is preferable. We have to stress the fact that even with the entire (N, E_t) the feasibility region of the scheme full CSI Simple we used, is already smaller than the statistical one, so limiting it even more so as to send pilots and learn the channel makes the full CSI scheme even less robust.

In Figure 2.4, we display again the Pareto frontier for the throughput and energy when HARQ is carried out or when one shot transmission is employed but with the same resources $(N, \mathrm{E_t})$. With full CSI a constant 37% percent, according to the figure, of energy can be saved for the same throughput by using HARQ instead of one-shot. This gain for statistical CSI scheme depends substantially on the channel quality K and it can become huge for poor channel conditions.

To explain this behavior we first discuss the optimal configuration of (n_1, n_2, P_1, P_2). The first packet is of significant importance since we measure average performance and the first packet is always sent whereas the second only ε_1 times. For throughput maximization n_1 should be kept as small as possible at the expense of power P_1. However, as we move to energy minimization, the situation is reversed, as larger n_1 with smaller P_1 reduces required energy. The role of the

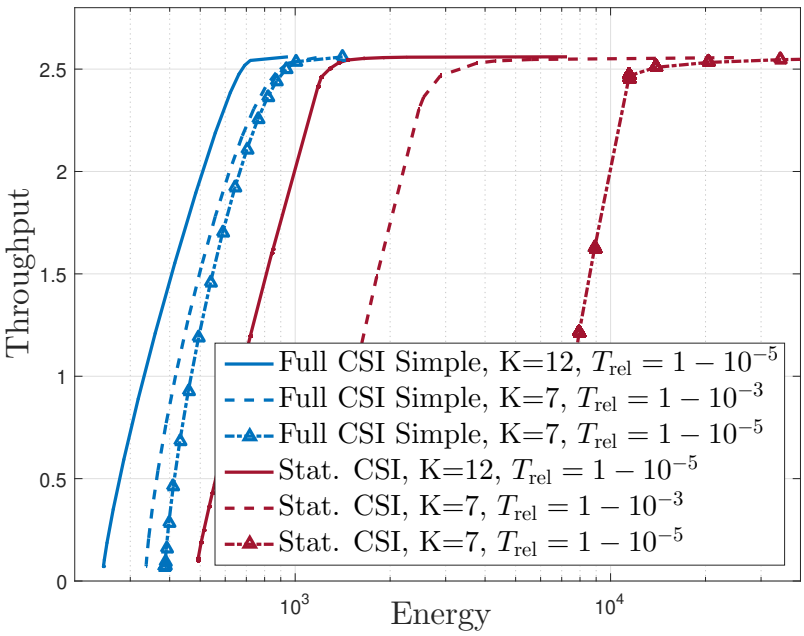

Figure 2.3: Pareto frontier for throughput and energy, with $\mathrm{E_t} = P_{\max}N$, $P_{\max} = 30dB$, and $N = 4000$.

second packet is mainly to successfully meet the constraints of the optimization problem and not to improve the objective (which is mainly the role of the first packet). This behavior is similar for both schemes using either full or statistical CSI.

Specifically for statistical CSI, where the channel coefficient is unknown, we see the mechanism of the optimized HARQ rendering the first packet responsible for achieving a good value of the objective function by focusing on the instances where the channel is good. The second packet will be employed only when the channel is bad and necessarily a lot of resources must be spent. In one-shot there is not this option of differentiating the good and bad realizations of the channel. Unfortunately, the bad channel realizations will be the ones to determine the amount of resources needed to spend for all cases (bad and good channel realizations). Reasonably, as channel statistics deteriorate (K decreases) the waste of resources in one-shot scheme becomes more profound since the bad channel realizations determining the expenditure of resources get worse. On the contrary, in the case of full CSI the surprising savings we see for statistical CSI do not happen. This is because the channel is known also for the one-shot scheme and there can be a distinction between good and bad channel realizations without having to rely on retransmissions. Moreover, this results to an almost constant save of energy given a specific throughput, independently of channel quality.

2.7 Conclusion

The focus of this chapter was the optimization of IR-HARQ under strict latency and reliability constraints in fading channels. We explored two different types of CSI, i.e. one where the exact value of channel is known and one where only its statistics is available. Even though full CSI can yield

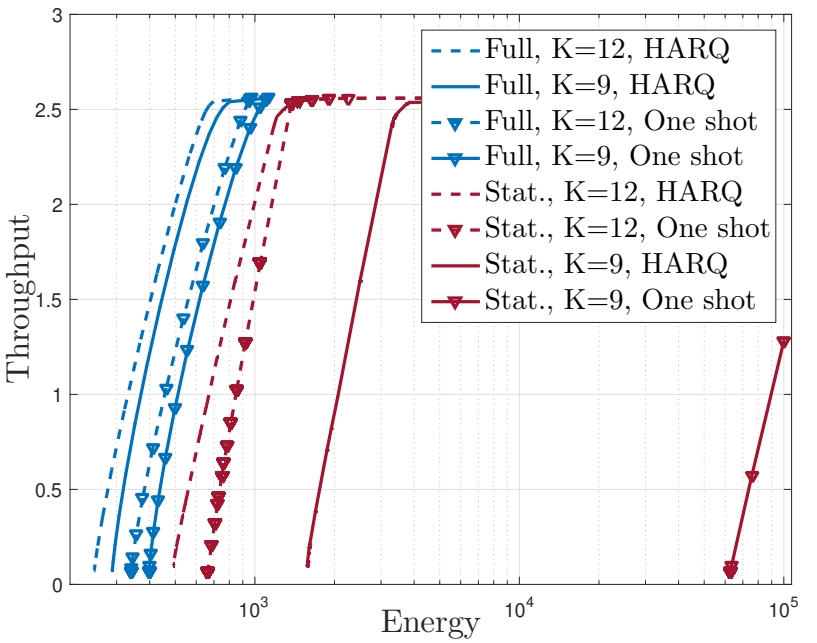

Figure 2.4: Pareto frontier for throughput and energy when HARQ or one shot transmission is used, with $\mathrm{E}_b = P_{\max}N$, $P_{\max} = 1000$ (30dB), and $N = 4000$. For readability in the legend we reduced “Stat. CSI” to “Stat.” and “Full CSI Simple” to “Full”.

considerable gains simultaneously in both energy and throughput compared to statistical CSI, it turns out that it is not necessarily more robust. Due to the difficulty of applying a scheme optimally exploiting full CSI, there can be a considerable shrinkage of the feasibility regime compared to statistical CSI under the same resources. We did not even account that a certain number of pilots have to be employed to acquire the full CSI which would definitely deteriorate the feasibility of a scheme depending on full CSI. So far, we focused only on a single user link under specific quality of service requirements. In the next chapter, we extend the resource allocation problem not only across transmission rounds of a single user but also across multiple users each having its own quality of service requirements.

Chapter 3

Deep Reinforcement Learning for Centralized Scheduling under Multi-class Traffic

3.1 Introduction

In the previous chapters, we mainly focused at the lower layers of the communication protocol and aim at optimizing the physical/link layer for transmitting packets that have to be delivered with high reliability and within a very short time interval. We investigated whether it is beneficial to break this short time interval into even smaller pieces in point-to-point communications. We would like now to consider multiuser downlink communications, moving as well one layer up, towards the media access control (MAC) layer. We retain the requirement that users have to be served within a specific time interval, and again with the possibility of using retransmissions within that interval. Nevertheless, unlike previous chapters, we consider that this delay constraint, even if it is a strict one, does not correspond to a extremely short time, enabling us to rely on the standard long blocklength assumption for deriving information-theoretic metric.

We consider a traffic of users that when they appear in the system, they require a specific amount data to be successfully delivered to them within a specific number of time slots. Since the base station to which they are connected to does not have infinite resources, it must carefully allocate resources to meet its users' requirements. Therefore the problem we consider here is that of centralized resource scheduling at each time slot so as to satisfy the service requirements of connected users. Specifically, we consider multi-class (heterogeneous) traffic, i.e., every user belongs to one traffic class that determines its service requirements and therefore the quality of service (QoS) that needs to be provisioned. Each class sets both the requested packet size and the maximum latency, measured in number of time slots a user is eager to wait until it gets the packet successfully. So every user has a strict latency constraint and within that time interval the base station has to spend the appropriate amount of resources so as to satisfy this user. If the resources were not enough and the transmission fails, then there is the possibility of retransmission but only within the strict (but not too short) latency constraint. The retransmission protocol we assume in this chapter is a simple Automatic Repeat Request (ARQ), according to which if the base station fails in one time slot to meet the user's requirements, then it has to spend again,

in a future time slot, resources to retransmit the whole packet without relying on any previous packet transmissions (as it is the case of IR-HARQ in the previous chapters). Since in every time slot different number of users, belonging to different classes, may appear, the scheduler has to take cumbersome decisions on how to distribute resources so as in the long term the number of satisfied users is maximized. These decisions depend on how many users are simultaneously active, how much data they require, how many more time slots are each user eager to wait and (if it is available) their channel quality. On top of that, the history of those parameters has to be considered in order to exploit possible time dependencies, which may improve the efficiency of the scheduler. For example, sometimes it may be better to spend more resources on a demanding user with bad channel quality because it is approaching its maximum latency constraint and sometimes to ignore that user because it is too demanding and at the same time some others appeared with surprisingly good channel quality and is more efficient to serve them.

In front of such complex problems a good approach is using more versatile tools. We use the combination of Reinforcement Learning (RL) and deep neural networks. This combination, which is called Deep Reinforcement Learning (DRL), has recently attracted significant attention and has been used in various scientific fields. To tackle the difficulties of our specific problem, we leverage on several existing ideas and develop a new DRL algorithm capable of competing and surpassing traditional approaches.

3.2 Traffic Model

The traffic model consider here is as follows: users appear and disappear continuously and each belongs to a specific class. The characteristics of those classes describe statistically the traffic. Each class has the following attributes:

- Data size D: the size of data a user of that class asks for.
- Maximum Latency L: the maximum number slots within which the user needs to successfully receive the packet of size D, so as to be satisfied.
- Arrival probability p: the probability a new user belonging to this class arrives in the system.

Additionally, we introduce a parameter to incorporate the notion of different importance, denoted by α, different classes may have. This parameter can be used to either dictate the scheduler to prioritize some classes needing higher success probability or maybe (from a service provider perspective) to prioritize a class of users with more privileged contracts (SLAs), demanding better service.

We denote $\mathcal{C}$ the set of classes. Every user that enters the system, belongs to a class $c \in \mathcal{C}$ with probability p_c and is characterized by the tuple (D_c, L_c, α_c). We assume that a maximum number K of users can exist per time slot. We also assume that a new user appears whenever a previous one has reached the maximum time it can be present in the system. For example, if a user appears at time $t = 1$, belonging to a class $c \in \mathcal{C}$ with $L_c = 4$, then even if it gets immediately and immediately successfully its requested packet of size D_c at $t = 1$, it will still remain in the system until a new user appears at $t = 5$ belonging to a class $c' \in \mathcal{C}$ with probability $p_{c'}$. Therefore, at every time slot a set U_t (with constant cardinality $|U_t| = K$) of users is observed, with some of them belonging to set $U_t^{act} \subseteq U_t$, with some demanding resources from the base station as being unsatisfied and some being already satisfied. To alleviate the assumption of

always K users and incorporate the fact that the number of user can fluctuate over time, we add a null class c_0 with $D_{c_0} = 0$, $\alpha_{c_0} = 0$ and $L_{c_0} > 0$. Therefore a user of the null class is equivalent to that no user has appeared. The null class effectively will result in occasionally having less than K users.

The rationale behind this specific traffic model is as follows: (i) we wanted to have a traffic model under which users with different strict data and latency requirements come and go, and that is both quite generic and also tractable enough so as to permit to build benchmarks against which we compare the DRL scheduler. We emphasize that the RL algorithm can adapt and train neural network models on other types of traffic and we believe that under proper parameter tuning, it will give good results. (ii) we wanted the traffic to remain uninfluenced by the scheduler decisions. For instance, if we made the common assumption that whenever a user is satisfied a new one arrives with some probability per time slot, then the scheduler performance affects the statistics of the traffic. This is due to the fact that at a given time interval, a scheduler with abundant resources will see more users than schedulers with poor resources, since more resources means satisfying users earlier leading to (statistically) more users appearing.

3.3 Channel and data rate models

The users are assumed to be uniformly distributed within a concentric ring. Therefore the distance of a user u from the base station is a random variable with a probability density function: $f_{\mathrm{d}}(\mathrm{d}_u) = \frac{2\mathrm{d}_u}{\mathrm{d}_{max}^2 - \mathrm{d}_{min}^2}, \mathrm{d}_u \in [\mathrm{d}_{min}, \mathrm{d}_{max}]$. Furthermore we assume that the mobility of the users is not high enough to change significantly within their limited time interval they are active. Consequently, their distances from the base station are kept constant. In contrast, the modification of the channel due to small scale fading is taken into account and described below.

Multiple users can be served simultaneously and in this work we assume that they are allocated on different orthogonal frequency bands and so there is no interference between them. We also assume that they experience flat block fading and therefore every user has a constant channel gain for a given time throughout all the available frequency band from which a user is served. Let a user u that appeared at time t_0, with channel gain at time t is $g_{u,t} = \frac{C_{pl}|h_{u,t}|^2}{\sigma_N^2}\mathrm{d}_u^{-n_{pl}}$ with n_{pl} being the pathloss exponent, C_{pl} a constant to account for the constant losses and σ_N^2 is the noise power spectrum density. The distance d_u remains constant throughout out the lifespan of user u but there is a small scale Rayleigh fading changing in every time slot according to the Markovian model:

$$h_{u,t_0} \sim \mathcal{CN}(0,1)$$
$$h_{u,t} = \rho h_{u,t-1} + Z, \quad \text{with} \quad Z \sim \mathcal{CN}(0, 1-\rho^2), t > t_0$$

where $\mathcal{CN}(0, v)$ represents a circular complex normal distribution with zero mean and variance v. The parameter $\rho = J_0(2\pi f_d T_{slot}) \in [0, 1]$ [40] determines the time correlation of the channel where $J_0(\cdot)$ being the zeroth-order Bessel function of the first kind, f_d the maximum Doppler frequency (determined by the mobility of the users) and T_{slot} the slot duration. If $\rho = 0$ (high mobility at the small scale level), in every time slot the user has an independent realization of the fading distribution. If $\rho = 1$ (absence of mobility), the fading is constant throughout the user's lifespan.

We consider two cases about the knowledge of the channels at the scheduler side:

- Statistical CSI: at current time, the scheduler knows the location of the active users and their

fading statistics. For future users coming in the system, only statistics of both the location and the channel are known.

- Full CSI: at current time, the scheduler knows the location and the fading of the active users.

We assume the Shannon rate formula is valid and that the base station operates on capacity level providing to user u at time t data equal to $w_{u,t}\log_2(1+g_{u,t}P_{u,t})$, where $P_{u,t}$ is the transmitted energy per channel use/symbol and $w_{u,t}$ the assigned bandwidth. An outage happens when the user's data requirement is higher than what the channel can support. For instance, in the statistical CSI case if we consider the first transmission of an active user (which means that previous channel realization is unknown) at given time t_u, then this user at distance d_u from the base station, belonging to class $c \in \mathcal{C}$ with resources $(w_{u,t}, P_{u,t})$ has a probability of failing to successfully decode its packet equal to:

$$\begin{aligned} P_u^{fail}(w_{u,t}, P_{u,t}; \mathrm{d}_u) &= \mathbb{P}(w_{u,t}\log_2(1+g_{u,t}P_{u,t}) < D_u | \mathrm{d}_u) \\ &= \mathbb{P}(|h_{u,t}|^2 < \zeta_{u,t}\mathrm{d}_u^{n_{pl}}) \\ &= 1 - e^{-\zeta_{u,t}\mathrm{d}_u^{n_{pl}}} \end{aligned} \tag{3.1}$$

with $\zeta_{u,t}=\dfrac{\sigma_N^2(2^{D_u/w_{u,t}}-1)}{C_{pl}P_{u,t}}$.

Now if the location d_u of this user is unknown by the scheduler, the error probability becomes

$$\begin{aligned} P_u^{fail}(w_{u,t}, P_{u,t}) &= \mathbb{P}(w_{u,t}\log_2(1+g_{u,t}P_{u,t}) < D_u) \\ &= \int_{\mathrm{d}_{min}}^{\mathrm{d}_{max}} P_u^{fail}(w_{u,t}, P_{u,t}; \mathrm{d}) f_{\mathrm{d}}(\mathrm{d}) d\mathrm{d} \\ &= 1 - \frac{\Gamma(\frac{2}{n_{pl}}, \zeta_{u,t}\mathrm{d}_{min}^{n_{pl}}) - \Gamma(\frac{2}{n_{pl}}, \zeta_{u,t}\mathrm{d}_{max}^{n_{pl}})}{n_{pl}\zeta_{u,t}^{2/n_{pl}}(\mathrm{d}_{max}^2 - \mathrm{d}_{min}^2)/2} \end{aligned} \tag{3.2}$$

where $\Gamma(s,x) = \int_x^\infty t^{s-1}\,\mathrm{e}^{-t}\,\mathrm{d}t$ is the upper incomplete gamma function. For the sake of simplicity, we overloaded notation by allowing x in D_x, α_x, L_x to either denote a class x or a user x belonging to a class with those characteristics.

