University of Padova

Department of Information Engineering

Ph.D. School in Information Engineering
Curriculum: Information Science and Technology
XXXI Class

Channel Access in Wireless Networks:

Protocol Design of Energy-Aware Schemes for the IoT and Analysis of Existing Technologies

Author:
Chiara Pielli

Supervisor:
Prof. Michele Zorzi

School Coordinator:
Prof. Andrea Neviani

Academic Year 2017/2018

To you, who have been keeping me on the track.

What's to gain by silence?
Ursula K. Le Guin

Abstract

The design of channel access policies has been an object of study since the deployment of the first wireless networks, as the Medium Access Control (MAC) layer is responsible for coordinating transmissions to a shared channel and plays a key role in the network performance. While the original target was the system throughput, over the years the focus switched to communication latency, Quality of Service (QoS) guarantees, energy consumption, spectrum efficiency, and any combination of such goals. The basic mechanisms to use a shared channel, such as ALOHA, Time Division Multiple Access (TDMA)- and Frequency Division Multiple Access (FDMA)-based policies, have been introduced decades ago. Nonetheless, the continuous evolution of wireless networks and the emergence of new communication paradigms demand the development of new strategies to adapt and optimize the standard approaches so as to satisfy the requirements of applications and devices.

This thesis proposes several channel access schemes for novel wireless technologies, in particular Internet of Things (IoT) networks, the Long-Term Evolution (LTE) cellular standard, and mmWave communication with the IEEE802.11ad standard.

The first part of the thesis concerns energy-aware channel access policies for IoT networks, which typically include several battery-powered sensors. In scenarios with energy restrictions, traditional protocols that do not consider the energy consumption may lead to the premature death of the network and unreliable performance expectations. The proposed schemes show the importance of accurately characterizing all the sources of energy consumption (and inflow, in the case of energy harvesting), which need to be included in the protocol design. In particular, the schemes presented in this thesis exploit data processing and compression techniques to trade off QoS for lifetime. We investigate contention-free and contention-based chanel access policies for different scenarios and application requirements.

While the energy-aware schemes proposed for IoT networks are based on a clean-slate approach that is agnostic of the communication technology used, the second part of the thesis is focused on the LTE and IEEE802.11ad standards. As regards LTE, the study proposed in this thesis shows how to use machine-learning techniques to infer the collision multiplicity in the channel access phase, information that can be used

to understand when the network is congested and improve the contention resolution mechanism. This is especially useful for massive access scenarios; in the last years, in fact, the research community has been investigating on the use of LTE for Machine-Type Communication (MTC). As regards the standard IEEE802.11ad, instead, it provides a hybrid MAC layer with contention-based and contention-free scheduled allocations, and a dynamic channel time allocation mechanism built on top of such schedule. Although this hybrid scheme is expected to meet heterogeneous requirements, it is still not clear how to develop a schedule based on the various traffic flows and their demands. A mathematical model is necessary to understand the performance and limits of the possible types of allocations and guide the scheduling process. In this thesis, we propose a model for the contention-based access periods which is aware of the interleaving of the available channel time with contention-free allocations.

Sommario

Fin dalla comparsa delle prime reti wireless, la progettazione di strategie di accesso al canale è stata oggetto di intenso studio, in quanto il livello MAC è responsabile di coordinare le trasmissioni su un canale condiviso e quindi svolge un ruolo fondamentale nelle prestazioni della rete intera. Originariamente la progettazione del livello MAC nelle reti wireless si proponeva di garantire un certo throughput, ma nel corso degli anni l'interesse si è spostato sulla latenza delle comunicazioni, assicurare un certo livello di QoS, ottimizzare il consumo energetico, garantire efficienza spettrale, e qualsiasi combinazione di questi obiettivi. I meccanismi classici di accesso al canale, come ALOHA, TDMA e FDMA, sono stati introdotte da decenni; ciononostante, la continua evoluzione delle reti wireless e la comparsa di nuovi paradigmi di comunicazione ha richiesto lo sviluppo di nuove strategie per adattare e ottimizzare gli approcci standard così da soddisfare i requisiti di dispositivi e applicazioni.

Questa tesi propone diversi schemi di accesso al canale per nuove tecnologie wireless, e in particolare per reti IoT, per lo standard cellulare LTE, e lo standard IEEE802.11ad per comunicazione con mmWaves.

La prima parte della tesi riguarda schemi di accesso al canale efficienti dal punto di vista energetico per reti IoT, che, di solito, comprendono molti sensori alimentati a batteria. In scenari con restrizioni energetiche i protocolli classici che non prendono in considerazione il consumo di potenza potrebbero portare alla morte prematura della rete e ad aspettative di prestazioni ottimistiche. Gli schemi proposti in questa tesi dimostrano l'importanza di caratterizzare tutte le fonti di consumo energetico (e di apporto energetico, nel caso di energy harvesting), che devono essere incluse nella progettazione del protocollo di comunicazione. In particolare, gli schemi proposti in questa tesi sfruttano tecniche di compressione ed elaborazione dati, le quali consentono di prolungare la vita della rete a discapito di una ridotta QoS. Abbiamo analizzato algoritmi di accesso sia basati sulla contesa del canale che non per diversi scenari e requisiti di applicazione.

