université
PARIS-SACLAY

NNT: xxxxxxxxxxxx

Offloading Strategies for Mobile Devices with Energy Harvesting Capabilities

Thèse de doctorat de l'Université Paris-Saclay
préparée à Télécom ParisTech

École doctorale n° 580 "Sciences et Technologies de l'Information et de la Communication" (STIC)
Spécialité de doctorat: Réseaux, Information et Communications

Thèse présentée et soutenue à Paris, le 22 Novembre 2019, par

Ibrahim FAWAZ

Composition du jury:

Luc VANDENDORPE Professeur, UCLouvain	Président
Elena Veronica BELMEGA Maîtresse de Conférences (HDR), ENSEA	Rapporteuse
Jean-Marie GORCE Professeur, INSA-Lyon	Rapporteur
Marceau COUPECHOUX Professeur, Télécom Paris	Examinateur
Deniz GUNDUZ Professeur, Imperial College London	Examinateur
Philippe CIBLAT Professeur, Télécom Paris	Directeur de thèse
Mireille SARKISS Maîtresse de Conférences, Télécom SudParis	Encadrante de thèse
Nouha OUALHA Ingénieure-Chercheuse, CEA	Invitée

Thèse de doctorat

Contents

List of Acronyms and Abbreviations

5G	Fifth Generation
AC	Alternating Current
ACK	Acknowledgment
AoI	Age of Information
AM	Amplitude Modulation
AR	Augmented Reality
ARQ	Automatic Repeat ReQuest
BS	Base Station
CC	Cloudlet Computing
CC-HARQ	Chase Combining Hybrid ARQ
CDMA	Code Division Multiple Access
CMDP	Constrained Markov Decision Process
CSI	Channel State Information
CSIT	Channel State Information at the Transmitter
DC	Direct Current
DDQN	Double Deep Q-Network
DL	Downlink
DP	Dynamic Programming
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
EH	Energy Harvesting
EM	Electromagnetic
ER	Experience Replay
ESIT	Energy State Information at the Transmitter
e.u	energy units
FC	Fog Computing
FDMA	Frequency Division Multiple Access
FM	Frequency Modulation
GPS	Global Positioning System
HAWT	Horizontal Axis Wind Turbine
ICT	Information and Communication Technology

i.i.d.	independent identically distributed
IoT	Internet of Things
IS	Importance Sampling
LP	Linear Programming
M2M	Machine-to-Machine
MCC	Mobile Cloud Computing
MDP	Markov Decision Process
MEC	Multi-Access Edge Computing
MIMO	Multiple-Input Multiple-Output
MISO	Multiple-Input Single-Output
MSE	Mean Squared Error
MU	Mobile Users
NACK	Negative Acknowledgment
NN	Neural Network
NOMA	Non-Orthogonal Multiple Access
OFDMA	Orthogonal Frequency Division Multiple Access
OMA	Orthogonal Multiple Access
PDF	Probability Density Function
PER	Prioritized Experience Replay
PI	Policy Iteration
POMDP	Partially observable Markov Decision Process
PV	Photo-Voltaic
QoS	Quality-of-Service
RAN	Radio Access Network
RF	Radio Frequency
RHS	Right Hand Side
RL	Reinforcement Learning
SIC	Successive Interference Cancellation
SWIPT	Simultaneous Wireless Information and Power Transfer
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TEG	Thermoelectric Generator
TEM	Thermoelectric Module
UL	Uplink
URLLC	Ultra-Reliable, Low-Latency Communication
V2V	Vehicle-to-Vehicle
VAWT	Vertical Axis Wind Turbine
VI	Value Iteration
VR	Virtual Reality
WiFi	Wireless Fidelity

WPT	Wireless Power Transfer

Introduction

This thesis is enrolled in the framework of the SCAVENGE (Sustainable CellulAr networks harVEstiNG ambient Energy) project supported by the "European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675891". It has been carried out since September 2016 at CEA LIST, Saclay, Communicating Systems Laboratory (LSC) in collaboration with Télécom ParisTech, Communications and Electronics department (COMELEC), under the supervision of Dr. Mireille SARKISS and Pr. Philippe CIBLAT.

Project Context

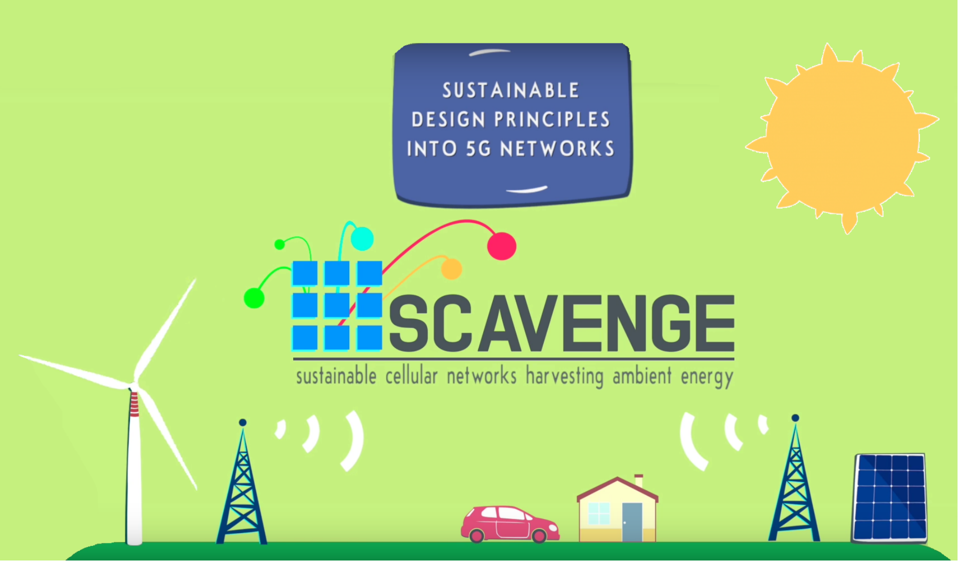
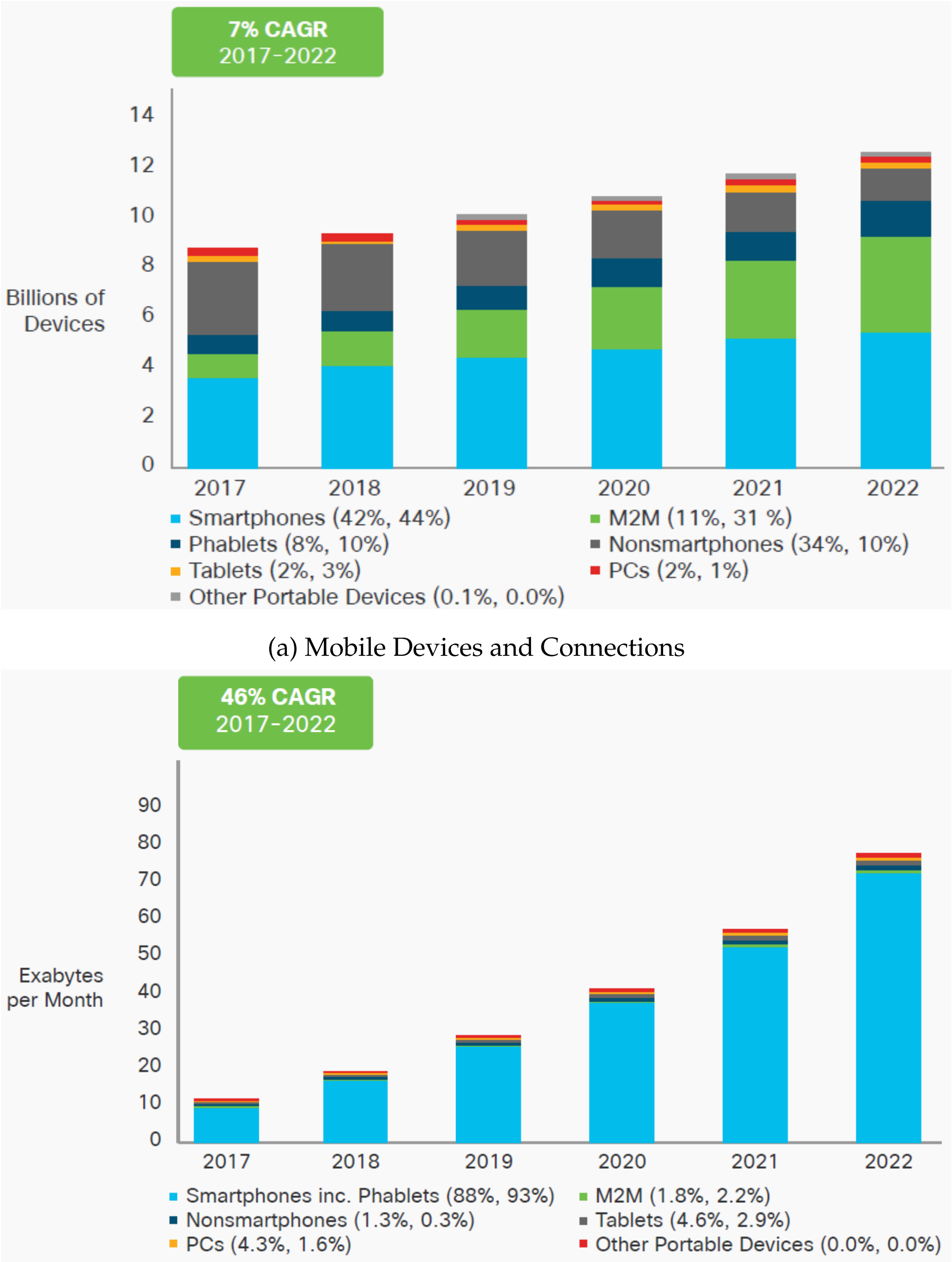


Figure 1 – SCAVENGE Project [1].

Information and Communication Technology (ICT) now penetrates all parts of society, with 6.8 billion of mobile subscribers approaching the global population, and 750 million households connected to the Internet forming 41% of the world's households [2]. The Fifth Generation (5G) of mobile technology will handle 1,000 times more capacity per unit area than previous mobile networks generation, for more than 100 billion devices with typical user rates of 10 Gb/s, and requirements for dramatically lower latency and higher reliability. This huge proliferation of devices (see Fig. 2) will also bring an equally significant increase in the carbon footprint of ICT. According to [3], the global ICT ecosystem already consumes about 1500 TWh of electricity per year, approximately 10% of the world's electricity production, 50% more than global aviation energy consumption, with an annual increase of 11 Mto of the emitted carbon footprint as seen in Fig. 3.



(a) Mobile Devices and Connections

(b) Mobile Traffic

Figure 2 – Global growth by device type [4].

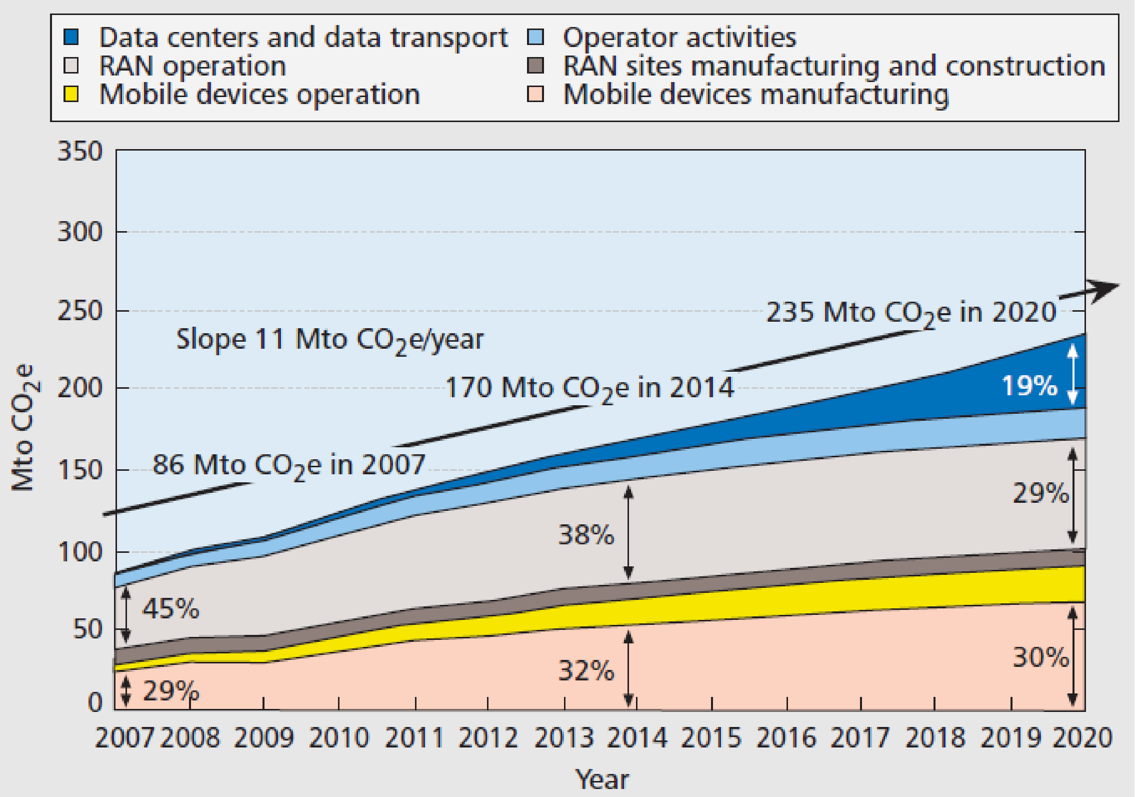


Figure 3 – Worldwide ICT Carbon footprint [5]

A promising solution to limit the negative environmental impact of the ICT sector and reduce its energy cost, while giving the 5G system sufficient autonomy from the electricity grid, is to harness the available ambient energy.

The SCAVENGE project proposes a sustainable paradigm for 5G mobile networks based on the premise that environmental energy can be scavenged through dedicated harvesting hardware, so as to power all network elements, i.e. Base Station (BS), mobile terminals, sensors, etc.

Problem Statement

5G on the go promises instant connectivity to billions of devices for the smart cities, the Internet of Things (IoT) and a truly connected world. There are three main areas of use cases for 5G

- **Massive Machine-to-Machine (M2M) communications:** also known as IoT, connecting billions of devices without human intervention on a scale never experienced before. Such a transformation has the potential to revolutionize modern industrial processes and applications, including agriculture, manufacturing and commercial communications.

- **Ultra-reliable low latency communications:** ranging from real-time applications to opening up a new world where medical care, procedures and remote treatments are all possible.
- **Enhanced mobile broadband:** offering considerably faster data speeds and greater capacity to keep the world connected. New applications include Virtual Reality (VR) and Augmented Reality (AR) and bring connected experiences beyond what was previously possible.

This explosive growth of mobile communications and IoT driven by the huge number of connected devices and resource-hungry mobile applications is increasing significantly the demands for high-volume delay-sensitive data traffic. It requires thus intensive computation and leads to high energy consumption. However, this expansion of wireless services is still restrained by mobile terminals limitations in terms of processing capacity, storage and energy. Recently, Energy Harvesting (EH) and offloading have been proposed as promising technologies to improve mobile devices computing capabilities and extend their battery lives.

On one hand, harvesting energy from surrounding environments allows to extend wireless devices lifetime by exploiting alternative renewable energy sources, such as solar power. This improves self-sustainability of wireless networks and limits their dependency on conventional grid power, reducing thus their growing carbon footprint. Depending on the environment conditions, the harvested energy arrives in intermittent amount at random times. It can be then stored in a capacity-limited energy storage device or battery for future use. The stochastic behavior and uncertainty of available energy may degrade the communication performance due to intolerable delay and packet loss. Therefore effective transmission policies are essential to satisfy Quality-of-Service (QoS) requirements and ensure the system reliability and sustainability.

On the other hand, offloading enables executing mobile devices tasks at remote cloud server or at BS with more energy and computation resources. This allows to reduce locally the consumed energy at the mobile devices and prolong their battery lives.

In Fig. 4, we roughly draw the communication scheme of the thesis context. A mobile device is able to harvest energy (see the right hand side of the figure). This device has heavy tasks to compute in order to satisfy the user's requirement (see the center of the figure). This task may be done locally or remotely to a BS with high computational ability (see the left hand side of the figure).

This thesis aims at studying optimal data transmission policies with offloading functionalities by leveraging on devices' energy harvesting capabilities in order to guarantee better performance in terms of energy consumption, packet error rate, delay, etc. There-

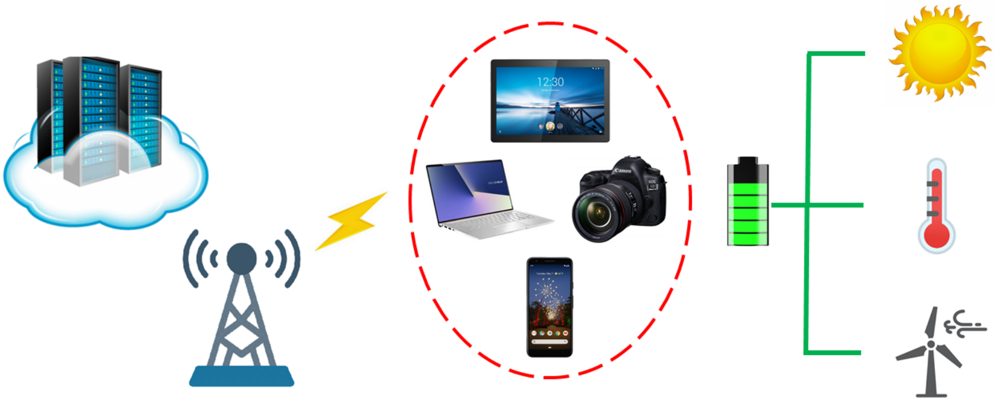


Figure 4 – Thesis context.

fore, the main objective of this thesis is to design and optimize resource scheduling and offloading strategies that account for random data arrivals, sporadic energy arrivals, channel conditions, application type, data buffer status and available energy in the battery.

The main contribution of this work is the introduction of the **strict delay constraint** instead of the average delay constraint used in the state-of-the-art. In fact, many applications (real-time control, industrial robotics, Vehicle-to-Vehicle (V2V) communications and safety systems, autonomous driving and safe transport networks, gaming, etc), the latency requirement is crucial for enabling application to work properly. For such applications, the average delay constraint is not strong enough anymore to ensure the well service. Notice that in this thesis, strict delay does not necessarly mean low delay, but just a strict deadline beyond which the packets are outdated and useless : therefore, this thesis does not deal with the so-called Ultra-Reliable, Low-Latency Communication (URLLC) of 5G systems.

Outline and Contributions

This thesis is composed of 4 chapters. Our original contributions are gathered in Chapters 2, 3 and 4 whereas Chapter 1 describes briefly the aforementioned technologies and presents an overview of the optimization framework used to solve our problems.

In Chapter 1, we first describe the underlying motivations for integrating EH into communication systems and present some of the available sources of this type of energy, in order to select the most appropriate ones for our system. Then, using the same approach, we show the importance of computation offloading in future communication systems, and among the available options, choose the most suitable one for our problem. The mathematical tool to exhibit powerful schedulers and offloading strategies will be the Markov Decision Process (MDP). Therefore the end of this Chapter is devoted to the description of the main results on MDP.

In Chapter 2, we investigate the joint optimization of resource scheduling and computation offloading for mobile networks, where EH-enabled devices are wirelessly connected to nearby BS, which can be endowed with some computational capabilities. In this first setup, we assume that the channel is acquired free-of-charge and error-free by the mobile device. This case stands for a perfect Channel State Information at the Transmitter (CSIT). Two special cases of this general problem are also presented, where the problem of resource scheduling is investigated for an EH-transmitter (without the offloading capabilities), and for a conventional transmitter (without EH and offloading capabilities), respectively.

In Chapter 3, we consider a more realistic scenario, where the channel is not perfectly known at the transmitter. In this setup, we assume that the receiver estimates the channel and feeds it back to the mobile device. In fact, this acquired Channel State Information (CSI) can be erroneous and can lead to a degradation of the system's performance. Therefore, we decide to assess the previously obtained optimal policy under imperfect CSIT conditions. We also consider imperfect CSIT assumption with Automatic Repeat ReQuest (ARQ) protocol, allowing thus packet re-transmission. This problem is also considered for an EH-transmitter without offloading ability to study the robustness of the resource scheduling derived policy under imperfect CSIT scenario.

In the Chapter 4, we investigate another approach dealing with the missing CSIT. We therefore assume that an outdated CSIT is only available, and we study the system of Chapter 2 under this condition, in which new types of errors are added. This assumption implies significant changes in the system description and the optimal policy to be able to adapt properly within this new context and maintain good performance.

Publications

The work conducted during this thesis has led to the following publications.

Journals

J1. I. Fawaz, M. Sarkiss and P. Ciblat, "Delay-Optimal Resource Scheduling of Energy Harvesting based Devices," IEEE Transactions on Green Communications and Networking (TGCN), accepted for publication in June 2019.

International conferences

C5. I. Fawaz, M. Sarkiss and P. Ciblat, “Packet Scheduling and Computation Offloading for Energy Harvesting Devices without CSIT,” submitted to IEEE Vehicular Technology Conference (VTC), September 2019.

C4. I. Fawaz, M. Sarkiss and P. Ciblat, “Delay-Optimal Resource Scheduling and Computation Offloading for Energy Harvesting Devices,” IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCom), September 2019.

C3. I. Fawaz, M. Sarkiss and P. Ciblat, “Joint Resource Scheduling and Computation Offloading for Energy Harvesting Communications,” IEEE International Conference on Telecommunication (ICT), June 2018.

C2. I. Fawaz, M. Sarkiss and P. Ciblat, “Optimal Resource Scheduling for Energy Harvesting Communications under Strict Delay Constraint,” IEEE International Conference on Communication (ICC), May 2018.

C1. I. Fawaz, P. Ciblat and M. Sarkiss, “Energy Minimization based Resource Scheduling for Strict Delay Constrained Wireless Communications,” IEEE Global Conference on Signal and Information Processing (GlobalSIP), November 2017.

Chapter 1

Preliminaries

Nowadays, mobile communication systems face unprecedented growth of connected devices which applications require data analysis and heavy computation as well as strict delay. These requirements lead to an increase in demands for processing capacity, storage and energy. In order to meet these challenges, energy harvesting and computation offloading schemes have been recently proposed.

In this chapter, we describe these two technologies and we present the mathematical optimization tools used to achieve the objectives of this work. Namely, we provide a review of the main MDP concepts and present several algorithms used to solve these problems.

1.1 Energy Harvesting

Global warming is one of the biggest threats to the natural world. This is caused primarily by the human use of fossil fuels (oil and coal), which emits greenhouse gases, in particular Carbon Dioxide (CO_2), into the atmosphere. According to [6], a 14% increase in ICT global emission is expected by the year 2040. Therefore, in an effort to reduce the operational expenses and environmental footprints, pushing for the integration of renewable energy technologies to power the communication infrastructures becomes essential.

Energy Harvesting or also known as Energy Scavenging is the process of collecting and storing ambient energy for future use [7]. The energy can be harvested from a lot of sources in the surrounding environment, such as solar, thermoelectric, vibration, wind power and electromagnetic radiations. In this section, we briefly describe each EH source together with its applicability to mobile phones, to ultimately select the ones that best suit our system.

1.1.1 Solar Energy

Figure 1.1 [8] shows the distribution of irradiation and thus the availability of solar power across the globe. Therefore, solar-powered generators can be deployed anywhere, even in northern countries (one can mention Germany where solar panels are well deployed on house's roofs). In southern countries, the solar energy is even more efficient because of more sun but also because the electric grid is not available anywhere, and the solar energy can be produced easily where it is used and does not require grid. Such a deployment is so easier and cheaper than conventional sources and reduces thus the dependence on expensive sources like oil. On the other hand, solar power depends on geolocation and weather. The stochastic nature and thus uncertainty of the available energy is a major obstacle to the large-scale use of solar energy in many regions. It requires a proper management to ensure its optimal use and avoid energy shortage.

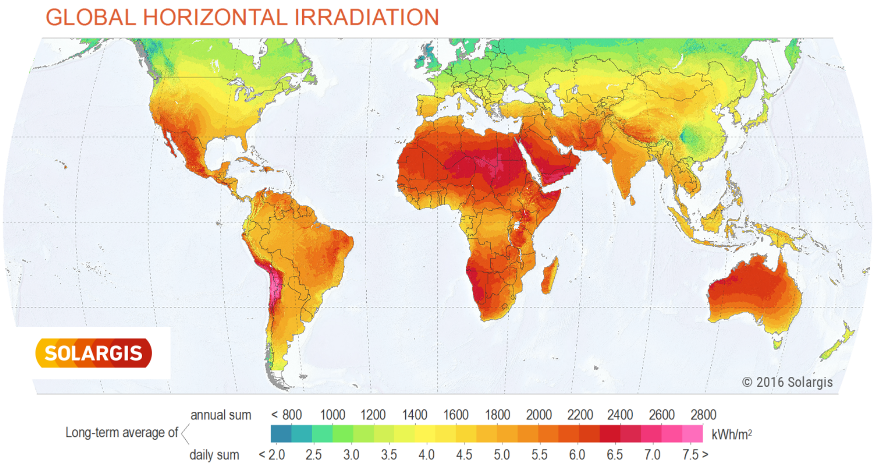


Figure 1.1 – Map of average solar irradiation across the globe [8].

Solar energy can be accessed by converting light radiation into electricity through a number of different technologies. Photo-Voltaic (PV) technology is the most used worldwide. Figure 1.2 shows an example of PV technology.

(a) Fixed tilt [9]

(b) Rooftop system [10]

(c) Car [11]

Figure 1.2 – Example of PV technology.

PV solar cells operate at near ambient temperature with no moving parts and their power efficiency is the same whatever the size of the panels [12]. A solar PV array consists of one or more electrically connected PV modules, where each module contains many individual cells [13]. Figure 1.3 shows the inside of a PV cell [14].

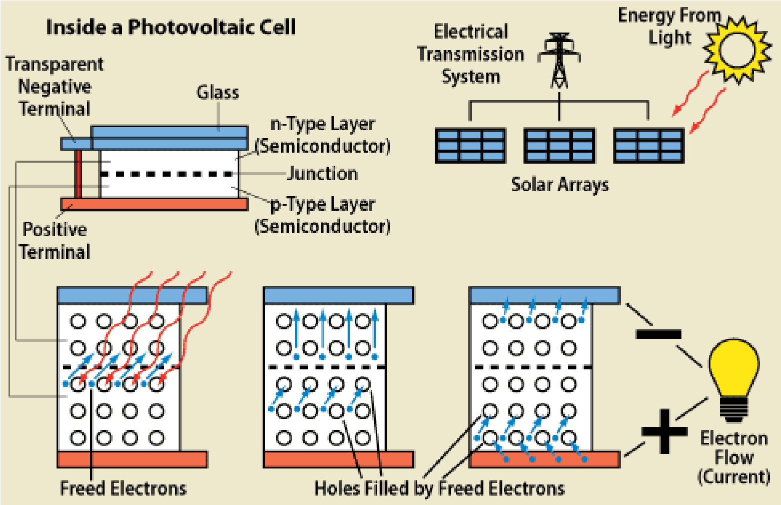

Inside a Photovoltaic Cell
Transparent Negative Terminal
Glass
n-Type Layer (Semiconductor)
Junction
p-Type Layer (Semiconductor)
Positive Terminal
Electrical Transmission System
Energy From Light
Solar Arrays
Electron Flow (Current)
Freed Electrons
Holes Filled by Freed Electrons

Figure 1.3 – Inside a PV cell [14].

When exposed to sunlight, a solar cell connected to an external circuit generates a Direct Current (DC), and using adequate components (combiners, inverters, and transformers), this current can be converted into grid-compatible Alternating Current (AC) if needed. Meanwhile, charge controllers and batteries could be also used to store energy during the day and provide on-demand power during the night. In peak sunlight, these PV cells are capable of generating over 50 – 100 mw of electrical power per square cm of area [15,16].

Application to mobile phones

Charging the smartphone using solar energy has been a long-standing consideration for many phone manufacturers to provide their customers with more sustainable alternative solutions to recharge their phones, especially in rural areas when it is not possible to plug into the electrical grid, or when the electricity supply is uncertain.

Samsung was officially the first manufacturer to launch a solar phone into the market in 2009. The "Solar Guru E1107" (Figure 1.4a), was launched in India to solve the problem of regular power outages, before it was expanded to Asia, Africa, and Latin America as well. The handset was able to provide between 5 and 10 minutes of talk time after one hour of solar charging using the built-in solar cells in its back plate. Samsung claimed that the battery is fully charged in about 40 hours of solar charging. Months later, Samsung released another solar device called "Blue Earth S7550" (Figure 1.4b), branded as an environmentally friendly product and launched with a bunch of advanced features (web access, YouTube, social networks, internet connection over Wireless Fidelity (WiFi) or 3G, Global Positioning System (GPS), etc) and in a much wider range of markets, including the United Kingdom. Similar to the previous model, this phone also had solar cells on its back providing 10 extra minutes of talk time after an hour of sunlight charging.

(a) Guru E1107

(b) Blue Earth S7550

Figure 1.4 – Samsung EH mobile phones [17].

Concurrently, LG introduced its mobile phone "POP GD510" (Figure 1.5a) integrated with solar cells in its battery cover. These cells charge in 10 minutes of sunlight enough energy to provide 2.25 minutes of talk time or 3 hours of standby time.

A year later, in 2010, Puma collaborated with Sagem to launch the "Puma Phone" (Figure 1.5b), a handset equipped with a solar panel designed to enhance the physical capabilities of this active phone equipped with step tracking and a GPS chip.

(a) LG POP GD510

(b) Puma

Figure 1.5 – LG and Puma EH mobile phones [18].

However, the solar panel was merely there to keep the battery charged, rather than to provide the main charging source. This is simply explained by the low efficiency of the solar panel despite the small energy requirements of these phones.

Since then, a number of manufacturers have been studying the feasibility of solar energy. For instance, Nokia developed in 2012 a prototype of a phone with a solar charging panel integrated in the back cover and tested it in different environments. The results confirmed that charging a mobile phone using just a solar panel mounted on the back cover is possible, but challenging. The prototype phones were able, at best, to collect enough energy to keep the phone in standby mode with very limited talk time, if they were carefully positioned. The biggest challenge is the limited size of the phone's rear cover, which limits the battery's charging capacity. In practice, the charging capacity is not solely dependent on the meteorological conditions and the amount of sunlight, but is also significantly affected by other factors such as lifestyle and the angle of incidence of the sunlight.

Recently, a Russian company called "Caviar", announced its new "iPhone X Tesla" (Figure 1.6) which can be fully charged with sunlight. The phone is equipped with a solar panel mounted on its back, allowing the smartphone to charge its battery from light using silicone semiconductors. The engineers of this company have performed more than 100 tests proving the efficiency and safety of this system, although the details regarding power specifications are not indicated on the product page. The smartphone is significantly thicker than its iPhone X counterpart and much more expensive.

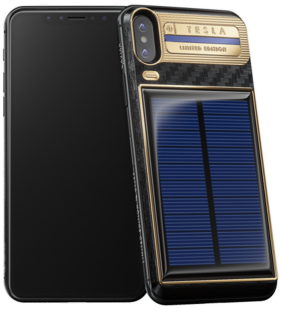


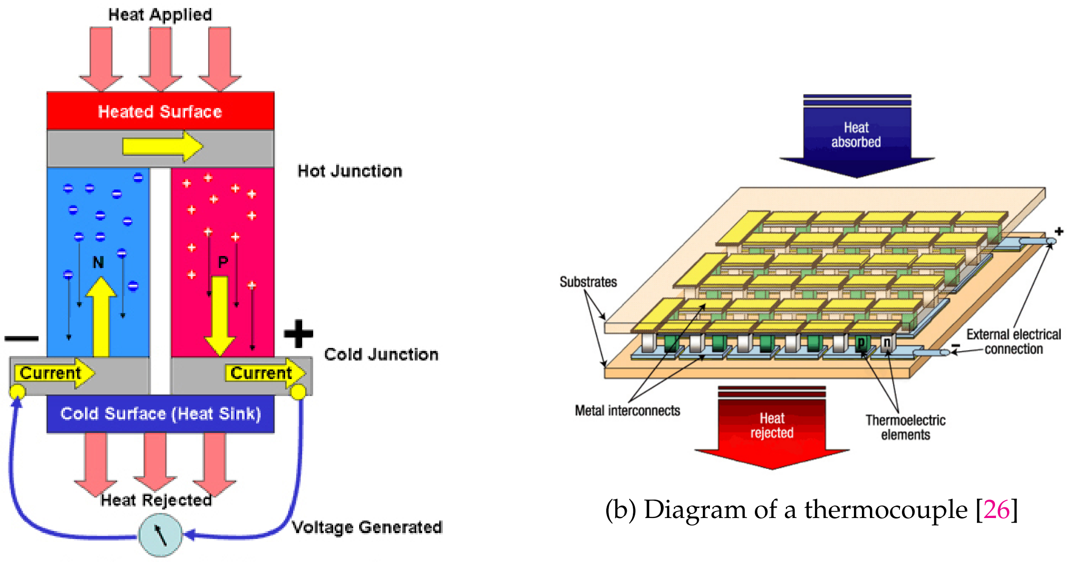
Figure 1.6 – IPhone X Tesla [19].

1.1.2 Thermoelectric Energy

Thermoelectric technology has been used in space missions for a long period by exploiting the radioisotope Thermoelectric Generator (TEG), which converts the heat of power plants, factories, motor vehicles, computers or even human bodies into electricity by the Seebeck effects, Peltier effect or Thomson effect [20].