3.4 Scheduling procedure

The base station is called to appropriately use in every time-slot its energy and bandwidth resources to satisfy its users. We concentrate only on the bandwidth distribution, assuming no power adaptation and simplifying the base station job that spends a fixed amount of energy per channel use, i.e., $P_{u,t} = P, \forall u, t$. If the available bandwidth at the base station's disposal is W then the scheduler aims to find the $(w_{u_1,t}, w_{u_2,t}, ...) \in \mathbb{R}_{\geq 0}^{|U_t^{act}|}$ with $u_1, u_2, ... \in U_t^{act}$ such that

$$\sum_{u \in U_t^{act}} w_{u,t} \leq W, \quad \forall t$$

and maximize over the time horizon the accumulated reward for every satisfied user which is described by the following objective "gain-function":

$$G = \sum_{t} \sum_{u \in U_t^{act}} \alpha_u \mathbb{1}\{w_{u,t} \log_2(1+g_{u,t}P) > D_u\}. \tag{3.3}$$

We stress out that a user u remains on the set U_t^{act} for a time interval less or equal to its maximum acceptable latency L_u. If not satisfied within that interval then he does not contribute positively to the objective G.

As implied by (3.3), the retransmission protocol adopted is Automatic Repeat reQuest (ARQ). If a user fails to correctly decode the received packet then this packet is ignored (no buffering at the receiver side) and the user waits until the base station sends again the same packet to try to decode. Finally, as stated previously, we consider two different CSI cases. In the first scenario, *statistical CSI*, only statistical properties of channel and location are known at current time t_c and the future, while, in the second scenario, *full CSI*, the exact value of the channels $h_{u,t_c}, \forall u \in U_{t_c}^{act}$ and the location of the users (and so $\mathrm{d}_u \forall u \in U_{t_c}^{act}$) are known at the current time t_c.

3.5 Benchmark procedures for the scheduler

In this section we describe the benchmark procedures/algorithms for both scenarios (statistical CSI and full CSI). The procedures will be compared to our DRL based algorithms in Section 3.10.

3.5.1 Case 1: Statistical CSI available only

We first concentrate on the case of a single user u_0 appearing at time t_0. The current time is $t_c \in [t_0, t_0 + L_{u_0} - 1]$. We denote by $\overrightarrow{w}_{u_0,t} = (w_{u_0,t_0}, w_{u_0,t_0+1}, ..., w_{u_0,t})$ the assigned bandwidth from time t_0 (beginning of transmission for user u_0). Additionally, let $A_{u_0,t}$ be a binary random variable which if $A_{u_0,t} = 1$ then u_0 is still unsatisfied at the end of time slot t (*after* receiving $\overrightarrow{w}_{u_0,t}$ resources) and $A_{u_0,t} = 0$ otherwise. Given that at the beginning of time t user u_0 is still unsatisfied and that we know the resource allocation $w_{u_0,t}$ is scheduled to be done at time t, we define $\Phi(\overrightarrow{w}_{u_0,t}; d_{u_0})$ to be the probability that $w_{u_0,t}$ is still not enough when the location d_{u_0} is known but the channel $h_{u_0,t}$ is unknown:

$$\Phi(\overrightarrow{w}_{u_0,t}; \mathrm{d}_{u_0}) = \begin{cases} \mathbb{P}(A_{u_0,t} = 1 | \overrightarrow{w}_{u_0,t-1}, \mathrm{d}_{u_0}, A_{u_0,t-1}{=}1), & t > t_0 \\ \mathbb{P}(A_{u_0,t} = 1 | \mathrm{d}_{u_0}), & t = t_c = t_0. \end{cases} \tag{3.4}$$

The average contribution of user u_0 to the gain function (3.3) on the time interval $[t_c, t]$ is given by the following equation, derived applying the chain rule for conditional probability:

$$\mathfrak{g}_{u_0}^{[t_c,t]}{=}\mathfrak{g}(w_{u_0,t_c}, ..., w_{u_0,t}; \mathrm{d}_{u_0}) = \begin{cases} 0, & \text{if } t_c > t_0 \text{ and } A_{u_0,t_c-1}{=}0 \\ \alpha_{u_0}\big(1 - \prod_{j=t_c}^{t} \Phi(\overrightarrow{w}_{u_0,j}; \mathrm{d}_{u_0})\big), & \text{else.} \end{cases} \tag{3.5}$$

Now we consider that the average contribution on the gain function (3.3) for the the future users following the user u_0. The next user (if it exists) appears at time $t_1 = t_0 + L_{u_0}$, and so on. Therefore

we consider the users noted as $u_1, u_2, \ldots$ that will appear at $t_1 = t_0 + L_{u_0}, t_2 = t_1 + L_{u_1}, \ldots$. We denote that with probabilities $p_{c_1}, p_{c_2}, \ldots$ they will belong to classes $c_1, c_2, \ldots$, respectively (and one of these classes may be the null class). These classes will determine the maximum latencies $L_{u_1}, L_{u_2}, \ldots$ and consequently the time arrivals $t_1, t_2, \ldots$ all being random variables. As we consider here future users, even their locations are unknown. Consequently we need to average over the locations the equations (3.4) and (3.5) to obtain their contribution on the gain function (3.3). So for $i \geq 1$ if $\overrightarrow{w}_{u_i,t} = (w_{u_i,t_i}, w_{u_i,t_i+1}, ..., w_{u_i,t})$, we have

$$\mathfrak{g}_{u_i}^{[t_i,t]} = \mathfrak{g}(w_{u_i,t_i}, ..., w_{u_i,t}) = \alpha_{u_0}\big(1 - \prod_{i=t_c}^{t} \Phi(\overrightarrow{w}_{u_i,i})\big) \tag{3.6}$$

where the contribution looking at time t with $t < t_i + L_{u_i}$ starts at time t_i for user u_i and where

$$\Phi(\overrightarrow{w}_{u_i,t}) = \begin{cases} \mathbb{P}(A_{u_i,t} = 1 | \overrightarrow{w}_{u_i,t-1}, A_{u_i,t-1}{=}1), & t > t_i \\ \mathbb{P}(A_{u_i,t} = 1), & t = t_i. \end{cases} \tag{3.7}$$

Hence, the averaged value of gain function for the sequence of users $u_0, u_1, ...$ (so when one user at most is active per time slot, ie, $K = 1$) starting at the current time t_c is:

$$\mathcal{G}(w_{u_0,t_c}, ..., w_{u_0,t_1-1}, w_{u_1,t_1}, ...) = \mathfrak{g}_{u_0}^{[t_c,t_1-1]}(.; d_{u_0}) + \sum_{c_1 \in \mathcal{C}} \left(p_{c_1} \cdot \mathfrak{g}_{u_1}^{[t_1,t_2-1]}(.) + \sum_{c_2 \in \mathcal{C}} \left(p_{c_2} \cdot \mathfrak{g}_{u_2}^{[t_2,t_3-1]} 5.) + \sum_{c_3 \in \mathcal{C}} (...) \right) \right). \tag{3.8}$$

From (3.8), we observe a tree structure[1] that when a user vanishes there is a summation over all the possibilities of the classes that the new user can belong to. Therefore a number of branches is equal to the number of possible classes ($|\mathcal{C}|$). To manage the scalability issue, we propose to cut the tree by considering only T future time slots, so to work with the finite horizon $[t_c, t_c + T - 1]$.

Finally, the general case with multiple users served simultaneously ($K > 1$) is easy to be considered by just computing K "parallel trees". With a slight abuse of notation, we consider that the first subscript of the variables w now refers to the index of the tree (and implicitly to a specific user). As a consequence, the variables for the scheduled bandwidth resources over an horizon of length T can be put into the following matrix:

$$\mathbf{W}_{t_c} = \begin{bmatrix} w_{1,t_c} & w_{1,t_c+1} & \cdots & w_{1,t_c+T-1} \\ w_{2,t_c} & w_{2,t_c+1} & \cdots & w_{2,t_c+T-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{K,t_c} & w_{K,t_c+1} & \cdots & w_{K,t_c+T-1} \end{bmatrix}$$

and the average gain for these resources takes the following form:

$$G(\mathbf{W}_{t_c}) = \sum_{k=1}^{K} \mathcal{G}(w_{k,t_c}, w_{k,t_c+1}, \cdots, w_{k,t_c+T-1}). \tag{3.9}$$

Finally our optimization problem whose solution constitutes the benchmark procedure for the sta-

[1] A simple way to be computed is recursively

tistical CSI case is the following one at current time t_c:

$$\max_{\mathbf{W}_{t_c} \in \mathbb{R}_{\geq 0}^{K \times T}} \quad G(\mathbf{W}_{t_c}) \tag{3.10}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} w_{k,t} \leq W, \quad \forall t \in \{t_c, \ldots, t_c+T-1\}. \tag{3.11}$$

It can be easily shown that the objective function $G(\cdot)$ is non-concave with multiple local optimums. In contrast, the constraints given by Eq. (3.11) describe a compact and convex domain set. Consequently, we may apply the so-called Frank-Wolfe algorithm [41]. The idea behind this algorithm is as follows: at each iteration, the algorithm starts from a point and approximates the objective function around it with a linear (first-order) approximation. Then it solves the corresponding Linear Programming problem (LP) to find the best solution which will be the starting point of the next iteration. The procedure terminates when the algorithm converges to a local optimum, i.e., when the objective function does not increase anymore significantly. In order to exhibit a solution close to the global optimum, the algorithm is repeated N_{init} times with different randomly chosen initial points. At the end, we peak the best local optimum. This Frank-Wolfe algorithm is known to have sublinear convergence speed. In our set-up, we remark that it always converges within few, reasonable number of iterations (≤ 20).

Before proceeding further, we provide here some general remarks.

- The above benchmark procedure takes into account the past through (3.4) since all the previously allocated resources are involved.
- The procedure at current time t_c proposes a solution for the scheduler for the current time t_c and also for the future $[t_c+1, t_c+T-1]$. Nevertheless, as this procedure will be recomputed at time t_c+1 (once the actions proposed for time t_c is applied and new information about the transmission's success or failure are available), the actions proposed at time t_c for time t_c+1 are generally not applied. Obviously we will apply at time t_c+1 the solution advocated by the procedure computed at time t_c+1.
- The Frank-Wolfe method is sublinear but the computation of the objective function (3.9) and its partial derivatives grow exponentially with T which leads in practice to a slow and cumbersome method (not to mention that to be sure to retrieve a good local optimum we repeat the process N_{init} times).
- Lastly, the algorithm treats the "mean" case. It does not specify what really happens in the future since it only evaluate what happens in the future on average. It would be possible to address every future scenario differently but by skyrocketing the number of variables and constraints, making the already-slow benchmark procedure.

Hereafter, we concentrate on calculating (3.4) and (3.7) for different channel model subcases. The rest of the benchmark procedure is straightforward[2].

[2]Perhaps it is tricky to also find the derivative of (3.2) which is required for the first-order approximation in the Franck-Wolfe algorithm. So we get

$$\frac{dP_u^{fail}}{dw} = \int_{\mathrm{d}_{min}}^{\mathrm{d}_{max}} \frac{d\mathbb{P}(|h|^2 < \zeta_{u,t}\mathrm{d}^{n_{pl}})}{d\zeta_{u,t}} f_{\mathrm{d}}(\mathrm{d}) d\mathrm{d} \frac{d\zeta_{u,t}}{dw} = \frac{\Gamma(\frac{2+n_{pl}}{n_{pl}}, \zeta_{u,t}\mathrm{d}_{min}^{n_{pl}}) - \Gamma(\frac{2+n_{pl}}{n_{pl}}, \zeta_{u,t}\mathrm{d}_{max}^{n_{pl}})}{n_{pl}\zeta_{u,t}^{(2+n_{pl})/n_{pl}}(\mathrm{d}_{max}^2 - \mathrm{d}_{min}^2)/2} \frac{d\zeta_{u,t}}{dw}$$

3.5.1.1 The i.i.d. fading channel case ($\rho = 0$)

It is the simplest subcase since there are no time dependencies on the fading, so (3.4) and (3.7) become

$$\Phi(\overrightarrow{w}_{u_0,t}; \mathrm{d}_{u_0}) = P_{u_0}^{fail}(w_{u_0,t}, P; \mathrm{d}_{u_0}), and \tag{3.12}$$

$$\Phi(\overrightarrow{w}_{u_i,t}) = P_{u_i}^{fail}(w_{u_i,t}, P), \quad i \geq 1. \tag{3.13}$$

We remind the users u_i for $i \geq 1$ follow the user u_0, and therefore we average over their unknown locations.

3.5.1.2 The constant fading channel case ($\rho = 1$)

Now the channel is the same for each retransmission on the user. For user u_0, the channel is invariant but unknown. Only its location is known. At time $t > t_0$, we have

$$\begin{aligned}\Phi(\overrightarrow{w}_{u_0,t}; \mathrm{d}_{u_0}) &= \mathbb{P}(w_{u_0,t}\log(1+g_{u_0}P) < D_{u_0} | w_{u_0,t'}\log(1+g_{u_0}P) < D_{u_0}, \forall t' \in [t_0, t-1], d_{u_0}) \\ &= \frac{\mathbb{P}(w_{u_0,t''}\log(1+g_{u_0}P) < D_{u_0}, \forall t'' \in [t_0, t] | d_{u_0})}{\mathbb{P}(w_{u_0,t'}\log(1+g_{u_0}P) < D_{u_0}, \forall t' \in [t_0, t-1] | d_{u_0})}.\end{aligned}$$

Therefore we obtain

$$\Phi(\overrightarrow{w}_{u_0,t}; \mathrm{d}_{u_0}) = \begin{cases} \dfrac{P_{u_0}^{fail}(\max\{\overrightarrow{w}_{u_0,t}\}, P; \mathrm{d}_{u_0})}{P_{u_0}^{fail}(\max\{\overrightarrow{w}_{u_0,t-1}\}, P; \mathrm{d}_{u_0})}, & \text{if } t > t_0 \\ P_{u_0}^{fail}(w_{u_0,t}, P; \mathrm{d}_{u_0}), & \text{if } \quad t = t_0. \end{cases} \tag{3.14}$$

For the case of the future users (u_i with $i \geq 1$), the equations remain the same with the only change that the location of the users is unknown as well. So in Eq. (3.14), we just need to omit the d_u similarly to the i.i.d. case.

3.5.1.3 The general Markovian case ($\rho \in (0, 1)$)

This case is much more complicated due to the correlation between the channel realizations. Actually, at time t, the distribution of $h_{u,t}$ given the past (which is not known in practice) is Ricean distributed. More precisely, if the user u is active at $t-1$ and t, we have $\mathbb{P}(|h_{u,t}|{=}x \,\big|\, |h_{u,t-1}|) = \mathrm{Rice}(x; v_R = \rho|h_{u_0,t-1}|, \sigma_R^2 = \frac{1-\rho^2}{2})$ where v_R and σ_R^2 are the so-called Ricean parameters.

Let us focus on the user u_0 and we are looking at the time $t = t_0 + 1$. According to [42, eq: 37], we have:

$$\begin{aligned}\Phi(\overrightarrow{w}_{u_0,t_0+1}; \mathrm{d}_{u_0}) &= \int_0^{x_{u_0,0}} \int_0^{x_{u_0,1}} \mathbb{P}(|h_{u_0,t_0+1}|{=}x \,|y) \mathbb{P}(|h_{u_0,t_0}|{=}y) dx dy \\ &= 1 - \frac{e^{-x_1^2} Q_1(\frac{x_{u_0,0}}{\sigma_R}, \frac{\rho x_{u_0,1}}{\sigma_R}) - e^{-x_{u_0,0}^2} Q_1(\frac{\rho x_{u_0,0}}{\sigma_R}, \frac{x_{u_0,1}}{\sigma_R})}{2(1 - e^{-x_{u_0,0}^2})}\end{aligned} \tag{3.15}$$

with $x_{u_i,j} = \sqrt{\zeta_{u_i,t_i+j}} d^{-\frac{n_{pl}}{2}}, i \in \{0, 1\}$ and Q_M be the marcum Q-function.

For the future users ($u_i, i \geq 1$), we have at time $t = t_i + 1$ (we remind that user u_i starts its

transmission at time t_i):

$$\Phi(\overrightarrow{w}_{u,t_i+1}) = \int_{\mathrm{d}_{min}}^{\mathrm{d}_{max}} \Phi(\overrightarrow{w}_{u_i,t_i+1};\mathrm{d}_{u_i}) f_{\mathrm{d}}(\mathrm{d}) d\mathrm{d}. \tag{3.16}$$

where $\Phi(\overrightarrow{w}_{u_i,t_i+1};\mathrm{d}_{u_i})$ is given by (3.15) by replacing u_0 with u_i. This equation (3.16) is already intractable whereas we are just focusing on the two first adjacent retransmissions. Obviously, it is even worse if we consider more retransmissions. Therefore, in the rest of the paper, the benchmark procedures will be only designed for $\rho = 0$ or $\rho = 1$, even if tested in the general case $\rho \in (0, 1)$. More precisely, for any ρ, we apply the benchmark procedure designed for either $\rho = 0$ or $\rho = 1$, and keep the best result.

3.5.2 Case 2: full CSI

Let us work on the user u_0 at the current time $t_c \geq t_0$. We remind that for this case, the channel h_{u_0,t_c} is known and the location d_{u_0} is known as well. Consequently, the channel gain g_{u_0,t_c} is also available to the base station. But the future channels $h_{u_0,t}$ for $t > t_c$ are only statistically known.

The user u_0 is unsatisfied at t iff the allocated bandwidth $w_{u_0,t}$ is smaller than the following threshold

$$w^{th}_{u_0,t} = \frac{D_{u_0}}{\log_2(1 + g_{u_0,t} \cdot P)}.$$

Consequently, the error probability of user u_0 defined by (3.4) can be expressed:

$$\Phi(\overrightarrow{w}_{u_0,t};\mathrm{d}_{u_0}) = \begin{cases} \mathbb{P}(w_{u_0,t} < w^{th}_{u_0,t} | A_{u_0,t-1} = 1, h_{u_0,t_c}, d_{u_0}), & \text{if } t > t_c \\ \mathbb{1}\{w_{u_0,t_c} < w^{th}_{u_0,t_c}\}, & \text{if } \quad t = t_c. \end{cases} \tag{3.17}$$

In this case, we remark that the probabilities are not necessary continuous due the indicator function in (3.17). Consequently, the gain function described in (3.9) is now non-continuous over the variables $w_{k,t_c} \forall k$ (because we know exactly the channel gains at t_c and indicator functions occur at this time), but continuous for $w_{k,t}, t > t_c$ corresponding to the future. To overcome this problem, we split the problem into two cases;

- Immediate horizon ($T = 1$): we focus only on the current time t_c and the effects on the future are omitted.