Mentre gli schemi proposti per le reti IoT non sono basati su tecnologie specifiche, la seconda parte della tesi riguarda gli standard LTE e IEEE802.11ad. Per quanto concerne LTE, lo studio proposto in questa tesi mostra come utilizzare tecniche di

machine-learning per stimare il numero di utenti che collidono durante l'accesso al canale; quest'informazione è utilizzata per capire quando la rete è congestionata e migliorare il meccanismo di risoluzione delle collisioni. Questo è particolarmente utile per scenari di accesso massivo: negli utlimi anni, infatti, si è sviluppato un forte interesse verso l'utilizzo di LTE per MTC. Per quanto riguarda IEEE802.11ad, invece, lo standard prevede un MAC ibrido con allocazioni da predefinire con e senza contesa per l'accesso al canale, e un meccanismo di allocazione dinamica che viene fatta al di sopra dello schema già stabilito. Nonostante ci si aspetti che questo schema ibrido possa soddisfare requisiti eterogenei, non è ancora chiaro come scegliere le allocazioni da usare in base ai vari flussi di traffico e i loro requisiti. Perciò, è necessario un modello matematico per capire le prestazioni e i limiti che possono essere ottenuti con le varie tipologie di accesso al mezzo previste dallo standard e guidare la fase di allocazione delle risorse. In questa tesi, proponiamo un modello per le allocazioni basate sulla contesa del canale di comunicazione che tiene conto della presenza di altre allocazioni di tipo diverso.

Contents

List of Acronyms

A-BFT	Association-Beamforming Training
AP	Access Point
AR	autoregressive
ARQ	Automatic Repeat Request
ATI	Announcement Transmission Interval
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BHI	Beacon Header Interval
BI	Beacon Interval
BRP	Beam Refinement Protocol
BS	Base Station
BTI	Beacon Transmission Interval
CBAP	Contention Based Access Period
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CP	Cyclic Prefix
CS	Compressive Sensing
CSI	Channel State Information
CSMA	Carrier Sense Multiple Access
CSMA/CA	CSMA with Collision Avoidance
CTS	Clear-To-Send
DCF	Distributed Coordination Function
DIFS	Distributed Interframe Space
DTI	Data Transmission Interval
EDCA	Enhanced Distributed Channel Access
EH	Energy Harvesting

EHF	Extremely High Frequency
eNB	Base Station
FC	Fusion Center
FDMA	Frequency Division Multiple Access
i.i.d.	independent and identically distributed
IoT	Internet of Things
LR	Logistic Regression
LTC	Lightweight Temporal Compression
LTE	Long-Term Evolution
MAC	Medium Access Control
MC	Markov Chain
MCS	Modulation and Coding Scheme
MDP	Markov Decision Process
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MSE	Mean Squared Error
MTC	Machine-Type Communication
NAV	Network Allocation Vector
NN	Neural Network
p.m.f.	probability mass function
PBSS	Personal Basic Service Sets
PCP	Personal Basic Service Sets (PBSS) Control Point
PDP	Power Delay Profile
PDR	Packet Delivery Ratio
PPP	Poisson Point Process
PRACH	Physical Random Access Channel
QO	quasi-omnidirectional
QoS	Quality of Service
r.v.	random variable
RACH	Random Access Channel
RTS	Request-To-Send
RVIA	Relative Value Iteration Algorithm
S-ALOHA	Slotted ALOHA
SIFS	Short Interframe Space

SINR	Signal-to-Interference-and-Noise Ratio
SLS	Sector-Level Sweep
SNR	Signal-to-Noise Ratio
SP	Service Period
STA	station
TDMA	Time Division Multiple Access
TSCH	Time-Slotted Channel Hopping
TXOP	Transmission Opportunity
UE	User Equipment
VBR	Variable Bit Rate
WSN	Wireless Sensor Network
WUR	Wake-Up Radio
ZC	Zadoff-Chu

1

Introduction

Circuitry and RF design are constantly improving to yield enhanced device capabilities, and an optimized design of all the layers of the protocol stack is necessary to fully exploit the potential of the underlying hardware. Channel access plays an important role in telecommunication networks: the MAC layer is in charge of coordinating transmissions to a shared channel, having a strong impact on throughput, energy consumption, and latency. It is responsible for defining how and when a node can attempt to transmit in such a way to limit all issues that can arise when multiple devices share the same medium.

At a macroscopic level, MAC schemes for wireless networks are typically classified as either contention-based or contention-free, according to the policy adopted to handle multiple accesses of the channel [1]. In contention-free channel access, a node transmits in dedicated resources (e.g., time, frequency, or using a specific code); this avoids interference from the other devices (from the same or other networks), but requires coordination among the devices for the generation and maintenance of the schedule. On the other hand, in contention-based channel access schemes (such as ALOHA and its numerous variants or Carrier Sense Multiple Access (CSMA)-based approaches), the devices can access the medium randomly; the channel access itself is simpler than in contention-free approaches as it requires little to no coordination, but demands a stable and fast algorithm for collision resolution. Both contention-based and contention-free approaches have advantages and disadvantages, and typically the choice of one over the other is driven by the use case, network topology, and application requirements. Hybrid schemes try to leverage on the advantages of both approaches; the access procedure, e.g., is then split into two phases characterized by different mechanisms, or dynamically adapts to the current network conditions.

The design of an efficient channel access mechanism should take into account the application goals and the target scenario, possibly including adaptive mechanisms to deal with changing network conditions and traffic patterns. Moreover, the devices may differ in computational, storage and energy capabilities, and the channel access scheme

should adapt to such heterogeneity. Clearly, to achieve a certain performance goal, it is necessary to accept a tradeoff on other aspects of network performance, such as bandwidth efficiency, latency, QoS, energy consumption, etc. The design of the MAC layer is also interrelated to that of other layers and communication aspects, such as routing and data processing.

The studies presented in this thesis are based on mathematical modeling. Although this entails the introduction of simplifications with respect to real-world deployments, it gives insight on the achievable system performance and on the role played by the various parameters. Modeling and analysis of communication networks allows one to study the system even before its actual implementation, and thereby optimize its design. This is helpful to assess bounds that can be used to gauge the performance of real deployments and to understand the tradeoffs involved in the design choices.