In comparison with other sources of energy, the thermoelectric energy source provides relatively low electrical power levels (few hundreds of μW to mW) [21]. In particular, a power density of 60 μW/cm^2 is achieved from human bodies and can reach 10 mW/cm^2 if the heat is extracted from the industry [16, 22]. Moreover, the main challenge of the thermoelectric power generation is the low heat-to-electricity conversion efficiency. A number of research efforts have been undertaken to improve the waste heat recovery efficiency using heat sources such as power plants, automobiles, geothermal energy and other heat-generating industries [23]. In addition, the TEG has many advantages such as no moving parts, no toxic residuals, no chemical reactions, being environmental friendly and deployable in unmaintained situations [23, 24]. These reasons make this source of power attractive in wide range of applications such as military, aerospace, biological systems (powering implanted pacemakers), etc [21].

The basic construction unit of a thermal harvester is a thermocouple shown in Figure 1.7a [25]. When a temperature difference is applied across this material, heat begins to flow from the hotter to the cooler side. To boost the output voltage and get more power, many of these legs are connected electrically in series and thermally in parallel to form a thermopile, as it is illustrated in Figure 1.7b. In order to create a thermal circuit, the needed elements are: Thermoelectric Module (TEM), heat source and heat sink. The heat source provides an elevated temperature and represents the hot side, while the heat sink represents the cold side and serves as heat dissipation in order to achieve a lower temperature [23].



(b) Diagram of a thermocouple [26]

(a) Semiconductor thermocouple [25]

Figure 1.7 – Thermoelectric harvester general scheme

Application to mobile phones

Since thermoelectric energy is based upon the fact that electrons escape from hot to cold in order to create a current, one can then extract this type of energy from the human body which acts as a permanent source of heat.

At the 2010 Glastonbury Festival, the French telecommunications operator Orange presented a prototype of the "Power Wellies" (Figure 1.8) rubber boots that convert heat into electricity through a thermoelectric panel fixed in the sole to charge mobile phones. They claimed that this model provides an extra hour for the cell phone after 12 hours of walking.

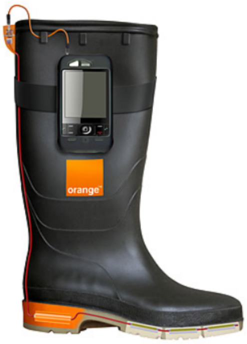


Figure 1.8 – Orange Power Wellies boots [27].

In turn, Vodafone stepped up the ante at the Isle of Wight Festival in 2013 with power shorts (Figure 1.9a) and sleeping bags (Figure 1.9b) both with thermoelectric pockets. These two prototypes are based on the Seebeck effect where the human body provides heat on the inner layer. The difference between this temperature and the colder temperature outside produces the power. They stated that 8 hours in the sleeping bag will provide 24 minutes of talk time and 11 hours of standby time, assuming that the inside of the sleeping bag is at 37° (the temperature of the human body). Likewise, a full day of walking in the power pocket shorts can recharge a smartphone for 4 hours.

1.1.3 Vibrations-based Energy

One of the most effective methods of implementing a vibration energy harvester system is to use mechanical vibration to apply a force to a transducer or displace an electromagnetic coil. Power generation from mechanical vibration usually uses ambient vibration around the harvesting device as an energy source and then converts it into electrical energy [29].

In some situations, vibrations can be very high, such as those of civil structures, railways, ocean waves and even human movements. The mechanical vibration energy is converted into electrical energy by electrostatic effect, magnetic field or stress on a piezoelectric material [30]. The amount of harvested power therefore depends on the vibration source and can provide about 4 μW/cm^3 from human motion, and around 800 μW/cm^3 from machine motion [16, 22].

Power Shorts

1. Smart shorts

These will be lined with ferroelectret foam, a unique polymer material which features internal voids.

2. Charging the voids

As the user moves, the size and shape of the voids change and a net charge is produced at the surface of the material

3. The Power Pocket

The charge from the voids is collected and then directed to the mobile devices held within.

(a) Power shorts

The Recharge Sleeping Bag

1. Harnessing thermal energy

Warmth generated from the user creates a temperature difference either side of thermoelectric modules built into the lining of the sleeping bag.

2. Thermoelectric material

Heat flows through a multi-layered flexible 'power felt', which contains insulating polymer films and two semi conductors. This generates an electric charge via the 'Seebeck effect'.

3. Power Pocket

A robust pocket, built to hold both a tablet and mobile phone, connects to the smart fabrics and directs the charge to the devices within and the LED lights integrated into the hood.

(b) Sleeping bags

Figure 1.9 – Vodafone smart fabric [28].

Application to mobile phones

The amount of energy extracted from the vibrations is directly related to the mass vibrating and the frequency (number of vibrations per second) with which it vibrates. In general, higher mass and frequency means more energy to collect. Harvesting vibrations with most harvesting technologies such as piezoelectrics is more appropriate for consistent high frequency vibrations, which is not the case for the human body, as it is sometimes active and sometimes completely inactive. Furthermore, the current materials and devices still cannot collect enough energy to charge mobile phones. This means that there is still some way to go before a workable and worry-free solution can be found. Therefore, vibration energy is not an option for our work.

1.1.4 Wind-based Energy

Wind power has been harvested for thousands of years, first to propel sailing boats, then to drive mechanical systems such as mill, pumps, etc. Nowadays, it can be harvested by a range of methods to produce electrical power.

Wind is a displacement of air due to different causes, the main one being the sun heating the non-uniform surface of the earth. The different elements at the surface of the earth (land, forest, desert, water, mountains, . . .) radiate the solar heat in the atmosphere in different fashions, such that hotter and colder masses of air appear, causing air to move around [31].

There are two main types of wind turbines designs: Horizontal Axis Wind Turbine

(HAWT) and Vertical Axis Wind Turbine (VAWT). HAWT are the most commonly used type of wind turbines and they are used almost exclusively in high-power installation. These turbines offer a range of power outputs from a few watts to thousands of kilowatts. On the other hand, VAWT generally have a lower efficiency than horizontal axis turbines, but present some advantages: the turbines are omnidirectional, i.e. they do not need to be oriented according to the wind direction and the alternator is located at ground level, reducing the weight on the top of the mast. The turbines are also well adapted to be installed on existing structures, like on top of buildings. There are however some drawbacks with the vertical axis turbine design: the omni-directionality of the VAWT presented as an advantage above requires stronger elements to withstand non constant torque due to gusts of wind. Thus, the difficulties in modelling them make the design of vertical axes turbines more complex compared to the horizontal ones.

The amount of harvested power depends primarily on the average wind speed. For instance, 100 mW of power can be produced at a wind speed between 2 and 9 m/s [32].

Application to mobile phones

Obviously, wind energy is not suitable for our work at all, since it requires large surfaces and turbines, which is not compatible with a small smartphone.

1.1.5 Electromagnetic Radiations-based Energy

Harvesting power from Electromagnetic (EM) sources, or Wireless Power Transfer (WPT), has attracted recent interest because of the broadcast nature of wireless communications and broadcasting infrastructure, such as analog/digital Television, Amplitude Modulation (AM)/Frequency Modulation (FM) radio, WiFi and cellular networks [32]. The system design is based on a simple concept of capturing the ambient Radio Frequency (RF) energy using the following equipment: appropriate antenna, tuning circuit, voltage multiplier and storage capacitor [33].

EM sources can be divided into two categories [33]:

- **Ambient:** These sources refer to the RF transmitters not destined for RF energy transfer. This type can provide a power ranging from 0.2 nW/cm^2 to 1 μW/cm^2 [34].

- **Dedicated:** These sources can be deployed to provide energy to network nodes when more predictable energy supply is needed. This type can provide 5 μW at a distance of 15 m from the source using a transmit power of 4 W [32].

Meanwhile, there are two main types of WPT [32]:

- **Near-field:** These systems rely on EM induction or magnetic resonance to transfer power wirelessly. They typically have high transfer efficiency exceeding 80% within a distance of a wavelength from the source. This approach is only suitable for short-range applications.

- **Far-field:** These systems utilize antennas to collect remotely radiated electromagnetic waves and rectifier circuits for the RF–DC conversion. This approach is suitable for long-range applications (up to a few kilometers).

However, the vision of embedding WPT into communication networks creates a technological need for a technique capable of transferring both information and power simultaneously to the end devices. For this reason, the concept of Simultaneous Wireless Information and Power Transfer (SWIPT) was first introduced in [35] from a theoretical point of view.

In the era of 5G communication, SWIPT technology can be vitally important for the transmission of information and energy in many types of modern information transmission systems and communication networks. However, fundamental design modifications are needed in the wireless communications sector to ensure an efficient SWIPT. In addition, the trade-off between information rate and harvested energy level becomes an important factor affecting the system's performance [36].

Application to mobile phones

Ambient electromagnetic radiation from WiFi transmitters, cell phone antennas, television towers and other sources could be converted into sufficient electrical power to sustain a battery charge.

In 2009, Nokia developed a prototype capable of harvesting from 3 to 5 mW. They pointed out that work is underway on an improved prototype that could collect up to 50 mW of energy, which is enough to slowly recharge a phone that is turned off. However, scientists believe that harvesting 50 mW could require about 1000 strong signals and an antenna capable of picking up such a wide range of frequencies, which means that a viable consumer device is still far from reaching the commercial market. Therefore, we do not rely in this work on this type of EH source.

1.1.6 Adopted EH sources

To sum up, we assume in this work that the energy can be harvested from the solar or thermoelectric sources. The field of EH remains an area of research and development, particularly for small devices such as mobile phones. The scavended energy is assumed to be intermittent, i.e., random. The conventional mathematical model for capturing this randomness (described in next Chapter) is not well supported by measurements (there is a lack of works in this topic) and especially the time-scale of the intermittence is not studied. Here we will assume that the scavenged energy time-variation scales with a slot (defined in next Chapter). We propose to design efficient transmission policies to use this random energy in the best possible way and avoid its waste and, of course, to prevent energy shortages and ensure sustainable communication.

1.2 Computation Offloading

With the recent expansion of services and applications embracing the Internet, the requirements for data storage and task processing have increased significantly. Meanwhile, despite all the advances in mobile devices, they are still resource constrained in terms of battery life, storage capacity and processor performance preventing to comute the tasks on time. Hence, they cannot meet the requirements of new communications systems. These constraints can be overcome by *computation offloading* that consists in sending extensive computations to resourceful servers to process remotely and send the results back.

This solution was first implemented using Mobile Cloud Computing (MCC), where Mobile Users (MU) can exploit computing and storage resources of powerful distant centralized clouds, accessible through the Core Network of a mobile operator and the Internet. However, all the advantages offered by MCC came at the cost of creating a massive overhead on both radio and mobile network backhauls. Moreover, it introduced high latency as the data is sent to powerful servers very far from users [37].

To tackle these challenges, Edge Computing has been proposed to provide an intermediate layer between the terminals and the cloud. This layer can be implemented in different ways depending on the devices acting as intermediate edge nodes, the communication protocols and networks, and the services offered to users. Therefore, edge computing systems can be classified into three categories:

- **Multi-Access Edge Computing (MEC) [38]:** This approach involves deploying intermediate nodes with storage and processing capabilities in the BSs of cellular networks. Thus it provides cloud computing capabilities in the Radio Access Network (RAN).

- **Fog Computing (FC) [39]:** This approach incorporates a computing layer that takes advantage of devices such as M2M gateways and wireless routers. Such nodes are called FC nodes and are used to compute and store data from end devices locally before forwarding it to the cloud.

- **Cloudlet Computing (CC) [40]:** This approach allows terminals to offload their computation to cloudlet devices located in their logical proximity and offering similar resources to that of a data center but on a smaller scale.

1.2.1 Adopted offloading approach

In this work, we are mainly dealing with mobile phones which are connected by default to a BS so that they can communicate, make/receive calls, etc. Therefore, we adopt the MEC approach for our problems. In other words, MU can use the cloud services embedded within each BS to execute some of their extensive computation.

1.3 Markovian Model

To study a system that changes over time, it is imperative to find a way to track these changes. A Markov chain is a particular model for tracking such systems that change over time according to given probabilities. It is a mathematical model that experiences transitions from one state to another according to certain probabilistic rules, where the *state* refers here to any particular situation that the system may take. The fundamental characteristic, or the *Markov property* of a Markov chain is that the probability of being in a particular state depends only on the current state, i.e.

$$P(S_{n+1} = s_{n+1} | S_n = s_n, S_{n-1} = s_{n-1}, \cdots, S_0 = s_0) = P(S_{n+1} = s_{n+1} | S_n = s_n) \tag{1.1}$$

Figure 1.10 shows an example of a Markov chain with two states G and B along with the respective probabilities of transitions r and w between them. If these two states refer to *Good* and *Bad* communication channels, this model is known as Gilbert-Elliot Channel [41].

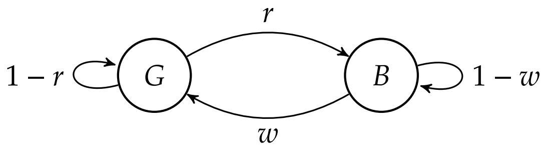


Figure 1.10 – Two-state Markov Chain.

A Markov chain is said to be **irreducible** if it is possible to get to any state from any state in a finite time. In other words, there exists a non-zero probability that any state j will be accessible from any state i at some point of the system's evolution.

A state i is said to be **transient** if there is a non-zero probability that the system will never visit i again after leaving it. If this condition does not hold, the state is said to be **recurrent**.

A state i has a period $k > 1$ if the system can return to this state within multiples of k time steps. If $k = 1$, then state i is said to be **aperiodic**.

A state i is said to be **ergodic** if it is aperiodic and recurrent at the same time. Consequently, if all states in an irreducible Markov chain are ergodic, then the chain is said to be ergodic [42]. This propriety has a direct implication in finding an optimal solution for the MDP presented in the following.

1.3.1 Markov Decision Process

An MDP is a discrete time stochastic control process for sequential decision making. It provides a mathematical framework for modeling decision making in a stochastic environment [43, 44]. It can be described formally with four components:

- **State Space $\mathcal{S}$:** A state is a combination of the most relevant information needed to characterize a particular situation of the system. All these possible situations form the State Space $\mathcal{S}$. In this thesis, we restrict to the case of discrete and finite $\mathcal{S}$.
- **Action Space $\mathcal{U}$:** Actions are used by decision maker to control the system. All the possible actions that the agent can perform form the Action Space $\mathcal{U}$. In this thesis, we restrict to the case of discrete and finite $\mathcal{U}$.
- **Transition Probabilities $p(s'|s,u)$:** The probability of transition is a function $\mathcal{S}\times\mathcal{U}\times\mathcal{S} \to [0,1]$ describing the probability of ending up in the state s' after taking action u in state s.
- **Cost $c(s,u)$:** It is the immediate outcome that the system gets after performing action u in state s.

The MDP is also known as *controlled Markov chain* in the sense that instead of having a process that transitions by itself, we now have a decision-making agent moving through the process. As shown in Figure 1.11, the agent is interacting with the environment by taking an action u, and then observe the next state s and the encountered cost c.

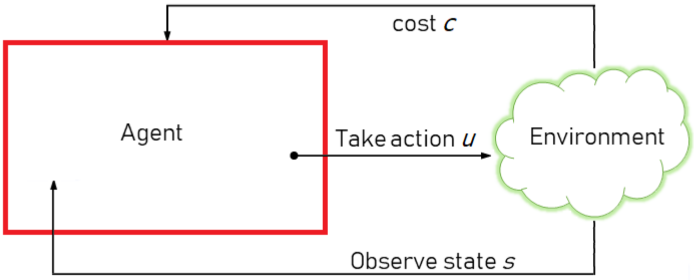


Figure 1.11 – MDP agent.

In MDP, the agent's objective is to minimize the total amount of the cost he receives. In other words, it aims at minimizing the expected long-term cumulative cost, rather than the immediate one, considering a:

- **Finite Horizon (T):** where the agent focuses on the sum of the costs up to stage T. This approach makes sense in applications where there is a natural notion of final time steps, i.e. when the agent-environment interaction naturally breaks down into sub-sequences, such as winning/losing a game and then restarting.
- **Infinite Horizon:** where the agent takes the long-run cost into account ($T \to \infty$). In other words, the agent–environment interaction does not break down naturally and continues without limit, such as on-going process-control task.

In its simplest form, the cumulative cost can be expressed as the **total cost** [45], i.e. the sum of the incurred cost in each time step t:

$$C = \mathbb{E}\Big[\sum_{t=0}^{T-1} c(s_t, u_t)\Big] \tag{1.2}$$

The cost function in Eq. (1.2) is straightforward when considering a finite horizon, but becomes problematic with infinite horizon problems, because the final time step would be $T = \infty$, and the total cost to be minimized, could easily be infinite as well [46]. For this reason, an alternative definition of cost is proposed for this type of problem, namely the **discounted cost**:

$$C = \lim_{T\to+\infty} \mathbb{E}\Big[\sum_{t=0}^{T-1} \gamma^t c(s_t, u_t)\Big] \qquad \text{where} \quad \gamma \in (0, 1) \tag{1.3}$$

The cost function in Eq. (1.3) is slightly more complex conceptually than the first definition (Eq. (1.2)) but much simpler mathematically for MDP. γ is the discount factor that determines the present value of the future cost. If $\gamma < 1$, the infinite sum in Eq. (1.3) has a finite value as long as the cost sequence $c(s, u)$ is bounded. If $\gamma = 0$, the agent is *myopic* and is only looking at maximizing the immediate cost. As $\gamma \to 1$, the objective function takes future costs into account more strongly. However, discounting is sometimes inappropriate and there is no natural cost-free termination state. In such situations, it is often meaningful to optimize the **average cost** defined as:

$$C = \lim_{T\to+\infty} \frac{1}{T}\mathbb{E}\Big[\sum_{t=0}^{T-1} c(s_t, u_t)\Big] \tag{1.4}$$

In this work, all our problems fall within the infinite horizon average cost formulation, since the problem keeps running indefinitely and both present and future costs have the same impact on the system's performance.

The goal of an MDP is to find an optimal policy $\mu^\star$, which is the decision-maker of the MDP, mapping the state space $\mathcal{S}$ to the action space $\mathcal{U}$. In this work, we are interested in **stationary** policies, in which the action u does not depend on the time at which the decision is made. Within this class of policies, we can identify [47]:

- **Randomized Policy:** It assigns the probability $p(u|s)$ to each action u in every state s.
- **Deterministic Policy:** This is one-to-one mapping from the state space $\mathcal{S}$ to the action space $\mathcal{U}$, performing a unique action u whenever a state s is visited.

Policies can also be classified according to how they are calculated:

- **Offline Policy:** When the system states and transitions are fully-known, the optimization problem can be solved once before running system. Then, this policy can be deployed in the system as long as the input parameters remains the same.
- **Online Policy:** When some information (transitions for example) is not available, the optimization problem can be solved by adapting the policy while the system is running and interacting with the environment.

According to [48], an optimal deterministic policy is guaranteed to exist and achieve the minimum average cost for an MDP if the underlying Markov chain is irreducible and aperiodic.

Therefore, to solve the MDP problem and find the optimal **offline deterministic** policy $\mu^\star$, one can resort to Dynamic Programming (DP) techniques. Most of DP algorithms rely on the estimation of *value functions*, i.e. state functions that evaluate how good it is for the agent to be in a given state. The concept of "quality" is defined here in terms of expected cost with respect to a particular policy. Therefore, the value function $v(s)$ of a state s under a policy μ is the expected cost when starting in s and following μ and is defined as follows

$$v_\mu(s) = \mathbb{E}_\mu[C|s_t = s] \tag{1.5}$$

where $\mathbb{E}$ is the expectation with respect to the policy μ and C refers to the discounted cost defined in Eq. (1.3) or the average cost in Eq. (1.4). The value function associated with the optimal deterministic policy $\mu^\star$ satisfies the following so-called Bellman's equation

$$v_{\mu^\star}(s) = \min_{u\in\mathcal{U}}\left[c(s,u) + \gamma\sum_{s'\in\mathcal{S}} p(s'|s,u).v_{\mu^\star}(s')\right] \tag{1.6}$$

Eq. (1.6) considers the discounted cost through γ. However, for average cost problems, the Bellman equation is slightly different [49] and writes as follows

$$v_{\mu^\star}(s) = \min_{u\in\mathcal{U}}\left[c(s,u) - \beta_{\mu^\star} + \sum_{s'\in\mathcal{S}} p(s'|s,u).v_{\mu^\star}(s')\right] \tag{1.7}$$

where $\beta_{\mu^\star}$ corresponds to the optimal average cost.

The optimal policy $\mu^\star$ is the one that achieves the lowest cost in the long term, which is why it is the action leading to lowest value function for all the states of the system, i.e.

$$\mu^\star(s) = \arg\min_{u\in\mathcal{U}}\Big[c(s,u) + \gamma\sum_{s'\in\mathcal{S}} p(s'|s,u).v_{\mu^\star}(s')\Big] \tag{1.8}$$

in the discounted case, and

$$\mu^\star(s) = \arg\min_{u\in\mathcal{U}}\Big[c(s,u) - \beta_{\mu^\star} + \sum_{s'\in\mathcal{S}} p(s'|s,u).v_{\mu^\star}(s')\Big] \tag{1.9}$$

In the following, two algorithms to compute the optimal offline deterministic policy will be described when **only average cost case** is considered.

1.3.1.1 Value Iteration

The first method known as Value Iteration (VI), given in Algorithm 1 [49], consists in finding the optimal value function $v(s)$ that defines the optimal cost-to-go from any state $s \in \mathcal{S}$ from the Bellman's equation given in Eq. (1.7). Belman's equation corresponds to a fixed-point solution of the operator $v \mapsto T_\beta(v) = \min_{u\in\mathcal{U}}\Big[c(s,u) - \beta + \sum_{s'\in\mathcal{S}} p(s'|s,u).v(s')\Big]$. So a fixed-point-like algorithm will be employed. This algorithm is iterative and we denoted these iterations by n. At iteration 0, we start by arbitrarily selecting the cost function $v^0(s)$ and then successively generates, based on Eq. (1.7), the corresponding optimal n-iteration costs using the following recursion until convergence

$$v^n(s) = \min_{u\in U}\Big[c(s,u) - \beta^{n-1} + \sum_{s'\in S} p(s'|s,u).v^{n-1}(s')\Big] \tag{1.10}$$

To avoid the divergence of some components in $v^n(s)$, β^{n-1} is chosen equal to the value of an arbitrary state s_0, i.e., $\beta^{n-1} = v^{n-1}(s_0)$. Thus, by simply setting $V^n(s) = v^n(s) - v^n(s_0)$, i.e. by subtracting the same constant from the components of $v^n(s)$, the difference remains bounded and the convergence of the algorithm is guaranteed. Finally, Eq. (1.10) becomes

$$v^n(s) = \min_{u\in U}\Big[c(s,u) + \sum_{s'\in S} p(s'|s,u).V^{n-1}(s')\Big] \tag{1.11}$$

The different steps of VI is summarized in Algorithm 1.

1.3.1.2 Policy Iteration

The second method known as Policy Iteration (PI), given in Algorithm 2 [49], consists in finding the optimal policy according to the following principle: given a policy, the Bellman's equation is solved with respect to the value function. Once we have this value

Algorithm 1 VI algorithm

```
1: Initialization
   Set $v^0(s) = 0\ \forall s \in S$
   Choose an arbitrary state $s_0$
   Fix a tolerance parameter $\epsilon > 0$
   Set $n = 1$
2: For each $s \in S$ compute
   $$v^n(s) = \min_{u \in U}\left[c(s,u) + \sum_{s' \in S} p(s'|s,u).V^{n-1}(s')\right]$$
   $$V^n(s) = v^n(s) - v^n(s_0)$$
3: If $sp(V^n - V^{n-1}) \leq \epsilon$, where $sp(V) = \max_{s \in \mathcal{S}} V(s) - \min_{s \in \mathcal{S}} V(s)$, let
   $$\mu^n(s) = \arg\min_{u \in U}\left[c(s,u) + \sum_{s' \in S} p(s'|s,u).V^{n-1}(s')\right]$$
   be the resulting policy and stop; else set $n = n + 1$ and go to step 2.
```

function, we update the policy, and so on. Consequently, the PI can be split into two steps:

1. **Policy Evaluation:** Using the current deterministic policy μ, compute the average cost and value function β and $\mathbf{v}$ from Eq. (1.7) by solving the following system of equations at iteration n

$$\begin{cases} \beta^{n-1}\mathbf{1} + (\mathbf{Id} - \mathbf{P})\mathbf{v}^{n-1} = \mathbf{c}^{n-1} \\ \sum_{s \in \mathcal{S}} v^{n-1}(s) = 0 \end{cases} \tag{1.12}$$

where

- $\mathbf{1}$ is the column-vector composed of $|\mathcal{S}|$ ones,
- $\mathbf{Id}$ is the identity with appropriate size,
- $\mathbf{P}$ is the transition matrix between states under policy μ^{n-1}. This matrix is of size $|\mathcal{S}| \times |\mathcal{S}|$.
- $\mathbf{v}^{n-1}$ is a vector whose the s-th component corresponding to $v^{n-1}(s)$ with the state $s \in \mathcal{S}$,
- and $\mathbf{c}^{n-1}$ is the cost vector whose the s-th component is the cost for the state $s \in \mathcal{S}$ and action $\mu^{n-1}(s)$.

Eq. (1.12) is a system of $|\mathcal{S}| + 1$ linear equations with $|\mathcal{S}| + 1$ unknowns.

2. **Policy Improvement:** Obtain an improved policy μ^n by computing an optimal action for each state s using

$$\mu^n(s) = \arg\min_{u \in \mathcal{U}} \left[c(s,u) + \sum_{s' \in \mathcal{S}} p(s'|s,u) v^{n-1}(s') \right] \tag{1.13}$$

The different steps of PI is summarized in Algorithm 2.

Algorithm 2 PI algorithm

```
1: Select arbitrary policy $\mu^0$ and fix a tolerance parameter $\epsilon > 0$
   Let $\beta^0 = 0$ and set $n = 1$
2: Given the policy $\mu^{n-1}$
   Compute the transition matrix $\mathbf{P}$ of size $|\mathcal{S}| \times |\mathcal{S}|$ when policy $\mu^{n-1}$ is applied
   Compute the cost vector $\mathbf{c}^{n-1}$ of length $|\mathcal{S}|$ for each state $s$ in $\mathcal{S}$ with action $\mu^{n-1}(s)$ .
3: Let a scalar $\beta^{n-1}$ and a vector $\mathbf{v}^{n-1}$ of length $|\mathcal{S}|$ be the solutions of
       $\beta^{n-1}\mathbf{1} + (\mathbf{Id} - \mathbf{P})\mathbf{v}^{n-1} = \mathbf{c}^{n-1}$
       $\sum_{\mathbf{s} \in \mathcal{S}} v^{n-1}(s) = 0$
4: Find a policy $\mu^n$ by computing an optimal action for each state $s$ using
       $\mu^n(s) = \arg\min_{u \in \mathcal{U}} \left[ c(s,u) + \sum_{s' \in \mathcal{S}} p(s'|s,u) v^{n-1}(s') \right]$
5: If $\mu^n(s) = \mu^{n-1}(s)$ for any state $s$ or $|\beta^n - \beta^{n-1}| < \epsilon$, then $\mu^n$ is the optimal policy estimate
   and stop; else $n = n + 1$ and go to step 2.
```

Roughly speaking, the VI and PI have similar complexities. There so is no strong argument to use one algorithm instead of the other one.

1.3.2 Constrained Markov Decision Process

Constrained Markov Decision Process (CMDP) is similar to an MDP with the difference that the admissible policies are those verifying additional cost constraints. An example of CMDP is given in the following

$$\begin{aligned} \mu^\star = \arg\min_{\mu} \quad & \underbrace{\lim_{T\to+\infty} \frac{1}{T}\mathbb{E}_\mu\Big[\sum_{t=0}^{T-1} c(s_t,u_t)\Big]}_{C^\mu} \\ \text{s.t.} \quad & \underbrace{\lim_{T\to+\infty} \frac{1}{T}\mathbb{E}_\mu\Big[\sum_{t=0}^{T-1} f(s_t,u_t)\Big]}_{F^\mu} \leqslant \epsilon \end{aligned} \tag{1.14}$$

where C^μ is the cost function to minimize, F^μ is the constraint function that the optimal policy $\mu^\star$ should respect and ϵ is a constant.

According to [50], deterministic policies are no longer guaranteed to be optimal for constrained problems. Therefore, the previous algorithms cannot be used to solve this type of decision process. Actually, the optimal policy is a **stationary random policy**. It can be obtained by solving a Linear Programming (LP) whose is looking for a probabilistic mapping from the state space $\mathcal{S}$ to the action space $\mathcal{U}$.

The optimal policy is exhibited through the so-called occupation measure defined as a probability measure over the set of state-action pairs and denoted by $\rho(s,u)$. The objective function as well as the constraints given in Eq. (1.14) can be expressed with respect to $\rho(s,u)$ as follows

$$\rho^\star = \arg\min_{\rho} \sum_{\mathbf{s}\in\mathcal{S},u\in\mathcal{U}(\mathbf{s})} \rho(\mathbf{s},u)c(s,u) \tag{1.15}$$

$$\text{s.t.} \quad \sum_{\mathbf{s}\in\mathcal{S},u\in\mathcal{U}(\mathbf{s})} \rho(\mathbf{s},u)f(s,u) \leqslant \epsilon \tag{1.16}$$

$$\sum_{\mathbf{s}\in\mathcal{S},u\in\mathcal{U}(\mathbf{s})} \rho(\mathbf{s},u) = 1 \tag{1.17}$$

$$\sum_{\mathbf{s}'\in\mathcal{S},u\in\mathcal{U}(\mathbf{s})} \rho(\mathbf{s}',u)p(\mathbf{s}|\mathbf{s}',u) = 1, \ \forall \mathbf{s}\in\mathcal{S} \tag{1.18}$$

where $\mathcal{U}(s)$ is the set of possible actions when the system is in state s. Eq. (1.17) comes from the fact that $\rho(s,u)$ is a probability measure over the set of state-action pairs, and Eq. (1.18) from the Markov property of the process (s,u).

The optimal stationary random policy $\mu^\star$ is then obtained from the optimal $\rho^\star$ according to the following equation

$$\mu^\star(u|s) = \frac{\rho^\star(s,u)}{\sum_{u'\in\mathcal{U}(s)}\rho^\star(s,u')} \tag{1.19}$$

whenever the denominator is non-zero. When it is zero, $\mu^\star(.|s)$ is chosen to be an arbitrary probability measure over $\mathcal{U}(s)$ [47]. Such states are known to be transient states and the taken actions do not affect the system in the long-term. Moreover the optimal policy for a CMDP with K constraints requires at most K randomizations [47]. In other words, the agent can choose between a maximum of $K + 1$ actions in each state.

1.4 Conclusion

In this chapter, we have started by presenting an overview of the different EH sources, and we have chosen among them those that best fits our model. Then, we have described the different offloading approaches and have chosen the one that was best adapted to our work. Finally, we have shown the mathematical optimization tools that will be used in the following chapters to solve the problems encountered in this thesis.

Chapter 2

Joint Resource Scheduling and Computation Offloading with perfect CSIT

2.1 Introduction

Nowadays, the unprecedented growth of mobile communications driven by the huge number of connected devices and new mobile applications is significantly increasing the demands for high-volume delay-sensitive data traffic, requiring thus intensive computation and leading to high energy consumption. Therefore, to overcome the limitations of mobile terminals in terms of processing capacity, storage and energy, EH [32, 51] and MEC [37, 52] have been recently proposed.

On one hand, EH from surrounding environments allows to prolong the operational life of mobile devices by leveraging alternative renewable energy sources. Resource scheduling with EH-enabled devices has been also widely studied during the past decade. In [53], the tradeoff between energy overflow and energy shortage was addressed by maximizing the scheduling throughput for capacity-limited EH system. Online algorithms were proposed to solve this problem for static and fading channels based on a new estimation method of future energy arrivals without any prior information. Both offline and online algorithms were also provided in [54] to maximize the throughput in finite-horizon scheduling with EH transmitter. The offline solution is expressed in terms of water levels and the online solution minimizes successively the expected throughput losses with respect to the offline optimal decision. Finite-horizon optimization problem was also considered in [55] to minimize the outage probability in an EH system. A low complexity fixed threshold transmission is proposed based on the offline mixed integer LP solution. Moreover, [56] minimized the weighted packet loss rate under an average delay constraint in wireless sensor networks. The constrained MDP was formulated with linear VI approximation that locally determines the energy allocation at every EH wireless node using multilevel water-filling. Near-optimal control policy was derived by applying stochastic online learning based on post-decision state framework. In [57], MDP modeling and online post-decision learning approach were also derived to maximize the data arrival rate at the transmitter queue under delay and energy constraints.

On the one hand, MEC enables offloading computation tasks from mobile devices to nearby BSs with more energy and computation resources. In recent years, MEC has attracted considerable attention [58], and particularly optimizing transmission strategies [59–64]. In [59], a dynamic offloading algorithm based on Lyapunov optimization is proposed to determine which software components to offload depending on wireless conditions. In [60], the energy consumption of the mobile device was minimized by jointly configuring the clock frequency for the mobile execution, and varying the transmission rate depending on the channel conditions for the cloud execution. The problem of energy consumption minimization is also studied in [61] by jointly optimizing the radio resource scheduling and the computation offloading under average delay constraints using DP techniques. In [62], a similar problem was considered to exploit the tradeoff between energy consumption and latency. In [63], a delay-optimal task scheduling policy for single-user systems was developed using MDP formulation. In [64], the trade-off between the energy consumption and the execution latency of performing mobile tasks was investigated by jointly considering the computation offloading and communication resource allocation for single and multi-cell MEC network scenarios.