- Finite horizon ($T > 1$): we take into account the future but unlike previously, we assume the channel realization and the location at time $t \in [t_c, t_c + T - 1]$ known in advance, i.e., when the algorithm is run at time t_c. The gain function obtained by this approach is an upper bound.

3.5.2.1 Immediate horizon: $T = 1$

In this case, the optimization problem can be entirely restated. The variables to be optimized are x_{u,t_c} which is 1 id user u is active at time t_c or 0 otherwise. The cost in bandwidth is $w^{th}_{u,t_c} x_{u,t_c}$ because we assume that if an user is active, then the scheduler provides to it the minimum bandwidth it required to do a transmission without failure. Then the contribution in the gain function

is $\alpha_u x_{u,t_c}$. Therefore the optimization problem can written as follows

$$\begin{aligned} \max_{x_{u,t_c}} \quad & \sum_{u \in U_{t_c}^{act}} \alpha_u x_{u,t_c} \\ \text{s.t.} \quad & \sum_{u \in U_{t_c}^{act}} w_{u,t_c}^{th} x_{u,t_c} \leq W \\ & x_{u,t_c} \in \{0,1\}, \quad \forall u \in U_{t_c}^{act}. \end{aligned}$$

This problem is a *Knapsack* problem, which corresponds to maximizing the total value by choosing from a set of objects a proper subset. Every object has its value but also a weight that prevents from picking all of them since the total weight of the chosen subset should not overreach the capacity level. It is a well known $\mathcal{NP}$-complete problem with various efficient algorithms for solving it.

3.5.2.2 Finite horizon: $T > 1$

As remarked previously, the original problem described by (3.17) is mixed, i.e. discrete over some variables and continuous over others. One idea is to approximate the indicator function with a continuous function[3] in order to apply the Frank-Wolfe algorithm again as in the case of statistical CSI. We do not follow this way since the number of bad local optimums grow up and also it is very dependent on the choice of the approximating function. Hereafter, we assume that for the future $T-1$ time slots, the base station knows exactly how many and where users will appear, of which class and what will be their channels. The base station thus acts as an *oracle* capable to perfectly calibrate the scheduling to future fluctuations. We obtain therefore an upper bound of the performance of our policies.

Connecting this problem with a knapsack problem is not successful. Let us assume in the time interval $[t_c, t_c+T-1]$ the oracle knows a set of $U_{t_c}^T$ users/objects appear in total. We can think of having T different knapsacks (one for each $t \in [t_c, t_c+T-1]$ and all of capacity W), which we aim to fill with users/objects from the set $U_{t_c}^T$. The goal is to maximize the overall value of the chosen objects, i.e. satisfied users. This corresponds to a "multiple knapsack problem" but with a crucial difference. In contrast to "multiple knapsack problem", the weight of each object/user fluctuates over time as a consequence of the channel variability which changes the required resources/weight. That means that every object has a different weight depending on the knapsack it will be put it. Even considering $\rho = 1$, the constant channel does not help much since for some time slots in $[t_c, t_c+T-1]$ a user can happen to be either "unborn" or "dead". In those time slots we have to assume a different weight at those time slots that will be something greater than W so as to make it impossible to fit in the knapsacks corresponding to those time slots.

Finally we address our problem using a more generic (and slower) approach after formulating it as a Integer Linear Programming (ILP) optimization one, and so we call this method *ILP oracle*. As mentioned, inside the lifespan $t \in I_{life} = [\max(t_c, t_u), \min(t_u+L_u-1, t_c+T-1)]$ of a user $u \in U_{t_c}^T$, $w_{u,t}^{th}$ are the (accurately predicted by the oracle) required bandwidth to satisfy u at time t given his channel gain $g_{u,t}$. Outside $t \in [t_c, t_c+T-1]/I_{life}$, $w_{u,t}^{th}$ is given a value greater than W so as to

[3] So, the form $\mathbb{1}\{w > w_{u,t_c}^{th}\}$ needs to be changed into continuous function for which when $w < w_{u,t_c}^{th}$ it is equal to 0 in order to avoid giving less than w_{u,t_c}^{th} resource at user u and then it goes as fast as possible to 1.

prevent any allocation. The formulation is

$$\begin{aligned}
\max_{x_{u,t}} \quad & \sum_{u\in U_{t_c}^T} \alpha_u \sum_{t=t_c}^{t_c+T-1} x_{u,t} \\
\text{s.t.} \quad & \sum_{U_{t_c}^T} w_{u,t}^{th} x_{u,t} \leq W, \quad \forall t \in [t_c, t_c{+}T{-}1] \\
& \sum_{t=t_c}^{t_c+T-1} x_{u,t} \leq 1, \quad \forall u \in U_{t_c}^T \\
& x_{u,t} \in \{0,1\}, \quad \forall t \in [t_c, t_c{+}T{-}1] \text{ and } \forall u \in U_{t_c}^T.
\end{aligned}$$

To solve this ILP optimization (which correspond to a "multiple choice knapsack") for every time step, we used the software CPLEX of IBM which relies on the Branch and Cut algorithm [43].

3.6 Deep reinforcement learning

The difficulty of finding efficient, not complex algorithms to tackle the problem motivates us to try a more versatile tool, namely deep reinforcement learning. The downside is the produced neural network model has a performance that is dependent on the training. It may take a great amount of time to converge, and eventually converge to a poorly performing model or even not converge at all. Trying to avoid those scenarios, we came up with a DRL algorithm combining several recently introduced ideas.

In standard RL setups, there is an agent who interacts with an environment. At every time step t this agent observes the current state of environment $s_t \in \mathcal{S}$ and takes an action $a_t \in \mathcal{A}$. In our case it is $s_t = \{\forall u \in U_t : D_u, L_u, \alpha_u, \mathrm{d}_u, l_{u,t}, A_{u,t}, h_{u,t}\}$ where $l_{u,t} \leq L_u$ is the number of time slots passed since appearance of u and $A_{u,t}$ is changed slightly to signify whether the user is active at the beginning (not the end) of time slot t. The action is simply the resource allocation at time t. After the action a_t the environment's state changes. The new state s_{t+1} is a realization of the random variable (RV) S_{t+1} with density $p(\bullet; s_t, a_t)$ (which for simplicity we will denote it as $p(; s_t, a_t)$) and the agent receives a reward $R(s_t, a_t, s_{t+1})$ which can be RV. This is a Markov Decision Process (MDP) [44]. The full CSI case conforms perfectly with this model and the reward is not RV but can be deterministically described as $R(s_t, a_t)$.

In the case of statistical CSI, it is much more complicated. Previously we assumed that the agent has full access to the state s_t but it does not stand anymore because its observation o_t lacks the $h_{u,t}$, so $o \subset s_t$. In that case where the observation o_t follows in general distribution $O(o_t; s_t, a_{t-1})$ the model is called Partially Observable Markov Decision Process (POMDP) [45]. In this situation, it is possible to come back to the MDP scheme, called belief MDP, by substituting the states with the "belief" b_t [46] of the value of s_t. Belief b_t preserves the Markovian property and follows $p(\bullet; o_{t-1}, a_{t-1}, b_{t-1})$. Equivalently one could not depend on the previous b_{t-1} to compute b_t but on the complete history $\{o_0, a_0, o_1, a_1, \cdots, a_{t-1}, o_{t-1}\}$. Hopefully, in our statistical CSI case, it suffices to keep only o_{t-1} together with some previous actions, which are the actions corresponding to the resources that had already been assigned to every current active user. Now we can revise the meaning of the state as $\mathfrak{s}_t = \{\forall u \in U_t : D_u, L_u, \alpha_u, \mathrm{d}_u, l_{u,t}, A_{u,t}, \overrightarrow{w}_{u,t-1}\}$[4] and keep the same formulation of the MDP. The algorithm below is described using s_t and the only

[4] Users just appearing at time t have no scheduling history, so $\overrightarrow{w}_{u,t-1}$ is omitted.

difference for the statistical case is that the reward depends on the new state, i.e. $R(\mathfrak{s}_t, a_t, \mathfrak{s}_{t+1})$, and is a RV[5].

Since the action in our setup is sharing the bandwidth in a continuous way a traditional Deep Q-learning Network (DQN) does not work. The reason is that in classic DQN, the trained Neural Network (NN) needs a number of outputs equal to the possible actions which in our case is infinite due to continuity. But even without the continuity and assuming serving every user with either a fixed amount of resources or not at all, the number of possible outputs is huge since we need one output for every different combination of the set of users chosen to be served. To resolve this problem we resort to Policy Gradient methods. A famous type of method of this category is the actor-critic where a NN, called critic, predicts how well the proposed policy provided by a second NN called actor will be. In general, the actor's policy gives a *probability distribution* over possible actions. So from a given state there are many probable actions that can be taken. If this taken action turns out to give higher (resp. lower) than the critic's expectation then the actor learns to increase (resp. decrease) the probability of taking again that action in similar situations. We choose a different type of method called "deterministic policy gradient" where the actor NN output provides *deterministically* a specific action.

3.6.1 Optimizing the Actor NN

Let us first concentrate on the actor. A deterministically parameterized agent is modeled as a Neural Network (NN) [47], named Actor NN, who aims to optimize his NN's parameters θ so that the resulting policy $\pi_\theta : \mathcal{S} \to \mathcal{A}$ maximizes an objective $J(\theta)$. Let us first define

$$Z^{\pi_\theta}(s_t, a_t) = R(s_t, a_t) + \sum_{i=1}^{\infty} \gamma^i R(S_{t+i}, \pi_\theta(S_{t+i})) \tag{3.18}$$

to be the RV representing the discounted reward accumulated when the agent starts from state s_t with action a_t and after follows the policy π_θ. Parameter $\gamma \in [0, 1]$ is the discount factor balancing the importance of future rewards. Let also the state-action value function $Q^{\pi_\theta}(s_t, a_t) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be:

$$Q^{\pi_\theta}(s_t, a_t) = \mathbb{E}[Z^{\pi_\theta}(s_t, a_t)] \tag{3.19}$$

If p_{t_0} is the density of the distribution of the initial state s_{t_0}, then we can now describe the objective function of the agent:

$$J(\theta) = \mathbb{E}_{s_{t_0} \sim p_{t_0}}[Q^{\pi_\theta}(s_{t_0}, \pi_\theta(s_{t_0}))] \tag{3.20}$$

To maximize it the gradient is needed which through the deterministic policy gradient theorem [48] can be written:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s_{t_0} \sim p_{t_0}, s \sim \rho^{\pi_\theta}_{s_{t_0}}}[\nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s, a) | a = \pi_\theta(s)] \tag{3.21}$$

with $\rho^{\pi_\theta}_{s_{t_0}}$ the discounted state (improper) distribution defined as $\rho^{\pi_\theta}_{s_{t_0}}(s) = \sum_{i=0}^{\infty} \gamma^i \mathbb{P}(s_{t+i} = s | s_{t_0}, \pi_\theta)$. In practice it is common that $\rho^{\pi_\theta}_{s_{t_0}}$ is approximated by the (proper) distribution $\varrho^{\pi_\theta}_{s_{t_0}}(s) := \sum_{i=0}^{\infty} \mathbb{P}(s_{t+i} = s | s_{t_0}, \pi_\theta)$. This gradient allows to gradually improve the Actor NN if, as assumed so far, the true function $Q^{\pi_\theta}(s, a)$ (and for every possible π_θ) is provided to the agent.

[5] It is random variable since there is the scenario of at time t a user to be at the last time slot he is eager to wait, he is served some resources but at $t + 1$ he disappears without knowing if got satisfied.

3.6.2 Optimizing the Value NN

However, it is intractable to get the true $Q^{\pi_\theta}(s,a)$ so we approximate it by building a second network, named Value NN and denoted $Q_\psi^{\pi_\theta}$ (ψ represents its NN's parameters). To train the Value NN, the Bellman equation is used which combines (3.18) and (3.19) into:

$$Q^{\pi_\theta}(s_t,a_t) = R(s_t,a_t) + \gamma \mathbb{E}_{s_{t+1}\sim p(;s_t,a_t)}[Q^{\pi_\theta}(s_{t+1},\pi_\theta(s_{t+1}))] \ ^{6} \tag{3.22}$$

The idea is in order $Q_\psi^{\pi_\theta}$ to be a good approximator of $Q^{\pi_\theta}(s,a)$ it has to satisfy (3.22). We push it by optimizing the parameters ψ to minimize the difference (named temporal difference) between the two sides of equation (3.22) when the $Q_\psi^{\pi_\theta}(s,a)$ is used. So if $d_2(x,y) = (x-y)^2$, to improve $Q_\psi^{\pi_\theta}$ we minimize the loss

$$\mathcal{L}_2(\psi) = \mathbb{E}_{s_{t_0}\sim p_{t_0}, s\sim \varrho_{s_{t_0}}^{\pi_\theta}, s'\sim p(;s,a)}[d_2(Q_\psi^{\pi_\theta}(s,a), R(s,a) + \gamma Q_\psi^{\pi_\theta}(s',\pi_\theta(s'))]. \tag{3.23}$$

The reasoning behind (3.23) is that we start from s_{t_0} and following the policy π_θ we arrive to different states s with probability $\varrho_{s_{t_0}}^{\pi_\theta}(s)$. From those states we want to know the validity to take the action a followed by actions done through π_θ in the expected discounted sum of rewards. Notice that action a is not necessarily the one proposed by π_θ since (as discussed later) we need to be able to input to the value NN different ones in order to explore the actions' space.

3.6.2.1 First trick: Target Networks

As $Q_\psi^{\pi_\theta}$ is optimized it affects both sides of the Bellman equation bringing instabilities. To limit them, a key idea was proposed in [49] which, when adopted to the deterministic policy gradient, introduces two separate NNs $Q_{\psi^-}^{\pi_{\bar{\theta}}}$ and $\pi_\theta-$, named here target Value NN and target Actor NN respectively. Those NNs either freeze the parameters of Value and Actor NN for a number of iterations and then they are "hard updated" $\psi^- \leftarrow \psi$ and $\theta^- \leftarrow \theta$, or are softly updated per iteration $\psi^- \leftarrow \rho_v^{sync}\psi + (1-\rho_v^{sync})\psi^-$ and $\theta^- \leftarrow \rho_v^{sync}\theta + (1-\rho_v^{sync})\theta^-$. In either case the target NN change slowly and are used to stabilize the right hand of bellman equation. The loss function becomes

$$\mathcal{L}_2(\psi) = \mathbb{E}_{s_{t_0}\sim p_{t_0}, s\sim \varrho_{s_{t_0}}^{\pi_\theta}, s'\sim p(;s,a)}[d_2(Q_\psi^{\pi_\theta}(s,a), R(s,a) + \gamma Q_{\psi^-}^{\pi_{\theta^-}}(s',\pi_{\theta^-}(s'))].$$

3.6.2.2 Second trick: Distributional perspective

A richer representation of the value NN to approximate is the distribution of Z^{π_θ} instead of only its mean value [50]–[52]. Therefore now instead of working with $Q_\psi^{\pi_\theta}(s,a)$ (which approximates the mean value of Z^{π_θ}), we propose to work with a new value NN, denoted by $Z_\phi^{\pi_\theta}(s,a)$ whose the outputs describe an approximation of the distribution of Z^{π_θ}. To accomplish that, we first have to revisit the Bellman equation through its distributional version

$$Z^{\pi_\theta}(s_t,a_t) \overset{D}{=} R(s_t,a_t) + \gamma Z^{\pi_\theta}(S_{t+1},\pi_\theta(S_{t+1})). \tag{3.24}$$

[6] In the statistical CSI the reward is a RV so we need to average over the rewards also and the equation becomes

$$Q^{\pi_\theta}(\mathfrak{s}_t,a_t) = \mathbb{E}_{\mathfrak{s}_{t+1}\sim p(;\mathfrak{s}_t,a_t)}[\mathbb{E}[R(\mathfrak{s}_t,a_t,\mathfrak{s}_{t+1})] + \gamma Q^{\pi_\theta}(\mathfrak{s}_{t+1},\pi_\theta(\mathfrak{s}_{t+1}))]$$

The idea is again to force the value network to satisfy Eq. (3.24).

Like in [51], [52] the value NN aims to approximate the real distribution of $Z^{\pi_\theta}(s,a)$ with one discrete RV $\mathcal{Z}_\phi^{\pi_\theta}(s,a)$ random variable taking values in $\{y_1, y_2, ..., y_{N_Q}\}$ (in strictly ascending order) with probabilities $\mathbb{P}(\mathcal{Z}_\phi^{\pi_\theta}(s,a) = y_i) = \mathrm{p}_i$. The relationship between the RV $\mathcal{Z}_\phi^{\pi_\theta}(s,a)$ and the NN $Z_\phi^{\pi_\theta}(s,a)$ is that the domain of $\mathcal{Z}_\phi^{\pi_\theta}(s,a)$ is defined as the output of $Z_\phi^{\pi_\theta}(s,a)$, i.e. $Z_\phi^{\pi_\theta}(s,a) = (y_1, y_2, ..., y_{N_Q})$. So if we well train the parameters ϕ of the NN $Z_\phi^{\pi_\theta}(s,a)$, the resulting random $\mathcal{Z}_\phi^{\pi_\theta}(s,a)$ will be a good approximation of $Z^{\pi_\theta}(s_t,a_t)$.