This thesis is made of two parts. The first part concerns energy-aware schemes for IoT networks. While several applications (environmental monitoring, tracking of goods, on-body measurement of physiological parameters, etc. [2]) benefit from the large availability of cheap and easy-to-install sensors, the limited resources of the devices pose a number of unique design challenges, in particular for dense networks. Even if connecting sensors to the electricity grid is feasible, it can be beneficial not to do so to simplify the installation process, facilitate changing the position of sensors or ensure devices are independent of the power grid. Many sensors are in fact designed to require very little maintenance and to be off-the-grid, in what has been called a *place-&-play* paradigm [3]. This requires the devices to be battery-powered, and possibly equipped with Energy Harvesting (EH) mechanisms to draw energy from the environment, in order to become energetically self-sufficient. In any case, the communication protocols should be designed to operate under strict constraints on the energy availability [4]. Ch. 2 introduces the energy efficiency issue and discusses how the MAC layer design is affected by energy constraints. It also introduces a parameterised model for the major sources of energy consumption, which is helpful in the design of energy-aware schemes. Chs. 3–5 present some channel access schemes for IoT networks. While they differ for channel access mechanism, devices capabilities and requirements, they share some similarities. All the proposed schemes target monitoring applications for single-hop networks, so that the purpose is to track some phenomena of interest and report data to a common receiver. The ultimate goal is to extend the devices lifetime as much as possible, i.e., to use the available energy optimally. To do so, all schemes leverage on data processing, which allows one to trade off some accuracy in the data representation for a reduced energy consumption and thus requires strategies to ensure a desired QoS.

Ch. 6, instead, addresses the energy issue at the MAC layer from a security perspective, and analyses energy-depleting jamming attacks using a game theoretical approach.

The second part of the thesis is related to the channel access mechanism of existing standards, in particular the 3GPP cellular standard LTE and the Wifi standard for mmWave communication, IEEE802.11ad. Ch. 7 concerns the Random Access Channel (RACH) procedure in LTE, which serves as synchronization mechanism between the base station and the other devices in the network. In particular, the study presents a machine-learning based technique to detect the collision multiplicity during the RACH. This information can be helpful for advanced collision resolution algorithms and thereby improve the channel access performance, since the current collision resolution mechanism can be a serious bottleneck in dense scenarios, in particular in the case of MTC. Ch. 8 is about the standard IEEE802.11ad, which targets short range mmWave communication at 60 GHz in local area networks. MmWave communication has been gaining increasing attention from both academia and industries, because of the new possibilities disclosed by the huge bandwidth and the mmWave frequencies. Unlike the previous WiFi standards, 802.11ad adopts a hybrid MAC layer, with contention-based and contention-free allocations. It is however not clear how to schedule such allocations to match the requirements of the various traffic flows, and the literature on this topic is still very sparse. Ch. 8 introduces a mathematical model for the contention-based access periods that allows one to evaluate the network performance based on the configuration chosen for the scheduling.

The studies presented in this thesis highlight that the MAC layer design is a challenging task that needs to take into account several conflicting factors and requirements. A well designed channel access scheme can significantly improve the network performance. It is key to adapt the channel access scheme to the scenario and network conditions and mathematical analysis can play a major role in determining the tradeoffs involved between the various system parameters.

We would like to highlight that some of the studies presented in this have been done in collaboration with other Ph.D. students. The model proposed in Ch. 8 has been realized during a six-month internship ad the National Institute of Standards and Technology (NIST) (Gaithersburg, MD, US).

Additional research. While the focus of this thesis is on the MAC layer for different scenarios and protocols, during my Ph. D. I also developed schemes for the rebalancing problem of bike-sharing systems, which represent one of the major services in Smart Cities and can be modeled as networks. The dissemination of sensors yields the collection of historical data whose analysis makes it possible to gain insight on the service usage and understand how it is affected by different conditions and external factors [C6]. This information can then be used to improve the existing service; one of the main problems

that affect bike-sharing systems is the unbalanced usage, which leads to have empty and full stations, where it is impossible to pick up and drop a bike, respectively. We developed a dynamic rebalancing scheme [J4] that adapts to the varying demand, and also proposed an analytical model that takes into account incentives given to users so that they are encouraged to self-rebalance the bike-sharing network [J7].

Part I

ENERGY-AWARE COMMUNICATION FOR THE IOT

2

The energy efficiency issue

The IoT has been one of the major drivers of innovation in the past few years, with new applications and technologies being invented on a daily basis [5]. Intelligent sensors and microcontrollers are extended into the world of everyday objects creating a ubiquitous interconnected world [6]. Unquestionably, the IoT is expected to grow significantly in the next future and drastically change many aspects of everyday life. Gartner predicts 20 billion internet-connected things by 2020[1] and McKinsey Global Institute estimates that the IoT could have an annual economic impact of $4 to $11 trillions by 2025[2].

However, many challenging issues still need to be addressed and both technological as well as social knots need to be untied before the vision of IoT becomes a reality [7]. From a technological perspective, these challenges include the heterogeneity of both devices and applications, the high volume of data collected, self-adaptability to dynamic scenarios and requirements, scalability and connectivity, energy management, privacy and security [8]. As traditional technologies are not able to solve these issues, there is an increasing need for novel solutions, and changes at both the architectural and silicon levels are required. An intelligent sensor and circuit design is in fact not sufficient: slim and lightweight implementations at all layers are also necessary. Ill-designed network, processing and resource management solutions may indeed hamper the effectiveness of an efficient PHY layer.