To take advantage of both technologies, recent studies have integrated EH capabilities into MEC systems. This recent field opens new opportunities to improve the performance of mobile devices but also brings new challenges in designing optimal and efficient policies, taking into account both radio and computation resources under EH constraints. In parallel to our work, a study on EH-MEC systems proposed a dynamic computation offloading policy for mobile devices using Lyapunov optimization techniques [65]. In [66], the resource management problem is also considered for EH-MEC but from the server side. In [67], Reinforcement Learning (RL) techniques are used to find the optimal offloading policy without requiring the knowledge of the MEC model, the local computation and energy consumption models.

Compared to previous works employing an average delay constraint, we force our system to respect a strict delay constraint, i.e. the age of the packets must not exceed the deadline. This constraint was initially introduced in [C1] that will be presented as a special case in Section 2.5.2.

The *age* of each packet here is different than the famous Age of Information (AoI). The

AoI concept was first introduced back in 2011 to evaluate the freshness of the information that is currently available on the status of a remote system [68]. More precisely, AoI is the time elapsed since the generation of the last successfully received message containing update information on its source system [69]. Based on this idea, a packet can be in the buffer but its content is not valid anymore since its AoI metric is abused. This definition is totally different from ours with regard to the strict delay. In other words, the content of each packet in the buffer remains valid as long as the deadline is not violated.

In this Chapter, we address joint resource scheduling and computation offloading for a single EH mobile user served by a BS with perfect CSIT. Then we minimize the number of discarded packets due to delay violation and buffer overflow, assuming random data and energy future arrivals, as well as time-varying channel. The problem is formulated as an MDP and solved using PI algorithm. We compare the proposed policy with different policies performing immediate scheduling, and two additional ones executing either only local processing or only offloading decisions. We also investigate the special case of scheduling resources for:

- an EH-transmitter (we remove offloading capabilities). We formulate the problem of minimizing the number of discarded packet as an MDP and solve it using VI algorithm. We compare the proposed policy with different policies performing immediate scheduling.
- a conventional transmitter (we remove EH and offloading capabilities). We formulate the problem of minimizing the power consumption as a CMDP and solve it using LP techniques. We compare the proposed policy with one performing immediate scheduling.

The remainder of this Chapter is organized as follows. In Section 2.2, we describe the system model. In Section 2.3, we formulate the MDP problem by defining its states, actions and transitions probabilities and we propose a PI algorithm to solve it. We provide and analyze numerical results in Section 2.4. In Section 2.5, we present two special cases as the initial steps of the general problem. Finally, concluding remarks are drawn in Section 2.6.

2.2 System Model

We consider in Figure 2.1 a MEC scenario where an EH mobile user is wirelessly connected to a BS endowed with some cloud resources. The mobile user stores the harvested energy from an external source in a limited-capacity battery and the data packets arriving from the upper layer in a finite buffer. The communication is slotted into consecutive epochs of equal duration T_s. Type of processing (locally or remotely) as well as the number of packets to be processed are determined at the beginning of each time slot, depending on the current channel state and the previous data and energy arrivals.

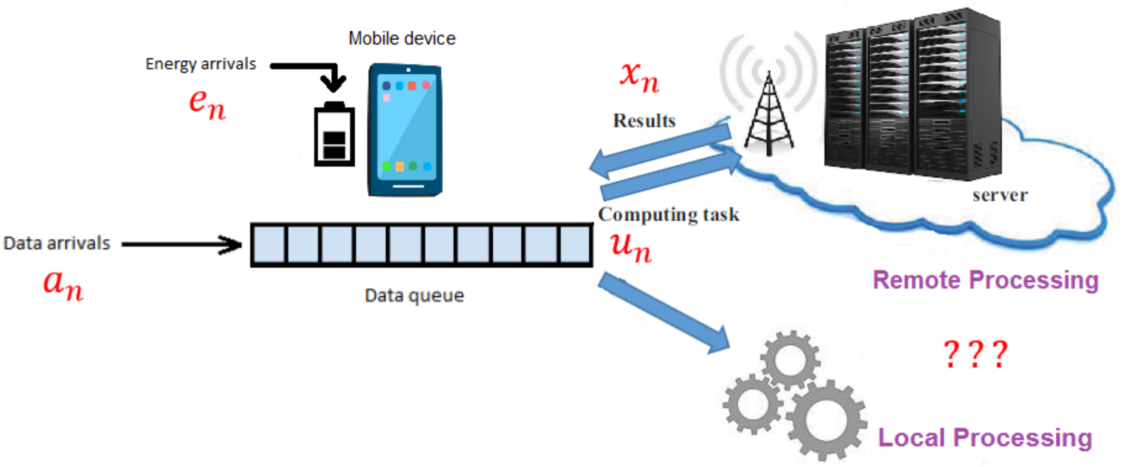


Figure 2.1 – MEC system with an EH mobile device.

In the following, we describe the energy, data and channel models, as well as the different types of scheduling decisions and their corresponding consumed energy.

2.2.1 Energy model

We assume that the energy arrives in multiple packets of energy units (e.u) of $\mathcal{E}_U$ Joules (J) [1]. The received energy is stored in a battery of limited capacity B_e, and is lost when it exceeds B_e. Then, the energy arrival process, counting as the number of e.u, is modeled as an independent identically distributed (i.i.d.) Poisson distributed process with an average arrival rate λ_e. Thus, let e_n denote the harvested energy at the beginning of time slot n. Its probability distribution is given by

$$p(e_n = e) = e^{-\lambda_e}.\frac{(\lambda_e)^e}{e!}. \tag{2.1}$$

Let b_n denote the energy level of the battery at the beginning of time slot n after receiving the harvested energy e_n. We have $b_n \in \{0, ..., B_e\}$. Let E_n be the energy consumed to execute packets during time slot n. By construction, $E_n \leqslant b_n$ $\forall n$. In addition, we suppose causal Energy State Information at the Transmitter (ESIT), i.e., b_n is known when scheduling at time slot n.

2.2.2 Data queue model and strict delay constraint

The mobile user receives data packets and stores them in a buffer of size B_d packets. We model the data arrival process as an i.i.d. Poisson distributed process with an average arrival rate λ_d. We assume that all packets are of the same size L bits. At the beginning of

[1] There is a huge amount of literature assuming i.i.d EH processes. We adopted this approach for sake of clarity. Nevertheless, this work can be easily extended to time-correlated EH processes. This will be presented in Section 2.4 to plot Figure 2.12.

time slot n, let a_n denote the number of received packets with the following probability distribution

$$p(a_n = a) = e^{-\lambda_d}.\frac{(\lambda_d)^a}{a!}. \tag{2.2}$$

A packet is discarded from the buffer if there is a

- **buffer overflow**, i.e., new packets are discarded if there is no space in the buffer to store them;
- **delay violation**, i.e., packets are discarded if they are stored in the queue more than the maximal delay K_0.

To describe the buffer configuration, the age of the packets, i.e., the time spent by the packets within the buffer is necessary. Therefore, we introduce the notation $k_i(n)$ corresponding to the age of the i-th packet at time n in the buffer. By definition, we have $k_i(n) \in \{-1, ..., K_0\}, \forall i, n$ and $k_i(n) = -1$ stands for an empty space in the buffer (i.e., when the i-th packet does not exist). In Figure 2.2, we provide a buffer state at time n, where q_n is the queue length in the buffer. Notice that $k_j(n) \leq k_i(n), \forall i \leqslant j$.

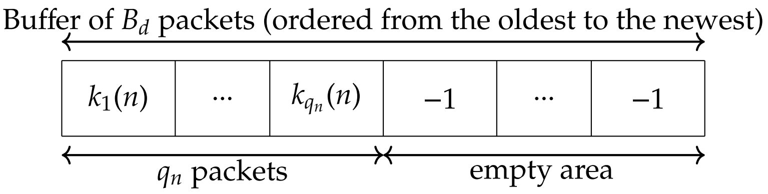


Figure 2.2 – Buffer configuration at time slot n.

2.2.3 Channel model

We consider the communication channel between the mobile device and its BS as a flat-fading channel with bandwidth W_{DL} (Hz) in the downlink and W_{UL} (Hz) in the uplink. The additive white Gaussian noise has power spectral density N_0. During time slot n, the channel response remains constant with complex-valued amplitude h_n, and varies independently along time slots [2]. We define the Rayleigh fading channel h_n and its gain by $g_n = |h_n|^2$. Then g_n is a continuous random variable distributed exponentially with probability density function $p(g) = \frac{1}{\xi}e^{-\frac{g}{\xi}}$ with mean ξ. For the sake of simplicity, we assume only quantized channel state $x_n = Q_g(g_n)$, where $Q_g(.)$ represents the quantization process. This assumption is generally justified in practice to account for the capacity-limited feedback and has also been adopted in the literature [70–77]. Fixing a sequence

[2] There is a lot of work involving i.i.d. channels. We have selected this approach for simplicity. However, this study can be easily generalized to time-correlated channels. In fact, this model will be adopted in Chapter 4.

of power quantization thresholds, the channel gain x_n is then a discrete variable taking values from a finite channel state space $\mathcal{X}$.

In order to define the discrete channel states, let M be the number of quantization levels, $\{t_m\}_{m=0}^{M-1}$ the set of thresholds and $\{\mathcal{L}_m\}_{m=0}^{M-1}$ the set of quantization levels for Q_g. The quantization regions of the channels are then given by the intervals $I_m = [t_m, t_{m+1}[$ with t_0 is fixed such that the transmission of 1 packet using 1 e.u. is guaranteed and $t_M = +\infty$. In our model, we consider a *Uniform* quantizer. So, let $t_{\max} = t_{M-1}$ be the maximal threshold such that the transmitter can send U_0 packets using B_e e.u., where U_o is the maximal number of offloaded packets. By applying $E_o(x_n, u_n) = B_e$ and $u_n = U_o$ in Equations (2.4) and (2.5) with equality (equations (2.4) and (2.5) are defined in next section 2.2.4), we obtain the corresponding value for x_n which is forced to $t_{\max}$. The uniform quantization thresholds are given by $t_{m+1} = t_m + \delta$ with $m = 0 \dots M-2$ and $\delta = \frac{t_{\max}}{M-1}$. We select the quantization levels as the lower bound of the regions, which is the worst case scenario. Thus, $\mathcal{L}_m = t_m$ for $m = 0 \dots M-1$ and a channel is said to be in state $x_n = \mathcal{L}_m$ if $g_n \in I_m = [t_m, t_{m+1}[$.

We also assume causal CSIT, i.e. x_n is perfectly known without errors at the beginning of slot n. This assumption is relaxed in Chapter 3 where we suppose that the system relies on an estimated version (perhaps erroneous) of the channel, i.e. an estimation of x_n is only known when taking the decision at time slot n, and time has been used for doing this estimation at the expense of the communication time. A more complicated problem is investigated in Chapter 4 where we consider that the system operates with an outdated CSI, i.e. x_n is totally unknown at the decision instant n, but in that case, a correlation over time for the channel is assumed.

2.2.4 Execution decisions and related energy consumption

At the beginning of time slot n, three scheduling decisions are possible:

- **Local processing:** The mobile device uses its own processor to execute u packets from the buffer ($u \leqslant q_n$) during time slot n. We assume that processing one packet locally consumes a power P_ℓ. Then the consumed energy, expressed as an integer multiple of the e.u, is given by
$$E_\ell(u) = \left\lceil u.P_\ell.\frac{T_s}{\mathcal{E}_U} \right\rceil. \tag{2.3}$$
- **Remote processing:** The mobile device transmits u packets to be executed at its BS and then receives the results. The mobile device thus consumes energy to send data, to wait for the remote processing and to receive the result. The energy consumption, expressed as an integer multiple of the e.u, is given by
$$E_o(x_n, u) = \left\lceil \frac{u}{\mathcal{E}_U}\left(\frac{L.P_t}{W_{UL}.\log_2\left(1 + \frac{P_t.x_n}{W_{UL}.N_0}\right)} + T_w.P_w + \frac{L_{DL}.P_r}{W_{DL}.\log_2\left(1 + \frac{P_s.x_n}{W_{DL}.N_0}\right)} \right) \right\rceil \tag{2.4}$$

where L_{DL} is the size in bits of the computation result, P_t is the transmission power of the mobile device, P_r is the power consumed by the mobile device to receive the result, and P_s is the power used by the BS to send the result. While waiting for the packets processing, the transmitter consumes a power P_w. Finally, T_w is the time for the BS to execute one packet. In addition, we require that this offloading procedure lasts at most one time slot leading to the following constraint

$$u\left(\frac{L}{W_{UL}.\log_2\left(1+\frac{P_t.x_n}{W_{UL}.N_0}\right)}+T_w+\frac{L_{DL}}{W_{DL}.\log_2\left(1+\frac{P_s.x_n}{W_{DL}.N_0}\right)}\right)\leqslant T_s. \tag{2.5}$$

Notice that W_{DL}, W_{UL}, N_0, L_{DL}, P_s, P_r and T_w are pre-fixed parameters. Forcing equality in Eq. (2.5) enables us to find P_t which so depends on u and the channel realization x_n and is time-varying along with the time slots.

- **Idle:** The mobile device does not execute any packet and decides to wait for the next time slot.For sake of simplicity, we assume that the device electronic circuitry is sleeping and so the consumed energy is given by

$$E_I = 0. \tag{2.6}$$

Note that only one of these three decisions is possible in each slot. In other words, the system cannot choose to execute part of its packets locally and the other part remotely in the same slot.

2.3 Problem Formulation and Resolution

Now, we aim at finding an optimal policy μ that minimizes the number of discarded packets due to buffer overflow and delay violation. The policy μ is a sequence of actions that specify the processing decisions (local processing, offloading or staying idle) and the number of packets **u** to be scheduled at each time slot, based on the past system states and actions. In this section, we characterize the appropriate states and actions and show that our problem can be formulated as an MDP. We define then the transition matrix and the reward of this MDP and propose to apply an offline PI algorithm to solve it.

2.3.1 State Space

The state space $\mathcal{S}$ is the set of $\mathbf{s} = (\mathbf{k}, b, x)$ where

- $\mathbf{k} = [k_1, \cdots, k_{B_d}]$ is the vector indicating the age of each packet in the data buffer,
- b is the battery level, and
- x is the channel gain.

Due to the strict delay constraint, we describe the data buffer states using **k** instead of the queue length q used in the state-of-the-art [56, 61, 65]. Actually, q can be extracted from **k**

$$q_n = \max\{i \mid k_i(n) \geqslant 0\}. \tag{2.7}$$

The state space $\mathcal{S}$ is finite, and the total number of possible states $|\mathcal{S}|$ is upper-bounded by $(K_0+2)^{B_d}.|B_e+1|.|\mathcal{X}|$. By assuming that the packets are queued in an increasing order of their age in the buffer, i.e. $k_1(n) \geq k_2(n) \geq \cdots \geq k_{q_n}(n)$, we can significantly reduce the state space by removing all impossible combinations of components in **k**. For instance, with $B_d = 8$, $K_0 = 3$, $B_e = 4$ and $|\mathcal{X}| = 5$, our system has only $12,375$ states compared to the upper-bound of $9,765,625$.

2.3.2 Action Space

The action space $\mathcal{V}$ denotes the processing decisions (local processing, offloading or staying idle) and the number of packets u that the mobile device can schedule during a slot. Let U_ℓ be the maximum number of packets that can be processed locally during a time slot and U_o be the maximum number of packets that can be offloaded during a time slot. U_o is a pre-fixed parameter constrained by the buffer size and U_ℓ is limited by the capacity of the internal processor of the mobile device, thus usually $U_\ell \leq U_o$. Finally, the action space is finite with cardinality $V = |\mathcal{V}| = U_\ell + U_o + 1$. The actions are ordered and the m-th action is as follows:

- if $m = 0$, idle processing is considered and $u = 0$.
- if $m = m_\ell$ with $m_\ell \in \{1, \cdots, U_\ell\}$, local processing is applied and $u = m_\ell$.
- if $m = m_o$ with $m_o \in \{U_\ell + 1, \cdots, V-1\}$, offloading is applied and $u = m_o - U_\ell$.

At time slot n, $v_n \in \{0, , \cdots, V-1\}$ corresponds to the decided action, and u_{v_n} is the number of processed packets, either locally or remotely.

2.3.3 Markov Decision Process

During time slot n, $w_n = \max(u_{v_n}, m_n)$ packets leave the buffer, either executed (locally or remotely) and/or discarded, where m_n is the number of packets reaching the maximal delay (K_0) in the buffer. The age of the remaining packets in the buffer is incremented by 1. Moreover, a_{n+1} new packets arrive to the buffer with age 0. Some of these arrival packets are discarded if there is no room in the buffer. Therefore, the vector **k** can be updated from slot n to slot $n+1$ according to the following rule.

```
for $i = 1$ to $q_n - w_n$ do
    $k_i(n+1) = k_{w_n+i}(n) + 1$
end for
```

```
2: for $i = q_n - w_n + 1$ to $q_n - w_n + a_{n+1}$ do
       $k_i(n+1) = 0$
   end for
3: for $i = q_n - w_n + a_{n+1} + 1$ to $B_d$ do
       $k_i(n+1) = -1$
   end for
```

At the same time, e_{n+1} e.u are harvested and stored in the battery while E_n e.u are used to execute u_{v_n} packets according to Eqs. (2.3), (2.4), or (2.6). Therefore, at time slot $n+1$, the battery state is updated according to

$$b_{n+1} = \min\{b_n - E_n + e_{n+1}, B_e\} \tag{2.8}$$

We remind that $E_n \leq b_n$, ensuring $b_{n+1} \geq 0$. We remark that $(\mathbf{k}_{n+1}, b_{n+1})$ only depends on previous state $(\mathbf{k}_n, b_n)$, action v_n (which provides E_n since x_n is known), and external perturbation (a_{n+1}, e_{n+1}), confirming that the problem turns to MDP.

2.3.4 Transition Matrix

The state transition probability of the MDP is defined by $p(\mathbf{s}'|\mathbf{s}, v)$ denoting the transition probability to fall in the future state $\mathbf{s}' = (\mathbf{k}', b', x')$ after taking action v in the current state $\mathbf{s} = (\mathbf{k}, b, x)$. Assuming that the buffer, battery and channel states are independent to each other and channel states are not time-correlated, the transition probability satisfies the following equation

$$p(\mathbf{s}'|\mathbf{s}, v) = p(\mathbf{k}'|\mathbf{k}, b, v).p(b'|b, x, v).p(x'), \tag{2.9}$$

where $p(x')$ is the distribution of the channel states, $p(\mathbf{k}'|\mathbf{k}, b, v)$ indicates the probability transitions between buffer states, and $p(b'|b, x, v)$ indicates the probability transitions between battery states.

We first exhibit the set of possible actions $\mathcal{A}(\mathbf{s}) \subset \mathcal{V}$ for each state $\mathbf{s}$. We have $\mathcal{A}(\mathbf{s}) = \mathcal{A}_0(\mathbf{s}) \cap \mathcal{A}_1(\mathbf{s}) \cap \mathcal{A}_2(\mathbf{s})$, where $\mathcal{A}_0(\mathbf{s})$, $\mathcal{A}_1(\mathbf{s})$ and $\mathcal{A}_2(\mathbf{s})$ are defined through their complementary sets as follows:

- the set $\overline{\mathcal{A}_0}(\mathbf{s})$ is composed by the actions of offloading using a transmit power $P_t > P_{\max}$ according to Eq. (2.5);
- the action v belongs to $\overline{\mathcal{A}_1}(\mathbf{s})$ by satisfying at least one of the following conditions

```
1: $u_v > q$ or $k'_i > k_i + 1$ or $q' < q - w$
2: $k'_i \neq k_{i+u_v} + 1$ and $k_{i+u_v} \neq -1$
3: $k'_i > 0$ and $k_{i+u_v} = -1$
4: $q = B_d$ and $u_v \neq 0$ and $k'_i > 0, \forall i \in \{q - w + 1, ..., B_d\}$
```

- the action v belongs to $\overline{\mathcal{A}_2}(\mathbf{s})$ by satisfying at least one of the following conditions

  1: $0 > b - E$
  2: $b' < b - E$

  where E is the energy consumed when the action v is applied.

Secondly, when $v \in \mathcal{A}(\mathbf{s})$, the transitions are as follows

```
if $q' < B_d$ then
    $p(k'_i|k_i, b, v) = e^{-\lambda_d} . \frac{(\lambda_d)^{q'-q+w}}{(q'-q+w)!}$
else
    $p(k'_i|k_i, b, v) = 1 - \mathrm{Q}(B_d - q + w, \lambda_d),$
```

and

```
if $b' < B_e$ then
    $p(b'|b, x, v) = e^{-\lambda_e} . \frac{(\lambda_e)^{b'-b+E}}{(b'-b+E)!}$
else
    $p(b'|b, x, v) = 1 - \mathrm{Q}(B_e - b + E, \lambda_e).$
```

where Q is the regularized Gamma function defined as

$$Q(x, y) = e^{-y} \sum_{i=0}^{x-1} . \frac{y^i}{i!} \tag{2.10}$$

2.3.5 Cost

In this thesis, we focus on infinite horizon MDP problem. We consider thus time-averaged cost, where at a given time slot $n \in \{1, \cdots, N\}$, the system state is denoted by $\mathbf{s}_n = (\mathbf{k}_n, b_n, x_n)$ and $\mu(\mathbf{s}_n) = v_n$ is the action. Our objective is to minimize the average number of discarded packets under policy μ. Hence, we define the cost function as

$$\overline{D}(\mu) = \lim_{N \to +\infty} \frac{1}{N} \mathbb{E}^{\mu} \left[\sum_{n=1}^{N} \Big(\varepsilon_d(\mathbf{s}_n, v_n) + \varepsilon_o(\mathbf{s}_n, v_n) \Big) \right], \tag{2.11}$$

where $\mathbb{E}$ is the expectation with respect to the policy μ and where $\varepsilon_d(\mathbf{s}_n, v_n)$ and $\varepsilon_o(\mathbf{s}_n, v_n)$ are the instantaneous number of discarded packets due to buffer overflow and delay violation, respectively.

At a given slot n, when the system state is $\mathbf{s}_n$ and the performed action is v_n, the number of discarded packets due to delay violation is given by

$$\varepsilon_d(\mathbf{s}_n, v_n) = \begin{cases} 0 & \text{if } m_n = 0 \text{ or } m_n \leqslant u_{v_n} \\ m_n - u_{v_n} & \text{otherwise.} \end{cases} \tag{2.12}$$

The buffer overflow occurs when $q_n - w_n + a_{n+1} > B_d$, thus the number of discarded packets due to buffer overflow is obtained as follows

$$\begin{aligned}\varepsilon_o(\mathbf{s}_n, \nu_n) &= \sum_{a=B_d-q_n+w_n+1}^{+\infty} (q_n - w_n + a - B_d).e^{-\lambda_d}.\frac{(\lambda_d)^a}{a!} \\ &= \lambda_d.(1 - \mathrm{Q}(B_d - q_n + w_n, \lambda_d)) + (q_n - w_n - B_d) \times (1 - \mathrm{Q}(B_d - q_n + w_n + 1, \lambda_d)).\end{aligned} \tag{2.13}$$

We can then state the MDP optimization problem as

$$\mu^\star = \arg\min_{\mu} \overline{D}(\mu) \tag{2.14}$$

2.3.6 Optimal Policy Computation

We propose an offline DP approach to solve this problem using PI algorithm presented in Section 1.3.1. This optimal offline deterministic policy consists in a one-to-one mapping from the state space $\mathcal{S}$ to the action space $\mathcal{V}$, performing a unique action ν whenever a state $\mathbf{s}$ is visited. We remind that it only depends on energy arrival and data arrival *a priori* distributions and current channel states at the mobile device.

2.4 Numerical Results

We evaluate numerically the optimal policy devised for the scheduling-offloading problem. We consider the system described in Section 2.2 and fix its design parameters as: the slot duration is $T_s = 1$ ms. The maximum available power at the transmitter is $P_{\max} = 2$ mW. Energy arrivals are stored as energy units in a battery of size $B_e = 4$ e.u, where $\mathcal{E}_U = 50$ nJ. During a time slot, a maximum of $U_\ell = 2$ packets can be processed locally or a maximum of $U_o = 4$ packets can be offloaded. The mean channel is $\xi = 1$ and the channel state x takes 5 possible values from the finite set $\mathcal{X} = \{-10, -3.98, -0.97, 0.792, 2.04\}$ dB. These channel values are obtained according to Section 2.2.3. The noise power spectral density is $N_0 = -87$ dBm/Hz and the allocated bandwidth is $W_{DL} = 5$ MHz in the downlink and $W_{UL} = 500$ kHz in the uplink. Data packets have equal size $L = 5000$ bits and are stored in a buffer of size $B_d = 6$ packets. The resulting packets from remote processing have also equal size $L_{DL} = 500$ bits. The maximum delay of packets in the buffer is $K_0 = 3$ (i.e., in absolute time $K_0 T_s = 3$ ms). The remaining values are chosen as: $P_\ell = 30\ \mu$W, $P_r = 0.2$ mW, $P_s = 1.6$ kW, $P_w = 0.1$ mW, and $T_w = 0.1$ ms.

Convergence of the long-term average cost

In Figure 2.3, we illustrate the average number of discarded packets for the optimal policy (obtained after the convergence of the PI algorithm) versus N (the horizon on which we average the cost) with various energy arrival rates λ_e and a data arrival rate $\lambda_d = 1.5$. We can notice that it takes only a few hundreds timeslots for the system to achieve the long-term cost. In addition, if a large amount of energy can be harvested, λ_e increases, and the system is able to execute more packets reducing hence considerably the average number of discarded packets.

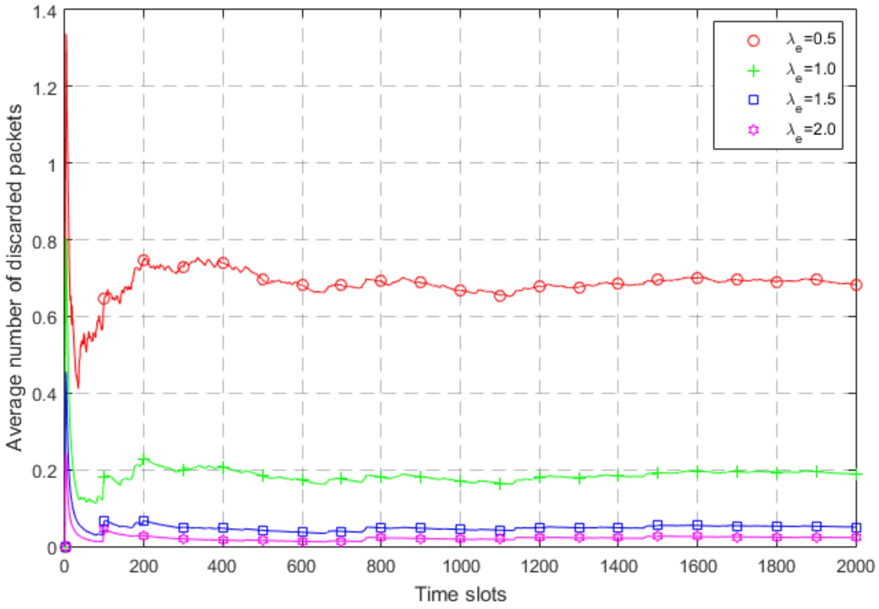


Figure 2.3 – Convergence analysis for the average rate of discarded packets for different energy arrival rates.

Comparison with other policies

In Figure 2.4, we show the percentage of discarded packets versus the data arrival rate λ_d for $\lambda_e = 0.5$ (small) and $\lambda_e = 2.0$ (large) energy arrival rates. We compare the performance of the optimal policy to three different policies:

- **Immediate:** This policy processes, locally or remotely, the maximum number of packets whenever energy is available in the battery.
- **Local:** This policy is obtained by running the offline PI algorithm taking into account only local decisions ($U_o = 0$).
- **Remote:** This policy is obtained by running the offline PI algorithm considering only remote decisions ($U_\ell = 0$).

We can observe that the proposed deterministic offline policy outperforms the other policies as it can adapt its processing to the energy and data arrivals as well as channel conditions. Yet, when the data arrival rate λ_d increases, the number of discarded packets of all the policies increases because the buffer overflow can happen more often. We can also see that for both λ_e values, the **local** policy discards the highest number of packets due to the limited capacity of the mobile device processor. Moreover, when λ_e is small, the **remote** policy is able to sustain more efficiently the system than the **immediate** policy by scheduling more packets depending on the channel states. This situation is reversed when λ_e is large since the scavenged energy is sufficiently available to process the maximum number of packets by the **immediate** policy irrespective of channel conditions.

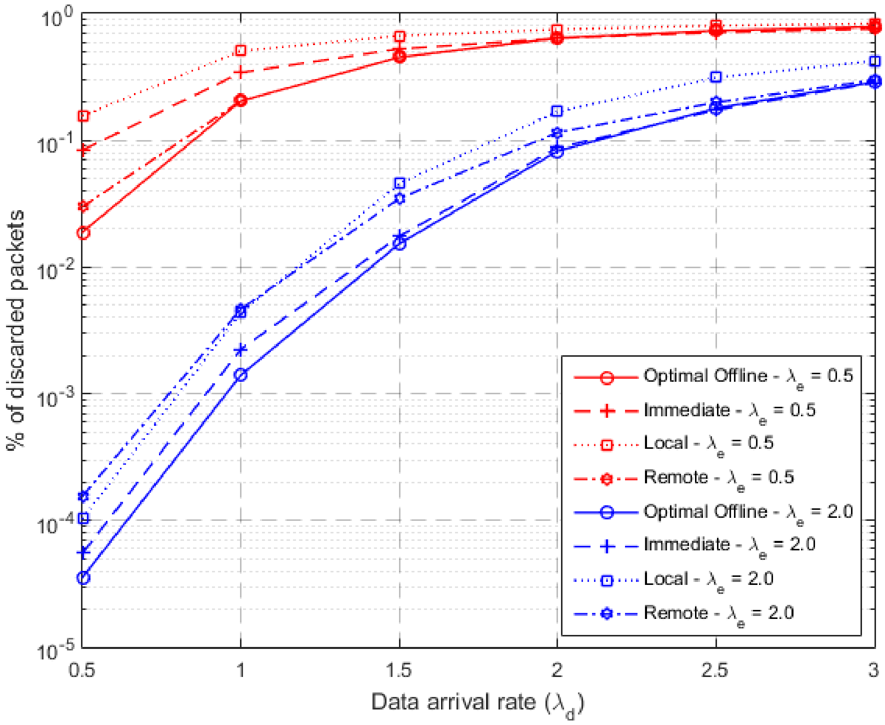


Figure 2.4 – Percentage of the discarded packets versus data arrival rate for energy arrival rates $\lambda_e = 0.5$ and 2.0.

Similar to Fig. 2.4, Fig. 2.5 compares the percentage of discarded packets of the optimal policy with two other variants of the **immediate** policy. Unlike the **immediate** policy that sends the maximum number of packets using the available energy in the battery, the introduced *p*-**immediate** policy restricts the number of packets sent by the **immediate** one by taking the buffer state into account through an additional parameter p in that way:

- **fixed** p: The policy sends only the packet i from the buffer if $k_i \geq p$.
- **variable** p: The policy performs a first step similar to the previous case (fixed p). If no packet satisfies the condition, p is decreased by 1, and the first step is repeated, until $p = 0$.

These two variants are studied to evaluate the influence of the forced number of packets to be executed on the system's performance. The **immediate** policy corresponds to a 0-**immediate** policy. Here, we choose $p = 2$ for the p-**immediate** policy. As we can see, taking only the age of the packets into account without adapting carefully the number of executed packets to the battery level and buffer state leads to decrease the number of sent packets, and therefore the **immediate** policy remains much better.

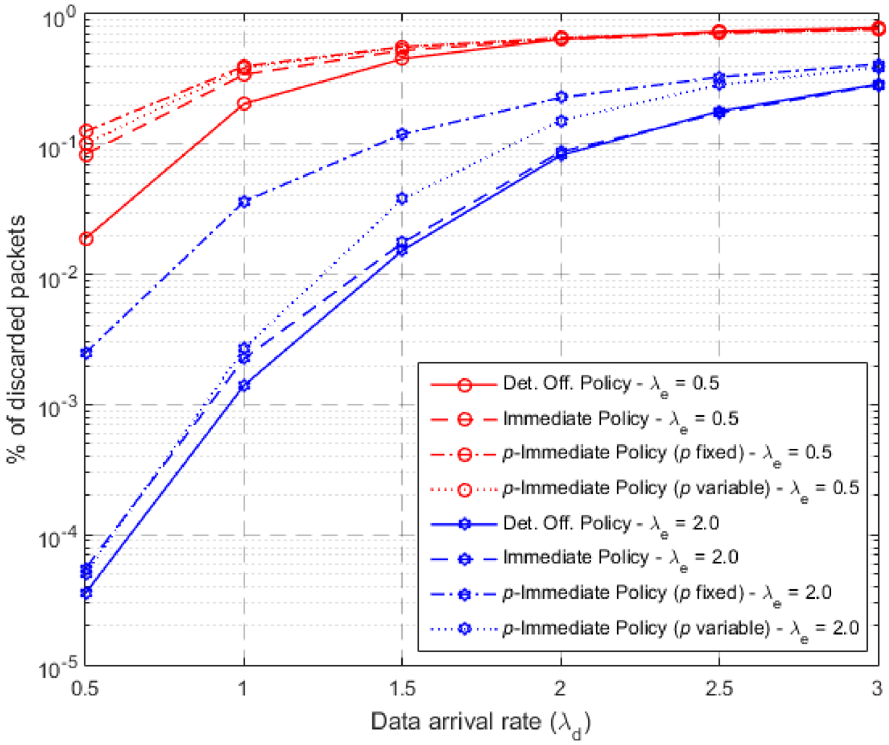


Figure 2.5 – Percentage of the discarded packets versus data arrival rate with different energy arrival rates and different immediate policies.