Let us go back to (3.24) to see how $\mathcal{Z}_\phi^{\pi_\theta}(s,a)$ can satisfy it. Instead of scalar (mean) values in the Bellman equation, distributions are compared. Therefore a distance metric between distributions is needed and in [51], [52] the 1-Wasserstein distance is used defined as $W_1(U,V) = \int_0^1 |F_U^{-1}(\omega) - F_V^{-1}(\omega)|\mathrm{d}\omega$ where U and V are RV with F_U^{-1} and F_V^{-1} being their inverse cumulative distribution function (CDF).

Incorporating also the first trick (target networks), we wish to optimize ϕ so as to minimize 1-Wasserstein distance $W_1(\mathcal{Z}_\phi^{\pi_\theta}(s_t,a_t), R(s_t,a_t) + \gamma\mathcal{Z}_{\phi^-}^{\pi_\theta}(S_{t+1}, \pi_{\theta^-}(S_{t+1})))$ which is easily shown to happen when

$$\mathbb{P}(y_i \geq R(s_t,a_t) + \gamma\mathcal{Z}_{\phi^-}^{\pi_\theta}(S_{t+1}, \pi_{\theta^-}(S_{t+1}))) = \tau_i \frac{\mathrm{p}_i}{2} + \sum_{j=1}^{i-1} \mathrm{p}_i \tag{3.25}$$

where $\tau_i := \frac{\mathrm{p}_i}{2} + \sum_{j=1}^{i-1} \mathrm{p}_i$ is obtained so that y_i is the τ_i-quantile $R(s_t,a_t) + \gamma\mathcal{Z}_{\phi^-}^{\pi_\theta}(S_{t+1}, \pi_{\theta^-}(S_{t+1}))$. This can be seen in Figure 3.1. The discrete approximation crosses the CDF of $R(s_t,a_t) +$

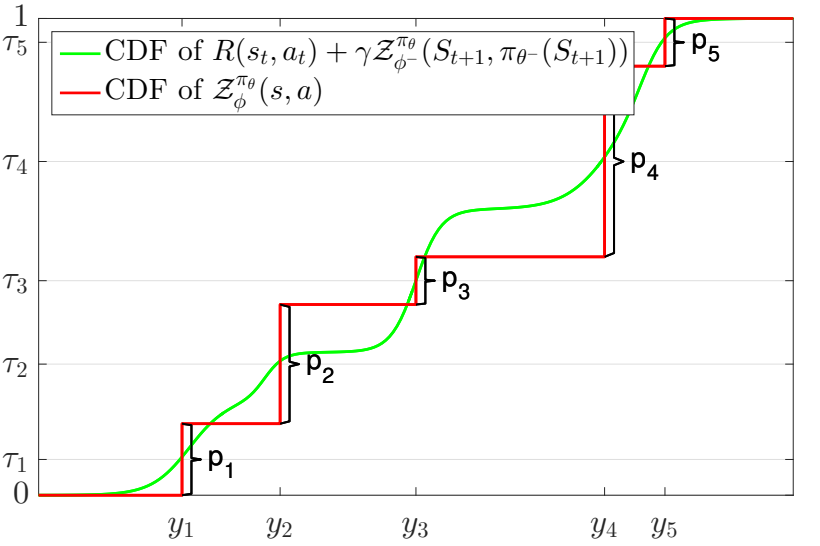

Figure 3.1: Minimizing the 1-Wasserstein distance

$\gamma\mathcal{Z}_{\phi^-}^{\pi_\theta}(S_{t+1}, \pi_{\theta^-}(S_{t+1}))$ is the middle of each vertical line (line corresponding to a jump from an discrete value to the next one). Finally it remains to optimize the parameters ϕ of $Z_\phi^{\pi_\theta}(s,a)$ so as the resulting $\mathcal{Z}_\phi^{\pi_\theta}(s,a)$ satisfies (3.25). This can be done by quantile regression [53]; more

precisely, if $d_1(y,z;\tau)=(\tau-\mathbb{1}\{z<y\})(z-y)$ then we just need to minimize the loss function[7]:

$$\mathcal{L}_1(\phi)=\mathbb{E}_{s_{t_0}\sim p_{t_0},s\sim\varrho_{s_{t_0}}^{\pi_\theta},s'\sim p_t(s,a),z\sim R(s,a)+\gamma\mathcal{Z}_{\phi^-}^{\pi_{\theta^-}}(s',\pi_{\theta^-}(s'))}[\sum_{i=1}^{N_Q}d_1(y_i,z;\tau_i)] \tag{3.26}$$

where the dependency in ϕ is hidden in the output $\{y_i\}_{i=1,\cdots,N_Q}$.

3.6.2.3 Third trick: N-steps update

A common trick is the value NN to be updated not only using the immediate reward but accumulated rewards from N_{st} steps [54], i.e. using:

$$Z^{\pi_\theta}(s_t,a_t)\overset{D}{=}\left[R(s_t,a_t)+\sum_{i=1}^{N_{st}-1}R(S_{t+i},\pi_\theta(S_{t+i}))\right]+\gamma^{N_{st}}Z^{\pi_\theta}(S_{t+N_{st}},\pi_\theta(S_{t+N_{st}})).$$

The motivation is the value NN to depend less on itself for the update since $N_{st}-1$ additional terms come straight from the environment rewards. This also enables the value NN to faster realize the contribution of some delayed rewards[55]. In our case the environment exhibits very big variance. From a specific state, one action can lead to a large variety of new states which are significantly different to each other (due to the new channel realizations and the traffic which can lead to the appearance of some new users with different requirements). Therefore every new term is also a source of large variance. On top of that on the statistical CSI case even the reward from a specific state after one action is a RV and can vary. Hence if N_{st} grows then adding those new terms together increases even more the variance resulting to convergence difficulties. Therefore we consider low values for the N_{st}: $N_{st}=1$ for the statistical CSI case, and $N_{st}=2$ for the full CSI case. Up to this point, the algorithm resembles to D4PG proposed in [56].

3.6.2.4 Fourth trick: Dueling

In [57], the "dueling network" architecture was proposed. This structure is similar to the red part of the Value NN in Fig. 3.2 where three outputs (coming from three subNN) are considered, namely, $Z_\phi^{\pi_\theta,M}(s,a)$, $Z_\phi^{\pi_\theta,S}(s,a)$ and $Z_\phi^{\pi_\theta,D}(s,a)$.

To explain its functionality firstly we concentrate on the setup of [57] where a traditional DQN (deep Q network) is used (i.e., no Actor NN or Deterministic Policy gradient, and the action space $\mathcal{A}$ is of finite cardinality). In that DQN setting, a NN $\mathcal{Q}_x^D(s_{in})$ depending on parameters x takes an input state s_{in} and provides $|\mathcal{A}|$ outputs (where $\mathcal{A}$ is the action space) where one output corresponds to a good approximation of the state-action values $Q(s_{in},a)$ for one possible action $a\in\mathcal{A}$. Consequently, we have a vector $\mathcal{Q}_x^D(s_{in})\in\mathbb{R}^{|\mathcal{A}|}$ where a component of $\mathcal{Q}_x^D(s_{in})$ corresponding to specific action a and is denoted by $\mathcal{Q}_x^D(s_{in})_a\in\mathbb{R}$ (we do the same for $\mathcal{Q}_x^S$). The $\mathcal{Q}_x^D(s_{in})$ is

[7] We illustrate the role of this specific loss function with an example. Suppose an RV Z following a probability density function $f_Z(z)$, then minimizing $J(y)=\mathbb{E}_{z\sim Z}[d_1(y,z;\tau)]$ equals to:

$$\min_\phi \quad \mathbb{E}_{z\sim Z}[\tau(z-y)]-\int_{-\infty}^{y}(z-y)f_Z(z)dz$$
$$\min_\phi \quad \tau\mathbb{E}_{z\sim Z}[z]-\tau y-\int_{-\infty}^{y}zf_Z(z)dz+y\int_{-\infty}^{y}f_Z(z)dz$$

Clearly, the derivative is $\frac{dJ}{dy}=-\tau-yf_Z(y)+\mathbb{P}(z\le y)+yf_Z(y)=\mathbb{P}(z\le y)-\tau$, hence $J(y)$ a unimodal with minimum when $\mathbb{P}(z\le y^\star)=\tau$, i.e. $y^\star$ the τ-percentile of the RV Z. Likewise for $d_2(y,z)=(z-y)^2$ we get the minimum when $y^\star=\mathbb{E}[z]$ which explains (3.23)

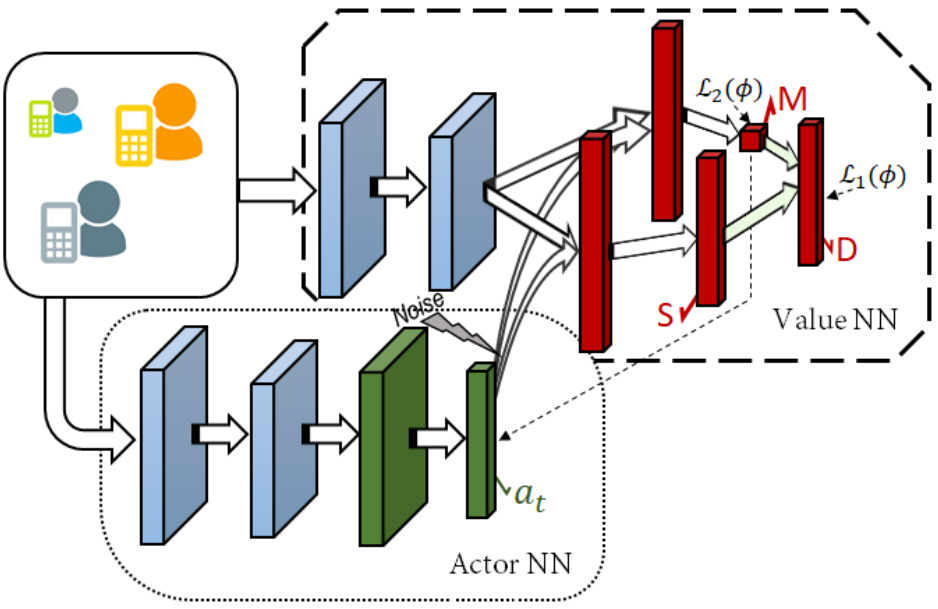

Figure 3.2: The architecture of the Neural Networks

a combination of $\mathcal{Q}_x^S(s_{in}) \in \mathbb{R}^{|\mathcal{A}|}$ and $\mathcal{Q}_x^M(s_{in}) \in \mathbb{R}$ and its components are computed in the following way:

$$\mathcal{Q}_x^D(s_{in})_a = \mathcal{Q}_x^M(s_{in}) + \Big(\mathcal{Q}_x^S(s_{in})_a - \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} \mathcal{Q}_x^S(s_{in})_a\Big), \forall a \in \mathcal{A} \tag{3.27}$$

This pushes $\mathcal{Q}_x^M(s_{in}) = \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} \mathcal{Q}_x^D(s_{in})_a \approx \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} Q(s_{in}, a)$ in a natural way (this can be seen by averaging (3.27) over every $a \in \mathcal{A}$). Therefore $\mathcal{Q}^M(s_{in})$ represents the mean value of the state s_{in}.

Coming back to our case, we use this idea in a different way. Instead of averaging for a specific state over the state-action value the different possible actions we have, we propose to average for a specific state-action over the quantiles of the (approximated) distribution of the discounted accumulated reward. The value NN $Z_\phi^{\pi_\theta}(s,a)$ outputs from $Z_\phi^{\pi_\theta,D}(s,a) = (y_1, ..., y_{N_Q}) \in \mathbb{R}^{N_Q}$ characterizing the RV $\mathcal{Z}^{\pi_\theta}(s,a)$ approximating the distribution of $Z^{\pi_\theta}(s,a)$ exactly as described previously and the parameters ϕ are optimized so as to minimize the loss $\mathcal{L}_1^D(\phi)$ as described by (3.26). Now we design the dueling architecture composed by $Z_\phi^{\pi_\theta,M}(s,a) \in \mathbb{R}$ and $Z_\phi^{\pi_\theta,S}(s,a) \in \mathbb{R}^{N_Q}$. Setting $\tau_i = \frac{2i-1}{2N_Q} \Rightarrow \mathrm{p}_i = \frac{1}{N_Q}, \forall i$ gives $Z_\phi^{\pi_\theta,M}(s,a)$ meaningfulness since it averages over the output $Z_\phi^{\pi_\theta,D}(s,a)$ equal to

$$\frac{\sum_{i=1}^{N_Q} y_i}{N_Q} = \sum_{i=1}^{N_Q} \mathrm{p}_i y_i = \mathbb{E}[\mathcal{Z}_\phi^{\pi_\theta}(s,a)] \approx \mathbb{E}[Z^{\pi_\theta}(s,a)] = Q^{\pi_\theta}(s,a). \tag{3.28}$$

From (3.28) we realize that an approximation of the objective which the Actor NN (3.20) seeks to maximize is provided through $Z_\phi^{\pi_\theta,M}(s, \pi_\theta(s))$ and therefore this output is used to compute the gradients in (3.21).

The dueling trick turns out to be helpful. We assume the reason is that it splits the work of the value NN into two sub-problems. One finding the center mean of the distribution of the approximated sum of discounted returns and another the shape of this distribution which happens to be very similar for different state-action input. Hence, this "shape" NN does not have to change much even if the mean changes.

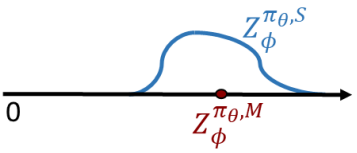

Figure 3.3: Dueling Explanation

So far, we implicitly forced $Z_\phi^{\pi_\theta,M}(s,a)$ to be an approximating mean of $Z^{\pi_\theta,S}(s,a)$ through the dueling architecture and optimizing the output $Z_\phi^{\pi_\theta,D}$ by minimizing $\mathcal{L}_1^D(\phi)$ as in (3.26). Obviously, it is possible to train $Z_\phi^{\pi_\theta,M}(s,a)$ directly by skiping the dueling and distributional approach, ie, by minimizing the quadratic loss $\mathcal{L}_2^M(\phi)$ as in (3.23). But, we can also think to do a trade off between both losses:

$$\mathcal{L}(\phi) = \rho^{losses}\mathcal{L}_1^D(\phi) + (1-\rho^{losses})\mathcal{L}_2^M(\phi),\ \rho^{losses} \in [0,1]$$

When $\rho^{losses} = 0$, the distributional approach is not applied and it leads to the worst results. When ρ^{losses} does not vanish, we have observed similar results to the case ρ^{losses}=1 but occasionally with faster convergence.

3.7 Common Features

In order to reduce the number of parameters and the sensitivity to the environment variation, we propose to "share" the first part of the value NN with the actor NN. Specifically as shown in Figure 3.2 we assume the value NN to consist of two sub-networks, the blue one with parameters ϕ_1 that we call feature sub-network[8] and the red one with ϕ_2. The total parameters are $\phi = [\phi_1, \phi_2]$. Respectively for the actor NN we have $\theta = [\theta_1, \theta_2]$. The idea is the feature sub-network of each to have the same structure and to push for $\phi_1 = \theta_1$. The feature sub-network converts the input state to a set of features that should be useful to both the Actor NN and Value NN. Those features are useful to both since if these features are enough for value NN to output a good approximation of state-value function $Q^{\pi_\theta}(s,a)$ then they should be adequate for the Actor NN to maximize this same function. The way we push for $\phi_1 = \theta_1$ is by training the complete value NN independently and we "synchronize" slowly the feature sub-network of the Actor NN by $\theta_1 \leftarrow \rho_{ft}^{sync}\theta_1 + (1-\rho_{ft}^{sync})\phi_1$. The actor does not use the gradients given in (3.21) to update its feature sub-network, but it is improved slowly as the feature sub-network of the value NN is improving. Important detail is ρ_{ft}^{sync} to be smaller than the learning rate of the actor so as to have "enough time" to adapt to the changes of his features imposed by the value NN.

3.8 Architecture, Multi-Agent, and Scalability

Assuming that the input to the actor NN is $s_t = (x_{t,1}, \ldots, x_{t,K})$ where $x_{t,i}$ is the input information related to the i^{th} user at time step t and the actor NN provides the action $a_t = (a_{t,1}, \ldots, a_{t,K})$, then we want any permutation of $x_{t,i}$ to lead to the same permutation of $a_{t,i}$. To force this, we need $a_{t,i} = f_a(F_{self}(x_{t,i}), F_{others}(\{x_{t,j} : j \neq i\}))$, $\forall i$ where $F_{others}(\cdot)$ is a vector valued function invariant to permutation of the input (commutative). Respectively, assuming the value NN takes input

[8] It is implemented with convolutional layers performing the same manipulations on the input information corresponding to each user without mixing them.

$(v_{t,1}, \cdots, v_{t,K})$ with $v_{t,i} = (x_{t,i}, a_{t,i})$ then if we denote the output as a function $f_v(v_{t,1}, \cdots, v_{t,K})$, $f_v(\cdot)$ has again to be commutative. Those remarks not only allow us to build networks that scale insignificantly as the number of users (and so the input) K increases, facilitating therefore the training, but also they provide interesting links to some ideas related multi-agent settings.

Our problem can be thought as a centralized fully co-operative multi-agent task. Specifically at each time slot one can think of K agents, each undertaking the responsibility to satisfy one user. The agents do not act selfishly to satisfy their own user but a global goal. Finally all agents have access to the information related to all the users. We use a Deep Deterministic Policy gradient approach like in [58] but here we prefer to use a single centralized critic for all agents. Also, previously we showed that each agent i takes a decision of the form $a_{t,i} = f_a(F_{self}(x_{t,i}), F_{others}(\{x_{t,j} : j \neq i\}))$, $\forall i$ which means (contrary again to [58]) that all agents actually use the same function to take their decisions. The function is implemented using a NN structure, which means that all the agents are sharing parameters, like what the authors of [59] do in a similar fully co-operative and partially observable setting.