This part of the thesis focuses on the energy efficiency problem, which is considered to be one of the main challenges of IoT networks, as many devices are expected not to be connected to any infrastructure, including the energy grid [9]. Most research so far has focused on low power circuit design and energy efficient PHY, with the goal of reducing the average energy per information bit required for communication. While any advances at the RF/PHY layer are expected to translate into a more energy-efficient device, it is not at all obvious that this is by itself sufficient for the whole system to make the best use of the available energy, and a more complete view of the system, including the application

1 https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf
2 https://www.mckinsey.com/industries/semiconductors/our-insights/whats-new-with-the-internet-of-things

(signal type and processing tasks), the lower networking layers (MAC scheduling and routing) and some basic network management functionalities (node discovery and sleep modes) can play a crucial role in identifying the main sources of energy consumption, revealing inefficiencies and providing opportunities for large gains [10].

Replacing sensors or recharging their batteries every few weeks may annihilate all the benefits of collecting data, and nodes failure due to power depletion may even lead to the breakdown of the whole architecture [11]. It is fundamental to have sensors that are completely stand-alone and can run for years between battery replacements. A typical periodically-reporting device is expected to operate for long periods without human intervention: in particular, the industry aims to achieve a minimum of 10 years of battery lifetime [12, 13, 14]. In such a scenario, keeping a node's transmitter turned on and transmitting continuously at maximum power is not realistic: duty-cycling and power control are fundamental techniques [15] to increase the lifespan of these devices, and several protocols have been studied and implemented [16].

EH is a promising technique through which sensors can scavenge energy from the environment and replenish their batteries, thereby fostering self-sustainability of the devices and, thus, of the IoT application. Ideally, EH could guarantee infinite lifetime; however, its intermittent nature requires the use of flexible protocols able to adapt to a stochastic energy availability. A general overview of recent advances in wireless communications with EH is presented in [17], while [18] discusses the challenges of designing an intelligent EH communication system and presents a general model that can be adapted to some specific contexts. Notice that EH requires a different approach to the energy management: when the battery represents the only source available to the device, the effort is put in the minimization of the energy consumption [19, 20]; instead, when devices have EH capabilities, energy-neutral operation modes are sought [21, 22]. Optimizing the energy consumption is a challenging task, sharpened by the numerous tradeoffs that need to be tackled when trying to provide reliability, security, and timeliness at the same time. The need for energy efficiency in the IoT has been gaining increasing attention in the last decade, and many efforts have been put in the design of energy-aware protocols at all layers of the protocol stack.

Notice that the energy constraints of many IoT devices, along with their reduced computational power, also pose a security challenge: devices are vulnerable to energy-depleting attacks and traditional security protocols may be too burdensome or complex. Gartner predicts that worldwide spending on IoT security will reach $1.5 billion in 2018, a 28 percent increase from 2017 spending of $1.2 billion.[3]

3 https://www.gartner.com/newsroom/id/3869181

In this thesis, the energy efficiency issue is investigated from a MAC layer perspective. The MAC layer is in charge of coordinating transmissions to a shared channel, and thus plays a key role in the IoT, where massive access and dynamicity are the rule rather than the exception. The MAC layer design attempts to improve energy efficiency by accepting a tradeoff on other aspects of network performance, such as bandwidth efficiency, latency, and reliability [23], and is crucial in the IoT because of its influence on the energy-hungry radio transceiver.

2.1 Sources of energy consumption at the MAC layer

It is possible to identify four major sources of energy waste that the design of the MAC layer must deal with [24] [25].

Collisions. When multiple radio signals overlap in time or frequency, they are said to collide. This situation can be observed especially in random access schemes where different sources transmit without any coordination, and in dense networks, where the same time slots have to be shared by multiple nodes. It is extremely important to avoid (or at least reduce) collisions, as they impinge on the energy efficiency (and the network performance in general), both because of the energy wasted to transmit and receive corrupted packets, and the following retransmissions that they may imply. In fact, even in case of advanced receivers with multi-packet reception capabilities, collisions may prevent the correct decoding of the signals involved, and this affects the energy consumption, the latency, the quality of the received data, and the throughput.

Overhead. Coordinating access schemes requires the exchange of signaling and control packets to maintain the synchronization among devices. These control packets do not carry information useful to the final application, but are only used by nodes to communicate and coordinate themselves, and clearly influence the energy consumption. Moreover, the payload size is often small in many IoT scenarios (less than 1 kByte), and thus the MAC overhead is significant and needs to be designed optimally. Notice that also the acknowledgment mechanism is included in the overhead, and in fact many IoT protocols try to limit the feedback from the receiver (e.g., in LoRa [26]).

Overhearing. In wireless networks, it may happen that a node receives a packet intended for a different destination, thereby wasting energy since the sensor uselessly listens to the channel and also spends some energy to decode. Usually, overhearing is reduced by filtering packets based on their destination address or exploiting their preamble.

Idle listening. When a node does not know when it will receive messages from other devices, it keeps listening to the channel waiting for potential data. It means that a node is ready to receive messages but in vain, and thus wastes its energy. This problem is

very important especially in networks with low traffic loads. The simplest way to reduce idle listening is to put nodes to sleep as long as possible, e.g., by adopting appropriate duty cycles. However, it is fundamental to ensure that a device is awake when it has to receive some message. Wake-Up Radios (WURs) are a novel hardware approach that eliminates these shortcomings: devices are provided with an ultra low power receiver that continuously listens to the channel and wakes up the main radio on demand [27]. WURs improve the overall network energy efficiency, but their design has to deal with several tradeoffs concerning sensitivity, resilience to interference, coverage area, wake-up speed, and power consumption [28].

2.2 Channel access schemes

At a macroscopic level, channel access schemes can be divided into *contention-free* and *contention-based*, depending on whether the devices try to use the same resources simultaneously or not, respectively.