In Figure 2.6 and Figure 2.7, we plot respectively the average consumed energy and the average battery state versus the data arrival rate λ_d for energy arrival rates $\lambda_e = 0.5$ and 2.0. We can observe that **local** and **immediate** policies experience the highest energy consumption since processing packets locally consume more energy, draining thus the battery level. The optimal proposed policy consumes approximately the same energy amount as the **remote** policy while sending more packets. Indeed, it ensures a better sustainable communication with less number of discarded packets by optimally using the available energy, leading hence to a higher energy level in the battery.

In Figure 2.8, we display the percentage of processing decisions for the optimal and **immediate** policies in Figure 2.8(a) and (c) respectively at $(\lambda_d = 1, \lambda_e = 1)$, and in Figure 2.8

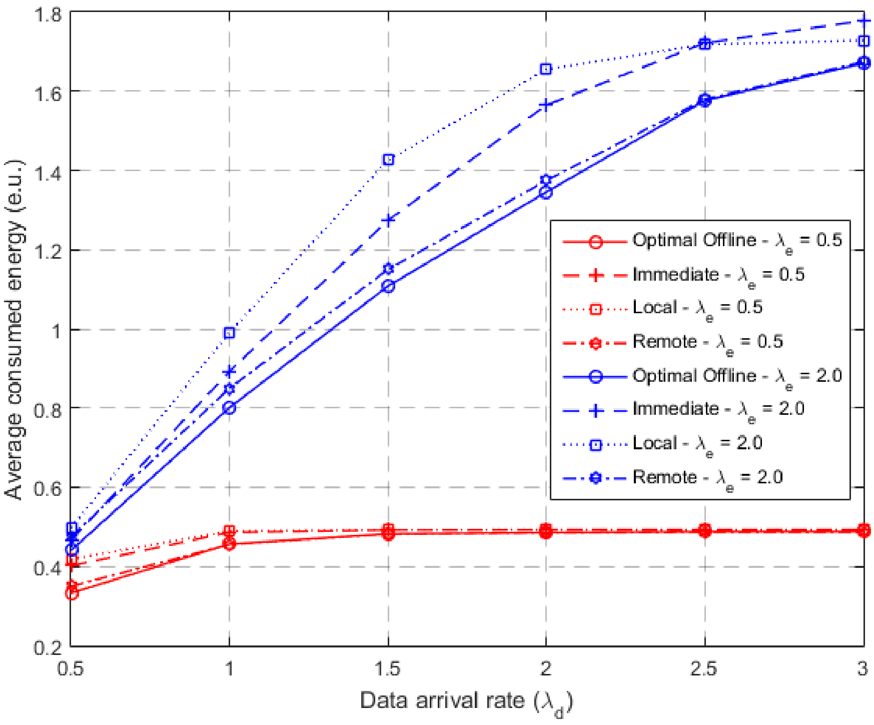


Figure 2.6 – Average consumed energy versus data arrival rate for energy arrival rates $\lambda_e = 0.5$ and 2.0.

(b) and (d) respectively at ($\lambda_d = 2, \lambda_e = 2$). As we can see, when the data arrival rate increases, the system schedules more packets either locally or remotely to minimize the number of discarded packets, which decreases **Idle** mode events. When more energy is available in the battery, the **immediate** policy processes the maximum number of packets, hence can offload more packets regardless of channel states. Therefore, energy shortage can occur more often forcing the system to enter **Idle** mode more than with optimal policy.

Comparison with the state-of-the-art

In Figure 2.9, 2.10 and 2.11, we compare the performance of our optimal policy to the optimal policy obtained by forcing the average (instead of the strict) delay to be less than a pre-defined threshold. Both policies are applied assuming buffer overflow and delay violation as the way to drop the packets, but the second policy is optimized just in order to minimize the buffer overflow and keep an average delay small enough.

Therefore, the policy with average delay constraint is obtained as follows: according to the Little's law, we propose to convert the average delay constraint D_{ct} into an average queue length constraint Q_{ct} since $Q_{ct} = \lambda_d.D_{ct}$ where λ_d is the data arrival rate. The policy

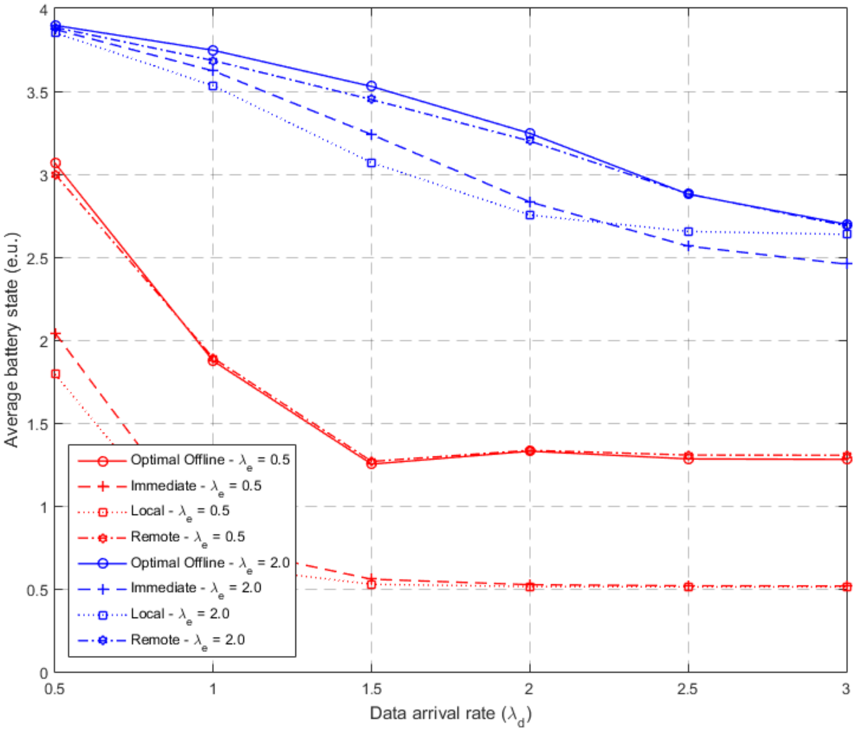
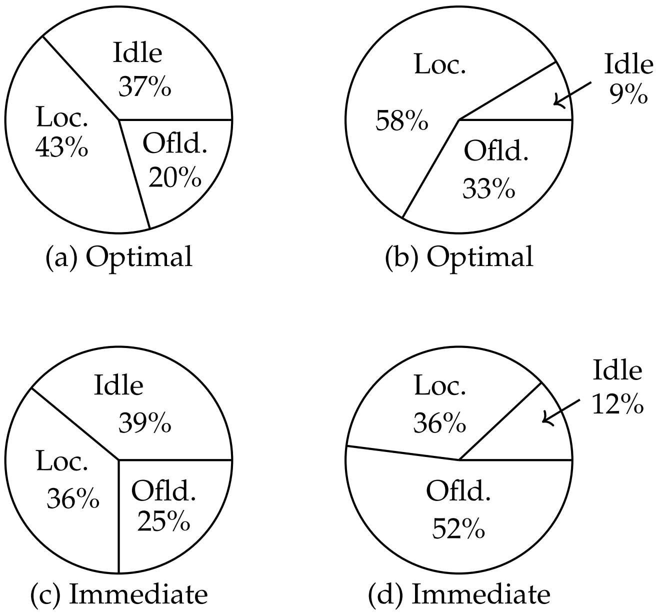


Figure 2.7 – Average battery state versus data arrival rate for energy arrival rates $\lambda_e = 0.5$ and 2.0.



(a) Optimal

(b) Optimal

(c) Immediate

(d) Immediate

Figure 2.8 – Percentage of processing decisions at $(\lambda_d = 1, \lambda_e = 1)$ (a) and (c), and $(\lambda_d = 2, \lambda_e = 2)$ (b) and (d).

ensuring a bounded average delay can be found by solving the following CMDP:

$$\tilde{\mu}_{Q_{ct}} = \arg\min_{\mu} \quad \lim_{N\to+\infty} \frac{1}{N}\mathbb{E}^{\mu}\left[\sum_{n=1}^{N} \varepsilon_o(\mathbf{s}_n, u_n)\right] \tag{2.15}$$

$$\text{s.t.} \quad \lim_{N\to+\infty} \frac{1}{N}\mathbb{E}^{\mu}\left[\sum_{n=1}^{N} q_n\right] \leqslant Q_{ct} \tag{2.16}$$

where q_n is the queue length. Notice that we do not consider the delay violation for this optimization since the strict delay is not taken into account in this policy as we just force the average delay to be less than a threshold. So the policy $\tilde{\mu}_{Q_{ct}}$ is done to handle properly the average delay and not the strict delay.

Our optimal policy adapted to strict delay has been computed with $K_0 = 3$. In order to compare both policies in the strict delay constraint set up (it means that the packet is dropped if the delay is strictly larger than K_0 even if we apply the policy $\tilde{\mu}_{Q_{ct}}$), we need to choose properly D_{ct}. It makes sense to force $D_{ct} \leq 3$ in order to have a small amount of dropped packets due to strict delay violation. In fact, we have found that $D_{ct} = 2$ or $D_{ct} = 3$ can lead to similar performance, thus we have fixed $D_{ct} = 3$.

As we can see, our policy outperforms the policy considering only the average delay in terms of percentage of discarded packets, consumed energy (in most cases), and battery levels (in most cases). So, it was worth to do the effort to optimize the policy by taking into account the strict delay into the state model rather than just using the optimal policy adapted to the average delay with a well-tuned threshold, especially for small λ_e.

Correlated EH process

We now consider that the EH process is time-correlated in order to be more general and better to capture the different timescales of the randomness. In order to cast this assumption into an MDP framework, we need to add EH process e to the state of the system, i.e., $\mathbf{s} = (\mathbf{k}, b, e, x)$ instead of $(\mathbf{k}, b, x)$ as done previously. Then, a new optimal policy taking into account the EH correlation is re-computed by using PI algorithm. Here, we assume that the transition probability of the Markov Chain satisfies the following equation

$$p(\mathbf{s}'|\mathbf{s}, u) = p(\mathbf{k}'|\mathbf{k}, b, u).p(b', e'|b, e, x, u).p(x'), \tag{2.17}$$

where $p(b', e'|b, e, x, u)$ is obtained according to the following rules:

```
1: if $E > b$ then
       $p(b', e'|b, e, x, u) = 0$
```

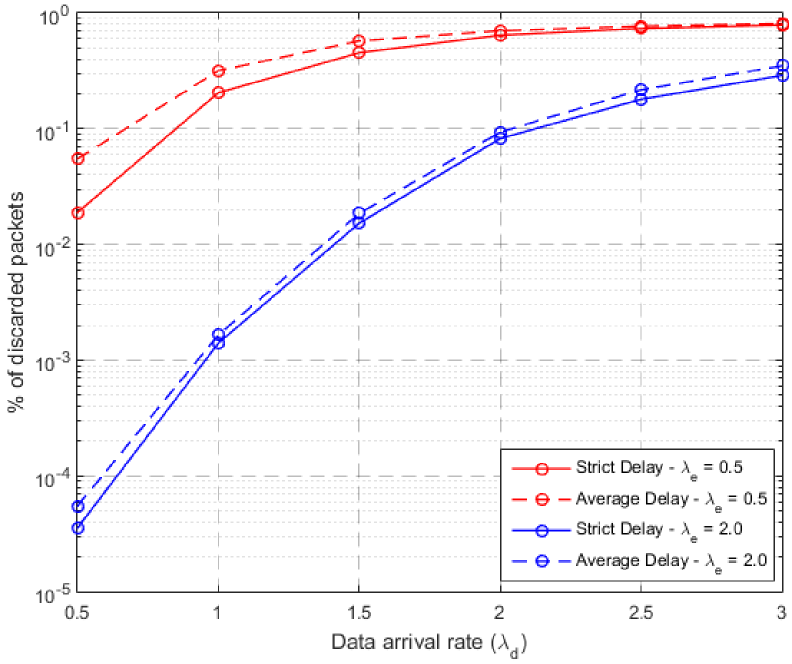


Figure 2.9 – Percentage of the discarded packets versus data arrival rate with different energy arrival rates between strict and average delay policies($\mu^\star$ and $\tilde{\mu}_3$, respectively).

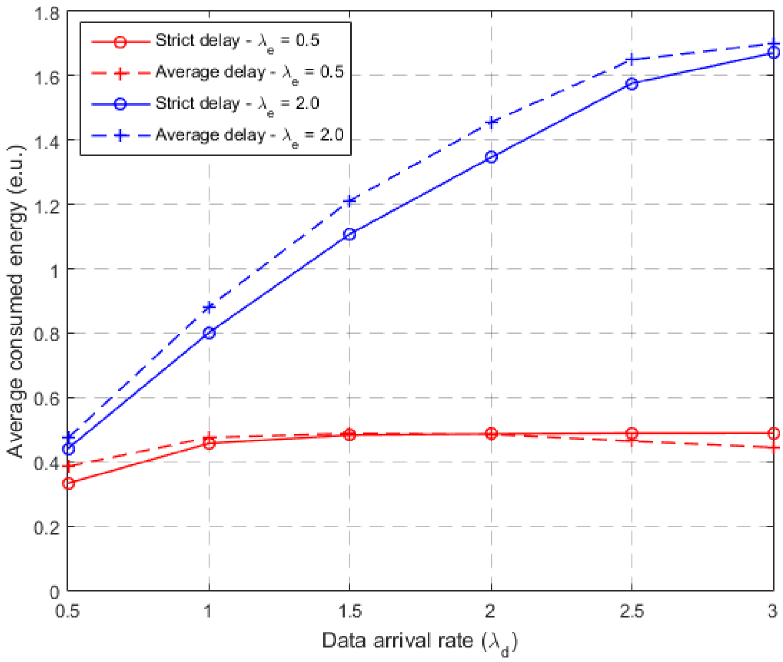


Figure 2.10 – Average consumed energy versus data arrival rate with different energy arrival rates between strict and average delay policies ($\mu^\star$ and $\tilde{\mu}_3$ respectively).

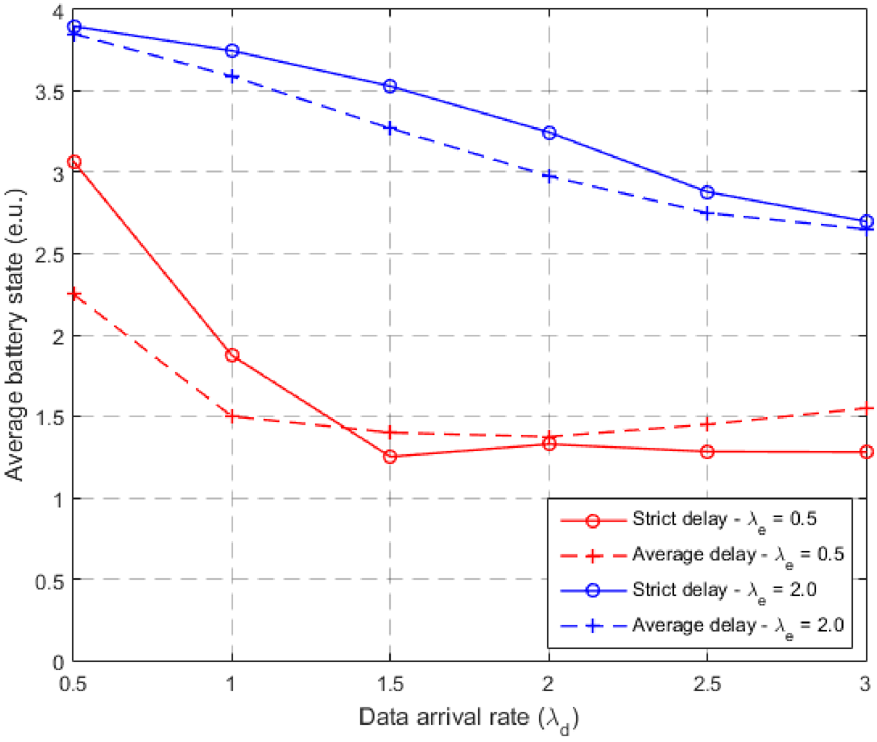


Figure 2.11 – Average battery state versus data arrival rate with different energy arrival rates between strict and average delay policies ($\mu^\star$ and $\tilde{\mu}_3$ respectively).

```
2: else if $b' < b - E$ then
       $p(b', e'|b, e, x, u) = 0$
3: else if $b' = \min(B_e, b - E + e)$ then
       $p(b', e'|b, e, x, u) = p(e'|e)$
4: else
       $p(b', e'|b, e, x, u) = 0.$
```

In addition, the transition probability from an energy arrival state j at time slot n to another energy arrival state i at time slot $n+1$ is given by

$$p(e_{n+1} = i|e_n = j) = \frac{(1-\rho_e)^{|i-j|}}{\sum\limits_{k=0}^{|\mathcal{H}_e|-1} (1-\rho_e)^{|k-j|}} \tag{2.18}$$

where ρ_e is the so-called correlation factor and $\mathcal{H}_e$ is the set of potential energy units harvested during one time slot.

In Figure 2.12, we compare the performance of the optimal policy (adapted to time-correlated EH process) with the **immediate** policy. We set $\mathcal{H}_e = \{0, 1, 2\}$ e.u. per slot. The proposed optimal policy is still better than the **immediate** policy. The performance

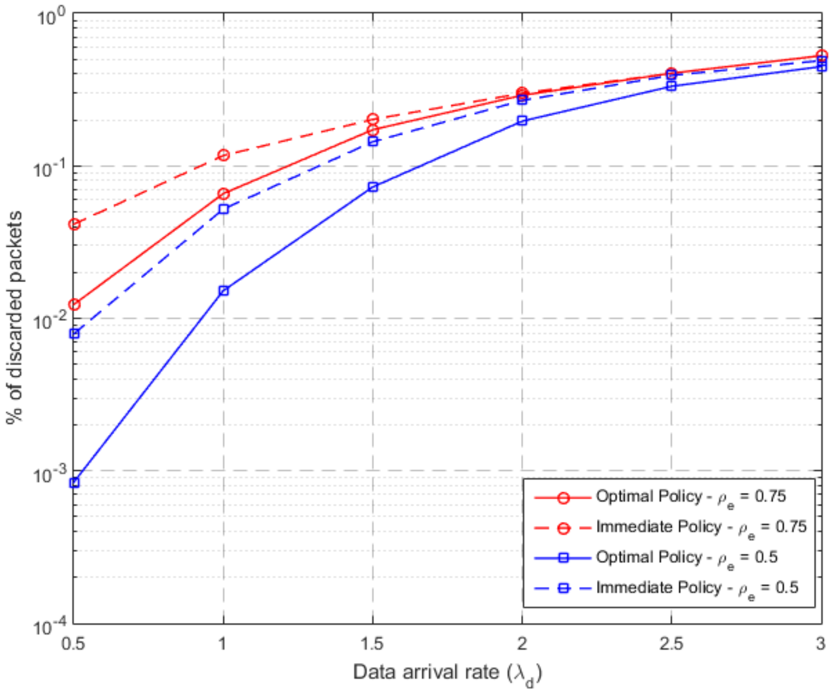


Figure 2.12 – Percentage of the discarded packets versus data arrival rate with different correlated energy arrival rates between strict and immediate policies.

of the system decreases when ρ_e increases because the system will be trapped in the state $e = 0$ for a longer period of time, leading to more discarded packets.

2.5 Special Cases

In this section, we present two simplified cases of the previous problem both under perfect CSIT and taking strict delay constraint into account. In subsection 2.5.1, an optimal policy minimizing the number of discarded packets is designed for an EH-transmitter (We remove the offloading capabilities from the main problem in Section 2.2). In subsection 2.5.2, an optimal policy minimizing the power consumption of a conventional transmitter powered by the grid is presented (We remove the EH and the offloading capabilities from the main problem in Section 2.2).

2.5.1 Packet Scheduling for EH-Transmitter

In the first case, the adopted system model depicted in Figure 2.13, is similar to the one described in Section 2.2, but without offloading capabilities. The main problem is reduced to a packet scheduling problem of an EH-transmitter. Therefore, the concept of executing packets locally or remotely is no longer applicable, and the system here only needs to transmit its packets to a receiver. To this end, we denote u_n ($u_n \leqslant q_n$) the number of packets to be transmitted during slot n of duration T_s, through the channel of gain x_n. The consumed energy to transmit these packets, previously stated in Eq. (2.3), (2.4) and (2.6), becomes

$$E(x_n, u_n) = \left\lceil \frac{P(x_n, u_n).T_s}{\mathcal{E}_U} \right\rceil, \tag{2.19}$$

where

$$P(x_n, u_n) = \frac{WN_0}{x_n}\left(2^{\frac{u_n L}{WT_s}} - 1\right). \tag{2.20}$$

is the required power for this transmission and W is the allocated bandwidth.

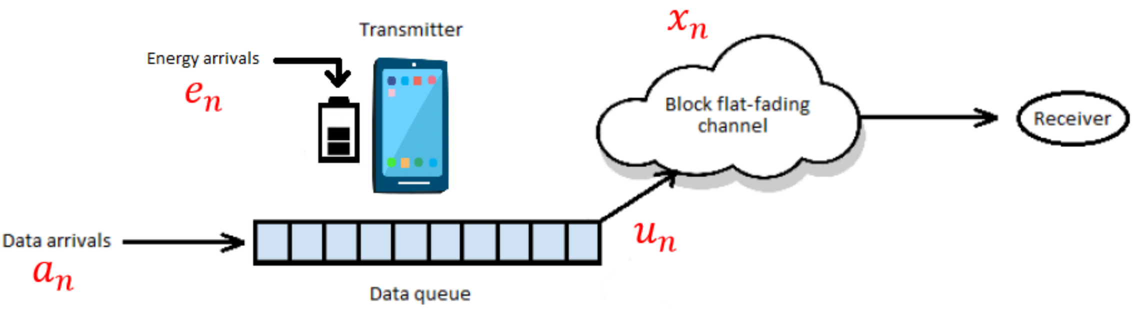


Figure 2.13 – Resource Scheduling for EH-Transmitter.

2.5.1.1 Problem Formulation and Resolution

Regarding the MDP formulation, the same state $\mathbf{s} = (\mathbf{k}, b, x) \in \mathcal{S}$ is valid here. However, the action space $\mathcal{U}$ in this case denotes the number of packets u that the transmitter can send during a time slot. This space is finite and the number of actions is $|\mathcal{U}| = U_0 + 1$, where U_0 is a pre-defined parameter. The transitions probabilities is again given by

$$p(\mathbf{s}'|\mathbf{s}, \nu) = p(\mathbf{k}'|\mathbf{k}, b, \nu).p(b'|b, x, \nu).p(x'), \tag{2.21}$$

Our goal is to find the optimal policy $\mu^\star$ minimizing the average number of discarded packets using the following MDP formulation

$$\mu^\star = \arg\min_{\mu} \left[\lim_{N \to +\infty} \frac{1}{N} \mathbb{E}^{\mu} \left[\sum_{n=1}^{N} \left(\varepsilon_d(\mathbf{s}_n, \nu_n) + \varepsilon_o(\mathbf{s}_n, \nu_n) \right) \right] \right] \tag{2.22}$$

We solve this problem using VI algorithm presented in Section 1.3.1.

2.5.1.2 Numerical Results

To evaluate numerically the obtained policy, we consider the same settings as in Section 2.4 with some minor differences: the allocated bandwidth is $W = 5$ MHz, $\mathcal{E}_U = 100$ nJ and U_0 = 6 packets per slot.

Convergence of the long-term average cost

In Figure 2.14, we illustrate the average number of discarded packets for the optimal policy (obtained after the convergence of the VI algorithm) versus N (the horizon) with various energy arrival rates λ_e and a data arrival rate $\lambda_d = 1.5$. We can notice that it takes only a few hundreds iterations for the system to stabilize its behavior. In addition, if a large amount of energy can be harvested, λ_e increases, and the system is able to execute more packets reducing hence considerably the average number of discarded packets.

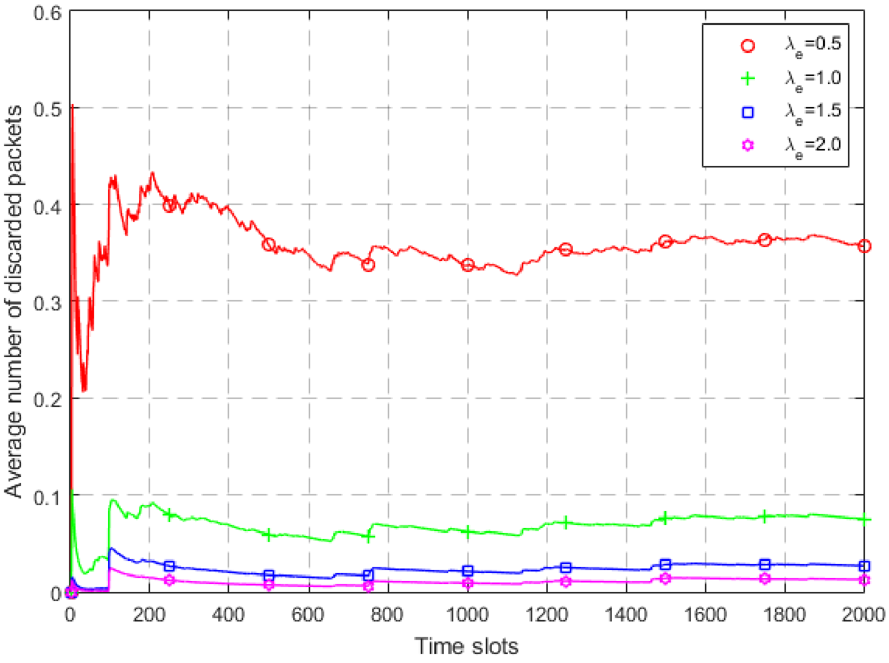


Figure 2.14 – Convergence analysis for the average rate of discarded packets with different energy arrival rates.

Comparison with other policies

In Figure 2.15, we display the percentage of discarded packets versus the data arrival rate λ_d for different energy arrival rates for two policies. we compare the optimal policy exhibited by the VI algorithm with the **immediate** one. As we can observe, the proposed optimal policy provides significantly better performance than the **immediate** one in terms of percentage of discarded packets. In fact, this policy can adapt the transmission rate according to the buffer, battery and channel conditions. In addition, we remark that the number of discarded packets increases when the data arrival rate λ_d increases because the buffer overflow could happen more often. On the one hand, when the energy available to scavenge is low (small λ_e), an efficient energy management becomes crucial to ensure the sustainability of the system, and the gap between both policies increases. On the other hand, when a large amount of energy is available (large λ_e), the system can survive even without controlling relevantly the energy consumption which leads to similar performance between the optimal and **immediate** policies.

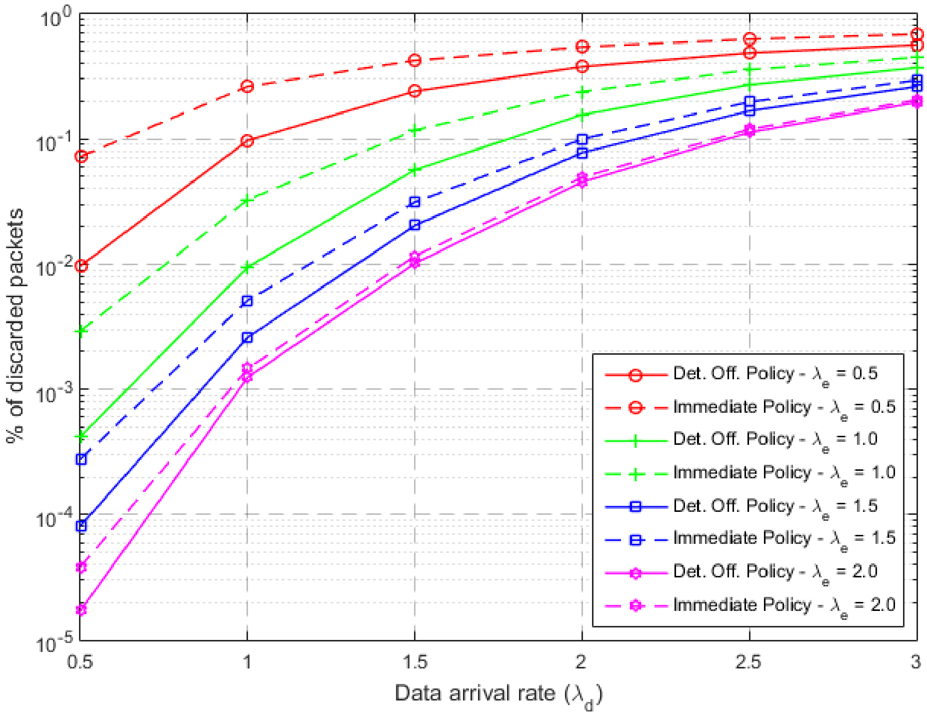


Figure 2.15 – Percentage of the discarded packets versus data arrival rate for different energy arrival rates.

Similar to Figure 2.15, Figure. 2.16 compares the percentage of discarded packets of the optimal policy with the two other variants of the **immediate** policy. As we can see, by considering only the age of the packets without adjusting carefully the number of transmitted packets to the battery level and buffer state leads to decrease the number of sent packets, and therefore the **immediate** policy remains much better.

In Figure 2.17, we plot the average consumed energy versus the data arrival rate λ_d with different energy arrival rates λ_e. We observe that the optimal policy consumes less energy than the **immediate** one while sending more packets because it adapts the number of transmitted packets per slot to the channel conditions and the battery state.

In Figure 2.18, we show the average battery state versus the packet arrival rate λ_d with different energy arrival rates λ_e. As the optimal policy offers a lower energy consumption

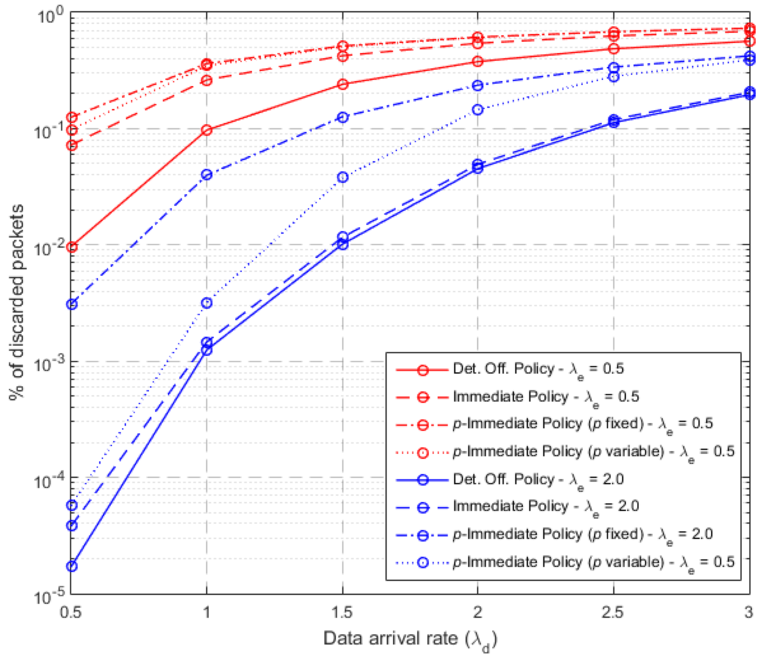


Figure 2.16 – Percentage of the discarded packets versus data arrival rate with different energy arrival rates and different immediate policies.

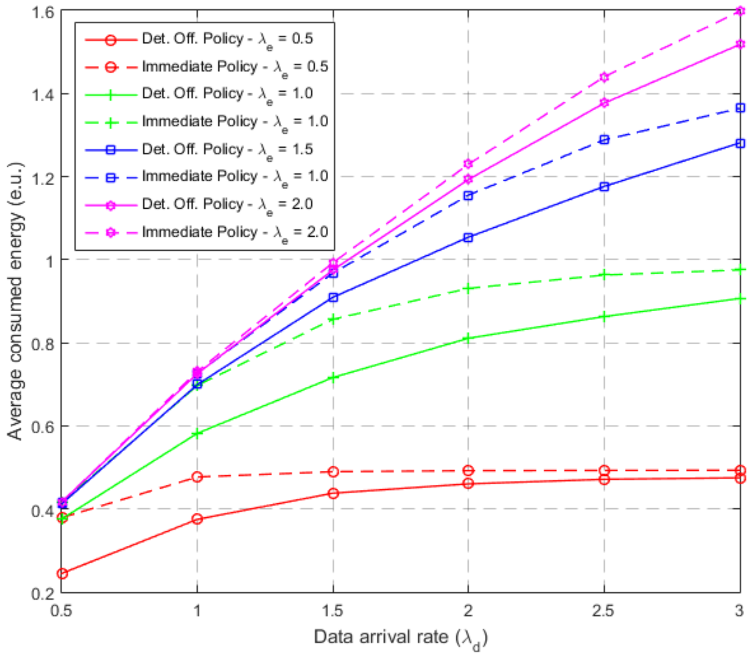


Figure 2.17 – Average consumed energy versus data arrival rate for different energy arrival rates.

(see Figure 2.17), the battery is less used and its energy level is thus higher. This ensures a better sustainable communication with less number of discarded packets.