Finally, there is some connection to [60] where again there is a co-operative multi agent setting and a centralized critic is used. Trying instead of using a global reward function to train each agent, they personalize it for every agent by introducing a counterfactual baseline. This baseline encapsulates the advantage a certain action of an agent brings to the system while keeping the actions of the rest of the agents the same. The idea of comparing one agent to the rest is also somehow apparent in our case but through the form the agents policy function has, i.e. $a_{t,i} = f_a(F_{self}(x_{t,i}), F_{others}(\{x_{t,j} : j \neq i\}))$.

In the full CSI since we can easily compute how much bandwidth $w_{u,t}^{th}$ each user needs at time t, for the scheduling we only need know which ones to prioritize. So the action a_t gives a sorting of the users and we satisfy accordingly as many as the total resources allow. The function $f_a(\cdot)$ gives a value to each user and the most valuable ones will be the first ones to be considered to be served and if the available resource allows it they will be served. We pick the $f_a(\cdot)$ to have a softmax form, specifically $f_a = \frac{e^{F_{self}(x_{t,i})}}{e^{F_{self}(x_{t,i})} + \sum_{j \neq i} e^{F_{self}(x_{t,j})}}$. For the statistical CSI the amount of resources each user needs cannot be straightforwardly estimated so each agent on top of the value of a user estimate also the (minimum) bandwidth it needs.

Finally to create a value NN that is commutative to its input and disassociate its output from the index identity of each user, i.e. by looking the output of the value of NN to not be able to infer any knowledge on who is the i-th user, we sort the users according to their value given by the actor. Therefore the value NN will get in his i-th input the information associated to the user who has the i-th bigger value (and since the values of each user is disassociated to their index identity we succeed to attain the commutative property). Now for example the value NN knows that at its first input will always have to process the information related to the most valuable user. We point out that the sort is by default a commutative function and that a linear transformation like $y = Ax$ can only be if all the entries of matrix A are equal to each other so a traditional feed forward neural network structure has troubles approximating a commutative behaviour.

3.9 Exploration issue

It is important that the Value NN not only correctly predicts $Z^{\pi_\theta}(s, a = \pi_\theta(s))$ but also approximates as precisely as possible the state-action function when "neighboring" actions are taken, i.e., $Q^{\pi_\theta}(s, a \in B_\epsilon(\pi_\theta(s)))$ (where $B_\epsilon(x)$ is a ball of radius ϵ around center x). This enables the

actor NN to correctly optimize its policy and improve gradually its actions. Therefore the Value NN has to be fed with not only the actions coming from the actor's policy $\pi_\theta(s)$ but also with randomly perturbed version of them so as to explore the neighborhood.

As previously stated, our environment exhibits randomness with large variance. For the same (s_t, a_t) (state-action) it can "easily" happen, especially in the statistical CSI case, the accumulated reward $Z^{\pi_\theta}(s_t, a_t)$ in one realization to be high and in another low, which means in the first time the actor will try inn state s_t to do again the action a_t and in the second instance to avoid it. Therefore the difficulty to estimate $\mathbb{E}[Z^{\pi_\theta}(s_t, a_t)]$ does an unavoidable exploration. Also due to the common features there can be an exterior influence on the policy of the actor leading to possibly changing some actions and therefore to some exploration. Nonetheless an additional perturbation is beneficial especially for the full CSI case.

The perturbation on an action $a_t \in \mathcal{A}$ should be done in a way that the final action a'_t is still a valid action, i.e. $a'_t \in \mathcal{A}$. In our case the action a_t is putting a value to each user using a softmax which by definition satisfy a simplex constraint. The so-called Dirichlet distribution obeys also these simplex constraints. Therefore, a straightforward way to disturb some actions is: $a'_t = (1-\rho^{noise})a_t + W\rho^{noise}U, \rho^{noise} \in [0,1]$, where $U \in \mathbb{R}^{|U_t^{act}|}$ a Dirichlet-distributed RV. For the statistical CSI the actor also output the bandwidth so we perturb this action by adding Gaussian noise $\mathcal{N}(0, \sigma^2_{bw})$.

3.10 Simulations

In our simulation We consider users located at a distance from the base station varying from $\mathrm{d}_{min} = 0.05$ to $\mathrm{d}_{max} = 1$ kilometers. The distance dependent path loss is set to be $120.9 + 37.6\log_{10}(\mathrm{d})$ in dB which is compliant to LTE standard [61] and in our setting it translates to the constant loss component $C_{pl} = 10^{-12.09}$ and path loss exponent $n_{pl} = 3.76$. The AWGN power is $\sigma^2_N = -149$dBm/Hz. The characteristic of the classes the users can belong to are summed up at the table 3.1. We examine first the full CSI case where we set the maximum number of users to be equal to $K = 100$ and then the statistical CSI case, for which we consider $K = 25$. One can estimate that in a time slot the average number of "real" users, i.e., not belonging to the null class, is equal to $100\frac{0.2\cdot 2+03\cdot 10}{0.2\cdot 2+0.3\cdot 10+0.5\cdot 1} \approx 87$.

	D_c	L_c	α_c	p_c
class 1	256 Bytes	2	1	0.2
class 2	2048 Bytes	10	1	0.3
null class	0	1	0	0.5

Table 3.1: Classes description

Obviously, with abundance of resources it is easy to always satisfy all users, rendering impossible to distinguish a good from a bad policy. We set the energy per symbol to be equal to $P = 10^{-6}\mathrm{Joule} = 1\mu\mathrm{J}$. So for example for the full CSI where $W = 2\mathrm{MHz}$ is used, it corresponds to power $P = 2\mathrm{Watt}$. The value of $W = 2\mathrm{MHz}$ is chosen so the maximum performance of the upper bound retrieved by the *ILP oracle* method to be equal to satisfying approximately 99% of the users. It can be easily confirmed from Figure 3.4 that this maximum performance is achieved when there is no time correlation between channel realizations, i.e., $\rho = 0$.

We proceed to a short description of the model and the hyperparameters. The function $F_{self}(\cdot)$ consists of two parts and the first one belongs to the feature NN. This part starts with a batch normalization [62] since the magnitude of the values of the different components of the input vary significantly. Then it is followed by two hidden layers with hyperbolic tangent function as activation function and 50 neurons each. The next part is another two hidden layers with 50 and 25 neurons respectively with both tanh as activation function and the output in the of size 1 in the full CSI or of size 2 in the statistical CSI (where both the value and resource each user requires is the output). The value NN shares the same first part for the feature NN and the second part has on the distributional side two hidden layers of 50 neurons each using a rectifier linear unit (ReLU) and the output is only a linear combination of 9 outputs (representing 9 quantiles). The other side is identical but outputs only one value (representing the mean). The optimizer used was RMSProp [63] with learning rate 10^{-4} and the gamma parameter equal to $\gamma = 0.6$. The $\rho^{losses} = 0.9$ and $\rho_{ft}^{sync} = 10^{-3}$. We perturbate 30 percent of the samples with a Dirichlet distribution with all of the concentration parameters equal to 1 and $\rho_{noise} = 0.1$ and as the performance improves we reduce to 5 percent of the samples to perturbate.

In Figure 3.4 we plot the probability of a user decoding its packet successfully during the requested latency. As the channel time correlation ρ increases each user experiences less time diversity throughout its lifespan resulting to a decrease of its likelihood of receiving successfully its packet. On the extreme case of $\rho = 1$, if a user experiences a very bad channel realization, then because it remains constant there is no chance of getting a better channel and so get satisfied throughout its lifespan.

Obviously when using an oracle that knows the channel values of the future $T - 1$ steps and uses that unrealistic information optimally, then we obtain an upper bound of the performance. As T increases there are diminishing performance gains. In our setup we cannot see any significant additional gain for values $T > 6$ so we keep $T = 6$. Interestingly it means that there is negligible impact of further in the future time steps on the present scheduling decision. This agrees on our choice of a small value of γ for the DRL.

In both Figures 3.4 and 3.5 we see the DRL algorithm to perform convincingly better than the knapsack benchmark. We remind that the knapsack benchmark is optimal when we want to maximize myopically only the present objective, neglecting any effect that the scheduling decision might have on the future. Contrary to the DRL method, it cannot take into consideration that a user might reaches his latency deadline so it needs to be prioritized. Also it cannot differentiate when currently a user u requires small w_{u,t_c}^{th} if the chance to be satisfied is high because, even though this user requires a lot of data D_u and is far from the base station, its channel gain $|h_{u,t}|^2$ is surprisingly large or when it can be postponed for a next round since the reason of the good w_{u,t_c}^{th} is due to the small distance d_u from the base station and the few data D_u the user requires. So in this case, most likely even in the future it is likely $w_{u,t}^{th}, t > t_c$ to be low. Overall we can see a 4% increase of satisfied users given a specific amount of resources. From another aspect given a required 96% success probability, Figure 3.5 shows that the DRL method saves 13% bandwidth resources. We point out that since we keep constant the energy per symbol the 13% bandwidth saving comes with an 13% energy saving also.

Next, we turn to the case of statistical CSI. Since only statistical properties of the channel are now available, the scheduler needs extra resources to compensate the lack of knowledge to successfully serve a smaller number of users. We set $K = 25$ and for Figure 3.6 increase to $W = 4$MHz. Looking from Figures 3.6 and 3.7 we see an advantage in performance for employing

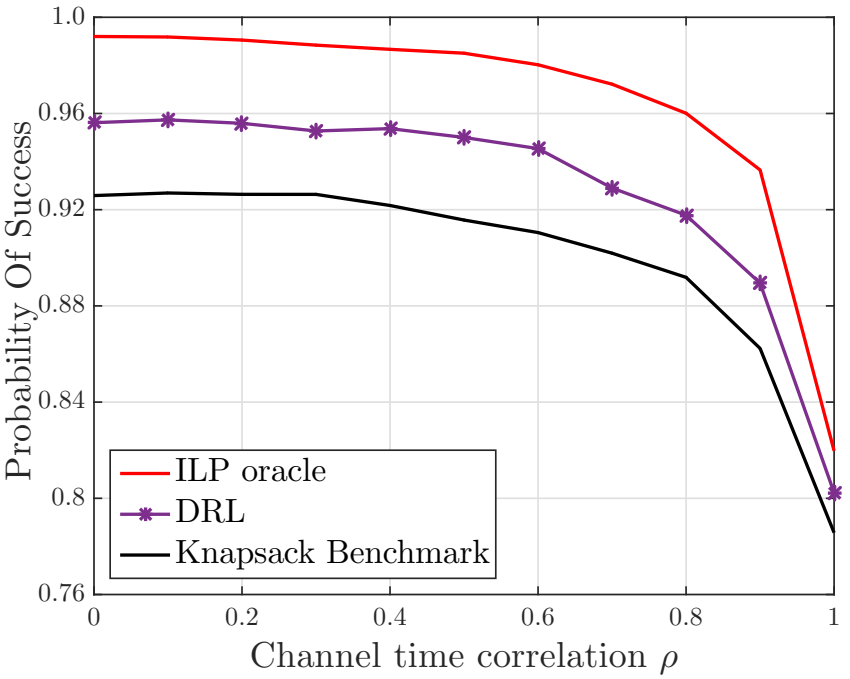

Figure 3.4: Probability of successfully satisfying a user versus the correlation factor ρ, Number of initialization for the Frank-Wolfe $N_{init} = 3$

a DRL scheduler but not a clear one.

As discussed in the section where Frank-Wolfe approach was presented, using the exact time correlation parameter ρ to perform the optimization is too complicate so we restricted to using the expressions retrieved by assuming either i.i.d. channel fading ($\rho = 0$) or constant ($\rho = 1$) even if the exact channel correlation has an intermediate value $\rho \in (0, 1)$. In Figure 3.6 we see that using the expression of either $\rho = 0$ or $\rho = 1$ does not bring much difference so we can reasonably assume that even using expressions with the exact value of ρ wouldn't change much the performance of the Frank-Wolfe algorithm.

We also want to test the impact of the number of future time steps we take into consideration for the Frank-Wolfe algorithm on the performance. From Figure 3.6 we see that to depend just on the present maximization of the objective, i.e., taking $T = 1$, is a bad strategy, but increasing a lot the time horizon and trying to form a scheduling plan including time steps much further on the future is not beneficial. This is not beneficial not only because the complexity of computing the objective function and its derivatives grows exponentially but also the performance does not necessarily improves. This consents with the choice of the γ parameter to be that low ($\gamma = 0.6$).

Finally since the Frank-Wolfe converges to a local optimum, a simple way to secure a good performance is to repeat the process multiple times starting from different random initialization points (in our simulation realized by a uniform distribution) and pick the best local optimum. Obviously, as confirmed in figure 3.7, increasing the number N_{init} of random initialization results to a bigger set of local optimum from which we choose the best one so the performance increases. Fortunately, increasing N_{init} gives diminishing returns and there is no need to search for many local optimums.

A natural question is the following: if it does not matter much that the Frank-Wolfe algorithm retrieves local optimums or that it takes a small future time horizon into consideration or that it

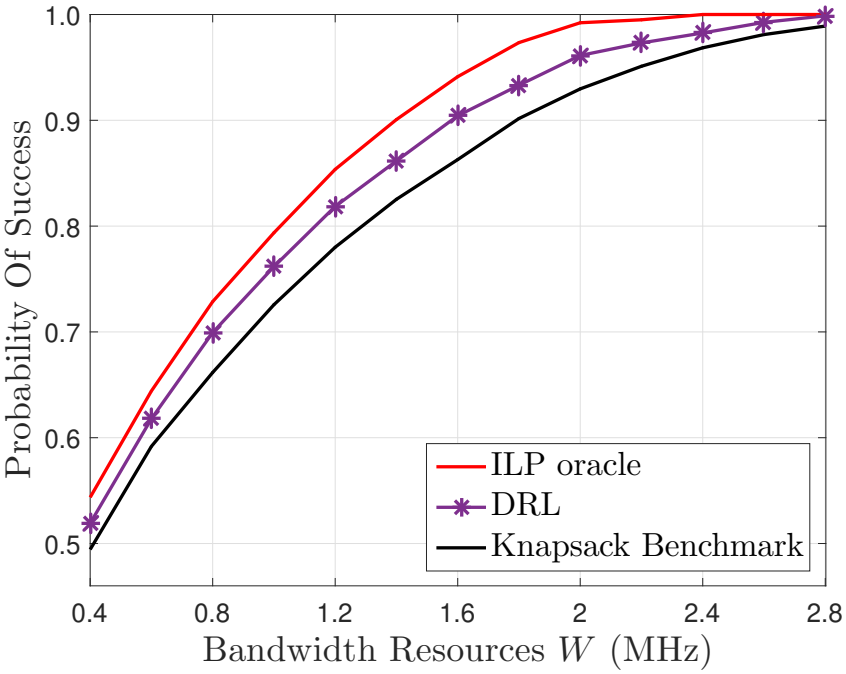

Figure 3.5: Probability of successfully satisfying a user versus the bandwidth W, Channel time correlation $\rho = 0$

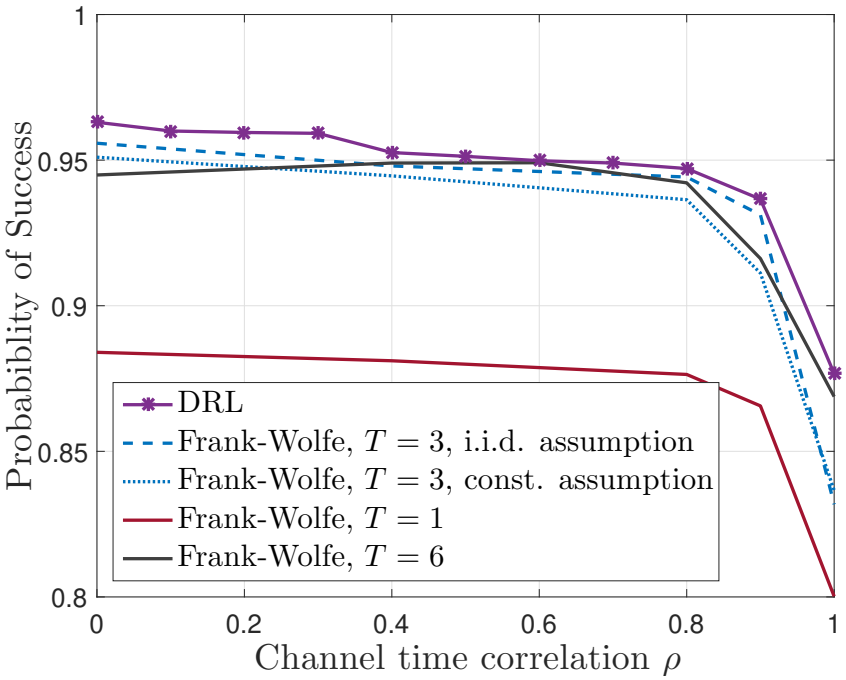

Figure 3.6: Probability of successfully satisfying a user versus the correlation factor ρ. Number of initialization for the Frank-Wolfe $N_{init} = 3$

uses in its expression an approximation of ρ to either equal to one or zero then, then how can it have a marginally worse performance than a DRL method? The last drawback of Frank-Wolfe that might contribute to the inferior performance, is that it does not consider the future users that

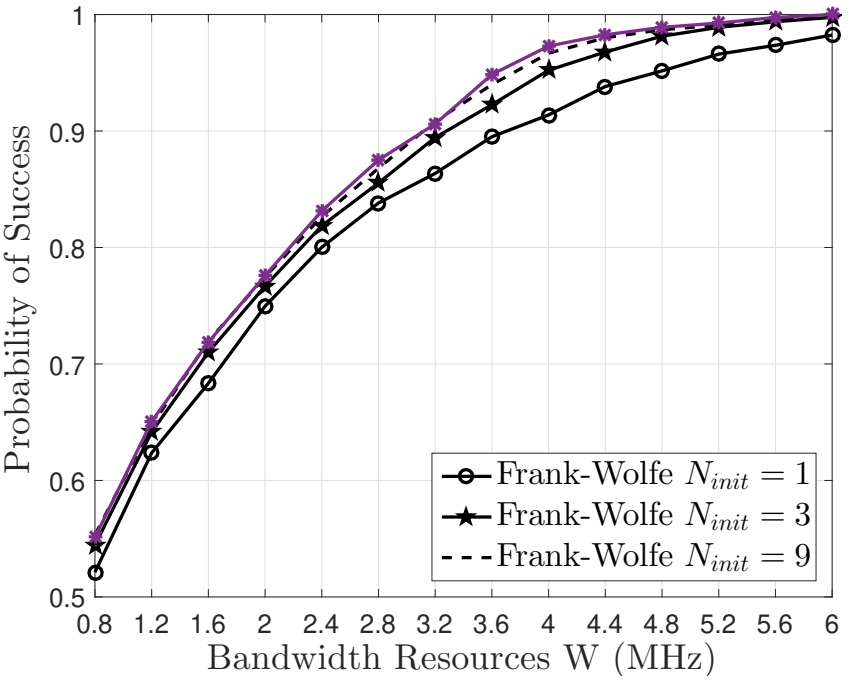

Figure 3.7: Probability of successfully satisfying a user versus the bandwidth W. Frank-Wolfe time horizon $T = 3$, channel time correlation $\rho = 0$

may appear in a way that it will plan differently according to what their characteristics will be but it treats them by averaging all the possible scenarios and we believe this is the reason it slightly falls back to DRL in performance.