Contention-free access. When the network topology and the traffic pattern can be known or predicted in advance, contention-free access mechanisms guarantee optimal performance because they allocate the available resources to nodes so as to avoid wastages and satisfy all users' requirements. Interference and collisions are avoided, and idle listening is reduced. Users can be allocated different time slots (TDMA-based approaches), frequency bands (FDMA-based approaches), or orthogonal codes (Code Division Multiple Access (CDMA) approaches). Coordinated access schemes are well suited for applications where the traffic pattern is known in advance, e.g., industrial Wireless Sensor Networks (WSNs) [29]. In 2012, for example, the Internet Engineering Task Force (IETF) introduced the Time-Slotted Channel Hopping (TSCH) [30] mode as an amendment to the MAC portion of the IEEE802.15.4e standard, which combines time synchronization and channel hopping and is intended for industrial automation. TDMA-based schemes can be effectively coupled with duty cycling, where nodes alternate active and sleeping phases to preserve energy [31]. However, pure coordinated access schemes may result in poor performance when dealing with event-based signals such as alarms, which have strict latency and QoS constraints. Traditional protocols should therefore be revisited in order to account for the different traffic types, like in [32], where the proposed access mechanism proactively tunes the number of used resources to meet the application requirements.

Contention-based access. In some IoT applications, using a coordinated scheme may be impractical or suboptimal, because i) synchronization and control messages burden too much the constrained nodes, ii) data transmission is event-triggered and thus difficult to predict, or iii) the device mobility and the likelihood of faults due to fluctuations of

the wireless channel make the topology highly dynamic. When at least one of these conditions occurs, random access schemes are generally preferable over coordinated ones. They are distributed and easier to implement and make channel resources available to more stations. The price to pay is interference among devices, which may cause packet losses or corruption, affecting the QoS, increasing latency, and wasting energy. Typically, transmission technologies in the IoT use some variant of ALOHA-based schemes, where devices access the channel whenever they have data to transmit, or CSMA schemes, where devices listen to the channel before transmitting to sense whether there are ongoing transmissions and reduce its probability of colliding with other packets. An interesting approach is represented by coded random access schemes, which map the structure of the access protocol to that of an erasure-correcting code defined on a graph, making it possible to achieve much better performance than simple ALOHA [33]. Another way to improve the performance of random access is to consider a receiver with interference cancellation or multiple packet reception capabilities [34]. Also duty cycling may lead to significantly reduced energy consumption, but has an impact on data latency and still wastes energy for idle listening [35].

Both contention-free and contention-based approaches have advantages and disadvantages, and typically the choice of one over the other is driven by the use case, network topology, and application requirements. Given the dynamic nature of many IoT scenarios, it may also be useful to leverage on the advantages of both schemes to achieve high performance under variable traffic patterns and network conditions [36, 37]. This leads to hybrid access, where the access procedure either is split into two phases characterized by different mechanisms, or dynamically adapts to the current network load and requirements.

2.2.1 *Data processing*

A well-designed channel access scheme may not itself be sufficient to ensure power savings. The resource limitations of the devices typically involved in an IoT scenario also pose a challenge, because they affect the complexity of the algorithms that can be run.

In monitoring applications, the devices constantly sense the surrounding environment and exchange a large amount of raw data, whose transmission would rapidly deplete the batteries of the nodes. A way to save some transmission energy consists in reducing the volume of data to send by applying source processing techniques. In-node signal processing/compression mechanisms can indeed reduce the amount of data to be transmitted, thus relieving channel contention, transport and interference issues. However, this comes at the cost of spending some energy for the compression operations and introducing a distortion with respect to the original signal.

A recent trend in IoT deployments is to move some of the processing from the network center to its edge, according to the *fog computing* paradigm [38]. In-node and in-network data processing are of paramount importance, as they can effectively reduce the amount of information that is to be sent to (and processed by) the higher levels of IoT systems [39].

In many IoT applications, like industrial and environmental monitoring, nodes periodically report measurements to a central entity (the sink) and their data volume can be highly reduced through predictive algorithms [40, 41], i.e., by sending data points only when they deviate from some expected pattern. The effectiveness of this approach has also been proved on real datasets [42]. When dealing with time series, *lossy compression* can be exploited to trade some accuracy in the signal's representation for improved energy efficiency. In this domain, a number of approaches like probabilistic, linear or autoregressive models, Fourier transforms and Kalman filters have been considered, although they are generally too computationally expensive and, in turn, power-hungry for constrained IoT devices [43]. The research community has thus started exploring lighter algorithms, e.g., the Lightweight Temporal Compression (LTC) algorithm [44].

Moreover, the heterogeneous and dynamic nature of IoT systems requires adaptability, and employing a traditional compression scheme may lead to suboptimal performance. The research focus is thus moving towards data-driven approaches, where the compression technique is automatically adjusted according to the type of signal and to the application requirements. For example, in [45] Compressive Sensing (CS) is combined with principal component analysis to capture the spatial and temporal characteristics of real signals, and a feedback control loop estimates the signal reconstruction errors on the fly and allows the system to self-adapt to changes in the signal statistics. [46] proposes another adaptive scheme that switches between lossless and lossy compression in an on-demand fashion according to a compression error bound derived from the application requirements.

Another promising approach consists in applying data mining techniques to extract features from time series, seeking feature-based classifiers [47]. Signal classification into groups with similar characteristics allows the sensors to choose the data processing technique that is most appropriate for their respective class (i.e., leading to the best performance for some metric, like the distortion of the compressed signal) [48, 49].

Moreover, the cost (energy and distortion) of the in-node processing algorithms shall be included in the optimization of the network protocols so as to allow the entire system to adapt, seeking a good tradeoff in terms of overall energy consumption (processing and communication) vs. quality of the information that is sent to the application (e.g., quality of an answer or representation accuracy of a measurement).