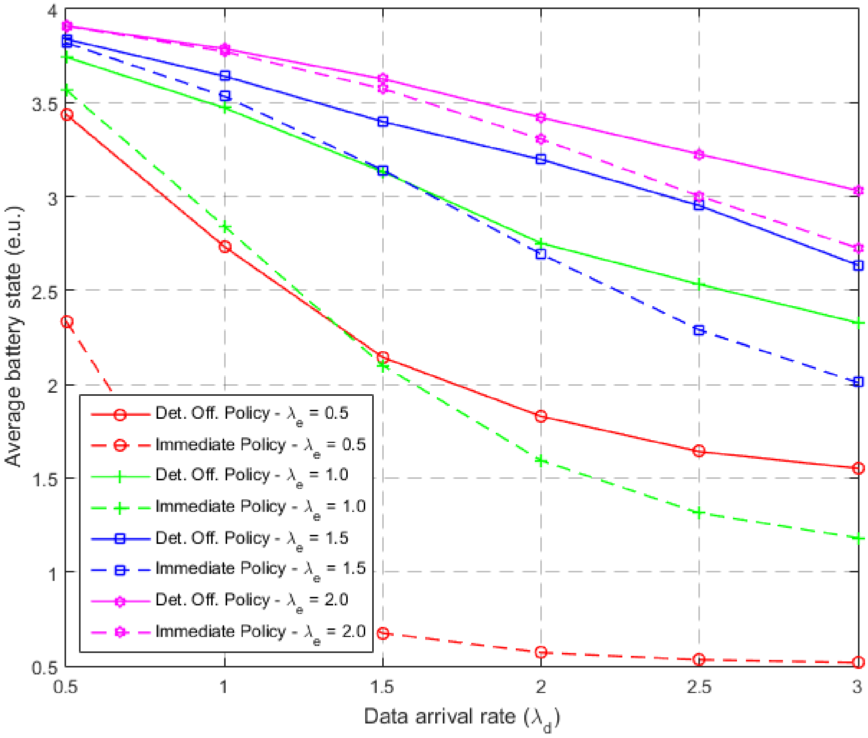


Figure 2.18 – Average battery state versus data arrival rate for different energy arrival rates.

In Figure 2.19, we show the percentage of discarded packets due to delay among the total number of discarded packets for different values of the energy arrival rate λ_e and the data arrival rate λ_d. As explained before, a packet can be discarded due to delay or buffer overflow. When the data arrival rate increases, the probability to discard a packet due to overflow increases, resulting in a lower contribution of the delay in discarding packets. On the other hand, when the energy rate decreases, the percentage of discarded packets due to the delay slightly increases because, in average, a packet remains more often in the buffer since there is not enough energy to transmit it. Hence, it is flushed from the buffer for delay's purpose.

Comparison with the state-of-the-art

In Figure 2.20, 2.21 and 2.22, we compare the performance of our optimal policy to the optimal policy obtained the same way as before, i.e. by forcing the average (instead of the strict) delay to be less than a pre-defined threshold using the following CMDP

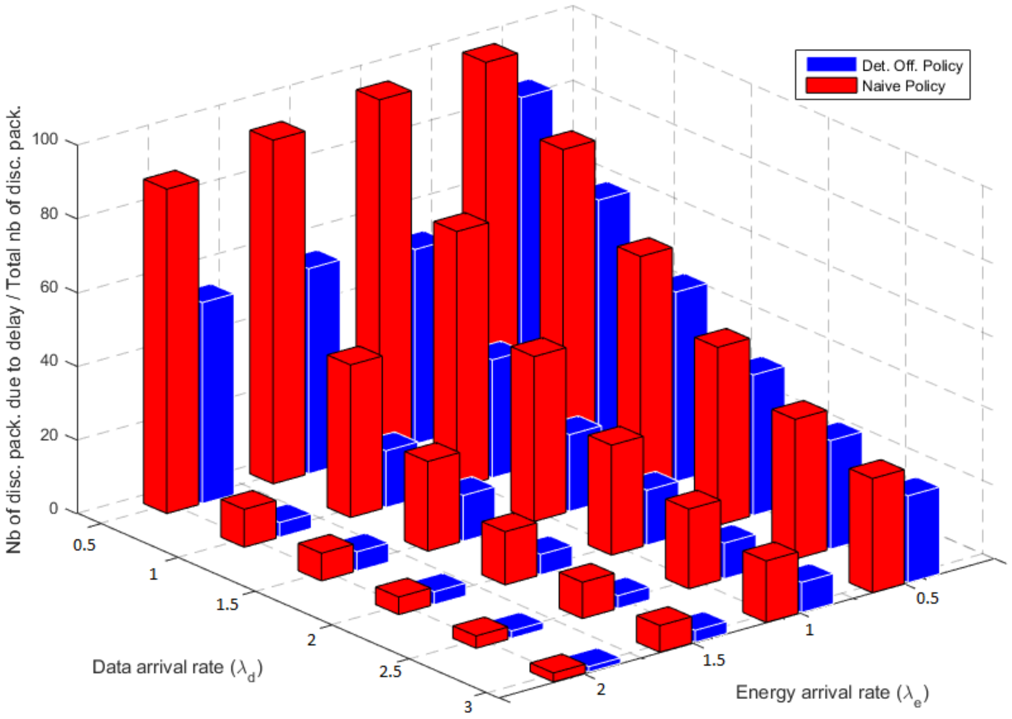


Figure 2.19 – Percentage of the discarded packets due to delay versus data arrival rate and energy arrival rate.

formulation:

$$\tilde{\mu}_{Q_{ct}} = \arg\min_{\mu} \quad \lim_{N\to+\infty} \frac{1}{N}\mathbb{E}^{\mu}\left[\sum_{n=1}^{N} \varepsilon_o(\mathbf{s}_n, u_n)\right] \tag{2.23}$$

$$\text{s.t.} \quad \lim_{N\to+\infty} \frac{1}{N}\mathbb{E}^{\mu}\left[\sum_{n=1}^{N} q_n\right] \leqslant Q_{ct} \tag{2.24}$$

As we can see, our policy outperforms the policy considering only the average delay in terms of percentage of discarded packets, consumed energy (in most cases), and battery levels (in most cases). It was therefore valuable to do the effort with the strict delay constraint and not simply use the optimal policy adapted to the average delay constraint with a well-adjusted threshold, especially for the small λ_e.

Correlated EH process

In Figure 2.23, we compare the performance of the optimal policy (adapted to time-correlated EH process) with the **immediate** policy. The proposed optimal policy is still better than the **immediate** policy. The performance of the system decreases when ρ_e increases because the system will be trapped in the state $e = 0$ for a longer period of time,

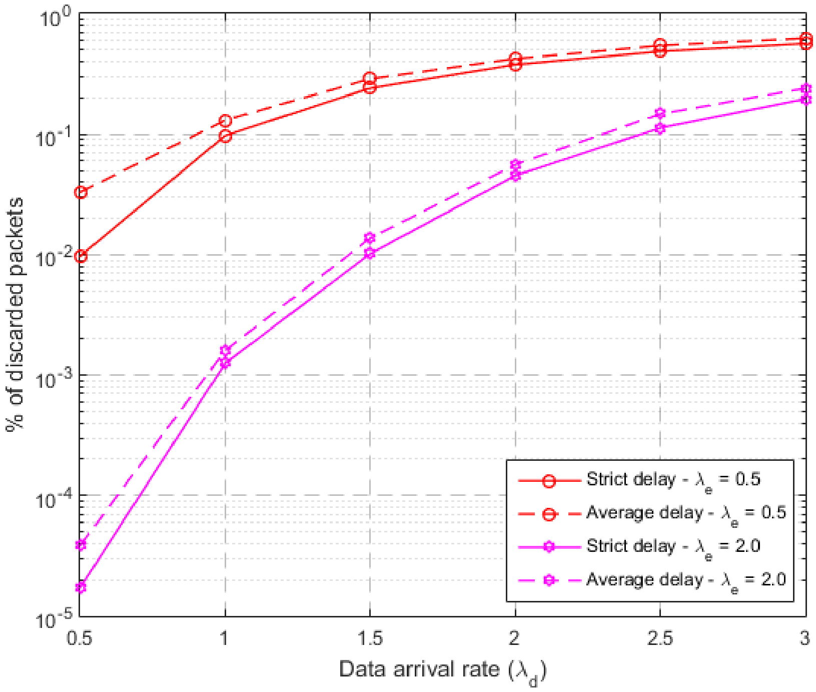


Figure 2.20 – Percentage of the discarded packets versus data arrival rate with different energy arrival rates between strict and average delay policies($\mu^\star$ and $\tilde{\mu}_3$, respectively).

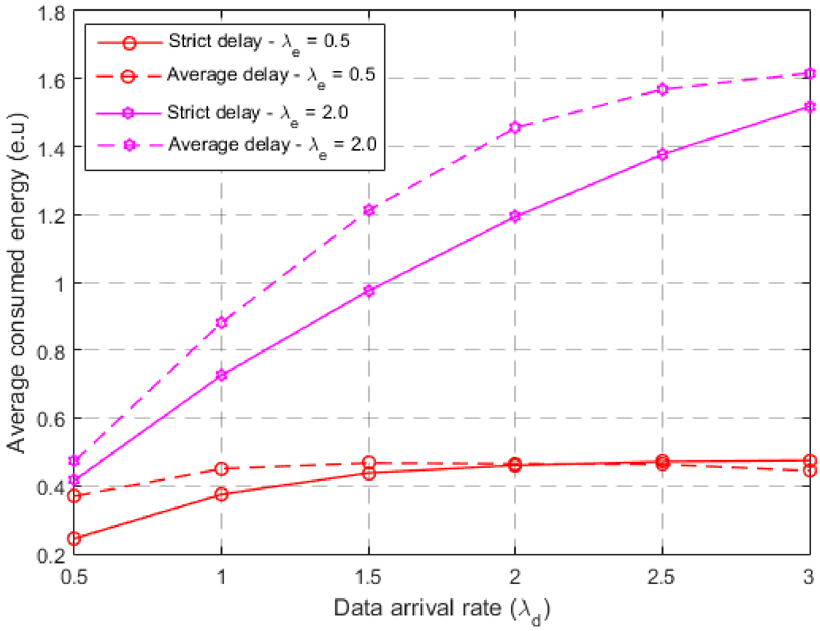


Figure 2.21 – Average consumed energy versus data arrival rate with different energy arrival rates between strict and average delay policies ($\mu^\star$ and $\tilde{\mu}_3$ respectively).

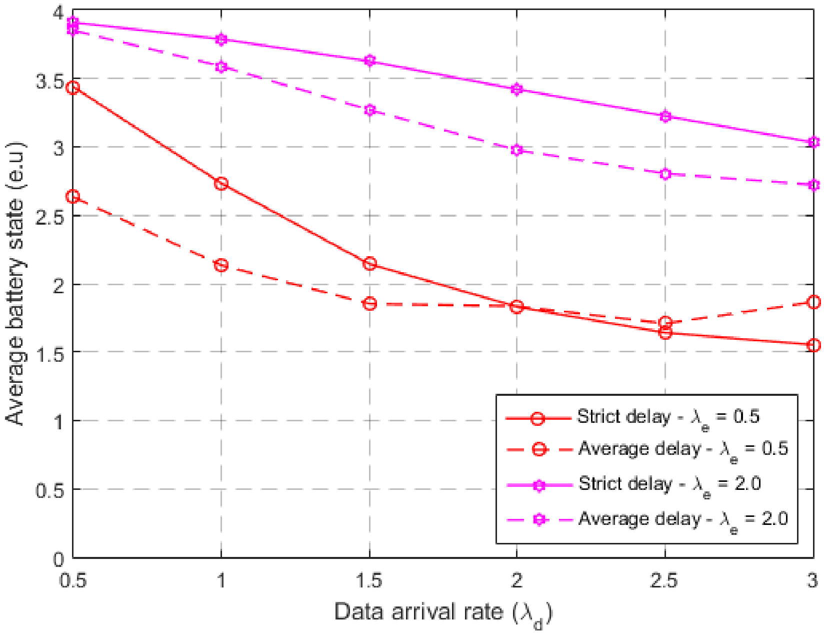


Figure 2.22 – Average battery state versus data arrival rate with different energy arrival rates between strict and average delay policies ($\mu^\star$ and $\tilde{\mu}_3$ respectively).

leading to more discarded packets.

2.5.2 Power Consumption Minimization under Strict Delay Constraint

We consider now in Figure 2.24 a conventional transmitter relying on its own battery. A different problem is studied here, since there is no replenishing mechanism of the battery. Therefore, our goal in this case is to minimize the power consumption of the device in order to prolong the life of its battery. The same system model of Section 2.5.1 is adopted while removing all the aspects related to EH.

2.5.2.1 Problem Formulation and Resolution

The MDP formulation considers the state space $\mathbf{s} = (\mathbf{k}, x)$ where $|\mathcal{S}| = (K_0+2)^B.|\mathcal{X}|$ with the same action space as Section 2.5.1, namely the number of transmitted packets u during a time slot. The battery is not taking part in the description of the states, so all transitions related to it are omitted.

The problem here is formulated as a CMDP problem. The objective then is to find the optimal policy $\mu^\star$ minimizing the average consumed power given by

$$\overline{P}^{\mu} = \lim_{N\to+\infty} \frac{1}{N}\mathbb{E}^{\mu}\left[\sum_{n=1}^{N} P(x_n, u_n)\right] \tag{2.25}$$

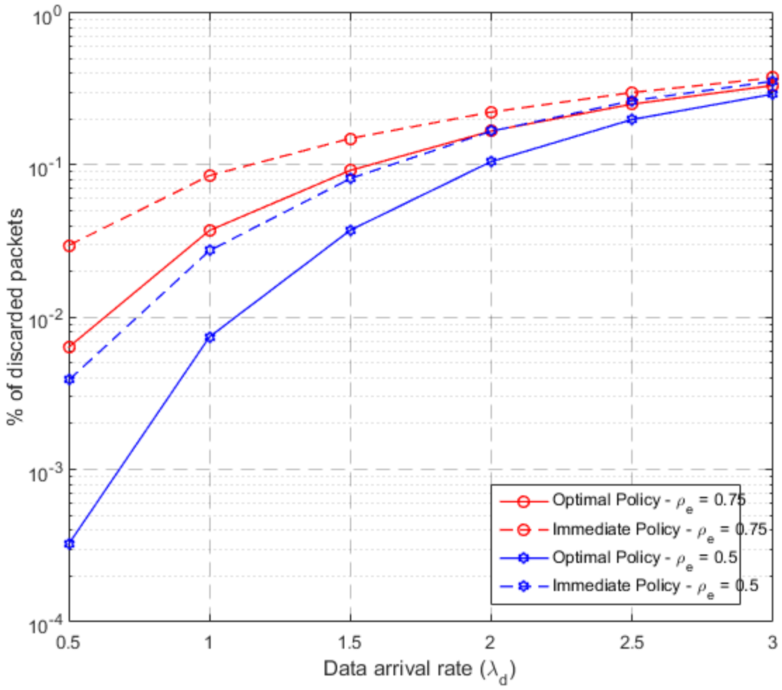


Figure 2.23 – Percentage of the discarded packets versus data arrival rate with different correlated energy arrival rates between strict and immediate policies.

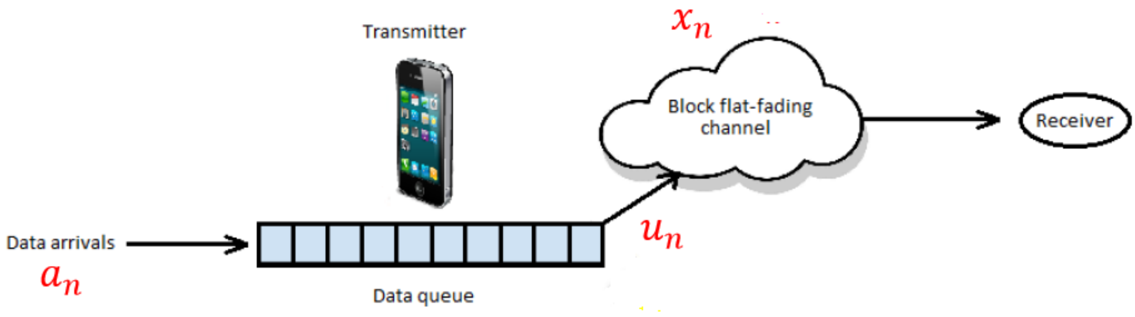


Figure 2.24 – Resource Scheduling for EH-Transmitter

where $P(x_n, u_n)$ is the instantaneous power cost when action u_n is performed at state $\mathbf{s}_n$ and can be derived using Eq. (2.20).

At a given slot n, when the system state is $\mathbf{s}_n$ and the performed action is u_n, the probability to discard packets due to delay violation is given by

$$\varepsilon_d(\mathbf{s}_n, u_n) = \begin{cases} 0 & \text{if } m_n = 0 \text{ or } m_n \leqslant u_n \\ 1 & \text{else.} \end{cases} \tag{2.26}$$

We define the average delay violation outage probability as

$$\overline{\varepsilon}_d^{\mu} = \lim_{N \to +\infty} \frac{1}{N} \mathbb{E}^{\mu}\Bigg[\sum_{n=1}^{N} \varepsilon_d(\mathbf{s}_n, u_n)\Bigg]. \tag{2.27}$$

At a given slot n, when the system state is $\mathbf{s}_n$ and the performed action is u_n, the buffer overflow occurs when $q_n - w_n + a_{n+1} > B$. The probability of overflow is then obtained as

$$\varepsilon_o(\mathbf{s}_n, u_n) = \sum_{a=B-q_n+w_n+1}^{+\infty} e^{-\gamma}.\frac{\gamma^a}{a!} = 1 - \mathrm{Q}(B - q_n + w_n + 1, \gamma). \tag{2.28}$$

where Q is the regularized Gamma function.

We define the average buffer overflow outage probability as

$$\overline{\varepsilon}_o^{\mu} = \lim_{N \to +\infty} \frac{1}{N} \mathbb{E}^{\mu}\Bigg[\sum_{n=1}^{N} \varepsilon_o(\mathbf{s}_n, u_n)\Bigg] \tag{2.29}$$

Finally, our CMDP optimization problem states as

$$\mu^{\star} = \arg\min_{\mu} \quad \overline{P}^{\mu} \tag{2.30}$$

$$\text{s.t.} \quad \overline{\varepsilon}_d^{\mu} \leqslant D_{out} \tag{2.31}$$

$$\overline{\varepsilon}_o^{\mu} \leqslant O_{out} \tag{2.32}$$

where D_{out} and O_{out} are the pre-defined thresholds for the delay and overflow outage probabilities respectively.

This problem can be solved using standard LP techniques to find the optimal offline randomized policy as shown in Section 1.3.1. Therefore, the CMDP problem can be formulated as an LP problem.

$$\rho^{\star} = \arg\min_{\rho} \sum_{\mathbf{s} \in \mathcal{S}, u \in \mathcal{U}(\mathbf{s})} \rho^{\mu}(\mathbf{s}, u) P(x, u) \tag{2.33}$$

$$\text{s.t.} \quad \sum_{\mathbf{s} \in \mathcal{S}, u \in \mathcal{U}(\mathbf{s})} \rho^{\mu}(\mathbf{s}, u) \varepsilon_d(\mathbf{s}, u) \leqslant D_{out} \tag{2.34}$$

$$\sum_{\mathbf{s} \in \mathcal{S}, u \in \mathcal{U}(\mathbf{s})} \rho^{\mu}(\mathbf{s}, u) \varepsilon_o(\mathbf{s}, u) \leqslant O_{out} \tag{2.35}$$

$$\sum_{\mathbf{s} \in \mathcal{S}, u \in \mathcal{U}(\mathbf{s})} \rho^{\mu}(\mathbf{s}, u) = 1 \tag{2.36}$$

$$\sum_{\mathbf{s}' \in \mathcal{S}, u \in \mathcal{U}(\mathbf{s})} \rho^{\mu}(\mathbf{s}', u) p(\mathbf{s}|\mathbf{s}', u) = 1, \ \forall \mathbf{s} \in \mathcal{S} \tag{2.37}$$

The optimal stationary policy $\mu^{\star}$ is then obtained as defined in Section 1.3.2 according to

$$\mu^{\star}(u|s) = \frac{\rho^{\star}(s,u)}{\sum_{u' \in \mathcal{U}(s)} \rho^{\star}(s,u')}$$

2.5.2.2 Numerical Results

In the following, the optimal policy obtained for the resolution of the LP Problem is evaluated numerically. We consider a system with the following characteristics: the slot duration is $T_s = 1$ ms, the channel states x takes 3 possible values from the finite set $\mathcal{X} = \{-5.41, -1.59, 3.18\}$ dB with respective probabilities 0.63, 0.32, and 0.05. The noise power spectral density is $N_0 = -87$ dBm/Hz and the allocated bandwidth is $W = 5$ MHz. We simulate i.i.d arrivals following a Poisson distribution with mean $\lambda_d = 2$ packets per slot. We assume that the packets are of equal size $L = 5000$ bits, and the buffer is of size $B = 5$ packets. The maximum delay is $K_0 = 2$ (i.e., in absolute time $K_0 T_s = 2$ ms) . The maximum available power at the transmitter is $P_{\max} = 2$ mW and we fix $U_0 = 5$ packets per slot.

Convergence of the long-term average cost

Figure 2.25 shows the convergence behavior (versus N) of the average power for various delay outage probabilities where the overflow outage probability O_{out} is fixed to 0.4. One can notice that as the delay constraint becomes tighter, the average consumed power significantly increases. Indeed, when the imposed delay is stricter, the system is forced to send more packets even if the channel is in a bad state.

Comparison with other policies

Figure 2.26 compares the average consumed power versus the delay outage probability for different overflow outage probabilities. Moreover, we compare the optimal and **immediate** policies. Notice that the system characteristics have been chosen such that the **immediate** policy satisfies the overflow outage probability constraint. The delay outage probability constraint is always satisfied by the **immediate** policy since the packets do not stay in the buffer. As one can observe, our policy gives strongly better performance in terms of power consumption as it adapts its transmission rate according to the channel conditions while satisfying the requirements of the system. The overflow constraint is inactive when $O_{out} = 0.4$, because satisfying the delay constraint is more crucial. In contrast, when $O_{out} = 0.05$, the delay outage probability is inactive. In general, requiring a stronger overflow constraint increases the energy consumption. Notice that after a certain value of D_{out}, the overflow constraint becomes more critical making the delay constraint inactive leading to an error floor.

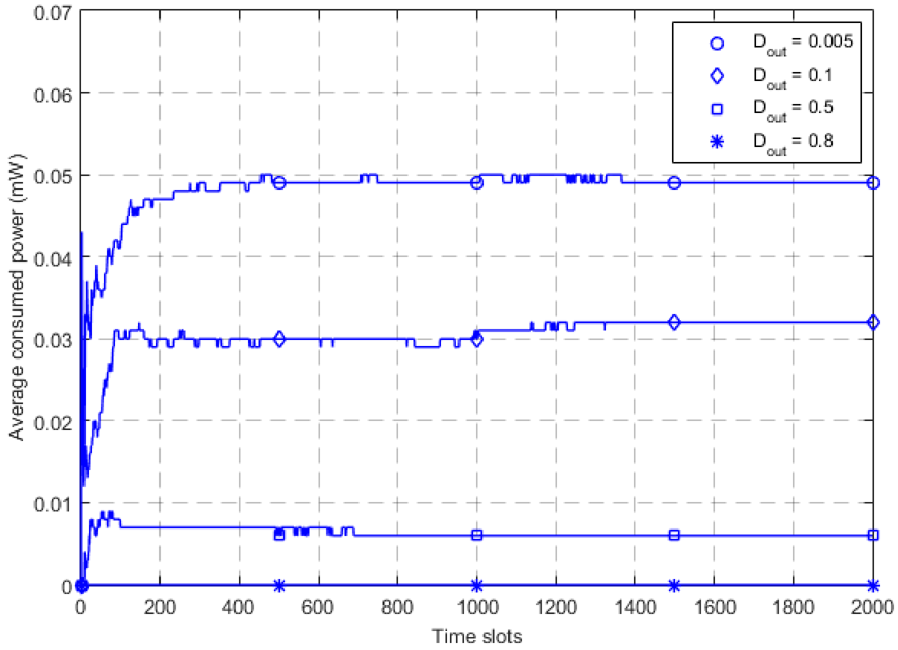


Figure 2.25 – Convergence of average power for various delay outage probability constraints.

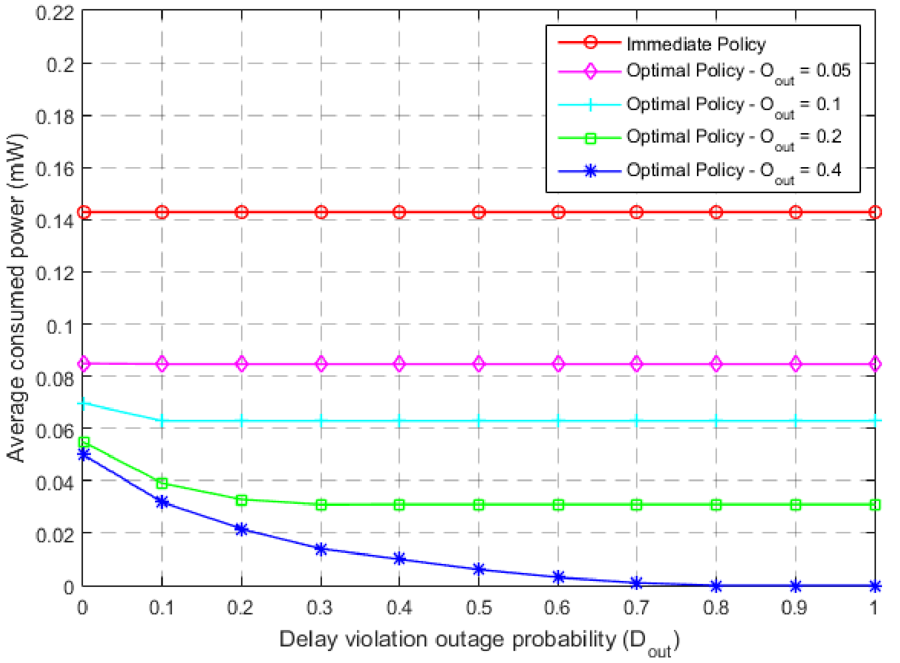


Figure 2.26 – Average power versus delay outage probability for different policies.

Comparison with the state-of-the-art

Figure 2.27 compares the average consumed power for average and strict delays. To do this, we first define the average queue length as

$$\overline{\varepsilon}_q^{\mu} = \lim_{N\to+\infty} \frac{1}{N}\mathbb{E}^{\mu}\left[\sum_{n=1}^{N} q_n\right] \tag{2.38}$$

And then solved the following CMDP problem devoted to average delay constraint:

$$\mu^{\star} = \arg\min_{\mu} \quad \overline{P}^{\mu} \tag{2.39}$$

$$\text{s.t.} \quad \overline{\varepsilon}_q^{\mu} \leqslant Q_{ct} \tag{2.40}$$

$$\overline{\varepsilon}_o^{\mu} \leqslant O_{out} \tag{2.41}$$

where $Q_{ct} = \lambda_d.D_{ct}$ is the average queue length constraint. Q_{ct} is chosen to achieve similar performance, in terms of delay and overflow, between the average and strict schemes.

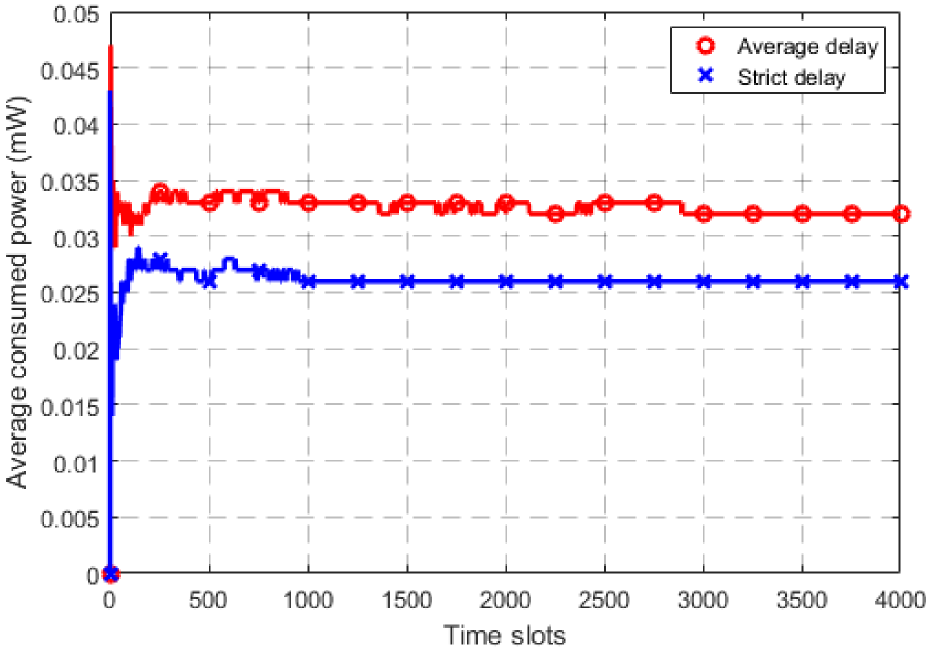


Figure 2.27 – Average consumed power by schemes with strict and average delay constraints.

As one can notice, for the same performance in terms of delay and overflow, the proposed scheme with the strict delay reduces the power consumption by 20% approximately compared to the average delay scheme.

2.6 Conclusion

In this chapter, we have investigated resource scheduling and computation offloading problem from an EH mobile user to its serving resourceful BS under strict delay constraint. We have proposed an optimal policy to minimize the packet loss rate using MDP framework and DP techniques. By leveraging on the knowledge of the available energy in the battery, the data and energy arrivals as well as the channel states, the optimal offline policy decides to process locally or remotely while specifying the number of packets to be processed. We have seen that taking into account the strict delay (at the expense of the complexity since the number of states dramatically increases) enables a significant gain in performance compared to the adaptation of the average delay based policy.

In addition, we have studied two special cases of the general problem. In the first one, we have addressed resource scheduling problem under energy harvesting capabilities with strict delay constraint. The optimal policy in this case adjusts the number of transmitted packets according to the channel conditions and the available energy in the battery, such that the number of discarded packets is minimized. In the second one, we have studied strict delay constrained scheduling problem. We have solved the power-efficient optimization problem using CMDP framework and LP techniques. The policy adjusts the number of transmitted packets according to the channel conditions. In this case, the power consumed is minimized while maintaining the delay and overflow outage probabilities below pre-defined thresholds given by the application.

The material presented in this Chapter has been published in [J1], [C1], [C2], and [C3].

Chapter 3

Joint Resource Scheduling and Computation Offloading with imperfect CSIT

3.1 Introduction

In Chapter 2, the scheduling-offloading problem and the desired transmission policies rely on the perfect knowledge of the CSIT and the relevant statistics without cost. In typical wireless systems, the receiver performs channel estimation and feeds back CSI on limited-capacity feedback channel to the transmitter in order to design channel-adapted transmission techniques. In practice, the acquired CSIT have errors due to different factors such as time-varying channel, inaccurate channel estimation, quantization and feedback errors, which can inevitably cause performance degradation. In addition, the balance between the time spent on acquiring the CSI and its quality will also have a significant impact on the performance of the system. For instance, a higher estimation time will result in a more accurate channel estimate, but at the cost of losing a considerable part of the communication time which may increase the communication error and so increase the delay. Therefore, in this chapter, we are interested in studying the scheduling-offloading problem under imperfect CSI.

Recently, some works have focused on new design strategies to deal with CSI imperfections in energy harvesting networks. For instance, in [78], the problem of power management for energy harvesting sensor nodes with packet retransmissions was addressed assuming partially observable CSI through an Acknowledgment (ACK)/Negative Acknowledgment (NACK) feedback. The problem was formulated as a Partially observable Markov Decision Process (POMDP) and efficient suboptimal approches were proposed based on the belief state of the channel and the solution of the underlying MDP. In [79], a beamforming problem for multi-antenna broadcast channel was studied under imperfect

CSI at the transmitter in a simultaneous information and power transmission system. A solution using relaxed semi-definite programming techniques was presented to maximize the worst-case harvested energy for an energy receiver while satisfying a required rate for an information receiver. A point-to-point feedback-enabled Multiple-Input Single-Output (MISO) channel was also considered in [80] where both the transmitter and the receiver are energy harvesting devices. The feedback policy and the transmission policy were jointly optimized to maximize the throughput using multivariate majorization theory. Considering EH only at the receiver, [81] investigated joint channel resource allocation and beamforming in Downlink (DL) MISO EH systems under imperfect CSI. The proposed strategy aimed at maximizing the DL data rate while involving both power and time consumed by the CSI training and feedback process. Moreover, an optimal transmission power policy based on only 1-bit feedback was proposed in [82] for EH communications over Rayleigh fading channels. The receiver sends bit 1 if the channel realization is above a certain threshold. Then, the transmitter does not transmit if the bit is 0 or transmits with a certain pre-defined power. The related data rate is chosen according to the threshold and not to the true value of the channel realization. Consequently, the selected data rate always ensures a safe transmission but with a pessimistic rate. The paper found out the optimal feedback channel threshold and the optimal policy that maximizes the throughput based on finite-horizon constrained MDP formulation. In [83], the problem of data amount maximization within a fixed duration was studied assuming imperfect CSIT in point-to-point communications with an EH transmitter. The authors proposed first a Markov process to model the energy arrivals and the channel impulse response with strong correlations and then derived the optimal online power scheduling policy using finite-horizon DP techniques. In addition, they studied the performance limits of EH systems under imperfect CSIT through an asymptotic analysis of the average throughput at low and high average energy recharge rates. In [84], they determined the optimal offline policy for a similar problem.