Of course we should not forget the fact that once a DRL model is trained, it takes no time to run as compared to the huge complexity of the Frank-Wolfe approach. Still the DRL approach has the problem of training which can take hours and the performance depends that the environment does not change from training to testing. So the question, which is the topic of future investigation, is if DRL is employed and some properties of the environment change, how fast would it be able to retrieve a good performance.

3.11 Conclusion

In this chapter we investigated the problem of centralized scheduling of bandwidth resources under multi-class traffic in terms of data requirements and latency constraints. We distinguished between two different cases depending on the information the scheduler has regarding the channel quality of the users: one of full CSI information and one of statistical only. We compared state-of-the-art combinatorial, integer linear programming programming and optimization algorithm with one based on deep reinforcement learning. The deep reinforcement learning method we developed manages to train a neural network model outperforming in the full CSI case a knapsack algorithm that optimally maximizes a myopic version of the objective. Using an oracle based solution we can get an tight upper bound of the performance and we verify that the neural network model is not far from it. Passing to the statistical CSI case the DRL method was compared to a Frank-Wolfe approach, which managed to marginally outperform, providing close to optimal

performance.

Conclusions and Perspectives

In this thesis we focused on communication over wireless channels under strict latency constraints. We investigated how to efficiently apply retransmission protocols and how to optimally distribute the resources among users on each round. We investigated the problem from two different perspectives in terms of network layers:

- The physical layer where we focused on a single user where a packet of a specific size has to be delivered with extremely reliability and within stringent latency constraints. We proposed a dynamic programming algorithm that splits the available time interval into smaller so as to allow retransmissions and distributes the power properly so as to optimize the IR-HARQ mechanism in terms of either energy consumption or throughput.

- The media access control where we take into consideration multi-user scenarios where each user can have its own service requirements in terms of data received and latency. We proposed a deep reinforcement algorithm that can train a neural network, which can outperform some traditional combinatorial or optimization approaches.

In the first as well as the second chapter we focused on the physical layer for a point-to-point communication where a fixed number of information bits needs to be transmitted under requirements of low latency and high reliability. The goal was to configure an HARQ scheme in an optimal way and assess its benefits. The low latency constraint enforces a limited number of symbols to be transmitted which compelled us to use finite blocklength information theoretic results to describe the optimization problem we cast. Unfortunately, the analytical expressions are more involved than in the widely used infinite blocklength (Shannon) framework, which consequently makes the problem harder to solve analytically. We managed to simplify it after mathematical manipulations and then illustrate an algorithm based on dynamic programming that solves optimally this complicate non-convex optimization problem.

Although we got useful intuitions and an algorithm to properly optimize an HARQ scheme and attain optimal energy or throughput in chapter 1, the problem that was tackled was restricted to additive white Gaussian noise channels. In chapter 2, we extended our scope considering a channel model where the signal is also distorted by Rician fading. We investigated the two distinct sides of Channel State Information where, on the one side, the transmitter knows exactly the channel fading coefficient (which can be estimated by the use of pilots), and on the other only statistical properties of the channel are available. Before proceeding on the evaluation of the IR-HARQ schemes we showed a way of finding the feasibility region. Interestingly, we found that even a reasonable scheme which efficiently exploits the full CSI has a smaller feasibility region than the one with statistical CSI, and that without even accounting for the need of sending pilots to get a full CSI. Therefore we questioned the robustness of full CSI schemes in occasions with

ultra reliable and low latency communication requirements. In terms of performance however the availability of full CSI is shown to yield huge energy and throughput benefits. We finally argued that pursuing solely energy minimization or throughput maximization can be detrimental for the throughput or energy respectively, hence, any optimization must take both of those dimensions into account.

We moved to the media access control layer in chapter 3 where we looked to a multi user scenario. We considered, as always in this thesis, the setup where users can only be satisfied if they get their packet within a strict time interval but retransmissions are allowed inside it. Following the paradigm of 5G, each user belongs into classes (in 5G those could represent eMBB, mMTC, and URLLC scenarios) which define their needs and therefore their requirements. The problem was once again how to distribute the resources but now not only over time but over the different users. We insisted again on the necessity to investigate the two opposite side of CSI, so we built benchmarks for the full CSI case and for the statistical CSI. Against those we built a reinforcement learning algorithm that applies to both CSI cases and manages to train a model that from our simulation results we verify that it outperforms the benchmarks.

Perspectives

Following the line of the work done in chapter 2 in the future we aim to consider the case of imperfect CSI. Increasing the number of pilots renders a more reliable CSI but, on the other hand, it pays a toll on the latency. We would like to test how many pilot symbols are needed in order the transmitter to get reliable enough information about the channel quality without compromising significantly the latency.

In chapter 3 there are multiple directions we aim to concentrate. The scheduling algorithm we proposed was targeting at the efficient distribution of the bandwidth across multiple users and over time. The first step is to add the possibility of power adaptation and serve the users with a personalized power consumption. Moreover, again we can consider the case of imperfect CSI. We would like to insist that under those more realistic but also more complicated conditions it is hard to find benchmarks that will have any optimality guarantee. On the other hand the Reinforcement learning will be able to train a model but whose performance has to be tested. Additional aspects are to consider a multi cell environment where users from different cells may be served under the same frequency band and so interfere each other. In those scenarios maybe an intercell communication can change the scheduling decision and mitigate or avoid interference.

Another two directions that we would like to investigate are the robustness of our proposed deep reinforcement learning algorithm and the possibility of using recurrent neural networks. To check the robustness we are planning after training a model to change the characteristic of the environment. Specifically we would like to see by changing the traffic and/or the channel model how much will deteriorate the performance and more importantly how long will it take to be trained and reach again maximum performance. Finally in multi-agent collaborative tasks it has been shown that a recurrent NN structure can be beneficial [59], [60], [64] so it is interesting to see whether this will also be beneficial in our setup.

Appendix A

A.1 Proof of Lemma 1

Let $\vec{P}^{\star}_M = (\vec{P}^{\star}_{M-1}, P^{\star}_M)$. If $\varepsilon_{M-1} \leq \varepsilon_M$ at $(\vec{n}^{\star}_M, \vec{P}^{\star}_M)$, then $(\vec{n}^{\star}_M, \vec{P}'_M)$ with $\vec{P}'_M = (\vec{P}^{\star}_{M-1}, 0)$ offers a lower consumed average energy since the last term in the sum of the objective function (1.2) can be removed while the other terms remain identical. This leads to a contradiction preventing $\varepsilon_{M-1} \leq \varepsilon_M$ at the optimal point.

A.2 Proof of Lemma 2

Let us denote by $\partial\varepsilon_M/\partial P$ the derivative function of $P \mapsto \varepsilon_M(\vec{n}^{\dagger}_M, \vec{P}^{\dagger}_{M-1}, P)$. We will prove that $\partial\varepsilon_M/\partial P_{|P=P^{\dagger}_M} < 0$. By change of variables $y = 1/(P+1)^2$ and setting $y^{\dagger} = 1/(P^{\dagger}_M + 1)^2$, we show that

$$\begin{aligned} \frac{\partial\varepsilon_M}{\partial P} < 0 \text{ at } P = P^{\dagger}_M \quad &\Leftrightarrow \quad \frac{\partial\varepsilon_M}{\partial y} > 0 \text{ at } y = y^{\dagger} \\ &\Leftrightarrow \quad h(y) > 0 \text{ at } y = y^{\dagger} \end{aligned} \tag{A.1}$$

where

$$h(y) = k_2 - yk_1 + n_M(1 - y + y\ln(y)/2), \quad \text{with}$$

$$k_1 = \sum_{i=1}^{M-1} n_i \ln(1+P_i) - B\ln 2 \quad \text{and} \quad k_2 = \sum_{i=1}^{M-1} n_i \left(1 - \frac{1}{(1+P_i)^2}\right).$$

It is easily proven $h(y)$ to be a monotonically decreasing function. If $h(1) \geq 0$, then (A.1) is straightforwardly satisfied. Now, if $h(1) < 0$, then it exists $y_0 \in (0,1)$ such that $h(y_0) = 0$. So for $y \in [y_0, 1]$, we get $h(y) \leq 0$, which implies that ε_M is decreasing with respect to y. As a consequence, for $y \in [y_0, 1]$ and so the corresponding $P(y)$, we have $\varepsilon_M(\vec{n}^{\dagger}_M, \vec{P}^{\dagger}_{M-1}, P(y)) \geq \varepsilon_M(\vec{n}^{\dagger}_M, \vec{P}^{\dagger}_{M-1}, 0) = \varepsilon_{M-1}(\vec{n}^{\dagger}_{M-1}, \vec{P}^{\dagger}_{M-1})$ which prevents to have $P(y) = P^{\dagger}_M$ according to the assumption $\varepsilon_{M-1} \geq \varepsilon_M$ on the analyzed point. Consequently, $y^{\dagger}$ does not belong to $[y_0, 1]$, and belongs to $(0, y_0)$ where (A.1) holds.

A.3 Proof of Lemma 3

Let $\varepsilon_1 = Q(F_1(a))$ and $\varepsilon_M = Q(F(a))$ where

$$F_1(a) = \frac{g_1(a) - c}{\sqrt{g_2(a)}} \text{ and } F(a) = \frac{g_1(a) + c_1 - c}{\sqrt{g_2(a) + c_2}},$$

with $g_1(a) = an_1 \ln(1 + \frac{P_1}{a})$, $g_2(a) = an_1\Big(1 - \frac{1}{(1+\frac{P_1}{a})^2}\Big)$, $c_1 = \sum_{m=2}^{M} n_m \ln(1 + P_m)$, $c_2 = \sum_{m=2}^{M} n_m\Big(1 - \frac{1}{(1+P_m)^2}\Big)$, and $c = B \ln 2$. As we consider a point in $\mathcal{B}$, we get

$$\varepsilon_1 < 0.5 \Leftrightarrow an_1 \ln(1 + \frac{P_1}{a}) > c \Rightarrow E_1 > B \ln 2 \tag{A.2}$$

where $E_1 = n_1 P_1$. To prove (A.2), we used the inequality $\ln(1 + x) \leq x$ when $x \geq 0$. Once again, belonging to $\mathcal{B}$ leads to

$$F_1(a) \leq F(a) \leq \frac{\sqrt{2B \ln 2}}{3}. \tag{A.3}$$

We want to show that ε_1 and ε_M are decreasing functions with respect to a, i.e., $F_1'(a) \geq 0$ and $F'(a) \geq 0$ where $f'(a)$ stands for $\frac{df}{da}$ for any mapping f. As $g_1(a), g_2(a), g_1'(a)$ and $g_2'(a)$ are strictly positive, we have

$$F_1'(a) \geq 0 \quad \Leftrightarrow \quad 2g_1'(a)g_2(a) \geq g_2'(a)(g_1(a) - c) \tag{A.4}$$

$$\Leftrightarrow \quad c \geq E_1 H\left(\frac{P_1}{a}\right) \tag{A.5}$$

and

$$F'(a) \geq 0 \Leftrightarrow 2g_1'(a)(g_2(a) + c_2) \geq g_2'(a)(g_1(a) + c_1 - c) \tag{A.6}$$

$$\Leftrightarrow c \geq E_1 H\left(\frac{P_1}{a}\right) + (c_1 - K\left(\frac{P_1}{a}\right) c_2) \tag{A.7}$$

where

$$x \mapsto H(x) = \frac{2x + 4 - \ln(1 + x)(\frac{4}{x} + x + 3)}{x(x + 3)},$$

and

$$x \mapsto K(x) = \frac{2(x + 1)^3\left(\ln(1 + x) - \frac{x}{x+1}\right)}{x^2(x + 3)}.$$

After some algebraic manipulations, (A.4) and (A.6) are equivalent to

$$F_1(a) \leq \quad \frac{2g_1'(a)\sqrt{g_2(a)}}{g_2'(a)} \quad = \sqrt{E_1} W\left(\frac{P_1}{a}, 0\right) \tag{A.8}$$

$$F(a) \leq \frac{2g_1'(a)\sqrt{g_2(a) + c_2}}{g_2'(a)} = \sqrt{E_1} W\left(\frac{P_1}{a}, \frac{c_2}{E_1}\right) \tag{A.9}$$

with

$$(x,y) \mapsto W(x,y) = K(x)\sqrt{y + \frac{x+2}{(1+x)^2}}. \tag{A.10}$$

We want now to prove that ((A.5) OR (A.8) holds for any $x > 0$) AND ((A.7) OR (A.9) holds for any $x > 0$). For that, we split the analysis into two intervals on x.

- If $x \in (0, 484)$: the function $x \mapsto W(x,0)$ is a positive unimodal function converging to zero when $x \to \infty$. For $x \in (0, 484)$, it is easy to check that $W(x,0) \geq W(0,0) = \frac{\sqrt{2}}{3}$. [1] As $W(x,y) > W(x,0)$ for any $y \geq 0$, we obtain that $\sqrt{E_1}W(x,y) \geq \sqrt{E_1}W(x,0) \geq \frac{\sqrt{2E_1}}{3}$. Due to (A.2), we have $\sqrt{E_1}W(x,y) \geq \sqrt{E_1}W(x,0) \geq \frac{\sqrt{2B\ln 2}}{3}$. According to (A.3), we check that $\sqrt{E_1}W(x,y) \geq \sqrt{E_1}W(x,0) \geq F(a) \geq F_1(a)$. Therefore, (A.8) and (A.9) hold.

- If $x \in [484, \infty)$: in that interval, we can see that $H(x) \leq 0$, which implies that (A.5) holds.

  It now remains to check that either (A.7) or (A.9) holds. For doing so, we distinguish two cases:

  - If $c_1 \leq 10.37c_2$: one can check that $K(x)$ is an increasing function. Therefore for $x \geq 484$, we get $K(x) \geq K(484) > 10.37$. Consequently, $c_1 - K(x)c_2 < 0$. As $H(x) \leq 0$ too for $x \geq 484$, it is easy to show that (A.7) holds.
  - If $c_1 > 10.37c_2$: this inequality leads to

$$\sum_{m=2}^{M} n_m \ln(1+P_m) - 10.37 n_m \left(1 - \frac{1}{(1+P_m)^2}\right) > 0$$

    which forces that there exists at least one $m_x \in \{2, ..., M\}$ such that:

$$n_{m_x} \ln(1+P_{m_x}) > 10.37 n_{m_x}\Big(1 - \frac{1}{(1+P_{m_x})^2}\Big) > 0 \Rightarrow P_{m_x} > 31866$$

    which implies that:

$$c_2 \approx n_{m_x} + \sum_{m \in \{2,..,M\} \setminus m_x} n_m\Big(1 - \frac{1}{(1+P_{m_x})^2}\Big) \Rightarrow c_2 > n_{m_x}.$$

    Consequently, according to (A.10), $\sqrt{E_1}W\left(x, \frac{c_2}{E_1}\right) \geq K(484)\sqrt{n_{i_x}} \geq 10.37 \cdot \sqrt{1}$. If (A.9) does not hold, one can see that $\varepsilon_M < Q(10.37) \approx 1.7{\cdot}10^{-25}$. As this error does not correspond to any reasonable operating point, we consider that (A.9) holds.

[1] In general for $x \in (0, 484)$ it holds $W(x,y) \geq \frac{\sqrt{y+2}}{3}$, so for (A.9) to hold it is enough to show $F(a) \leq \frac{\sqrt{c_2 + 2E_1}}{3}$

A.4 Proof of Result 2

Consider the last round M where for the optimal point $(\vec{n}_M^\star, \vec{P}_M^\star)$, we know that $\varepsilon_{M-1} > \varepsilon_M$ (see Lemma 1 and its related proof for more details). For $x \in [0, n_M^\star]$, let

$$F(x) = Q\left(\frac{x\ln(1+P_M^\star) + \sum_{i=1}^{M-1} n_i^\star \ln(1+P_i^\star) - B\ln 2}{\sqrt{x\frac{P_M^\star(P_M^\star+2)}{(P_M^\star+1)^2} + \sum_{i=1}^{M-1} n_i \frac{P_i^\star(P_i^\star+2)}{(P_i^\star+1)^2}}}\right).$$

We know that $F(0) = \varepsilon_{M-1} > \varepsilon_M = F(n_M^\star)$ and that $F(\cdot)$ is a continuous (not necessary monotonically decreasing) function. Therefore, it exists $x_0 \in (0, n_M^\star)$ such that $F(0) < F(x_0) < F(n_M^\star)$. If F is smooth enough, it exists an integer $\overline{n} \in \{1, 2, ..., n_M^\star - 1\}$ (typically equal to $\lfloor x_0 \rfloor$ or $\lceil x_0 \rceil$) such that $\varepsilon_{M-1} > F(\overline{n}) > \varepsilon_M$. Then, the new point of $M+1$ rounds, which is $(\vec{n}_{M-1}^\star, \overline{n}, n_M^\star - \overline{n}, \vec{P}_{M-1}^\star, P_M^\star, P_M^\star)$, leads to the following average energy

$$\sum_{m=1}^{M-1} n_m^\star P_m^\star \varepsilon_{m-1} + \overline{n} P_M^\star F(\overline{n}) + (n_M^\star - \overline{n}) P_M^\star \varepsilon_{M-1},$$

which is smaller that the average energy provided by the point $(\vec{n}_M^\star, \vec{P}_M^\star)$. Obviously the reliability constraint (given by ε_M) remains unaltered and the latency constraint does not change since $D(\vec{n}_m) = 0$. So increasing the number of transmissions to $M+1$ improves the optimal operating point of M transmissions.