2.3 A model for the energy consumption

It is extremely important to accurately estimate the lifetime of a sensor to gauge the performance of IoT energy-constrained systems. In fact, although a wide variety of network lifetime definitions are adopted in the literature, they all ultimately depend on the lifetime of individual sensors. To design energy efficient algorithms and protocols it is necessary to identify and characterize all the sources of energy consumption and supply. It is hard to define an exhaustive and general model for the energy dynamics of an IoT device, since its energy consumption highly depends on the technology it employs, its operating conditions, and the algorithms it uses. Next, we describe a parameterized model that tries to capture all the major sources of energy expenditure, namely, communication, data acquisition, processing, and circuitry [10].

Sensing. Let N_s be the number of sensing events performed in a given time window T_s. The sensing energy is defined as:

$$E_{\mathrm{s}} = N_s E_{\mathrm{sens}} \tag{2.1}$$

where E_{sens} is the energy spent by the device to collect one sample. For periodic sensing, $N_s \simeq \mathrm{round}(T_s/T_p)$, where T_p is the nominal sensing period, while for aperiodic sensing, where the acquisition of samples is triggered by some event, N_s is a random variable whose distribution depends on the specific sensing process and on the observation window T_s. Often, E_{sens} is very small compared to the energy drained by the RF architecture, and E_{s} becomes negligible. However, there exist devices such as cameras that may spend non negligible amounts of energy to collect a new image every few tens of milliseconds.

Data processing. Nodes may collect endogenous or exogenous data, which can both be processed either at an intra-node level or at an inter-node level with data aggregation and data fusion techniques. The latter approach highly depends on the network topology, the amount of information exchanged among nodes and the way data is processed. It is thus difficult to derive a general model; moreover, it is not used in any of the studies of this thesis, which only consider in-node compression operations. For data compression the energy cost can be quantified using the results in [43]:

$$E_{\mathrm{p}} = E_0 L_0 N_{\mathrm{p}}(\eta_{\mathrm{p}})\,. \tag{2.2}$$

E_0 is the energy consumption per CPU cycle (that depends on the micro-controller unit), L_0 is the number of bits used to represent the original signal, and $N_{\mathrm{p}}(\eta_{\mathrm{p}})$ represents the

number of clock cycles per bit needed to compress the input signal and is a function of the compression ratio η_{p}. Note that $N_{\mathrm{p}}(\eta_{\mathrm{p}})$ depends on the compression algorithm.

In both the studies of Chs. 3 and 4, it is assumed that the IoT nodes use the LTC algorithm or the Fourier-based Low Pass Filter (DCT-LPF) algorithm, which are very lightweight and suitable for constrained devices. In this case, the function $N_p(\eta_{\mathrm{p}})$ is increasing and concave in η_{p}:

$$N_{\mathrm{p}}(\eta_{\mathrm{p}}) = \alpha_p \eta_{\mathrm{p}} + \beta_p \tag{2.3}$$

with $\alpha_p, \beta_p > 0$. Notice that the more compressed the packet, the less the energy spent. This seemingly counterintuitive fact is due to implementation details; interested readers can refer to [43] for an explanation.

For what concerns channel coding, typically the energy it requires is assumed to be negligible with respect to the overall energy consumption and only the energy needed by the receiver for decoding is taken into account [50], hence this contribution may be considered only in terms of variation of the number of bits to be transmitted over the air (i.e., redundancy bits added for FEC/CRC).

Transmission. The energy cost of any wireless transmission period can be modeled as:

$$E_{\mathrm{t}x} = \frac{\tau P_{\mathrm{t}x}}{\eta_{\mathrm{A}}}, \tag{2.4}$$

where τ is the transmission duration, $P_{\mathrm{t}x}$ is the average radiated power, and $\eta_{\mathrm{A}} \in (0, 1]$ is a constant that models the efficiency of the antenna's power amplifier. This source of energy consumption should be considered for all transmissions performed by the IoT device, and thus also includes retransmission attempts and control messages, e.g., related to the maintenance/generation of the access schedule in coordinated access schemes.

Reception. When receiving a packet, the device spends energy to receive the radio signal, which can be modeled analogously to Eq. (2.4), and to reconstruct the original data from its compressed/encoded version. This latter contribution is highly algorithm-dependent and, to the best of our knowledge, there exists no general expression to characterize it. Also the energy required by advanced decoding algorithms (e.g., interference cancellation) should be taken into account. However, in the studies proposed here the focus is on the energy consumed for transmission, because the considered applications assume single-hop networks where the data sink is an energy-rich device.

Circuitry. Circuits spend some "basal" energy spent by the circuit in each of the node's possible operating states, $x \in \{\text{sleep}, \text{idle}, \text{active}\}$. A simple way to model it is the following:

$$E_{\mathrm{c}} = T_x \varepsilon_{\mathrm{c},x}, \tag{2.5}$$

where $\varepsilon_{c,x}$ is the rate of circuitry energy consumption when the node is in mode x, and T_x is the time spent by the device in that mode. Also going from mode x_1 to mode x_2 consumes energy, which is modeled as a constant contribution only depending on the two modes:

$$E_{\text{switch}} = k_{x_1,x_2}, \tag{2.6}$$

The switching time is assumed to be negligible, and for this reason E_{switch} does not depend upon it.

Energy harvesting. Devices that are not connected to the energy grid may harvest energy from the surrounding environment. Typically, the energy arrivals are assumed to follow a certain probability distribution, e.g., deterministic or Poisson. In case of energy supply that exhibits time correlation, some studies validated against real data have proved Markov Chains (MCs) to well model the energy inflow [51, 52]. Each possible state entails a different distribution of the energy income. Note that, when dealing with EH, the model is generally discrete, i.e., the energy inflow is quantized. Thus, the dynamics of the source can be tracked through an X-state MC: the source is in state $x \in \mathcal{X} = \{0, \dots, X-1\}$ and scavenges e_x quanta of energy from the environment, according to some probability mass function (p.m.f.).