In this chapter, for realistic imperfect CSI scenario, we consider that acquiring channel estimates incurs some time and energy costs on the system performance. We assess the previously obtained optimal policy (in Section 2.3.6) under imperfect CSI conditions due to channel estimation errors. We also consider imperfect CSI assumption with ARQ protocol, allowing thus packet re-transmission. Therefore, in these cases, the packet loss rate is affected twofold:

- On one side, with respect to the imposed strict delay because of a smaller transmission period of data packets, or because of a longer duration of packets in the buffer for re-transmission (with ARQ protocols);
- On the other side, with respect to the erroneous channel estimation which can lead to an increase in the number of discarded packets.

We analyze the system taking into account these errors and show through numerical results that an appropriate trade-off is needed between the channel estimation accuracy and the transmission period in order to reduce the dropped packets depending on the available energy, energy arrivals and data arrivals.

The chapter is organized as follows. In Section 3.2, we describe the system model. In Section 3.3, we present the framework of the imperfect CSI scenario. We provide and analyze numerical results in Section 3.4. In Section 3.5, we present the packet scheduling problem under imperfect CSI as a special case of the general problem. Finally, we give some concluding remarks in Section 3.6.

3.2 System model

In wireless communication systems, CSI is not perfectly known at the transmitter and can include errors. Indeed, in a Time Division Duplex (TDD) Uplink (UL) transmission between an EH device and a BS, the CSI can be obtained at the EH device by first estimating the channel at the BS via an UL training process and then feeding back a quantized version of the estimate to the transmitter. We assume that the feedback channel is error-free and instantaneous as soon as the receiver has estimated the channel. Therefore, accounting for the channel estimation phase, the time slot structure, shown in Figure 3.1 is divided into two parts:

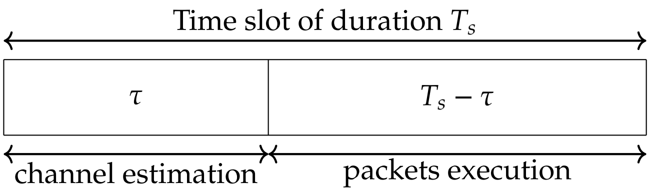


Figure 3.1 – Time slot structure under imperfect CSI scenario.

- A duration of τ ms to acquire CSI at the mobile device
- The remaining $(T_s - \tau)$ ms to execute data packets.

In particular, the EH device exploits the acquired CSI to send data whenever offloading decisions are made. In this section, we aim at evaluating the optimal policy $\mu^\star$ obtained with PI Algorithm from Section 1.3.1 when the CSI are imperfect which means that the current states used for computing the output of $\mu^\star$ are not necessary correct.

To this end, the same system model from Section 2.2 is adjusted to the imperfect CSI scenario, as shown in Figure 3.2. However, with respect to Chapter 2 where the defined quantization process and parameters are used by default, i.e. the values of x correspond to the perfect discrete channel states, in this chapter and under the imperfect CSI scenario, the channel is first estimated before being quantized. So, let $\hat{h}_n$ and $\hat{g}_n = |\hat{h}_n|^2$ denote the estimated channel and the estimated channel gain. Then, the estimated discrete (quantized) channel states are defined by $\hat{x}_n$ accordingly. In this case, a channel is said to be in state $\hat{x}_n \neq x_n$ if $\hat{g}_n \in I_{m'}$ while $g_n \in I_m$ with $m' \neq m$.

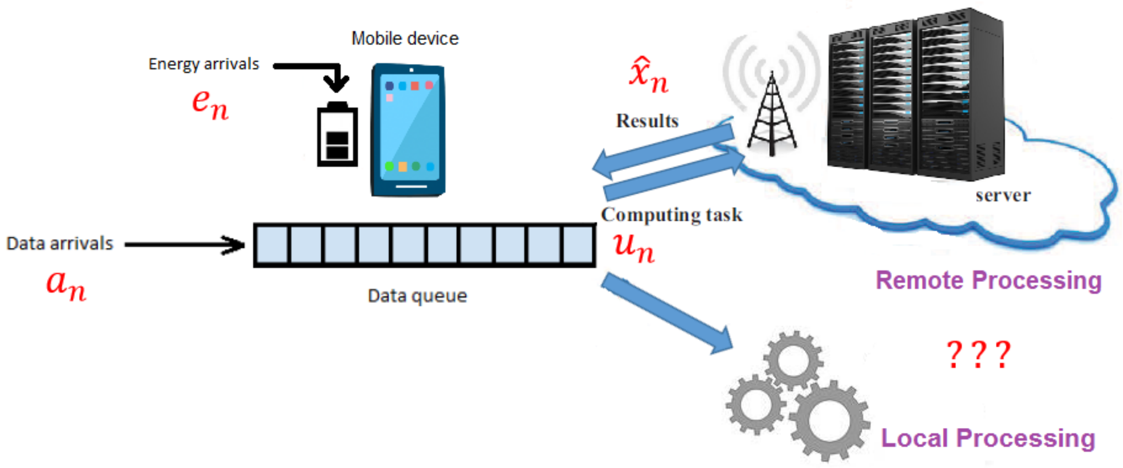


Figure 3.2 – MEC system with an EH mobile device.

3.2.1 Execution decisions and related consumed energy

Under imperfect CSI scenario, similar expressions to those derived in Section 2.2.4 are obtained by replacing x_n by the estimated channel gain $\hat{x}_n$ and T_s by $T_s - \tau$ where τ is the time required to perform channel estimation.

Therefore, at the beginning of time slot n, the consumed energy for **Local** and **Offload** decisions become:

- **Local processing:** This decision is only affected by the estimation time τ since neither x_n nor $\hat{x}_n$ is relevant in this case. The consumed energy, expressed as an integer multiple of the e.u, is then given by

$$E_\ell(u) = \left\lceil u.P_\ell.\frac{T_s - \tau}{\mathcal{E}_U} \right\rceil. \tag{3.1}$$

- **Remote processing:** This decision is affected by the estimation time τ and the estimated channel $\hat{x}_n$ between the mobile device and its BS. The energy consumption, expressed as an integer multiple of the e.u, is then given by

$$E_o(\hat{x}_n, u) = \left\lceil \frac{u}{\mathcal{E}_U}\left(\frac{L.P_t}{W_{UL}.\log_2\left(1 + \frac{P_t.\hat{x}_n}{W_{UL}.N_0}\right)} + T_w.P_w + \frac{L_{DL}.P_r}{W_{DL}.\log_2\left(1 + \frac{P_s.\hat{x}_n}{W_{DL}.N_0}\right)} \right) \right\rceil \tag{3.2}$$

Finally, we still require that the offloading procedure lasts at most one time slot, but now taking into account that a part of the time slot is reserved for channel estimation, therefore

$$u\left(\frac{L}{W_{UL}.\log_2\left(1+\frac{P_t.\hat{x}_n}{W_{UL}.N_0}\right)}+T_w+\frac{L_{DL}}{W_{DL}.\log_2\left(1+\frac{P_s.\hat{x}_n}{W_{DL}.N_0}\right)}\right)\leqslant T_s-\tau. \tag{3.3}$$

3.3 Channel estimation

At τ ms after the beginning of time slot n, we consider that the EH mobile device has an estimated discrete channel state $\hat{x}_n$. This estimated channel can be obtained through a training sequence of η pilot symbols using a total training power P_{tr} during the period τ of the time slot. Then, the required energy to perform this channel estimation is

$$E_{ce}(\hat{x}_n)=\left\lceil\frac{P_{\text{tr}}.\tau}{\mathcal{E}_U}\right\rceil \tag{3.4}$$

In fact, recalling the channel model described in Section 2.2.3, the channel response is h_n at the BS. Due to the imperfect channel estimation at this BS, we have now

$$\hat{h}_n=h_n+e_{h_n} \tag{3.5}$$

where e_{h_n} is the estimation error independent of h_n and it is a zero-mean i.i.d. complex-valued Gaussian process with variance σ_e^2 per complex dimension. According to [85], this error variance can be expressed in terms of energy per pilot symbol E_s, the number of pilot symbols used for estimation η and the Gaussian noise variance per complex dimension σ_w^2 as

$$\sigma_e^2=\mathbb{E}[|\hat{h}-h|^2]=\frac{\sigma_w^2}{\eta E_s}=\frac{N_0}{\tau P_{\text{tr}}}. \tag{3.6}$$

As we have defined the channel gain $g_n=|h_n|^2$, the estimated channel gain is then $\hat{g}_n=|h_n+e_{h_n}|^2$. It is a non central χ^2 random variable with 2 degrees of freedom in which the Gaussian variables are independent with common variance $\sigma_e^2/2$ and mean $g_n=|h_n|^2$. It has a Probability Density Function (PDF) of the form

$$P_{\hat{G}|H}(\hat{g}|h)=P_{\hat{G}|G}(\hat{g}|g)=\frac{1}{\sigma_e^2}e^{-\frac{g+\hat{g}}{\sigma_e^2}}I_0\left(\frac{2}{\sigma_e^2}\sqrt{g\hat{g}}\right) \tag{3.7}$$

where I_0 is the zero-order modified Bessel function of the first kind [86].

The estimated channel state $\hat{x}_n$ is then defined by $\hat{x}_n=Q_{\hat{g}}(\hat{g}_n)$ according to the quantization process defined in Section 2.2.3. This quantized value is sent back to the EH device and will be available at the beginning of each time slot.

3.3.1 Error probability and packet loss rate

In this section, we analyze the impact of channel estimation on the system performance, in particular on the packet loss rate. In fact, channel estimation can affect the number of discarded packets in three ways:

- The transmission period is reduced which offers less time to transmit the same amount of data, increasing thus the packet loss rate due to delay and buffer overflow.
- If the channel estimate is smaller than the actual channel, less packets can be scheduled at decision instants. Thus, more packets can be queued in the data buffer with higher delays, and may lead to more delay violation and buffer overflow occurrences.
- If the channel estimate is higher than the actual channel, the scheduled packets are all dropped. This condition incurs additional loss rate besides the delay violation and buffer overflow losses given in equations (2.12) and (2.14).

Therefore, we need to take into account such errors in the total error probability. This extra error probability (called, channel mismatch probability in the rest of the paper) can be expressed as

$$P_{e,\mathrm{CSI}} = \mathrm{Prob}\ (\hat{x} > x) = \sum\nolimits_{m'|m'>m} \mathrm{Prob}\ (\hat{g} \in I_{m'}, g \in I_m) \tag{3.8}$$

where $I_m = [t_m, t_{m+1}[$ and $I_{m'} = [t_{m'}, t_{m'+1}[$, $m = 0, \ldots, M-1$, $m' > m$ are the quantization regions of the perfect channel state and the estimated channel state, respectively.
Then, using Bayes rule and some derivations, we can compute

$$\begin{aligned}
\mathrm{Prob}\ (\hat{g} \in I_{m'}, g \in I_m) &= \int_{\hat{g}\in I_{m'}} \int_{g \in I_m} P(\hat{g}, g)\, d\hat{g}\, dg && (3.9)\\
&= \int_{\hat{g}\in I_{m'}} \int_{g \in I_m} P_{\hat{G}|G}(\hat{g}|g) P_G(g)\, d\hat{g}\, dg && (3.10)\\
&= \int_{g\in I_m} \left(\int_{\hat{g} \in I_{m'}} P_{\hat{G}|G}(\hat{g}|g)\, d\hat{g} \right) P_G(g)\, dg && (3.11)\\
&= \int_{t_m}^{t_{m+1}} \left(Q_1\left(\frac{\sqrt{2g}}{\sigma_e}, \frac{\sqrt{2}t_{m'}}{\sigma_e}\right) - Q_1\left(\frac{\sqrt{2g}}{\sigma_e}, \frac{\sqrt{2}t_{m'+1}}{\sigma_e}\right)\right) P_G(g)\, dg && (3.12)
\end{aligned}$$

where $P_{\hat{G}|G}(\hat{g}|g)$ is given in (3.7), Q_1 is the Marcum function, and $P_G(g)$ is the PDF of the channel gain with

$$P_G(g) = \begin{cases} \frac{1}{\sigma_h^2} e^{-\frac{g}{\sigma_h^2}} & \text{if } g \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

At a given time slot n, when the action u_n is done by applying the optimal policy $\mu^\star$ (obtained for the perfect channel knowledge case) on the estimated channel state $\hat{x}_n > x_n$, the number of discarded packets due to CSI errors is computed as

$$\varepsilon_e(u_n, P_{e,\mathrm{CSI}}) = u_n \times \mathbb{1}(P_{e,\mathrm{CSI}} \neq 0), \tag{3.14}$$

and the cost function of our MDP problem under policy $\mu^\star$ and imperfect CSI is given by

$$\overline{D'}(\mu^\star) = \lim_{N\to+\infty} \frac{1}{N}\mathbb{E}^{\mu^\star}\left[\sum_{n=1}^{N}\Big(\varepsilon_d(\mathbf{s}_n, u_n) + \varepsilon_o(\mathbf{s}_n, u_n) + \varepsilon_e(u_n, P_{e,\mathrm{CSI}})\Big)\right]. \tag{3.15}$$

3.4 Numerical Results

In this section, our goal is to evaluate the proposed optimal policy when the transmitter relies on an estimated version of the channel state. We consider an estimation phase duration equal to $\tau = 10$ μs (1% of T_s), and a power of $P_{\mathrm{tr}} = 4$ mW. The corresponding energy consumption for the estimation phase is thus $E = 40$ nJ which can be neglected to the energy unit, and therefore we assume $E_{ce} = 0$ e.u.

Comparison with other policies

In Figure 3.3, we compare the percentage of discarded packets between perfect and imperfect CSI scenarios. For low data arrival rate λ_d, the gap between both scenarios is large. Indeed, in our setup, the smallest channel mismatch probability is between 10^{-3} and 10^{-2} which implies that the percentage of discarded packets is necessary worse since as soon as the channel is over-estimated, the packets are dropped. However, when the data arrival rate increases, the buffer overflow can happen more often and the channel mismatch probability has less impact, which lead both scenarios to behave similarly.

In Figure 3.4, we compare the optimal and naive policies under perfect and imperfect CSI scenarios. For small energy arrival rate λ_e, the optimal policy under imperfect CSI is better than the naive policy with perfect CSI, because the latter sends packets without any adaptation to the energy and data arrivals, so energy shortage can happen more often and the number of discarded packets increases. For high energy arrival rate, imperfect CSI has stronger impact since the energy has to be controlled in a smarter way and knowing the channel accurately is more required.

In Figure 3.5, we compare the percentage of discarded packets for different estimation times τ (expressed in % of T_s). For low energy arrival rate λ_e, increasing the estimation time leads to a better channel estimation, which slightly reduces the number of discarded packets due to imperfect CSI, but at the cost of increasing the number of discarded packets

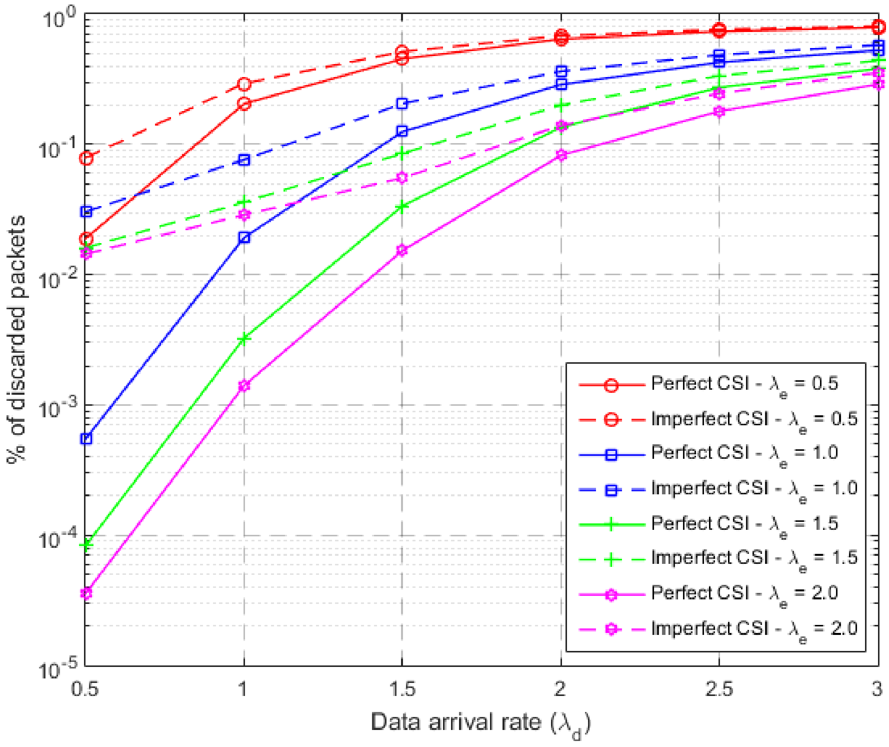


Figure 3.3 – Percentage of the discarded packets versus data arrival rate with different energy arrival rates between perfect and imperfect CSI scenarios.

due to delay and overflow. Since energy is not available in this situation, and executing packets locally is costly in most cases, the overall number of discarded packets increases because the system does not have enough energy to execute packets. For high energy arrival rate, the system has enough energy to maintain a balance between the decrease number of discarded packets due to imperfect CSI (when τ increases, the channel is better estimated) and the increase number of discarded packets due to delay and overflow (the system has less time for data packets transmission).

In Figure 3.6, we display the nature of discarded packets in percentage due to delay violation, buffer overflow and channel mismatch with different data and energy arrival rates. The number of discarded packets due to channel mismatch is significant for low data arrival rate because the delay violation or the buffer overflow can happen less often. However, for high data arrival rate, the number of packets discarded due to channel mismatch is negligible and the policy behaves approximately in the same way for perfect and imperfect CSI. Nevertheless, the imperfect CSI degrades the whole system (on the delay violation and buffer overflow) since a part of the time slot is now devoted to perform the estimation rather than the transmission.

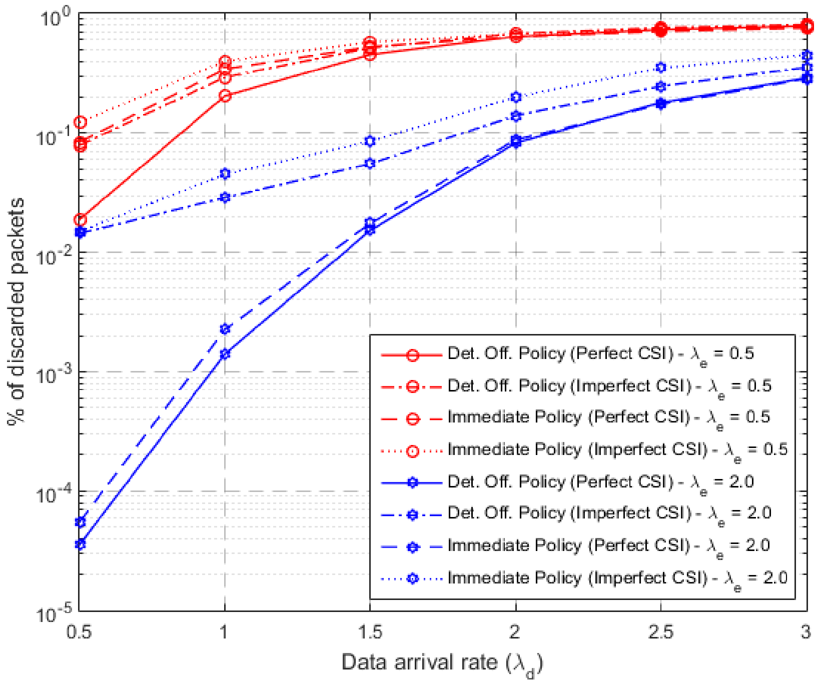


Figure 3.4 – Percentage of the discarded packets versus data arrival rate with different energy arrival rates for different policies and between perfect and imperfect CSI scenarios.

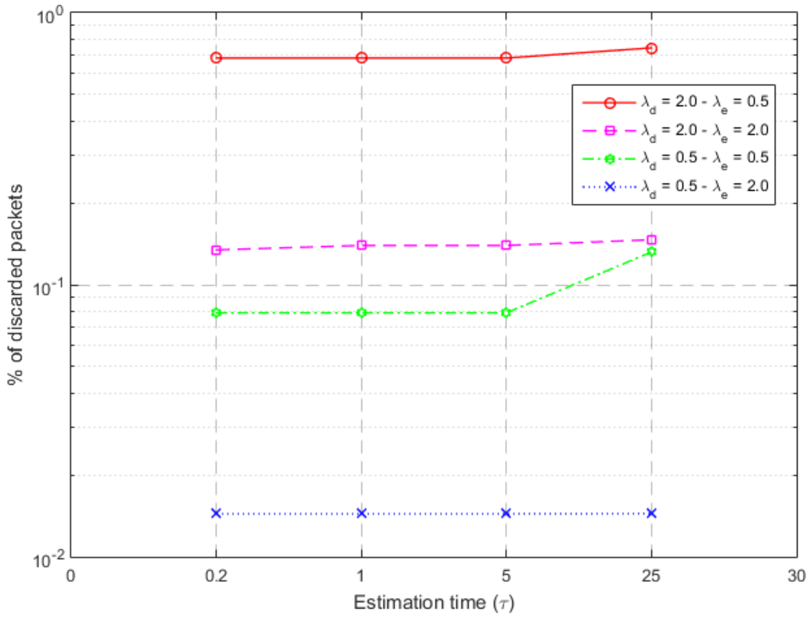
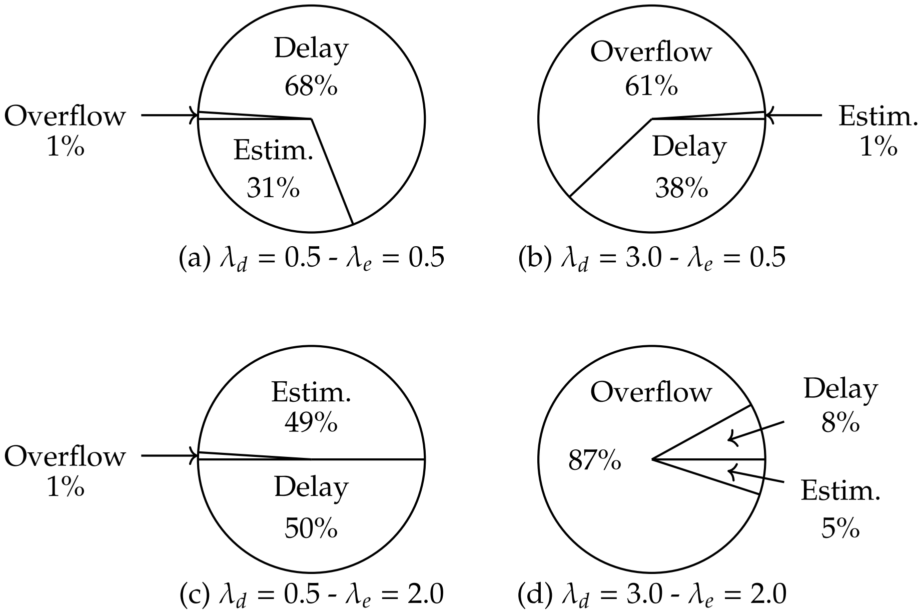


Figure 3.5 – Percentage of the discarded packets versus the estimation time τ (expressed in % of T_s) with different data and energy arrival rates.



(a) λ_d = 0.5 - λ_e = 0.5

(b) λ_d = 3.0 - λ_e = 0.5

(c) λ_d = 0.5 - λ_e = 2.0

(d) λ_d = 3.0 - λ_e = 2.0

Figure 3.6 – Percentage of the discarded packets due to delay violation, buffer overflow, and channel mismatch with different data and energy arrival rates.

Re-transmission with ARQ protocols

Under imperfect CSI assumption, it is usual to allow packet re-transmission through an ARQ protocol [87], where the receiver sends an ACK if the packet is received successfully and a NACK if not, so that the transmitter can re-transmit the packet in the following time slots. An improvement of this protocol is the Chase Combining Hybrid ARQ (CC-HARQ), where the receiver keeps the packet even if it is not received correctly. Then, when the same packet is re-transmitted, all the previous packets in memory will be combined together on the receiver side to decode it, increasing the correction capability at each re-transmission. Here, we just run our policy (the optimal one described in Section 2.3) when ARQ and CC-HARQ protocols are carried out. The packets are not discarded directly because we keep them fopr a possible retransmission. So different updating rules for the buffer state are necessary. So there is a trade-off between the higher probability for each packet to be correctly decoded at the receiver, the higher duration for the packet to stay in the buffer while waiting for the feedback and the higher energy consumed for re-transmitting the packet.

In Figure 3.7, ARQ and CC-HARQ are implemented with at most two transmissions (one re-transmission is allowed only). When λ_e is low, using ARQ and CC-HARQ is not efficient because re-transmitting the same packet twice consumes energy while it is not available in large quantities. However, when λ_e is large, these two protocols significantly reduce the number of discarded packets due to imperfect CSI.

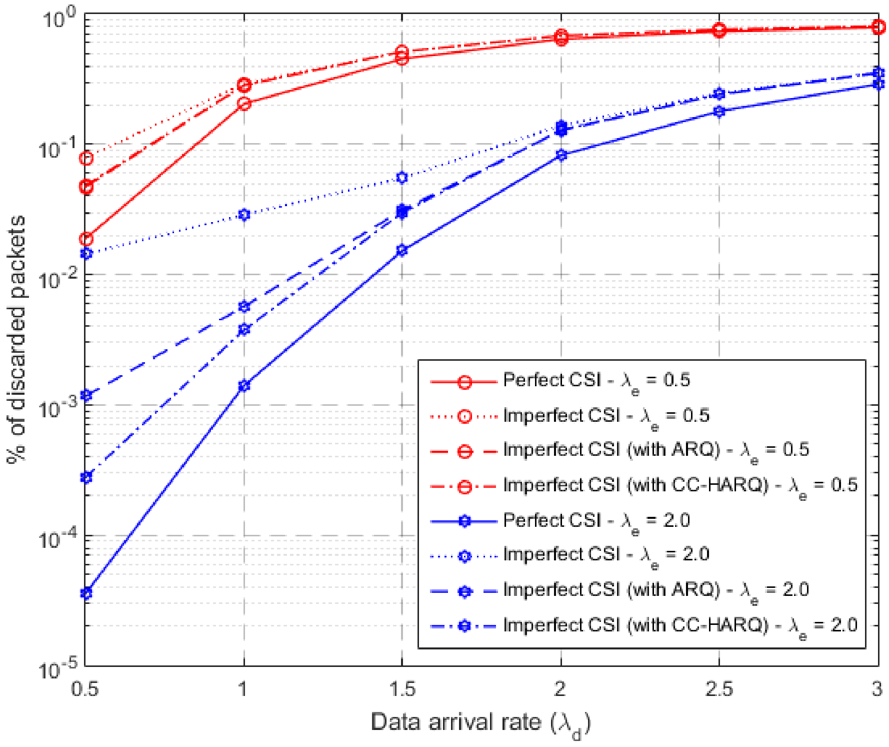


Figure 3.7 – Percentage of the discarded packets versus data arrival rate with different energy arrival rates between perfect and imperfect CSI scenarios.

3.5 Special Case: Packet Scheduling for EH-Transmitter under imperfect CSI

This section simplifies the model in Section 3.2 in a similar way to Section 2.5.1 as shown in Figure 3.8, where the offloading capabilities are removed. We aim now at studying the robustness of the derived optimal policy under imperfect CSI scenario. Again, the transmitter starts by performing channel estimation at the BS before taking the decision.

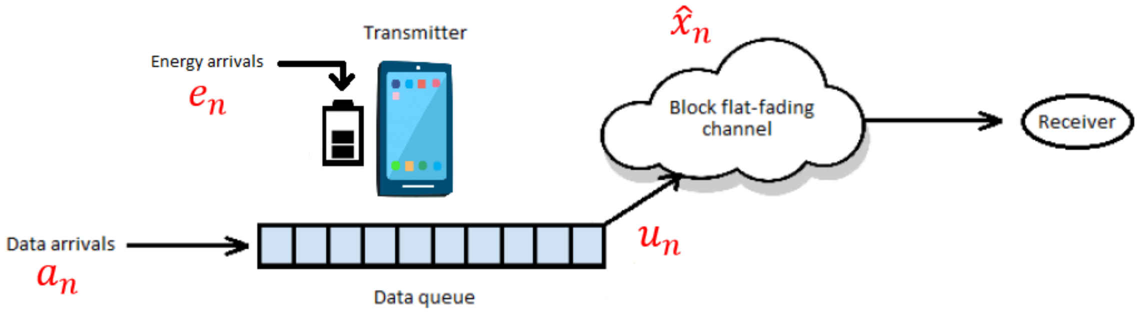


Figure 3.8 – Resource Scheduling for EH-Transmitter.

In this case, the energy consumption expressions are similar to those obtained in Section 2.5 by replacing x_n by the estimated channel gain $\hat{x}_n$ and T_s by $T_s - \tau$, i.e.

$$E(\hat{x}_n, u_n) = \left\lceil \frac{P(\hat{x}_n, u_n).(T_s - \tau)}{\mathcal{E}_U} \right\rceil \tag{3.16}$$

where

$$P(\hat{x}_n, u_n) = \frac{WN_0}{\hat{x}_n}\left(2^{\frac{u_n L}{W(T_s-\tau)}} - 1\right). \tag{3.17}$$

3.5.1 Numerical Results

In this section, our goal is to evaluate the proposed optimal policy when the transmitter relies on an estimated version of the channel state. The estimation phase duration is equal to $\tau = 10$ μs (1% of T_s), and a power of $P_{\text{tr}} = 4$ mW is used. The corresponding energy consumption for the estimation phase is thus $E = 40$ nJ which can be neglected to the energy unit, and therefore we assume $E_{ce} = 0$ e.u.

Comparison with other policies

In Figure 3.9, we compare the percentage of discarded packets between perfect and imperfect CSI scenarios. For low data arrival rate λ_d, the gap between both scenarios is large. Indeed, in our set up, the smallest channel mismatch probability is between 10^{-3} and 10^{-2} which implies that the percentage of discarded packets is necessary worse since as soon as the channel is over-estimated, the packets are dropped. However, when the data arrival rate increases, the buffer overflow can happen more often and the channel mismatch probability has less impact, which lead both scenarios to behave similarly.

In Figure 3.10, we compare the optimal and naive policies under perfect and imperfect CSI scenarios. For small energy arrival rate λ_e, the optimal policy under imperfect CSI is better than the naive policy with perfect CSI, because the latter sends packets without any adaptation to the energy and data arrivals, so energy shortage can happen more often and the number of discarded packets increases. For high energy arrival rate, imperfect CSI has stronger impact since the energy has to be controlled in a smarter way and knowing the channel accurately is more required.

In Figure 3.11, we compare the percentage of discarded packets for different estimation times τ (expressed in % of T_s). For low data arrival rate λ_d, increasing the estimation time leads to a better channel estimation, which slightly reduces the number of discarded packets since the impact of estimation error is high in this configuration (see Figure 4). Nevertheless, after a certain threshold, for instance $\tau \approx 5\%$, the number of discarded packets will increase because the remaining communication time of the slot is smaller. This leads to decrease the number of sent packets and so to increase the number of packets into the buffer, exhibiting thus more delay violation and buffer overflow. For high

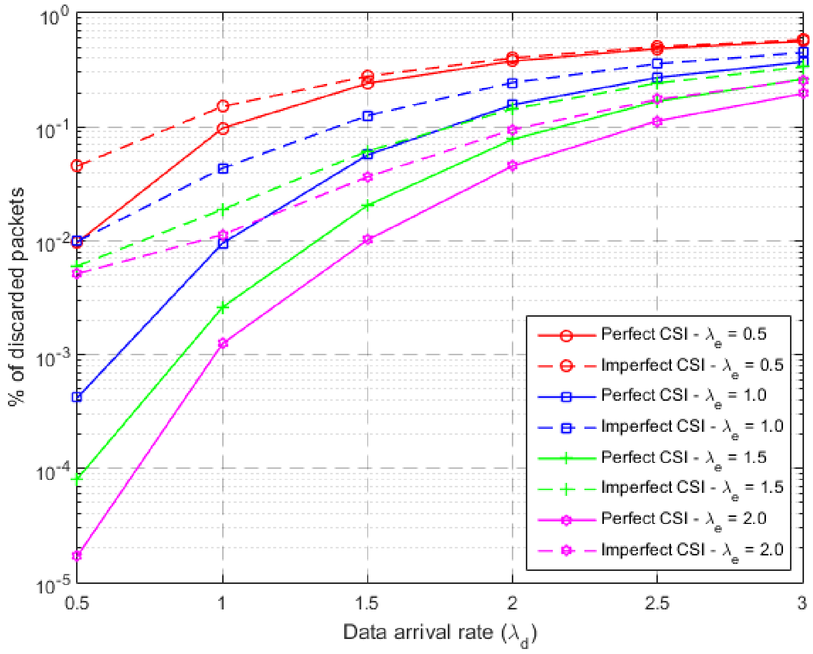


Figure 3.9 – Percentage of the discarded packets versus data arrival rate with different energy arrival rates between perfect and imperfect CSI scenarios.