A.5 Proof of Lemma 4

To prove the lemma, we will prove that if for some solution the states $\tilde{S}_{i-1}$, $\tilde{S}_i$,$\tilde{S}_{i+1}$ satisfy $\tilde{c}_{i-1} \geq \tilde{c}_i$ and $\tilde{c}_i < \tilde{c}_{i+1}$, then there exists a better solution, thus it cannot be the optimal one. Therefore, if for the optimal solution for some i we know $c_{i+1}^\star > c_i^\star$ then it must $c_i^\star > c_{i-1}^\star$ and since from Lemma 1 we know $c_M^\star > c_{M-1}^\star$, this lemma is proved by induction.

To prove the existence of a better solution we only have to prove the superiority of a configuration of $M-1$ rounds that goes directly from the state $\tilde{S}_{i-1}$ to $\tilde{S}_{i+1}$ using one fragment of blocklength $n_i + n_{i+1}$ and has exactly the same configuration before and after those states (then due to Proposition 2 there exists an even better configuration with M rounds) Hence, we only need to prove:

$$\Delta E(\tilde{S}_{i-1}, \tilde{S}_i) + \Delta E(\tilde{S}_i, \tilde{S}_{i+1}) \geq \Delta E(\tilde{S}_{i-1}, \tilde{S}_{i+1}). \tag{A.11}$$

Since a zero delay penalty is assumed, using (1.3) and (1.4) with equalities allows us to derive that

$$\Delta E(S_{k-1}, S_k) = n_k P_k \varepsilon_{k-1} = n_k (e^{\frac{\gamma_k}{n_k}} - 1) Q(c_{k-1})$$

where $\gamma_k = c_k\sqrt{V_k} - c_{k-1}\sqrt{V_{k-1}} > 0$. Since $\tilde{c}_{i-1} \geq \tilde{c}_i$, to prove (A.11) it suffices to prove that

$$\tilde{n}_i e^{\frac{\tilde{\gamma}_i}{\tilde{n}_i}} + \tilde{n}_{i+1} e^{\frac{\tilde{\gamma}_{i+1}}{\tilde{n}_{i+1}}} \geq (\tilde{n}_i + \tilde{n}_{i+1}) e^{\frac{\tilde{\gamma}_i + \tilde{\gamma}_{i+1}}{\tilde{n}_i + \tilde{n}_{i+1}}}.$$

Changing variables as $\lambda_l = \dfrac{\tilde{n}_l}{\tilde{n}_i + \tilde{n}_{i+1}}$ and $x_l = \dfrac{\tilde{\gamma}_l}{\tilde{n}_l}$, $l \in \{i, i+1\}$:

$$\lambda_i e^{x_i} + \lambda_{i+1} e^{x_{i+1}} \geq e^{\lambda_i x_i + \lambda_{i+1} x_{i+1}},$$

which holds due to the convexity of the exponential function.

A.6 Proof of Result 3

We consider $E_i = n_i^\star P_i^\star$ where $n_i^\star$ and $P_i^\star$ are the i-th blocklength and power components of the optimal solution $(\vec{n}_M^\star, \vec{P}_M^\star)$ respectively. Notice that each E_i depends on N. Let us assume that it exists at least one $i_0 \in \{1, 2, ..., M\}$ such that $\lim\limits_{N\to\infty} E_{i_0} = \infty$. According to Lemma 4, we know that $\varepsilon_1 > \varepsilon_2 > ...\varepsilon_M = 1 - T_{\text{rel}} > 0$ at the optimal point. Consequently, the minimum average energy $\lim\limits_{N\to\infty}\left(E_1 + \sum\limits_{i=2}^{M} \varepsilon_{i-1} E_i\right) = \infty$ too. For at least one finite N, say N_f, the optimal point leads to a finite minimum average energy. For any $N > N_f$, the optimal solution cannot increase the minimum average energy since the optimal solution at N_f is a feasible point of Problem 1 for N. So the minimum average energy is upper bounded when $N \to \infty$. Therefore,

$$\lim_{N\to\infty} E_i < \infty, \forall i \in \{1, 2, ..., M\}.$$

When $N \to \infty$, both the delay feedback model don't not have an impact on the latency constraint (1.3) (since $D(\cdot)$ doesn't allow infinite values for finite input), so we can generally apply the results obtained for $D = 0$. According to Lemma 3, we also know that it is preferable to increase the blocklength rather than the power in order to save energy. Therefore, when $N \to \infty$, we have to take $n_1^\star$ as large as possible, i.e., $\lim\limits_{N\to\infty} n_1^\star = \infty$. Similar arguments can be applied to the other rounds, i.e., $\lim\limits_{N\to\infty} n_i^\star = \infty$ with $i \in \{2, \cdots, M\}$. We prove below that $E_i = \lim\limits_{N\to\infty} n_i^\star \ln(1 + P_i^\star)$:

$$\begin{aligned}
\frac{P_i^\star}{P_i^\star + 1} \leq \quad & \ln(1 + P_i^\star) \quad \leq P_i^\star \\
\Rightarrow \lim_{N\to\infty} \frac{n_i^\star P_i^\star}{P_i^\star + 1} \leq & \lim_{N\to\infty} n_i \ln(1 + P_i^\star) \leq \lim_{N\to\infty} n_i^\star P_i^\star \\
\Rightarrow \frac{E_i}{0+1} \leq & \lim_{N\to\infty} n_i \ln(1 + P_i^\star) \leq E_i.
\end{aligned}$$

Now we can easily confirm (1.1) leading to

$$\begin{aligned}
\lim_{N\to\infty} \varepsilon_m &= \lim_{N\to\infty} Q\left(\frac{\sum_{i=1}^{m} n_i^\star \ln(1+P_i^\star) - B\ln 2}{\sqrt{\sum\limits_{i=1}^{m} n_i^\star P_i^\star \dfrac{P_i^\star + 2}{(P_i^\star + 1)^2}}}\right) \\
&= Q\left(\frac{\sum_{i=1}^{m} E_i - B\ln 2}{\sqrt{2\sum_{i=1}^{m} E_i}}\right). \qquad \text{(A.12)}
\end{aligned}$$

Putting $m = M$ in (A.12) and using the reliability constraint (1.4) with equality, we have

$$\sum_{i=1}^{M} E_i = \frac{(Q^{-1}(1-T_{\rm rel}))^2}{2}\Big(1+\sqrt{1+\frac{2B\ln 2}{(Q^{-1}(1-T_{\rm rel}))^2}}\Big)^2 \tag{A.13}$$

where its right hand side corresponds to the energy when no HARQ ($M = 1$) is used and is denoted by $E^{\infty}_{\rm No-HARQ}$.

A.7 Proof of Result 4

The function $E \mapsto Q\Big(\frac{E - B\ln 2}{\sqrt{2E}}\Big)$ is plotted in Figure A.1. We also draw the m-th component of the objective function (1.26) of Result 3, which corresponds to the area of the partially grey partially green rectangular box located from $\sum_{i=1}^{m-1} E_i$ to $\sum_{i=1}^{m} E_i$ with a level ε_{m-1} (see (A.12)). According to (1.25), the final point is $E^{\infty}_{\rm No-HARQ}$. Consequently, the sum of the green and the grey areas gives the value of the objective function (1.26). It is evident that the function $E \mapsto Q\Big(\frac{E - B\ln 2}{\sqrt{2E}}\Big)$ coincides at the upper left corner of each rectangular box and is always inside each rectangular box (due to its decreasing monotonicity). Therefore, the value of the objective function (1.26) cannot be lower than the green area. When $M \to \infty$, we can decrease the width of each rectangular box converging to a solution that includes only the green area. Consequently, the minimum energy spent converges to the green area, which is identical to the Riemann integral of $E \mapsto Q\Big(\frac{E - B\ln 2}{\sqrt{2E}}\Big)$ from 0 to $E^{\infty}_{\rm No-HARQ}$.

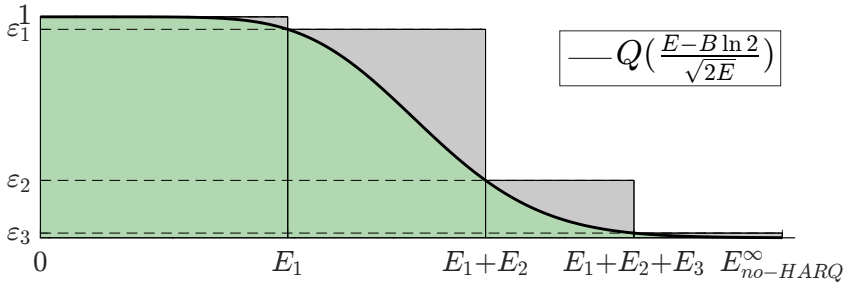

Figure A.1: Geometrical interpretation of Result 3 for $M = 4$.

A.8 Proof of Proposition 2

The constraints of the problems 5 and 4 are the same, therefore they share the same feasible domain that we denote by $\mathbb{D}$. Thus, $(B^{mod}, \vec{n}_M^{mod}, \vec{P}_M^{mod})$ is a feasible point of Problem 4. Since $Th^{\star}$ is the optimal value and Th just a feasible one, we have that $Th \leq Th^{\star}$. Furthermore, the solution of Problem 5 guarantees that for every point in $\mathbb{D}$ it holds $\frac{B}{\sum_{m=1}^{M} n_m \varepsilon_{m-1}} \leq \frac{Th}{T_{\rm rel}}$. Therefore if $x^{\star} \in \mathbb{D}$ is the optimal point of Problem 4 and gives an error probability of $\varepsilon_M^{\star}$, then $\frac{Th^{\star}}{(1-\varepsilon_M^{\star})} \leq \frac{Th}{T_{\rm rel}}$ from which we can easily show that $Th^{\star} \leq \frac{Th}{T_{\rm rel}}$.

A.9 Proof of Proposition 3

The proof is a combination of the proofs at A.1 and A.4 but applied now at the problem 6. If $(\vec{n}_M^\star, \vec{P}_M^\star)$ is the optimal solution of problem 6 and $\varepsilon_m^\star = \varepsilon(\vec{n}_m^\star, \vec{P}_m^\star)$ (where $\vec{n}_m^\star$ (resp. $\vec{P}_m^\star$) is an extracting vector from the m-th first components of $\vec{n}_M^\star$ (resp. $\vec{P}_M^\star$)) then it holds that $\varepsilon_{M-1}^\star > \varepsilon_M^\star$ since if not, by just removing the M-th round (or equivalently setting $n_M = 0$) we arrive to a better than optimal solution, i.e. giving smaller objective function (1.36) while satisfying all the constraints (1.37-1.39).

So by not wanting to reduce the number of rounds we proved $\varepsilon_{M-1}^\star > \varepsilon_M^\star$ but it remains to show that if we increase M, we can do stricly better. Following exactly the same steps as in proof at A.4, with $M+1$ rounds available, it is possible to find a point $(\vec{n}_{M-1}^\star, \overline{n}, n_M^\star - \overline{n}, \vec{P}_{M-1}^\star, P_M^\star, P_M^\star)$ with average consumed energy:

$$\left(\sum_{m=1}^{M-1} n_m^\star P_m^\star \varepsilon_{m-1}^\star\right) + \overline{n} P_M^\star F(\overline{n}) + (n_M^\star - \overline{n}) P_M^\star \varepsilon_{M-1} < \sum_{m=1}^{M} n_m^\star P_m^\star \varepsilon_{m-1}^\star \leq \mathrm{E_t}$$

so that the energy constraint (1.39) is satisfied. It also attains the same error probability as with M rounds (namely $\varepsilon_M^\star$) so also the reliability constraint (1.38) isn't violated. On top of being feasible it gives a lower objective function (1.36)

$$\left(\sum_{m=1}^{M-1} n_m^\star \varepsilon_{m-1}^\star\right) + \overline{n} F(\overline{n}) + (n_M^\star - \overline{n}) \varepsilon_{M-1} < \sum_{m=1}^{M} n_m^\star \varepsilon_{m-1}^\star$$

and this concludes the proof.

A.10 Proof of Proposition 4

Assume that for $m_0 < M$ we have $\varepsilon_{m_0}^\star < 1 - T_{\mathrm{rel}}$. Then if we reduce the available number or retransmissions from M to $M' = m_0$, then $(\vec{n}_{M'}^\star, \vec{P}_{M'}^\star)$ is a feasible point of the reduced problem with a bigger value of the objective function (1.36) which according to proposition 3 is a contradiction.

Proving $\varepsilon_M^\star \leq 1 - T_{\mathrm{rel}} < \varepsilon_{M-1}(\vec{n}_M^\star, n_M^\star - 1, \vec{P}_M^\star)$ is fairly simple since the first inequality is the reliability constraint (1.38) and the second cannot be violated; otherwise the point $(\vec{n}_{M-1}^\star, n_M^\star - 1, \vec{P}_M^\star)$ is better than the optimal solution, which again leads to a contradiction.

Appendix B

B.1 Proof of Lemma 5

We first consider $M = N$ and so $n_i = 1, \forall i$ and we have the general case where each symbol chooses its own average power P_i.We want to prove that $P_i = \frac{\mathrm{E_t}}{N},\ \forall i$ is the solution of the optimization problem. If it is true, these P_i can get out of the sums of the error formula (2.1), leaving $\sum_i^N 1 = N$, i.e.

$$Q\left(\frac{\sum_{i=1}^{M} n_i \log(1+gP_i) - B\log 2}{\sqrt{\sum_{i=1}^{M} n_i\left(1-\frac{1}{(1+gP_i)^2}\right)}}\right) \overset{P_i=\frac{\mathrm{E_t}}{N}}{=} Q\left(\frac{N\log(1+g\frac{\mathrm{E_t}}{N}) - B\log 2}{\sqrt{N\left(1-\frac{1}{(1+g\frac{\mathrm{E_t}}{N})^2}\right)}}\right) \tag{B.1}$$

This way the optimal error probability could be expressed versus N and $\frac{\mathrm{E_t}}{N}$ which is equivalent to choose one block of size N with identical power $\frac{\mathrm{E_t}}{N}$.

First of all since using full resources is beneficial for reliability, which means the constraints become equalities (proof similar to the ones of lemmas 2 and 3). Moreover since Q-function is strictly decreasing and the logarithm is increasing, we can alter the objective function and we end up to

$$\max_{x_1,\ldots,x_N} \log\big(\sum_{i=1}^{N}\log(\frac{1}{x_i}) - B\log 2\big) - \frac{1}{2}\log\big(\sum_{i=1}^{N}(1-x_i^2)\big) \tag{B.2}$$

$$s.t. \quad \sum_{i=1}^{N}\frac{1}{x_i} = \tilde{\mathrm{E}} \tag{B.3}$$

where $x_i = \frac{1}{1+h^2P_i}$ and $\tilde{\mathrm{E}} = N + h^2\mathrm{E_t}$. So $x_i \in [1/\tilde{\mathrm{E}}, 1]$. The domain on which we maximize is a compact set, thus a global maximum should exist. Additionally, the interval boundary, i.e. $x_i \in \{1/\tilde{\mathrm{E}}, 1\}$ represents the cases where all but one symbol vanish (if $x_i = 1 \Leftrightarrow P_i = 0$ and if $x_i = 1/\tilde{\mathrm{E}} \overset{(B.3)}{\Leftrightarrow} x_j = 0 \quad \forall j \neq i$) which yield suboptimal error probabilities because it is equivalent to the case of sending only one symbol with all the energy and so wasting most of the blocklength resources (i.e. if $x_i = 1/\tilde{\mathrm{E}}$ then $n_j = 0\ \forall j \neq i$) . Hence, the global maximum cannot be on the interval boundary. We use KKT conditions to prove that there is only one stationary point for the above problem and this point is when all x_i are equal to each other, and so these x_i are optimal.

Applying the KKT conditions with λ the Lagrangien multiplier associated with (B.3), we get the set of equations

$$-\frac{x_i^3}{V}+\frac{x_i}{A}=\lambda, \quad \forall i \in \{1,2,...,N\} \tag{B.4}$$

with $A=-\sum_{i=1}^{N}\log(x_i)-B\log 2$ and $V=\sum_{i=1}^{N}(1-x_i^2)$. Let us assume that the solution of (B.4) is $\vec{x}^\star=(x_1^\star,...,x_N^\star)$ and denote $A^\star=A(\vec{x}^\star)$, $V^\star=V(\vec{x}^\star)$. $A^\star$ and $B^\star$ are the same for each equation in (B.4). If we can find more than three different elements of $\vec{x}^\star$, then the cubic polynomial $-\frac{x^3}{V^\star}+\frac{x}{A^\star}-\lambda=0$ has more than three roots which is impossible. Additionally as $A^\star$, $B^\star$, and $x_i^\star$ are positive by construction, we can show that $x_i^\star$ can at most take two different values. Let us denote them by $(\tilde{x}_1,\tilde{x}_2)$. The value $\tilde{x}_1$ is taken by n_1 out of N x_i-variables while the value $\tilde{x}_2$ is taken by $n_2=N-n_1$ x_i-variables. Then (B.3) and (B.4) can be transformed into

$$n_1+n_2=N \tag{B.5}$$

$$\frac{n_1}{\tilde{x}_1}+\frac{n_2}{\tilde{x}_2}=\tilde{\mathrm{E}} \tag{B.6}$$

$$-\frac{\tilde{x}_1^3}{V}+\frac{\tilde{x}_1}{A}=-\frac{\tilde{x}_2^3}{V}+\frac{\tilde{x}_2}{A} \tag{B.7}$$

For instance, the case $\tilde{x}_1=\tilde{x}_2=\dfrac{\tilde{\mathrm{E}}}{N}$ is a solution. Actually, it corresponds to our desired stationary point. It just remains to prove that this is the only solution.