Battery dynamics. Let B be the battery size of a device, which may be finite or infinite. A widely-adopted model for the battery charge b_n in slot n

$$b_n = \min\{\max\{0, b_{n-1} + e - u\}, B\} \tag{2.7}$$

where e and u are the harvested and used energy in the last slot, respectively. When the device has no EH capabilities, $e \equiv 0$. In case of EH, it is important to prevent battery outage (empty battery) and overflow (waste of excess energy because of full battery) situations; it is thus necessary to design schemes that dynamically adapts to the randomness of the energy inflow, so as to ensure acceptable performance on a long-term horizon.

Typically, the battery consumption is assumed to be linear as in (2.7), but actually it depends on the current battery charge and external factors like the temperature. This idealistic assumption leads to naïve lifetime estimations which often cause premature depletion in real deployments [53]. Nonetheless, it is extremely challenging to use more realistic models in the protocol design.

2.4 Proposed channel access schemes

We designed several channel access schemes for energy-constrained nodes in different scenarios. All schemes are based on a mathematical optimization and are described in the following chapters. Modeling and analyzing the performance of communication networks plays in fact an important role in the design of communication networks. The purpose of a mathematical model is to serve as guideline for the development of an efficient protocol and the performance analysis is beneficial to gain insight on the role played by the various system parameters and infer the tradeoffs among them. This allows one to optimize the network parameters and assess bound on the performance that can be obtained in practice.

This part of the thesis is structured as follows.

Ch. 3 discusses an adaptive TDMA scheme where the network resources are dynamically assigned to battery-powered devices according to their requirements (in terms of targeted QoS) and capabilities (channel conditions and energy availability) so that the network lifetime is maximized. The considered scenario includes data compression at the source as well as power control to counteract the channel fading; the two cases of complete and incomplete Channel State Information (CSI) at the transmitter are analyzed.

Ch. 4 also presents a TDMA approach, but in this cases the time resources assigned to each user are fixed. Devices are battery powered and can harvest energy from the environment; the goal is to optimally use the available energy to guarantee the desired QoS in terms of quality of the received data. Data compression and channel coding are optimized jointly as they induce a tradeoff between the accuracy in the data representation and the communication robustness, i.e., success probability of the transmission.

Ch. 5 analyzes the QoS/energy tradeoff for three different random access schemes targeting different monitoring applications with battery-powered sensors. In this case, collisions with other transmissions impinge on the energy efficiency, thus the channel access needs to be carefully designed taking into account possible interference. All schemes make use of some compression algorithms (sampling compression, data compression and communication compression) to adapt the sampling and transmission rates.

Finally, Ch. 6 studies a jamming attack, where an energy constrained jammer tries to disrupt the communication and deplete the battery of a legitimate transmitter. The attack is modeled with a game-theoretic approach and both the cases of complete and incomplete information available to the victim are analyzed.

3

Adaptive TDMA for IoT sensor networks

This chapter discusses a scheduling strategy for IoT sensors that adapts the data processing and the transmission parameters (transmission duration and power) to the energy and QoS requirements. Devices with heterogeneous capabilities and requirements access the channel in a TDMA fashion and the ultimate goal is to extend the network lifetime while guaranteeing a low overall distortion of the transmitted data with respect to their uncompressed version.

Other studies in the literature consider joint source coding and transmission policies and investigate the tradeoff between energy efficiency and data quality [54, 55]. In [56], an online joint compression and transmission optimization strategy is investigated for sensors with EH capabilities that generate correlated information, but how to schedule transmissions in a time slot is not treated. In [57], the authors derive optimal compression policies for a single sensor in order to minimize the long-term average distortion subject to the energy sustainability of the sensor, where power control is used to adapt the transmission to the status of the fading channel. In [58], energy allocation strategies are proposed with the goal of minimizing the signal distortion when several sensors measure the same process of interest and exploit data fusion techniques, but analytical results are derived only for a two-node system. Finally, [59] proposes a TDMA scheduling where time slots are allocated in a dynamic fashion based on the spatial correlation of the transmitted signals.

This study aims at determining the optimal operating point in the tradeoff between network lifetime and signal quality in order to derive a TDMA-based scheduling strategy for resource-constrained nodes. The devices have heterogeneous characteristics and requirements, and data compression and transmission are optimized jointly. Users are dismissed from transmission when it is impossible to have all devices meet their requirements in a frame. A set of theoretical results are derived to define processing and transmission policies when CSI is known perfectly (full knowledge) or statistically at the transmitters.

This work has been presented in [60] and [61].

Notation: In this study, matrices and vectors are represented with boldface letters, and the subscript and superscript refer to the row and column index, respectively; accordingly, $\mathbf{E}_i$ refers to the i-th row of matrix $\mathbf{E}$, $\mathbf{E}^{(k)}$ to its k-th column, and $E_i^{(k)}$ is the (i,k) element.

3.1 System Model

N heterogeneous sources send data to a central Base Station (BS), accessing the uplink channel in a TDMA fashion. Time is partitioned into frames, where frame k corresponds to the time interval interval $[t_k, t_{k+1})$. Each node periodically generates data, decides whether and how much to compress it, and finally transmits it to the common receiver.