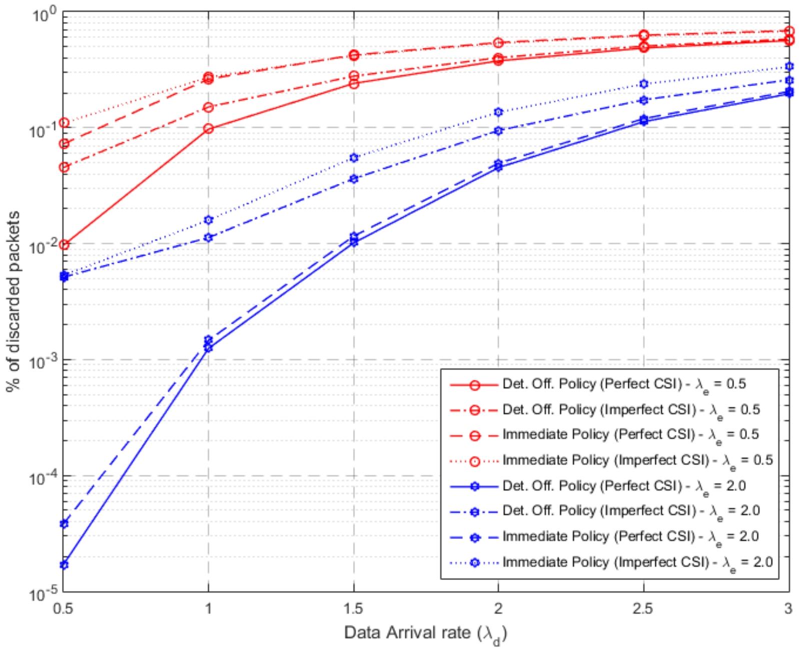


Figure 3.10 – Percentage of the discarded packets versus data arrival rate with different energy arrival rates for different policies and between perfect and imperfect CSI scenarios.

data arrival rate, we know that the estimation accuracy is not required (see Figure 3.9). Therefore, increasing the estimation time directly decreases the performance since the system has less time for data packets transmission.

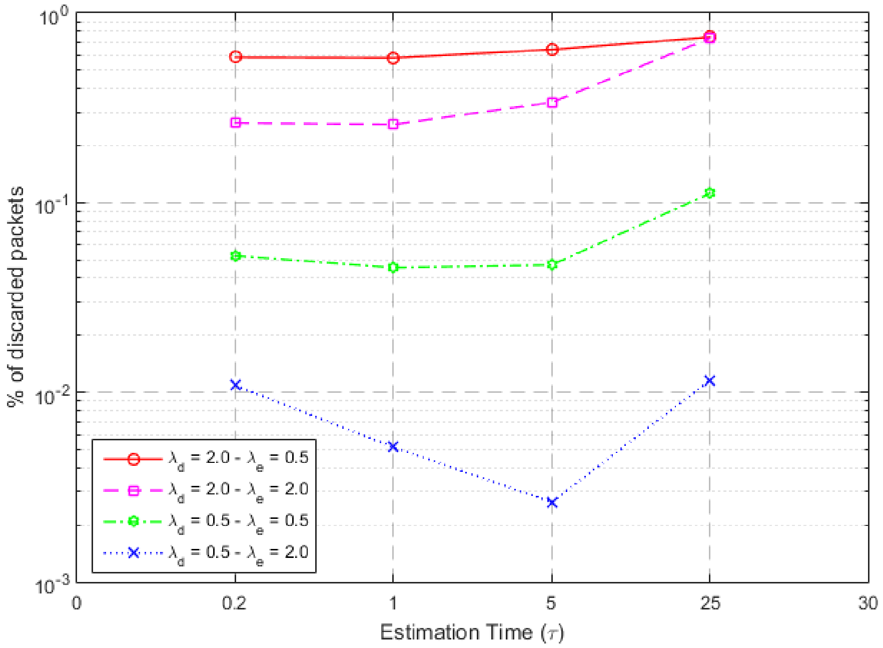
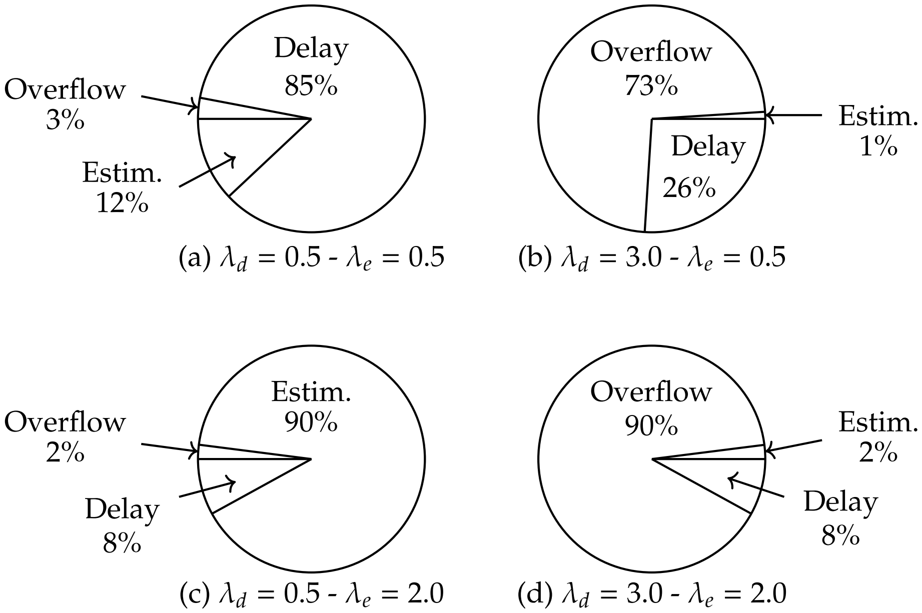


Figure 3.11 – Percentage of the discarded packets versus the estimation time τ (expressed in % of T_s) with different data and energy arrival rates.

In Figure 3.12, we display the nature of discarded packets in percentage due to delay violation, buffer overflow and channel mismatch with different data and energy arrival rates. The number of discarded packets due to channel mismatch is significant for low data arrival rate because the delay violation or the buffer overflow can happen less often. However, for high data arrival rate, the number of packets discarded due to channel mismatch is negligible and the policy behaves approximately in the same way for perfect and imperfect CSI. Nevertheless, the imperfect CSI degrades the whole system (on the delay violation and buffer overflow) since a part of the time slot is now devoted to perform the estimation rather than the transmission.

Re-transmission with ARQ protocols

In Figure 3.13, ARQ and CC-HARQ are implemented with at most two transmissions (one re-transmission). When λ_e is low, using ARQ and CC-HARQ is not efficient because re-transmitting the same packet twice consumes energy while it is not available in large quantities. However, when λ_e is large, these two protocols significantly improve the performance by reducing the number of discarded packets due to imperfect CSI.



(a) $\lambda_d = 0.5$ - $\lambda_e = 0.5$

(b) $\lambda_d = 3.0$ - $\lambda_e = 0.5$

(c) $\lambda_d = 0.5$ - $\lambda_e = 2.0$

(d) $\lambda_d = 3.0$ - $\lambda_e = 2.0$

Figure 3.12 – Percentage of the discarded packets due to delay violation, buffer overflow, and channel mismatch with different data and energy arrival rates.

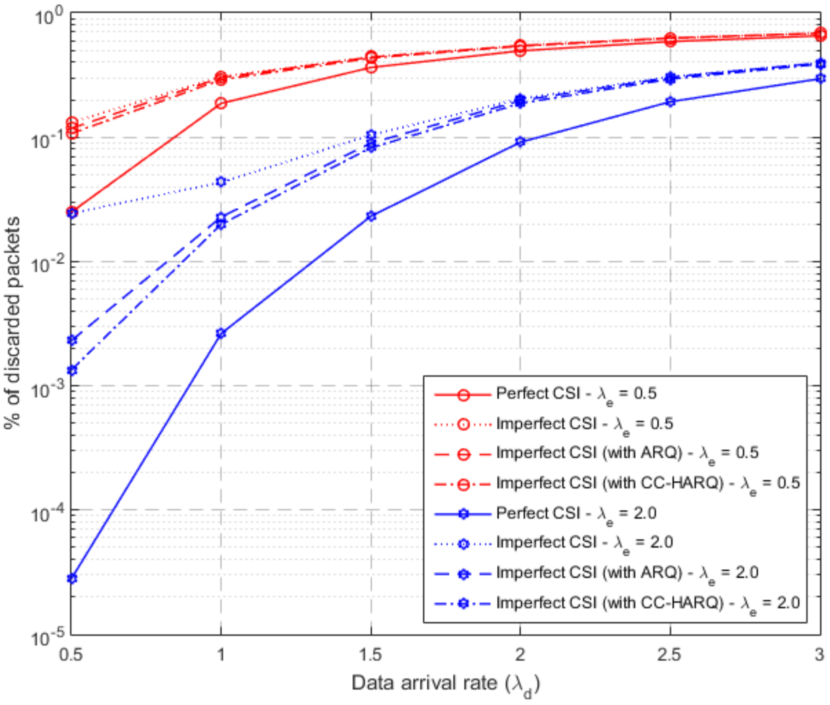


Figure 3.13 – Percentage of the discarded packets versus data arrival rate with different energy arrival rates between perfect and imperfect CSI scenarios.

3.6 Conclusion

In this chapter, we have evaluated the impact of imperfect CSI without and with ARQ protocols on the optimal policy in terms of additional packet loss due to the channel estimation time and errors. We have also studied the scheduling special case of the general one under imperfect CSI. The numerical results showed that an appropriate balance is required between the channel estimation accuracy and the transmission period in order to reduce the discarded packets depending on the available energy, energy and data arrivals.

The material presented in this Chapter has been published in [J1] and [C4].

Chapter 4

Joint Resource Scheduling and Computation Offloading with unknown CSIT

4.1 Introduction

As we have seen in the previous chapters, the CSI is an important information for our resource scheduling and computation offloading problem with a single EH device served by its BS. It affects the decisions and therefore the performance of the whole system. Indeed, in Chapter 2, we have considered that the CSI is known perfectly at the EH device before taking any decision which is an ideal assumption, usually unpractical in real wireless system. We have relaxed this assumption in Chapter 3 considering that the channel is estimated before taking the decision at the cost of losing time and energy since a part of the time slot is devoted to perform channel estimation. In this chapter, we are interested in studying a much complicated problem assuming that the current CSIT is not known to the mobile device and the only available information is an outdated CSIT, i.e. the CSIT of the previous slot. But we assume correlation between channels over timeslot.

Lately, some research has concentrated on designing new approaches to cope with the lack of CSI in the communication system. For instance, in [88], the problem of scheduling in Multi-Round Multi-User Multiple-Input Multiple-Output (MIMO) systems was considered under outdated CSI. The authors showed that an adequate training is needed to achieve the Degree-of-Freedom of these schemes. In [89], the problem of user selection for downlink of MIMO cognitive radio network was studied assuming that the cross interference channel between the BS and the primary users is not known. The proposed selection scheme, based on a best-effort interference mitigation, aimed at either minimizing the transmission rank of secondary users or the rank-power product for a given secondary user data rate requirement. In [90], the problem of layered interference network was studied under delayed CSI at all nodes. The authors claimed that the achievable Degree-of-Freedom under this assumption scales with the number of users. In [91], the problem of crosstier interference between femto and macro cells BS was studied. The authors proposed a generalized inverse precoder at the macro BS that can achieve perfect interference nulling to offloaded femto-cell users under perfect CSI and significant interference suppression while maintaining zero macro inter-user interference under imperfect CSI. In [92], the problem of offloading in heterogeneous cellular networks from congested macro BS to less congested femto BSs using Non-Orthogonal Multiple Access (NOMA). The authors analyzed the offloading process under imperfect CSIT and claimed that the outage probability increases compared to the perfect CSI scenario, but implementing NOMA in such situation can reduce this outage probability. In [93], the scheduling and offloading problem under imperfect CSI is studied. The authors proposed an optimal resource scheduling and dynamic offloading to balance throughput and fairness using Lyapunov optimization and Lagrange dual decomposition techniques. In [94], the resource allocation and computation task offloading optimization problem was studied in MIMO-based MEC systems considering both perfect and imperfect CSI. The goal was to minimize the energy consumption subject to available radio and computing resources and allowable latency. The results showed that the proposed design can achieve significant energy saving.

In this chapter, we assume that the current channel state is not available at the device side before taking the action neither perfectly nor imperfectly. In this case, the decision is made based on the previous encountered channel which is assumed to be correlatec with the current one. To our best knowledge, such situation is not studied before especially for EH-MEC systems. This assumption adds a new type of packet loss, called channel mismatch. Therefore, we aim now at minimizing the number of discarded packets due to the strict delay violation, the buffer overflow, and the channel mismatch. We formulate the problem as an MDP and solve it using PI algorithm. We compare the obtained optimal offline stationary policy as in previous chapters with the immediate scheduling and the two policies taking into account either local processing or offloading decisions.

The chapter is organized as follows. In Section 4.2, we describe the system model. In Section 4.3, we formulate the MDP problem by defining its states, actions and transition probabilities and we propose a PI algorithm to solve it. In Section 4.4, we provide and analyze the numerical results. Finally, we give some concluding remarks in Section 4.5.

4.2 System Model

The same MEC system model from Section 2.2 is adopted (Figure 4.1). However, since the CSI is neither known nor estimated, we adopt here a correlated model for the channel states to keep track of the fluctuations statistically.

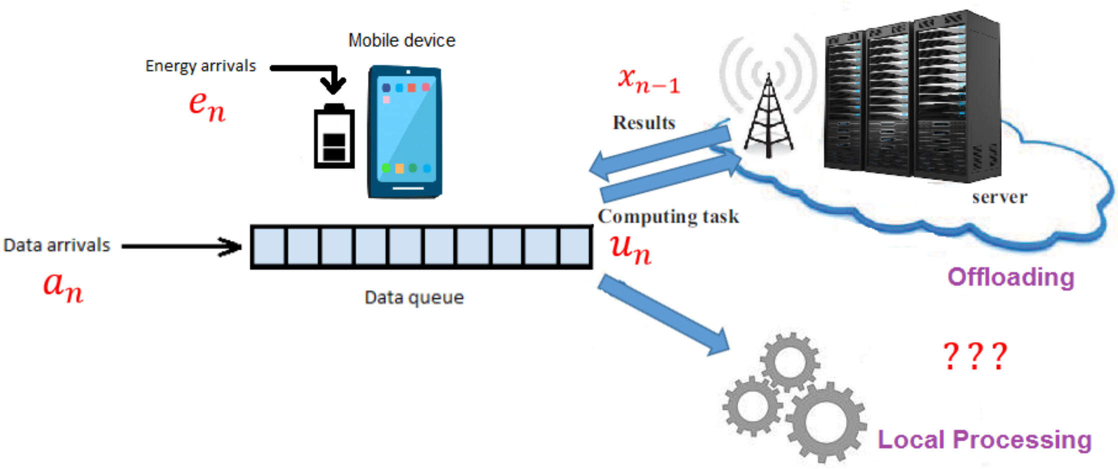


Figure 4.1 – MEC system with an EH mobile device.

4.2.1 Channel model

Recalling the channel model described in Section 2.2.3, we consider here that the channel gains are time-correlated following a Markovian model, i.e. the current channel state depends only on the previous channel state. The transition probability from a channel state j at time slot n to a channel state i at time slot $n+1$ is given by

$$p(x_{n+1} = i | x_n = j) = \frac{(1-\rho)^{|i-j|}}{\sum_{k=0}^{|\mathcal{X}|-1} (1-\rho)^{|k-j|}}, \tag{4.1}$$

where $\rho \in [0, 1[$ is the correlation coefficient. We assume outdated CSIT, i.e., only x_{n-1} is known when taking decision at time slot n. Packets can thus be lost due to a **channel mismatch** since the mobile user takes the scheduling decision based on the previous (and so wrong) channel state.

4.2.2 Execution decisions and related consumed energy

When CSI is unknown, at the beginning of time slot n, the consumed energy for **Local** and **Remote** decisions become:

- **Local processing:** This decision is not affected at all since using the local processor is independent from the channel state. Therefore, the same expression under perfect CSI scenario is applicable here.

- **Remote processing:** As the current channel state x_n is not known, packets are sent at a rate tuned according to the previous encountered channel state x_{n-1}. As a consequence, the energy consumption, defined in Section 2.2.4 as an integer multiple of the e.u is obtained by replacing x_n by x_{n-1} as

$$E_o(x_{n-1}, u) = \left[\frac{u}{\mathcal{E}_U}\left(\frac{L.P_t}{W_{UL}.\log_2\left(1+\frac{P_t.x_{n-1}}{W_{UL}.N_0}\right)} + T_w.P_w + \frac{L_{DL}.P_r}{W_{DL}.\log_2\left(1+\frac{P_s.x_{n-1}}{W_{DL}.N_0}\right)}\right)\right] \quad (4.2)$$

The first term in the Right Hand Side (RHS) of Eq. (4.2) corresponds to the energy to send the packets. In fact, each packet has L bits and the number of bits sent in one second is the rate tuned with the only Shannon capacity we know, i.e., $W_{UL}.\log_2\big(1+\frac{P_t.x_{n-1}}{W_{UL}.N_0}\big)$ adapted to the previous realization of the channel. Obviously, rate mismatch may occur and lead to error. For the return link (downlink) corresponding to the third term of the RHS of Eq. (4.2), whereas the BS may adapt its rate to the current channel (since it can estimate the current channel via the training sequence of the uplink received packet), we force the BS to select the downlink rate with the channel that the transmitter has used for optimizing its action (i.e., the rate depending on x_{n-1} since the transmitter only knows x_{n-1} at the beginning of the time slot when it makes its decision).

As in Chapter 2, this offloading procedure has to be done within the time slot leading to the following constraint with the channel state x_{n-1}

$$u\left(\frac{L}{W_{UL}.\log_2\left(1+\frac{P_t.x_{n-1}}{W_{UL}.N_0}\right)} + T_w + \frac{L_{DL}}{W_{DL}.\log_2\left(1+\frac{P_s.x_{n-1}}{W_{DL}.N_0}\right)}\right) \leqslant T_s. \quad (4.3)$$

Notice that $W_{DL}, W_{UL}, N_0, L_{DL}, P_s, T_w$ are pre-fixed parameters. Forcing equality in Eq. (4.3) enables us to find P_t with respect to u and x_{n-1}.

4.3 Problem Formulation and Resolution

We aim at finding an optimal policy μ that minimizes the number of discarded packets due to buffer overflow, delay violation and **channel mismatch**. The policy μ specifies now the processing decisions (local processing, offloading or staying idle), the number of packets u to be scheduled and the corresponding **transmission power** P_t at each time slot, based on the past system states and actions. In this section, we characterize the appropriate states and actions and present the MDP formulation of our problem. We define then the transition matrix and the reward of this MDP and propose an offline policy iteration algorithm to solve it.

4.3.1 State Space

The state space $\mathcal{S}$ is the set of $\mathbf{s} = (\mathbf{k}, b, x)$ where

- $\mathbf{k} = [k_1, \cdots, k_{B_d}]$ is the vector indicating the age of each packet in the data buffer,
- b is the current battery level, and
- x is the previous channel gain.

4.3.2 Action Space

The action space $\mathcal{V}$ denotes the processing decisions (local processing, remote processing or staying idle), the number of packets u that the mobile device schedules during a time slot and the corresponding transmission power P_t. Recall that U_ℓ is the maximum number of packets that can be processed locally during a time slot and U_o be the maximum number of packets that can be offloaded during a time slot. It is obtained from Eq. (4.3) with equality, using the maximum transmit power $P_{\max}$ and the best channel gain $x_{\max} = \max_{x \in \mathcal{X}} x$. For each remote processing action, the mobile device can choose the transmission power among those calculated using Eq. (4.3) or to transmit using $P_{\max}$ if possible. Finally, the action space is finite with cardinality $V = |\mathcal{V}| = U_\ell + U_o \times (|\mathcal{X}| + 1) + 1$. The actions are ordered and the m-th action is as follows:

- if $m = 0$, idle processing is considered and $u = 0$.
- if $m = m_\ell$ with $m_\ell \in \{1, \cdots, U_\ell\}$, local processing is applied and $u = m_\ell$.
- if $m = m_o$ with $m_o \in \{U_\ell + 1, \cdots, V - 1\}$, remote processing is applied. Note that each block of $|\mathcal{X}| + 1$ elements transmit the same number of packets but using different transmission power.

At time slot n, $v_n \in \{0, \cdots, V - 1\}$ corresponds to the decided action specifying the transmission power P_{v_n} and the number of packets u_{v_n} processed either locally or remotely.

4.3.3 Transition Matrix

The state transition probability of the MDP is defined as Section 2.3.4 by

$$p(\mathbf{s}'|\mathbf{s}, v) = p(\mathbf{k}'|\mathbf{k}, b, v).p(b'|b, x, v).p(x'|x), \tag{4.4}$$

where the channel x is now unknown and $p(x'|x)$ is the distribution of the future channel states x' given x exhibited using Eq. (4.1).

4.3.4 Cost

At a given time slot $n \in \{0, \cdots, N\}$, the system state is denoted by $\mathbf{s}_n = (\mathbf{k}_n, b_n, x_{n-1})$ and the action is $\mu(\mathbf{s}_n) = v_n$. To minimize the average number of discarded packets under policy μ, we define the cost function as

$$\overline{D}(\mu) = \lim_{N \to \infty} \frac{1}{N} \mathbb{E}^{\mu} \left[\sum_{n=1}^{N} (\varepsilon_d(\mathbf{s}_n, v_n) + \varepsilon_o(\mathbf{s}_n, v_n) + \varepsilon_c(\mathbf{s}_n, v_n)) \right], \tag{4.5}$$

where $\varepsilon_d(\mathbf{s}_n, v_n)$, $\varepsilon_o(\mathbf{s}_n, v_n)$ and $\varepsilon_c(\mathbf{s}_n, v_n)$ are the instantaneous number of discarded packets due to delay violation, buffer overflow and channel mismatch, respectively.

The first two types of packet loss are derived similarly to Eq. (2.12) and (2.14).
The channel mismatch occurs since the mobile user does not know the current channel state x_n, and takes decisions based on the knowledge of the previous channel state x_{n-1}. Therefore, the outdated CSI generates another type of errors, called **Non-Compatible Rate**. This situation arises when the mobile user decides to offload at the beginning of time slot n with a rate $R(P_t, x_{n-1})$ higher than the optimal channel rate. An error also occurs when the BS uses the rate $R_{DL}(P_s, x_{n-1})$ higher than the current channel rate.

The rate $R(P_t, x_{n-1})$ is calculated by knowing only x_{n-1} and is equal to

$$R(P_t, x_{n-1}) = \log_2\left(1 + \frac{P_t x_{n-1}}{W_{UL} N_0}\right) \tag{4.6}$$

while the optimal rate R_n for the transmission according to the current channel state x_n is

$$R_n = \log_2\left(1 + \frac{P_t x_n}{W_{UL} N_0}\right). \tag{4.7}$$

For the downlink, $R_{DL}(P_s, x_{n-1})$ is equal to

$$R_{DL}(P_s, x_{n-1}) = \log_2\left(1 + \frac{P_s x_{n-1}}{W_{DL} N_0}\right) \tag{4.8}$$

while the current achievable rate is

$$R_{n,DL} = \log_2\left(1 + \frac{P_s x_n}{W_{DL} N_0}\right). \tag{4.9}$$

Therefore, packets are lost when $R(P_t, x_{n-1}) > R_n$ or $R_{DL}(P_s, x_{n-1}) > R_{n,DL}$. Both conditions are equivalent to $x_{n-1} > x_n$.

Finally, the total number of discarded packets is

$$\varepsilon_c(\mathbf{s}_n, \nu_n) = u_n \times \text{Prob}(x_{n-1} > x_n), \tag{4.10}$$

and the MDP optimization problem is as follows

$$\mu^\star = \arg\min_{\mu} \overline{D}(\mu). \tag{4.11}$$

4.3.5 Optimal Policy Computation

We solve our MDP problem using PI algorithm. Similarly to known channel cases, the obtained policy is an optimal offline deterministic policy, that depends on the channel, energy arrival and data arrival *a priori* distributions.

4.4 Numerical Results

We evaluate numerically the optimal policy designed for our scheduling/offloading problem. We consider the system described in Section 4.2 and choose the same design parameters as Section 2.2.

In Figure 4.2 and Figure 4.3, we plot the percentage of discarded packets versus the data arrival rate λ_d for two correlation factors, $\rho = 0.99$ and $\rho = 0.75$, respectively. Two values of energy arrival rate are also considered, $\lambda_e = 1$ (red) and $\lambda_e = 2.0$ (blue). We compare the performance of the optimal policy to the three different policies: **immediate** policy, **local** policy and **Remote** policy. We observe that the proposed policy outperforms the other policies as it can adapt its processing to the energy and data arrivals. More precisely, when the data arrival rate λ_d increases, the number of discarded packets for any policy increases because the buffer overflow happens more often. We also see that for both values λ_e, the **local** policy approaches the optimal policy as channel correlation decreases since the probability of losing a packet during the transmission increases. When the channel correlation is high, the performance of the **local** policy decreases when the data arrival rate increases due to the limited capacity of the mobile device processor. Moreover, the gap between the optimal and the **immediate** policy increases when ρ decreases. Finally, the performance of the **Remote** policy decreases when ρ decreases because packets can be lost at each decision due to channel fluctuation.

In Figure 4.4 and Figure 4.5, we plot the average consumed energy and the average battery state respectively versus the data arrival rate λ_d for $\rho = 0.99$ and energy arrival rates $\lambda_e = \{1, 2\}$. We observe that **local** and **immediate** policies lead to the highest energy consumption since processing packets locally consume more energy, draining thus the battery level. The optimal proposed policy consumes approximately the same energy amount as the **Remote** policy while sending more packets. Indeed, it ensures a better sustainable communication with less discarded packets by optimally using the available energy, leading hence to a higher energy level in the battery.

In Figure 4.6, we display the percentage of processing decisions of the optimal policy at $\lambda_d = \{1, 2\}$ for $\rho = 0.99$ in Figure 4.6(a)(c), and for $\rho = 0.75$ in Figure 4.6(b)(d). When the channel correlation decreases, the system executes more packets locally to minimize the number of discarded packets. When the energy is available (λ_e large), the system reduces the offloading decisions in order to prevent the packet loss due to channel mismatch. However, when λ_e is small, the system is forced to use offloading because executing packets locally is costly in terms of energy.

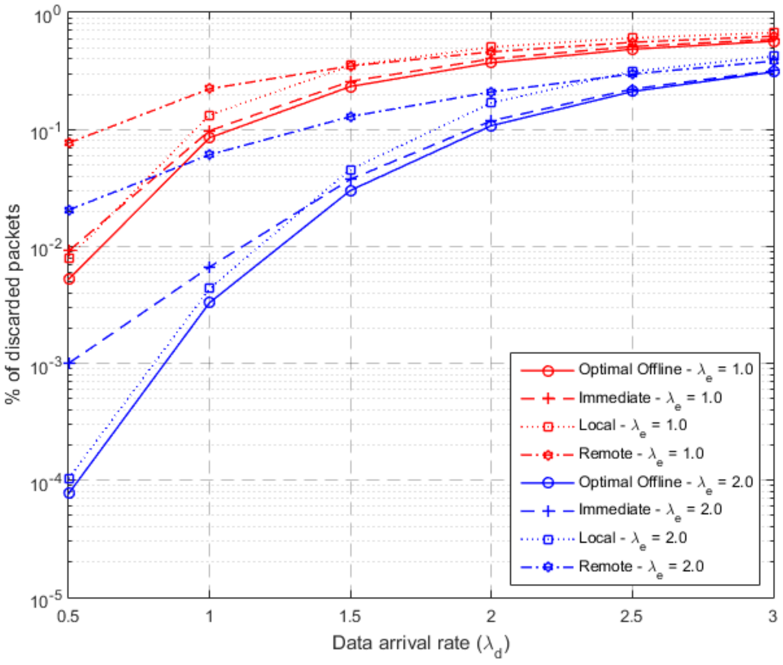


Figure 4.2 – Percentage of the discarded packets versus data arrival rate for energy arrival rates $\lambda_e = \{1, 2\}$ and $\rho = 0.99$.

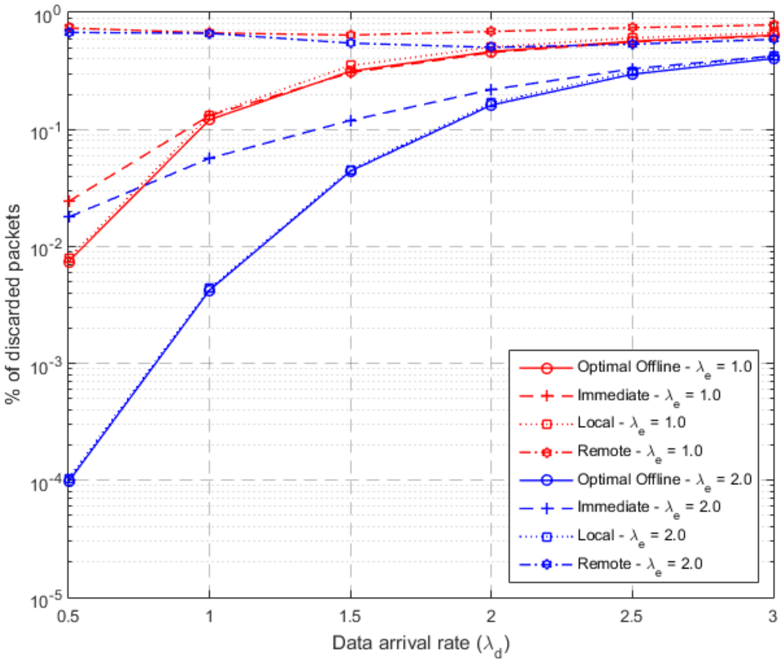


Figure 4.3 – Percentage of the discarded packets versus data arrival rate for energy arrival rates $\lambda_e = \{1, 2\}$ and $\rho = 0.75$.

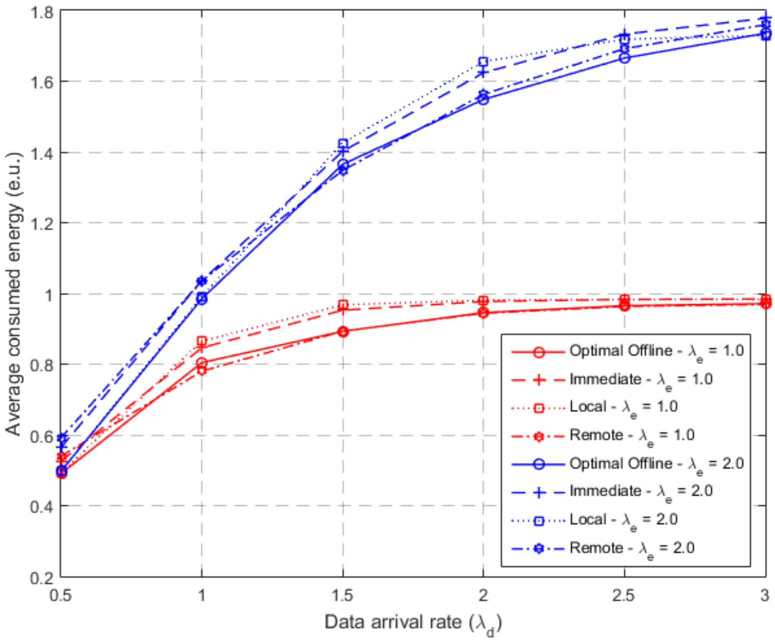


Figure 4.4 – Average consumed energy versus data arrival rate for energy arrival rates $\lambda_e = \{1, 2\}$ and $\rho = 0.99$.

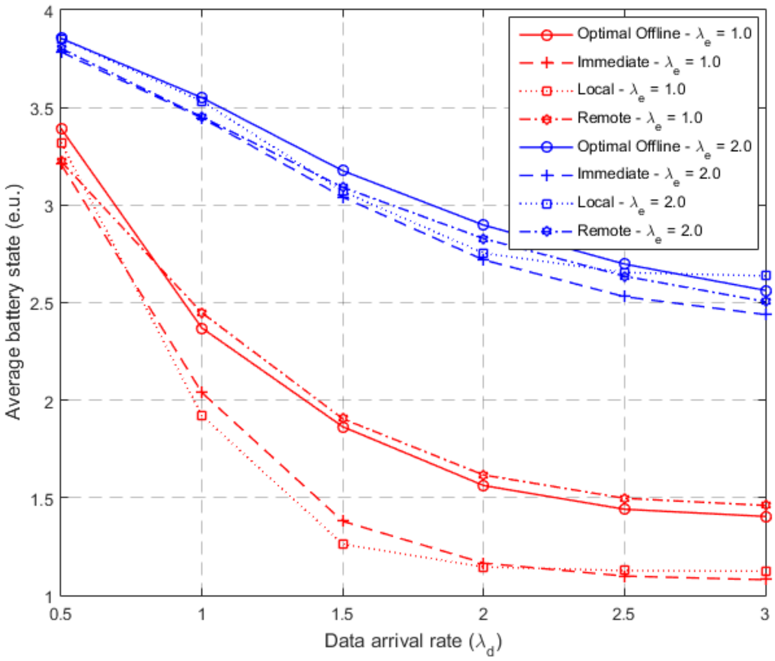
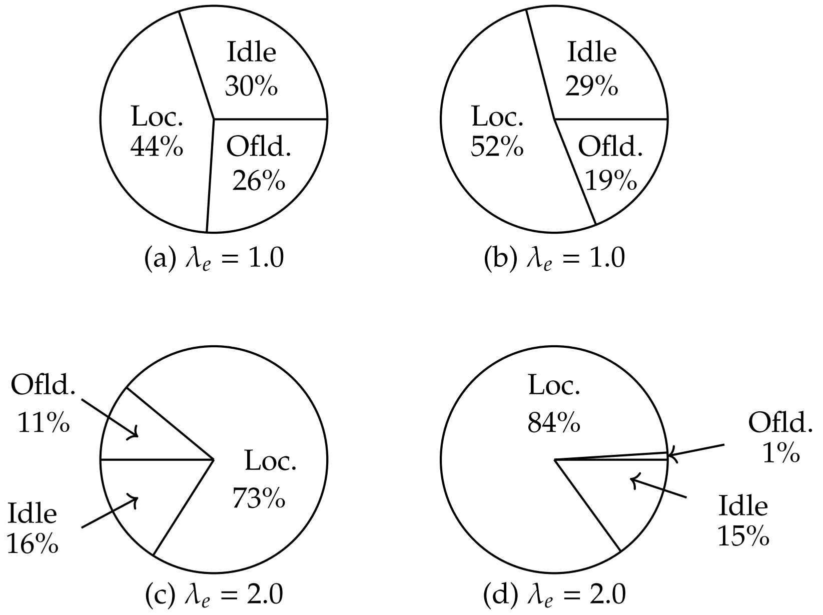


Figure 4.5 – Average battery state versus data arrival rate for energy arrival rates $\lambda_e = \{1, 2\}$ and $\rho = 0.99$.