For $\tilde{x}_1\neq\tilde{x}_2$:

$$\text{(B.7)} \Leftrightarrow A(\tilde{x}_1^2+\tilde{x}_1\tilde{x}_2+\tilde{x}_2^2)=V. \tag{B.8}$$

We will show that (B.8) and the assumptions (2.25) and (2.26) cannot all hold at the same time. Using (2.26) we get:

$$A>0 \Leftrightarrow n_1\log(\tilde{x}_1)+n_2\log(\tilde{x}_2)<-B\log 2 \tag{B.9}$$

$$\frac{A}{\sqrt{V}}<F \overset{\text{(B.8)}}{\Leftrightarrow} \sqrt{n_1(1-\tilde{x}_1^2)+n_2(1-\tilde{x}_2^2)}<F(\tilde{x}_1^2+\tilde{x}_1\tilde{x}_2+\tilde{x}_2^2) \tag{B.10}$$

with $F=\min\{0.45\sqrt{B\log 2}, Q^{-1}(10^{-9})\}$ and the change from $\max$ of (2.25) to $\min$ is due to the decreasing monotonicity of $Q(\cdot)^{-1}$.

In Figure B.1, we display the area where (B.9) holds in blue, and the area where (B.10) holds in grey. We want to prove that both blue and black areas are disjoint in order to have no solution satisfying both inequalities. It is easy to prove that the boundary-curve C_1 (resp. C_2) is convex (resp. concave). So to avoid common points between both areas, the points K_2 and K_3 (intersection point of curve C_2 with $\tilde{x}_2=1$ and $\tilde{x}_1=1$ respectively) have not to belong in the blue area.

The point $K_2=(e^{-\frac{B\log 2}{n_1}},1)$ does not belong in the blue area if it does not satisfy (B.10), i.e. for $n_1\log(\tilde{x}_1)=-B\log 2$ we want either (B.11) or (B.12) to hold:

$$\sqrt{n_1(1-\tilde{x}_1^2)}>0.45\sqrt{B\log 2}(\tilde{x}_1^2+\tilde{x}_1+1), \tag{B.11}$$

$$\sqrt{n_1(1-\tilde{x}_1^2)}>Q^{-1}(10^{-9})(\tilde{x}_1^2+\tilde{x}_1+1). \tag{B.12}$$

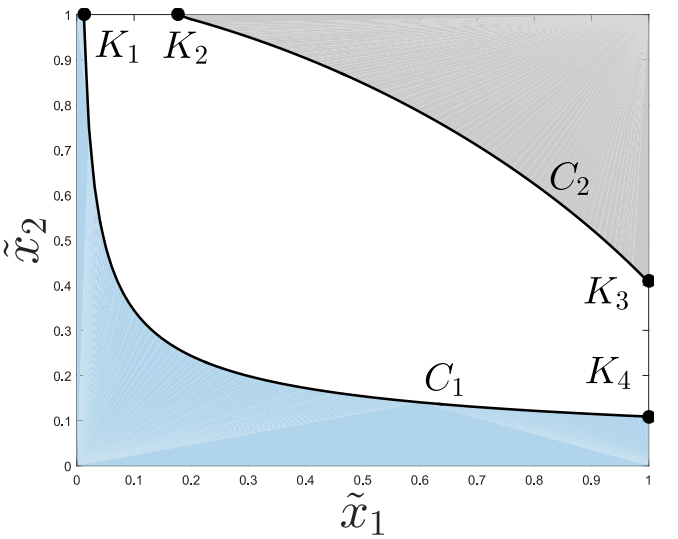

Figure B.1: Inequalities description for $\tilde{x}_1$ and $\tilde{x}_2$.

First we concentrate on (B.11). After substitution we want to show that:

$$\sqrt{\frac{\tilde{x}_1^2 - 1}{\log \tilde{x}_1}} > 0.45(\tilde{x}_1^2 + \tilde{x}_1 + 1). \tag{B.13}$$

A known inequality is $\log(x) \geq \frac{x-1}{\sqrt{x}}$, for $x \leq 1$. By dividing with $1 - x^2 (> 0)$ we can get $\sqrt{\frac{x^2-1}{\log x}} \geq \sqrt{\sqrt{x}(1+x)}$. Furthermore for $0 < x \leq 1$, we have $2x + 1 \geq x^2 + x + 1$. If

$$\sqrt{\sqrt{\tilde{x}_1}(1 + \tilde{x}_1)} \geq 0.45(2\tilde{x}_1 + 1) \tag{B.14}$$

holds, then (B.13) holds. Proving (B.14) is equivalent to show $\sqrt{\tilde{x}_1}^4 - 1.2346\sqrt{\tilde{x}_1}^3 + \sqrt{\tilde{x}_1}^2 - 1.2346\sqrt{\tilde{x}_1} + 0.25 \leq 0$. The roots of this fourth-order polynomial can analytically be found and the inequality is satisfied when $\tilde{x}_1 \geq \rho^2 = 0.0563$. So (B.13) is satisfied for $\tilde{x}_1 \geq \rho^2$. For $\tilde{x}_1 < \rho$, one can see it is equivalent to $0.45\frac{\tilde{x}_1^2 + \tilde{x}_1 + 1}{\sqrt{1 - \tilde{x}_1^2}} < 0.45\frac{\rho^2 + \rho + 1}{\sqrt{1 - \rho^2}}$. If $\tilde{x}_1 > e^{-\frac{1-\rho^2}{0.45^2(\rho^2+\rho+1)^2}} (\approx 0.0125)$, then $0.45\frac{\rho^2 + \rho + 1}{\sqrt{1 - \rho^2}} < \sqrt{\frac{-1}{\log \tilde{x}_1}}$ and again (B.13) holds. To sum up, when $\tilde{x}_1 > 0.0125$ the point K_2 is outside the blue area.

Now we will concentrate on (B.12) to treat the case of $\tilde{x}_1 \leq 0.0125$. From (B.12), we have:

$$n_1 > Q^{-1}(10^{-9})\frac{(\tilde{x_1}^2 + \tilde{x}_1 + 1)^2}{1 - \tilde{x}_1^2} \approx Q^{-1}(10^{-9})$$

which holds according to the assumption done in the Lemma. Similar procedure can be applied for the point K_3 which concludes the proof.

Bibliography

[1] (Jul. 2018). Technical Report: 3GPP TR 38.913 v15.0.0: Study on Scenarios and Requirements for Next Generation Access Technologies, [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.913.

[2] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, *et al.*, "Latency critical iot applications in 5g: Perspective on the design of radio interface and network architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, 2017.

[3] N. A. Johansson, Y.-P. E. Wang, E. Eriksson, and M. Hessler, "Radio access for ultra-reliable and low-latency 5g communications," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, IEEE, 2015, pp. 1184–1189.

[4] J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, "Millimeter-wave vehicular communication to support massive automotive sensing," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, 2016.

[5] M. A. Lema, K. Antonakoglou, F. Sardis, N. Sornkarn, M. Condoluci, T. Mahmoodi, and M. Dohler, "5g case study of internet of skills: Slicing the human senses," in *2017 European Conference on Networks and Communications (EuCNC)*, IEEE, 2017, pp. 1–6.

[6] G. Amitabha. (Sep. 2017). 5G mmWave Revolution and New Radio, [Online]. Available: https://5g.ieee.org/images/files/pdf/5GmmWave_Webinar_IEEE_Nokia_09_20_2017_final.pdf.

[7] (Aug. 2014). Report ITU-T Technol., Geneva: The tactile Internet, [Online]. Available: https://www.itu.int/oth/T2301000023/en.

[8] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "The 5g-enabled tactile internet: Applications, requirements, and architecture," in *2016 IEEE Wireless Communications and Networking Conference*, IEEE, 2016, pp. 1–6.

[9] (Oct. 2010). ''Communications requirements of smart grid technologies," U.S. Dept. Energy, Washington, DC, USA, [Online]. Available: http://www.smartgrid.gov/.

[10] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5g: Ran, core network and caching solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3098–3130, May 2018.

[11] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[12] M. Maier, M. Chowdhury, B. P. Rimal, and D. P. Van, "The tactile internet: Vision, recent progress, and open challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 138–145, May 2016.

[13] P. Petar, N. Jimmy J., S. Cedomir, D. C. Elisabeth, S. Erik, T. Kasper F., B. Alexandru-Sabin, K. Dong Min, K. Radosla, P. Jihong, and S. Rene B., "Wireless access for ultra-reliable low-latency communication: Principles and building blocks," *IEEE Network*, vol. 32, no. 2, pp. 16–23, Apr. 2018.

[14] (Nov. 2017). Report ITU-R M.2410-0(11/2017): Minimum requirements related to technical performance for IMT-2020 radio interface(s), [Online]. Available: `https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2410-2017-PDF-E.pdf`.

[15] Y. Polyanskiy, "Channel coding: Non-asymptotic fundamental limits," PhD thesis, Princeton University, Nov. 2010.

[16] M. Hayashi, "Information spectrum approach to second-order coding rate in channel coding," *IEEE Trans. on Inf. Theory*, vol. 55, no. 11, pp. 4947–4966, Nov. 2009.

[17] B. Makki, T. Svensson, and M. Zorzi, "Finite block-length analysis of the incremental redundancy HARQ," *IEEE Wireless Commun. Lett.*, vol. 3, no. 5, pp. 529–532, Oct. 2014.

[18] P. Wu and N. Jindal, "Coding versus ARQ in fading channels: How reliable should the PHY be?" *IEEE Trans. on Commun.*, vol. 59, no. 12, pp. 3363–3374, Dec. 2011.

[19] S. H. Kim, D. K. Sung, and T. Le-Ngoc, "Performance analysis of incremental redundancy type hybrid ARQ for finite-length packets in AWGN channel," in *Proc. IEEE Global Commun. Conf. (Globecom)*, Atlanta, GA, USA, Dec. 2013.

[20] A. R. Williamson, T. Chen, and R. D. Wesel, "A rate-compatible sphere-packing analysis of feedback coding with limited retransmissions," in *Proc. IEEE ISIT*, Cambridge, MA, USA, Jul. 2012.

[21] S. Xu, T.-H. Chang, S.-C. Lin, C. Shen, and G. Zhu, "Energy-efficient packet scheduling with finite blocklength codes: Convexity analysis and efficient algorithms," *IEEE Trans. on Wireless Commun.*, vol. 15, no. 8, pp. 5527–5540, May 2016.

[22] O. L. A. López, H. Alves, and M. Latva-aho, "Joint power control and rate allocation enabling ultra-reliability and energy efficiency in simo wireless networks," *IEEE Trans. on Commun.*, vol. 67, no. 8, pp. 5768–5782, Aug. 2019.

[23] L. Szczecinski, S. R. Khosravirad, P. Duhamel, and M. Rahman, "Rate allocation and adaptation for incremental redundancy truncated HARQ," *IEEE Trans. on Commun.*, vol. 61, no. 6, pp. 2580–2590, Jun. 2013.

[24] R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone, and E. Uysal, "Reliable transmission of short packets through queues and noisy channels under latency and peak-age violation guarantees," Oct. 2019. [Online]. Available: `https://arxiv.org/abs/1806.09396`.

[25] Y. Hu, M. Ozmen, M. C. Gursoy, and A. Schmeink, "Optimal power allocation for qos-constrained downlink multi-user networks in the finite blocklength regime," *IEEE Trans. on Wireless Commun.*, vol. 17, no. 9, pp. 5827–5840, Sep. 2018.

[26] H. Wang, N. Wong, A. M. Baldauf, C. K. Bachelor, S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "An information density approach to analyzing and optimizing incremental redundancy with feedback," in *Proc. IEEE Int. Symp. Inf. Theory*, Aachen, Germany, May 2017.

[27] K. F. Trillingsgaard and P. Popovski, "Generalized HARQ protocols with delayed channel state information and average latency constraints," *IEEE Trans. on Inf. Theory*, vol. 64, no. 2, pp. 1262–1280, Jan. 2017.

[28] D. Djonin, A. Karmokar, and V. Bhargava, "Joint rate and power adaptation for type-I hybrid ARQ systems over correlated fading channels under different buffer-cost constraints," *IEEE Trans. Veh. Technol.*, vol. 57, no. 1, pp. 421–435, Jan. 2008.

[29] E. Visotsky, V. Tripathi, and M. Honig, "Optimum ARQ design: A dynamic programming approach," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2003.

[30] M. Jabi, M. Benji, L. Szczecinski, and F. Labeau, "Energy efficiency of adaptive HARQ," *IEEE Trans. on Commun.*, vol. 64, no. 2, pp. 818–831, Feb. 2016.

[31] J. Gibson, *The Communications Handbook*. CRC press, 2002.

[32] G. Caire and D. Tuninetti, "The throughput of hybrid ARQ protocols for the Gaussian collision channel," *IEEE Trans. on Inf. Theory*, vol. 47, no. 5, pp. 1971–1988, Jul. 2001.

[33] C. L. Martret, A. Leduc, S. Marcille, and P. Ciblat, "Analytical performance derivation of hybrid ARQ schemes at IP layer," *IEEE Trans. on Commun.*, vol. 60, no. 5, pp. 1305–1314, May 2012.

[34] J. Park and D. Park, "A new power allocation method for parallel AWGN channels in the finite block length regime," *IEEE Wireless Commun. Lett.*, vol. 16, no. 9, pp. 1392–1395, Sep. 2012.

[35] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Feedback in the non-asymptotic regime," *IEEE Trans. on Inf. Theory*, vol. 57, no. 8, pp. 4903–4925, Aug. 2011.

[36] K. Vakilinia, S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Optimizing transmission lengths for limited feedback with nonbinary LDPC examples," *IEEE Trans. on Commun.*, vol. 564, no. 6, pp. 2245–2257, Jun. 2016.

[37] A. Martinez and A. G. i Fàbregas, "Saddlepoint approximation of random-coding bounds," in *Proc. Inf. Theory Applicat. Workshop (ITA)*, CA, USA, Aug. 2011.

[38] S. Khosravirad and H. Viswanathan, "Analysis of feedback error in automatic repeat request," Oct. 2017. [Online]. Available: `https://arxiv.org/abs/1710.00649`.

[39] R. Wolff, *Stochastic modeling and the theory of queues*. Upper Saddle River, NJ, U.S.A.: Prentice Hall, 1989.

[40] C. C. Tan and N. C. Beaulieu, "On first-order markov modeling for the rayleigh fading channel," *IEEE Transactions on Communications*, vol. 48, no. 12, pp. 2032–2040, 2000.

[41] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956. DOI: `10.1002/nav.3800030109`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800030109`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800030109`.

[42] A. H. Nuttall, "Some integrals involving the q_m function," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 95–96, Apr. 1975.

[43] J. E. Mitchell, "Branch-and-cut algorithms for combinatorial optimization problems," *Handbook of applied optimization*, vol. 1, pp. 65–77, 2002.

[44] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, pp. 679–684, 1957.

[45] K. J. Åström, "Optimal control of markov processes with incomplete state information i," Journal of Mathematical Analysis and Applications, vol. 10, pp. 174–205, 1965. [Online]. Available: `https://lup.lub.lu.se/search/ws/files/5323668/8867085.pdf`.

[46] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[47] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *International Conference on Learning Representations, ICLR*, San Juan, Puerto Rico, May 2016.

[48] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," *Proceedings of the 31st International Conference on Machine Learning, PMLR*, vol. 32, no. 1, pp. 387–395, Jun. 2014.

[49] 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, Feb. 2015.

[50] S. C. Jaquette, "Markov decision processes with a new optimality criterion: Discrete time," *The Annals of Statistics*, vol. 1, no. 3, pp. 496–505, 1973.

[51] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *International Conference on Machine Learning, ICML*, Syndey, Australia, Aug. 2017.

[52] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, USA, Feb. 2018.

[53] R. Koenker and K. F. Hallock, "Quantile regression," *Journal of economic perspectives*, vol. 15, no. 4, pp. 143–156, 2001.

[54] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[55] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[56] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, A. Muldal, N. Heess, and T. Lillicrap, "Distributed distributional deterministic policy gradients," in *International Conference on Learning Representations, ICLR*, Vancouver, Canada, May 2018.

[57] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," in *International Conference on Machine Learning, ICML*, New York, USA, Jun. 2016.

[58] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in neural information processing systems, NIPS*, California, USA, Dec. 2017.

[59] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *International Conference on Autonomous Agents and Multiagent Systems*, São Paulo, Brazil, May 2017.

[60] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Thirty-second AAAI conference on artificial intelligence*, Louisiana, USA, Feb. 2018.

[61] (Jun. 2018). Technical Report: 3GPP TR 36.913 v15.0.0: Requirements for further advancements for E-UTRA (LTE-Advanced), [Online]. Available: `https://www.3gpp.org/ftp/Specs/archive/36_series/36.913/`.

[62] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[63] H. Geoffrey. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent, [Online]. Available: `http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`.

[64] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in *International Conference on Machine Learning, ICML*, Syndey, Australia, Aug. 2017.