(a) $\lambda_e = 1.0$ (b) $\lambda_e = 1.0$

(c) $\lambda_e = 2.0$ (d) $\lambda_e = 2.0$

Figure 4.6 – Percentage of processing decisions for $\rho = 0.99$ (a) and (c), and $\rho = 0.75$ (b) and (d).

4.5 Conclusion

We have addressed resource scheduling-offloading problem for an energy harvesting mobile device to its serving resourceful base station under a strict delay constraint and without CSIT. We have proposed an optimal policy to minimize the packet loss rate using MDP framework. The optimal policy, which decides the number of packets to be processed locally or remotely, relies on the channel, energy arrival and packet arrival distributions as well as the current battery level, the current packets' delays, and the outdated channel gain.

The material presented in this Chapter has been published in [C5].

Conclusions and future work

The main objective of this thesis was to propose optimal data transmission policies for 5G mobile systems having the following characteristics:

- **Offloading capabilities:** The MU can execute some of his tasks at its serving BS. It is an important feature for future mobile systems to overcome the devices' limitations in terms of processing capacity and storage.
- **Energy harvesting:** The MU relies on collecting energy from renewable sources to power his communications and computations. It is also a crucial aspect in future mobile systems to tackle the fast depletion of the devices' batteries.
- **Strict delay:** The MU should execute his tasks before a certain deadline. This is an extremely challenging requirement in future mobile systems requiring more and more stringent delays and very low error rates.

In Chapter 1, we started by describing the underlying motivations for integrating EH and computation offloading into communication systems. We presented some of the available EH sources and offloading options, and chose the most suitable schemes for our problem. Finally, we provided the framework of MDP and detailed various algorithms used to solve this type of problems.

In Chapter 2, we investigated the joint optimization of resource scheduling and computation offloading for mobile networks, where EH-MU are wirelessly connected to nearby BS, which can be endowed with some computational capabilities. The main contribution of this work was the introduction of the strict delay constraint instead of the average delay constraint used in the state-of-the-art. Two special cases of this general problem were also presented. In all three cases, we found under the perfect CSI assumption that our optimal policy outperforms the other policies by adapting the number of executed packets to the buffer state, the available energy in the battery and the channel conditions.

In Chapter 3, we assessed the previously obtained optimal policy considering imperfect CSI with and without ARQ protocol. We found that our optimal policy remains robust compared to the other policies, and that an appropriate trade-off should be kept between channel estimation accuracy and the communication period to maintain good performance.

In Chapter 4, we addressed the same problem as in Chapter 2, but considering unknown CSI. We showed that it is possible to achieve good performance compared to the other policies by relying on an outdated version of the channel state.

Future Works

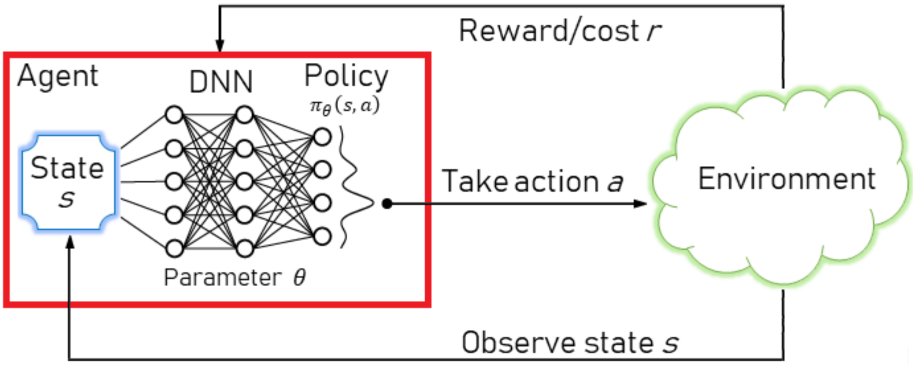
The following proposals deserve to be addressed in future works.

Deep Reinforcement Learning Approach

Solving the optimization problem using DP yields the optimal solution, but its derivation becomes impractical when the system state is large. One way to overcome this constraint is by using function approximation techniques to estimate the action-value function. This function, denoted $Q_\mu(s,u)$, represents the expected cost starting from state s and taking action u under the policy μ [46]. It is given by

$$Q_\mu(s,u) = \mathbb{E}_\mu[C_i|s_i = s]$$

The approximation can be done by using a non-linear function to represent the Q-function via a Neural Network (NN). This scheme is known as Deep Reinforcement Learning (DRL) where the NN takes the state vector as input and outputs the Q-value for each possible action. The DRL field is receiving increasing interest in various systems and applications [95–100].



DRL Agent.

To this end, let $Q(s,u;\theta)$ be an approximate action-value function with parameter θ. The NN or equivalently the Deep Q-Network (DQN) can be trained to learn the weights θ of $Q(s,u;\theta)$ by minimizing the Mean Squared Error (MSE) between:

- **Target** = $c(s_n, u_n) + \min_u Q(s_{n+1}, u; \theta)$
- **Prediction** = $Q(s_n, u_n; \theta)$

Note that $Q(s_{n+1}, u; \theta)$ and $Q(s_n, u_n; \theta)$ are predicted through the Q-Network for the input s_{n+1} and s_n respectively. Next, the loss function may be optimized using *Adam* optimizer [101] with a learning rate α.

However, RL techniques tend to diverge when used with NN. So, in order to solve this problem, two methods are proposed [102]:

- **Experience Replay (ER):** The system will interact with the environment and store the experience $(s_n, u_n, c(s_n, u_n), s_{n+1})$ in replay memory $\mathcal{M}$. When training the network, random mini-batches of size B_m from $\mathcal{M}$ are used instead of the most recent transition, which will greatly improve the stability. In this way, the system will be more robust by learning from more varied past experiences, instead of learning from the immediate interaction with environment.
- **Fixed target Network:** This method makes use of a second network during the training phase to predict $Q(s_{n+1}, u; \theta)$. The reason behind this method is that the Q-network's values shift at every step of training. The value estimations can easily diverge if the set of values used to adjust the network values is constantly shifting, and the network can become unstable by falling into feedback loops between the target and estimated Q-values. So, in order to mitigate that risk, the second network's weights are fixed, and only periodically or slowly updated to the primary Q-network values. In this way, training can proceed in a more stable manner.

During the training phase, the system makes use of an ϵ-greedy strategy to ensure adequate exploration of the state space. Therefore, the system chooses an action $u_n = \min_u Q(s_n, u; \theta)$ with probability $1 - \epsilon$ and selects a random action with probability ϵ.

The deep Q-learning or algorithm is presented in Algorithm 3.

Nevertheless, the Q-learning algorithm performs poorly in some stochastic environments, due to the large over-estimations of action values. These over-estimations result from a positive bias that is introduced because Q-learning uses the maximum action value as an approximation for the maximum expected action value. The reason is that the same samples are used to estimate the action-value function and to decide which action is the best, i.e., with lowest expected cost. As a consequence, at the beginning of the training the system does not have enough information about the best action to take. Therefore, taking the minimum Q value (which is noisy) as the best action to take can lead to false positives. If non-optimal actions are regularly given a lower Q value than the optimal

Algorithm 3 Deep Q-Learning algorithm

```
Initialize replay memory $\mathcal{M}$ to capacity $M$
Initialize Q-network with random weights $\theta$
Initialize target Q-network with random weights $\theta'$
for t = 1, $T$ do
    Generate random state $s_0$
    for n = 0, N do
        $u_n = \arg\min_u Q(s_n, u; \theta)$ with probability $1 - \epsilon$
        Otherwise, $u_n$ is selected randomly
        Execute $u_n$ and observe $c(s_n, u_n)$ and $s_{n+1}$
        Store experience $(s_n, u_n, c(s_n, u_n), s_{n+1})$ in $\mathcal{M}$
        Sample random minibatch of $B_m$ transitions from $\mathcal{M}$
        Set the target to $c(s_n, u_n) + \min_u Q(s_{n+1}, u; \theta')$
        Perform *Adam* update on $\theta$
    end for
    Update target network, i.e. $\theta' = \theta$
end for
```

best action, the learning will be complicated, because the accuracy of Q values depends on what action is tried and what neighboring states is explored.

Thus, to overcome the over-estimation problem of the Q-learning algorithm, **Double Deep Q-Network (DDQN)** is proposed [103] where two networks are used to decouple the action selection from the target Q value generation:

- use the primary DQN network to select what is the best action to take for the next state (the action with the lowest Q value).
- use the target network to calculate the target Q value of taking that action at the next state.

Mathematically, this is reflected by the following rule

$$\textbf{Target} = c(s_n, u_n) + Q(s_{n+1}, \arg\min_u Q(s_{n+1}, u; \theta); \theta')$$

Moreover, ER mechanism allows the RL agent to remember and reuse experiences, i.e., transitions, from the past. In particular, transitions are uniformly sampled from the replay memory $\mathcal{M}$. However, this approach simply replays transitions at the same frequency as that the agent was originally experienced, regardless of their significance. Therefore, to learn more efficiently, a **Prioritized Experience Replay (PER)** framework was proposed [103] in order to replay important transitions more frequently.

In other words, the idea is to take in priority experience where there is a big difference between the prediction and the target, since it means that there is a lot to learn from it. Therefore, the priority γ of entry i in the replay memory is given by:

$$\gamma_i = |\text{target} - \text{prediction}| + \epsilon$$

where ϵ is a constant to ensure that no experience has 0 probability to be taken.

However, performing just greedy prioritization, will lead to always training on the same experiences (that have big priority), and thus over-fitting. Therefore, stochastic prioritization is introduced to generate the probability $P(i)$ of being chosen for a replay

$$P(i) = \frac{{\gamma_i}^{\alpha}}{\sum_k {p_k}^{\alpha}}$$

where α is a hyper-parameter used to introduce some randomness in the experience selection. As consequence, during each time step, a batch of samples with this probability distribution is used to train the network.

Indeed, with normal experience replay, the experiences are selected randomly. Therefore, there is no bias, because each experience has the same chance to be taken, so the weights can be updated normally. But with priority sampling, bias toward high-priority samples is introduced, and updating the weights normally can cause over-fitting. As a consequence, the weights will be updated with only a small portion of experiences considered to be really interesting. Therefore, to correct this bias, Importance Sampling (IS) weights is used to adjust the updating by reducing the weights of the often seen samples.

$$IS = \left[\frac{1}{MP(i)}\right]^{\beta}$$

where β is used to control how much the IS will affect the learning.

Following this, the weights corresponding to high-priority samples have very little adjustment (because the network will see these experiences many times), whereas those corresponding to low-priority samples will have a full update.

Thus, this kind of DRL techniques may be used to solve the dimensionality issues of our problem and should be done in a future work.

Non-Orthogonal Multiple Access protocols

In practical MEC schemes, several users can be served by the same BS, where each of them has its own tasks and attempts to execute its packets remotely within a reasonable delay. This scenario can be managed using multiple access protocols.

Over the past decades, Orthogonal Multiple Access (OMA) has been used, in which every user can exploit orthogonal communication resources within a particular:

- frequency band, like Frequency Division Multiple Access (FDMA) in the first Generation of mobile networks (1G)
- time slot, like Time Division Multiple Access (TDMA) in 2G
- code, like Code Division Multiple Access (CDMA) in 3G

Despite the fact that OMA techniques can offer good performance even with simple receivers due to the absence of mutual interference between users in an ideal environment (as done for 4G that uses Orthogonal Frequency Division Multiple Access (OFDMA)), NOMA s a key technology for pursuing the growth of the trafic.

In NOMA, several users can use non-orthogonal resources simultaneously. NOMA schemes can be classified into two main categories [104]:

- **Power-based multiplexing:** several users are assigned different power coefficients depending on their channels' condition. On the transmitter side, the information signals of multiple users are superimposed. On the receiver side, Successive Interference Cancellation (SIC) is applied to decode the signals one at a time until the intended user's signal is reached.
- **Code-based multiplexing:** multiple users are assigned different codes and are multiplexed on the same time-frequency resources.

In fact, the advantages of NOMA over OMA can be highlighted in the following aspects [105]:

- **Spectral efficiency and throughput:** In OMA (OFDMA), a specific frequency resource is assigned to each user, regardless of its channel state, which results in low spectral efficiency and throughput in the entire system. In contrast, NOMA assigns the same frequency resource to several mobile users, with good and bad channel conditions, at the same time. Hence, the resource allocated to the weak user is also used by the strong user, and the interference can be mitigated through SIC processes at the user's receivers.
- **User fairness, low latency, and massive connectivity:** Scheduling in OMA prioritizes the user with a good channel state over the user with a poor one, which leads to an equity issue and high latency. This approach cannot therefore cope with massive connectivity. This problem can be solved in NOMA because it can simultaneously serve many users with different channel conditions.

Therefore, our work can be extended to multi-user case using NOMA, espcially the objective will be to schedule of subset of users at each timeslot. But each user has its own delay constraints to satisfy. a way to solve this problem is to use the so-called Whittle's index which regains a lot of attenton recently in the topic of Age of Information where the goal is to schedule users' updates into a capacity-limited link. So adaptation of Whittle's index to our NOMA-based context should be a future work [106, 107].

Bibliography

[1] “SCAVENGE.” http://www.scavenge.eu/. Cited page 1

[2] B. Sanou, “ICT Facts and Figures,” tech. rep., ITU Telecommunication Development Bureau, 2013. Cited page 1

[3] M. Mills, “The Cloud Begins With Coal,” tech. rep., Digital Power Group, 2013. Cited page 2

[4] “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022,” tech. rep., Cisco, 2019. Cited page 2

[5] A. Fehske, G. Fettweis, J. Malmodin, and G. Biczok, “The global footprint of mobile communications: The ecological and economic perspective,” *IEEE Communications Magazine*, vol. 49, no. 8, pp. 55–62, 2011. Cited page 3

[6] L. Belkhir and A. Elmeligi, “Assessing ICT global emissions footprint: Trends to 2040 & recommendations,” *Journal of Cleaner Production*, vol. 177, pp. 448–463, 2018. Cited page 9

[7] “Energy Harvesting.” https://www.sagentia.com/insight/energy-harvesting/, 2014. Cited page 9

[8] “SOLARGIS Maps.” http://solargis.com/products/maps-and-gis-data/free/download/world. Cited page 10

[9] “Fixed Tilt Solar Panel.” https://commons.wikimedia.org/wiki/File:Fixed_Tilt_Solar_panel_at_Canterbury_Municipal_Building_Canterbury_New_Hampshire.jpg. Cited page 10

[10] B. Coxworth, “Starting in 2020, solar panels will be required on all new California houses.” https://newatlas.com/california-homes-solar-panels/54569/, May 2018. Cited page 10

[11] “Cars with Solar Panels.” https://www.buyacar.co.uk/cars/871/cars-with-solar-panels, 2018. Cited page 10

[12] S. Kim, R. Vyas, J. Bito, K. Niotaki, A. Collado, A. Georgiadis, and M. M. Tentzeris, “Ambient RF Energy-Harvesting Technologies for Self-Sustainable Standalone Wireless Sensor Platforms,” *Proceedings of the IEEE*, vol. 102, no. 11, pp. 1649–1666, 2014. Cited page 11

[13] “PN junction – Solar Cell Central.” http://solarcellcentral.com/junction_page.html. Cited page 11

[14] “How Solar Works.” http://www.solarconnexion.com/how-solar-works/. Cited page 11

[15] J. Gruetter, “Solar Energy Harvesting: Low Power in a Compact Footprint,” tech. rep., Analogue Devices. Cited page 11

[16] S. Cao and J. Li, “A survey on ambient energy sources and harvesting methods for structural health monitoring applications,” *Advances in Mechanical Engineering*, vol. 9, no. 4, p. 168781401769621, 2017. Cited pages 11, 13, and 15

[17] “Samsung EH phone comparison.” https://www.gsmarena.com/compare.php3?idPhone1=2829&idPhone2=2931. Cited page 12

[18] "EH phone comparison." https://www.gsmarena.com/compare.php3?idPhone1=3140&idPhone2=2949. Cited page 12

[19] "IPhone X Tesla." https://caviar.global/caviar-iphone-x-en/tesla-en-2/. Cited page 13

[20] M. Hamid Elsheikh, D. A. Shnawah, M. F. M. Sabri, S. B. M. Said, M. Haji Hassan, M. B. Ali Bashir, and M. Mohamad, "A review on thermoelectric renewable energy: Principle parameters that affect their performance," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 337–355, 2014. Cited page 13

[21] D. Enescu, "Thermoelectric Energy Harvesting: Basic Principles and Applications," in *Green Energy Advances* (D. Enescu, ed.), IntechOpen, 2019. Cited pages 13 and 14

[22] I. Sil, S. Mukherjee, and K. Biswas, "A review of energy harvesting technology and its potential applications," *Environmental and Earth Sciences Research Journal*, vol. 4, no. 2, pp. 33–38, 2017. Cited pages 13 and 15

[23] J. Chen, *Design and Analysis of a Thermoelectric Energy Harvesting System for Powering Sensing Nodes in Nuclear Power Plant*. PhD thesis, Virginia Polytechnic Institute and State University, 2015. Cited page 14

[24] X. Zhang and L.-D. Zhao, "Thermoelectric materials: Energy conversion between heat and electricity," *Journal of Materiomics*, vol. 1, no. 2, pp. 92–105, 2015. Cited page 14

[25] X. Wenyu, *Experimental and Numerical Analysis of Thermoelectric MagnetoHydroDynamic Driven Liquid Lithium Flow in Open Channels for Fusion Applications*. PhD thesis, University of Illinois, 2015. Cited page 14

[26] Y. K. Ramadass and A. P. Chandrakasan, "A batteryless thermoelectric energy-harvesting interface circuit with 35mv startup voltage," in *IEEE International Solid-State Circuits Conference (ISSCC)*, (San Francisco, CA, USA), pp. 486–487, 2010. Cited page 14

[27] "Green Energy - Body Powered Devices." http://content.time.com/time/specials/packages/article/0,28804,2029497_2030623_2029815,00.html, 2010. Cited page 15

[28] "Vodafone unveils the future of festival season tech: Charge your phone while you sleep." https://blog.vodafone.co.uk/2013/06/12/vodafone-unveils-the-future-of-festival-season-tech-charge-your-phone-while-you-sleep/, 2013. Cited page 16

[29] W. K. Seah, Z. A. Eu, and H.-P. Tan, "Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges," in *IEEE International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, (Aalborg, Denmark), pp. 1–5, May 2009. Cited page 15

[30] C. Williams and R. Yates, "Analysis Of A Micro-electric Generator For Microsystems," in *IEEE International Solid-State Sensors and Actuators Conference - TRANSDUCERS*, vol. 1, (Stockholm, Sweden), pp. 369–372, 1995. Cited page 15

[31] E. Hau, "The Wind Resource," in *Wind Turbines*, pp. 505–547, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Cited page 16

[32] M.-L. Ku, W. Li, Y. Chen, and K. J. Ray Liu, "Advances in Energy Harvesting Communications: Past, Present, and Future Challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1384–1412, 2016. Cited pages 17 and 29

[33] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless Networks With RF Energy Harvesting: A Contemporary Survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 757–789, 2015. Cited page 17

[34] M. Pinuela, P. D. Mitcheson, and S. Lucyszyn, "Ambient RF Energy Harvesting in Urban and Semi-Urban Environments," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 7, pp. 2715–2726, 2013. Cited page 17

[35] L. R. Varshney, "Transporting information and energy simultaneously," in *IEEE International Symposium on Information Theory*, (Toronto, ON, Canada), pp. 1612–1616, 2008. Cited page 18

[36] T. D. Ponnimbaduge Perera, D. N. K. Jayakody, S. K. Sharma, S. Chatzinotas, and J. Li, "Simultaneous Wireless Information and Power Transfer (SWIPT): Recent Advances and Future Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 264–302, 2018. Cited page 18

[37] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017. Cited pages 19 and 29

[38] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5g Network Edge Cloud Architecture and Orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017. Cited page 19

[39] H.-J. Hong, "From Cloud Computing to Fog Computing: Unleash the Power of Edge and End Devices," in *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, (Hong Kong), pp. 331–334, 2017. Cited page 19

[40] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," in *IEEE Global Internet of Things Summit (GIoTS)*, (Geneva, Switzerland), pp. 1–6, 2017. Cited page 19

[41] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell System Technical Journal*, vol. 39, no. 5, pp. 1253–1265, 1960. Cited page 20

[42] A. Tolver, *An Introduction to Markov Chain*. Copenhagen, Denmark: Department of Mathematical Sciences University of Copenhagen, 1 ed., 2016. Cited page 21

[43] B. Givan and R. Parr, "An Introduction to Markov Decision Processes." Cited page 21

[44] F. Spieksma and R. Nunez-Queija, *Markov Decision Processes*. 2015. Cited page 21

[45] S. Lall, "Infinite Horizon Markov Decision Problems," 2016. Cited page 22

[46] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. Adaptive computation and machine learning, Cambridge, Mass: MIT Press, 1998. Cited pages 22 and 92

[47] E. Altman, *Constrained Markov Decision Processes*. Chapman & Hall, 2004. Cited pages 23, 27, and 28

[48] S. Mahadevan, "Average reward reinforcement learning: Foundations, algorithms, and empirical results," *Machine Learning*, vol. 22, no. 1-3, pp. 159–195, 1996. Cited page 23

[49] D. P. Bertsekas, *Dynamic programming and optimal control. Vol. 1*. No. 3 in Athena scientific optimization and computation series, Belmont, Mass: Athena Scientific, 3. ed., 2005. Cited pages 23 and 24

[50] P. Geibel, "Reinforcement Learning for MDPs with Constraints," in *Machine Learning: ECML 2006* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, eds.), vol. 4212, pp. 646–653, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. Cited page 27

[51] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy Harvesting Wireless Communications: A Review of Recent Advances," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 360–381, 2015. Cited page 29

[52] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A Survey of Computation Offloading for Mobile Systems," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, 2013. Cited page 29

[53] W. Wu, J. Wang, X. Wang, F. Shan, and J. Luo, "Online Throughput Maximization for Energy Harvesting Communication Systems with Battery Overflow," *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 185–197, 2017. Cited page 29

[54] B. T. Bacinoglu, E. Uysal-Biyikoglu, and C. E. Koksal, “Finite-Horizon Energy-Efficient Scheduling With Energy Harvesting Transmitters Over Fading Channels,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 6105–6118, 2017. Cited page 29

[55] A. O. Isikman, M. Yuksel, and D. Gunduz, “A Low-Complexity Policy for Outage Probability Minimization With an Energy Harvesting Transmitter,” *IEEE Communications Letters*, vol. 21, no. 4, pp. 917–920, 2017. Cited page 29

[56] L. Lei, Y. Kuang, X. S. Shen, K. Yang, J. Qiao, and Z. Zhong, “Optimal Reliability in Energy Harvesting Industrial Wireless Sensor Networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 8, pp. 5399–5413, 2016. Cited pages 29 and 36

[57] I. Ahmed, K. T. Phan, and T. Le-Ngoc, “Optimal Stochastic Power Control for Energy Harvesting Systems With Delay Constraints,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3512–3527, 2016. Cited page 30

[58] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, “Communicating While Computing: Distributed mobile cloud computing over 5g heterogeneous networks,” *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 45–55, 2014. Cited page 30

[59] Dong Huang, Ping Wang, and D. Niyato, “A Dynamic Offloading Algorithm for Mobile Computing,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012. Cited page 30

[60] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, “Energy-Optimal Mobile Cloud Computing under Stochastic Wireless Channel,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013. Cited page 30

[61] W. Labidi, M. Sarkiss, and M. Kamoun, “Energy-optimal resource scheduling and computation offloading in small cell networks,” in *IEEE International Conference on Telecommunications (ICT)*, (Sydney, Australia), pp. 313–318, 2015. Cited pages 30 and 36

[62] O. Munoz, A. Pascual-Iserte, and J. Vidal, “Optimization of Radio and Computational Resources for Energy Efficiency in Latency-Constrained Application Offloading,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4738–4755, 2015. Cited page 30

[63] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, “Delay-optimal computation task scheduling for mobile-edge computing systems,” in *IEEE International Symposium on Information Theory (ISIT)*, (Barcelona, Spain), pp. 1451–1455, 2016. Cited page 30

[64] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, “Energy-Latency Tradeoff for Energy-Aware Offloading in Mobile Edge Computing Networks,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2633–2645, 2018. Cited page 30

[65] Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016. Cited pages 30 and 36

[66] J. Xu, L. Chen, and S. Ren, “Online Learning for Offloading and Autoscaling in Energy Harvesting Mobile Edge Computing,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 361–373, 2017. Cited page 30

[67] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, “Learning-Based Computation Offloading for IoT Devices With Energy Harvesting,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1930–1941, 2019. Cited page 30

[68] S. Kaul and R. Yates, “Real-time status: how often should one update?,” in *IEEE International Conference on Computer Communications (INFOCOM)*, May 2012. Cited page 31

[69] A. Kosta, N. Pappas, and V. Angelakis, “Age of Information: A New Concept, Metric, and Tool,” *Foundations and Trends® in Networking*, vol. 12, no. 3, pp. 162–259, 2017. Cited page 31

[70] H. Wang and N. Mandayam, "A Simple Packet-Transmission Scheme for Wireless Data Over Fading Channels," *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1055–1059, 2004. Cited page 33

[71] Y. Chen and Q. Zhao, "An Integrated Approach to Energy-Aware Medium Access for Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3429–3444, 2007. Cited page 33

[72] D. V. Djonin and V. Krishnamurthy, "MIMO Transmission Control in Fading Channels—A Constrained Markov Decision Process Formulation With Monotone Randomized Policies," *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 5069–5083, 2007. Cited page 33

[73] N. Salodkar, A. Bhorkar, A. Karandikar, and V. Borkar, "An on-line learning algorithm for energy efficient delay constrained scheduling over a fading channel," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 732–742, May 2008. Cited page 33

[74] N. Salodkar, A. Karandikar, and V. S. Borkar, "A Stable Online Algorithm for Energy-Efficient Multiuser Scheduling," *IEEE Transactions on Mobile Computing*, vol. 9, no. 10, pp. 1391–1406, 2010. Cited page 33

[75] R. A. Berry, "Optimal Power-Delay Tradeoffs in Fading Channels—Small-Delay Asymptotics," *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3939–3952, 2013. Cited page 33

[76] K. T. Phan, T. Le-Ngoc, M. van der Schaar, and F. Fu, "Optimal Scheduling over Time-Varying Channels with Traffic Admission Control: Structural Results and Online Learning Algorithms," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4434–4444, 2013. Cited page 33

[77] R. Berry and R. Gallager, "Communication over fading channels with delay constraints," *IEEE Transactions on Information Theory*, vol. 48, pp. 1135–1149, May 2002. Cited page 33

[78] A. Aprem, C. R. Murthy, and N. B. Mehta, "Transmit Power Control Policies for Energy Harvesting Sensors With Retransmissions," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 895–906, 2013. Cited page 63

[79] Z. Xiang and M. Tao, "Robust Beamforming for Wireless Information and Power Transmission," *IEEE Wireless Communications Letters*, vol. 1, no. 4, pp. 372–375, 2012. Cited page 63

[80] R. Gangula, D. Gesbert, and D. Gunduz, "Optimization of Energy Harvesting MISO Communication System With Feedback," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 396–406, 2015. Cited page 64

[81] T.-D. Le, G. Kaddoum, and O.-S. Shin, "Joint Channel Resources Allocation and Beamforming in Energy Harvesting Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 884–887, 2018. Cited page 64

[82] R. Ma and W. Zhang, "Optimal power allocation for energy harvesting communications with limited channel feedback," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, (Atlanta, GA, USA), pp. 193–197, 2014. Cited page 64

[83] M. R. Zenaidi, Z. Rezki, and M.-S. Alouini, "Performance Limits of Online Energy Harvesting Communications With Noisy Channel State Information at the Transmitter," *IEEE Access*, vol. 5, pp. 1239–1249, 2017. Cited page 64

[84] M. R. Zenaidi, Z. Rezki, H. Tembine, and M.-S. Alouini, "Performance limits of energy harvesting communications under imperfect channel state information," in *IEEE International Conference on Communications (ICC)*, (Kuala Lumpur, Malaysia), pp. 1–6, May 2016. Cited page 64

[85] P. Ciblat and L. Vandendorpe, "On the Maximum-Likelihood based data-aided frequency offset and channel estimates," in *European Signal Processing Conference (EUSIPCO)*, 2002. Cited page 67

[86] J. G. Proakis and M. Salehi, *Digital communications*. Boston: McGraw-Hill, 5th ed ed., 2008. Cited page 67

[87] S. Lin and D. J. Costello, *Error control coding: fundamentals and applications*. Upper Saddle River, N.J: Pearson-Prentice Hall, 2nd ed ed., 2004. Cited page 72

[88] A. Adhikary, H. C. Papadopoulos, S. A. Ramprashad, and G. Caire, "Multi-user MIMO with outdated CSI: Training, feedback and scheduling," in *IEEE Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, (Monticello, IL), pp. 886–893, 2011. Cited page 79

[89] W. Xiong, A. Mukherjee, and H. M. Kwon, "Underlay MIMO Cognitive Radio Downlink Scheduling with Multiple Primary Users and No CSI," in *IEEE 81st Vehicular Technology Conference (VTC Spring)*, (Glasgow, United Kingdom), pp. 1–5, May 2015. Cited page 79

[90] M. J. Abdoli and A. S. Avestimehr, "Layered Interference Networks With Delayed CSI: DoF Scaling With Distributed Transmitters," *IEEE Transactions on Information Theory*, vol. 60, no. 3, pp. 1822–1839, 2014. Cited page 79

[91] Y. Huang, S. Durrani, and X. Zhou, "Interference nulling for offloaded heterogeneous users using macro generalized inverse precoder," in *IEEE International Symposium on Communications and Information Technologies (ISCIT)*, (Nara, Japan), pp. 53–56, 2015. Cited page 80

[92] P. Swami, V. Bhatia, S. Vuppala, and T. Ratnarajah, "Performance Analysis of Offloading in NOMA-HetNets using Imperfect CSI," in *IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, (Indore, India), pp. 1–6, 2018. Cited page 80

[93] L. Jiao, H. Yin, Y. Lyu, H. Huang, J. Dong, and D. Guo, "Optimal Schedule of Mobile Edge Computing Under Imperfect CSI," in *Algorithms and Architectures for Parallel Processing* (J. Vaidya and J. Li, eds.), vol. 11335, pp. 32–45, Cham: Springer International Publishing, 2018. Cited page 80

[94] T. T. Nguyen, L. Le, and Q. Le-Trung, "Computation Offloading in MIMO Based Mobile Edge Computing Systems Under Perfect and Imperfect CSI Estimation," *IEEE Transactions on Services Computing*, pp. 1–1, 2019. Cited page 80

[95] R. Atallah, C. Assi, and M. Khabbaz, "Deep reinforcement learning-based scheduling for roadside communication networks," in *IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, (Paris, France), pp. 1–8, May 2017. Cited page 92

[96] H. Ye and G. Y. Li, "Deep Reinforcement Learning for Resource Allocation in V2v Communications," in *IEEE International Conference on Communications (ICC)*, (Kansas City, MO), pp. 1–6, May 2018. Cited page 92

[97] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Performance Optimization in Mobile-Edge Computing via Deep Reinforcement Learning," *arXiv:1804.00514 [cs]*, 2018. Cited page 92

[98] L. Huang, X. Feng, A. Feng, Y. Huang, and L. P. Qian, "Distributed Deep Learning-based Offloading for Mobile Edge Computing Networks," *Mobile Networks and Applications*, 2018. Cited page 92

[99] M. K. Sharma, A. Zappone, M. Debbah, and M. Assaad, "Deep Learning Based Online Power Control for Large Energy Harvesting Networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Brighton, United Kingdom), pp. 8429–8433, May 2019. Cited page 92

[100] M. K. Sharma, A. Zappone, M. Assaad, M. Debbah, and S. Vassilaras, "Distributed Power Control for Large Energy Harvesting Networks: A Multi-Agent Deep Reinforcement Learning Approach," *arXiv:1904.00601 [cs, math, stat]*, 2019. Cited page 92

[101] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, 2014. Cited page 93

[102] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," *arXiv:1312.5602 [cs]*, 2013. Cited page 93

[103] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *arXiv:1810.07862 [cs]*, 2018. Cited page 94

[104] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo, "A Survey of Non-Orthogonal Multiple Access for 5g," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2294–2323, 2018. Cited page 96

[105] M. Aldababsa, M. Toka, S. Gökçeli, G. K. Kurt, and O. Kucur, "A Tutorial on Nonorthogonal Multiple Access for 5g and Beyond," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–24, 2018. Cited page 96

[106] P. Whittle, "Restless bandits: activity allocation in a changing world," *Journal of Applied Probability*, vol. 25, pp. 287–298, 1988. Cited page 97

[107] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, E. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2637–2650, 2018. Cited page 97