3.1.1 *Data Generation and Compression*

Nodes generate data by collecting measurements from the surrounding environment or by serving as relays for farther nodes. Let $L_{0,i}^{(k)}$ be the size of the data generated in frame k by node i. Each user $i \in \mathcal{N}$ is capable of compressing its data using a lossy compression scheme, which may be source-specific. The compression operation affects the quality of the transmitted information and introduces a distortion $D_i^{(k)}$, which is a function of the compression ratio $L_i^{(k)}/L_{0,i}^{(k)}$, where $L_i^{(k)} \leq L_{0,i}^{(k)}$ is the size of the compressed packet in frame k. It is thus possible to define a function that maps the distortion to the transmission rate or, equivalently, to the compression ratio. Typically, closed-form expressions for the rate-distortion curves are only available for idealized compression techniques operating on Gaussian information sources [62], whereas for practical algorithms such curves are generally obtained experimentally. An example of *rate-distortion* curve, which will be used as a baseline in the numerical evaluation, is

$$D_i^{(k)} \triangleq D_i\left(L_i^{(k)}\right) = \left[b_i \left(\left(\frac{L_{0,i}^{(k)}}{L_i^{(k)}} \right)^{a_i} - 1 \right) \right]^+, \tag{3.1}$$

where $a_i, b_i > 0$ and $[\cdot]^+ \triangleq \max\{\cdot, 0\}$. It represents the maximum absolute error between the original and the compressed signal normalized to the amplitude range of the signal in the considered time window [10]. Notice that the distortion is zero when the packet is not compressed, i.e., $L_i^{(k)} = L_{0,i}^{(k)}$. This model is derived from the results obtained in [63], where a general parameterized expression for the rate-distortion curve was derived by compressing realistic time series with practical algorithms. The framework proposed in this study and its analysis do not rely on the particular shape of Eq. (3.1), but any other convex and decreasing function of the compression ratio $L_i^{(k)}/L_{0,i}^{(k)}$ could also be used.

The data collected in a frame is lost if not transmitted in the next frame, which is equivalent to imposing a strict limit on latency, or to considering finite data buffers at the devices. Hence, no retransmission mechanism is implemented.

Finally, there is a QoS requirement on the data quality: $D_i^{(k)} \leq D_{\text{th},i}^{(k)}$, where $D_{\text{th},i}^{(k)} < \infty$ is a threshold distortion level. If the reconstruction error exceeds this threshold, the signal generated by the source node is no longer useful for the final destination. The thresholds may depend on the size of the network, the transmission parameters (e.g., modulation), the data itself and other factors.

3.1.2 *Channel Model*

The average physical rate of user $i \in \mathcal{N}$ in frame k is approximated by Shannon's bound

$$r_i^{(k)} = W \log_2 \left(1 + \gamma_i^{(k)}\right) = W \log_2 \left(1 + h_i^{(k)} \frac{P_{\text{tx},i}^{(k)}}{N_0}\right), \tag{3.2}$$

where W is the bandwidth, $\gamma_i^{(k)}$ the Signal-to-Noise Ratio (SNR), $P_{\text{tx},i}^{(k)}$ the transmission power, $h_i^{(k)}$ the channel gain, and N_0 the noise power. The N channel gains $h_1^{(k)}, \ldots, h_N^{(k)}$ are affected by fast fading, which evolves independently over time and is independent among users. The transmission rate is approximated with Shannon's capacity and CSI is assumed at the transmitter. Therefore, in the absence of interference among devices, it is possible to ignore packet losses, that would have a negative impact on the QoS and require a retransmission or error-recovery mechanism [64].

3.1.3 *Energy Consumption Model*

All devices are battery-powered, and $B_i^{(k)}$ denotes the battery level of node i in frame k. The initial battery level $B_i^{(0)}$ represents the only energy available to node i, which therefore has a strong impact on the system performance. In every frame, a non-negative amount of energy $E_i^{(k)} \in [0, B_i^{(k)}]$ is used for processing and transmission tasks. The diverse sources of energy consumption can be characterized as described in 2.3. The ones used in this study are reported here for the sake of clarity.

Data Processing. The expression for the processing energy is given in (2.2); in this study each term has to be refereed to the corresponding device i and time frame k, and the compression ratio is given by $L_i^{(k)} / L_{0,i}$. Considering the expression given in (2.3), it is:

$$E_{\text{p},i}^{(k)} = E_{0,i}\, L_i^{(k)}\, \alpha_{p,i} + E_{0,i}\, L_{0,i}^{(k)}\, \beta_{p,i}. \tag{3.3}$$

Notice that the second term is independent of the compression ratio.

Data Transmission. The expression given in (2.4) needs to be considered for every node i in each frame k. Thus, the amount of energy consumed by the radio module is:

$$E_{\mathrm{tx},i}^{(k)} = \tau_i^{(k)} \frac{P_{\mathrm{tx},i}^{(k)}}{\eta_{A,i}}. \tag{3.4}$$

Note that no energy is wasted because of collisions and overhearing, since a TDMA-based access mechanism is adopted and devices are granted exclusive use of the communication channel in their slot (a single frame is composed by N slots).

Data Sensing and Circuitry Costs. As discussed in 2.3, the circuitry energy consumption can be modeled as

$$E_{\mathrm{c},i}^{(k)} = \beta_i^{(k)} + \varepsilon_{c,i}\tau_i^{(k)}, \tag{3.5}$$

where $\varepsilon_{c,i}$ is the circuitry power consumption during data transmission (see (2.5)). The constant term $\beta_i^{(k)}$ takes into account the energy spent to generate the sensor data (Eq. (2.1) with a constant number of samples per sensing period), the synchronization costs, and the energy spent to switch between sleep and active modes (see Sec. 2.3).

Total Energy Consumption. The total energy consumption of node i in frame k is obtained by summing Eqs. (3.3), (3.4) and (3.5): $E_{\mathrm{used},i}^{(k)} = E_{\mathrm{p},i}^{(k)} + E_{\mathrm{tx},i}^{(k)} + E_{\mathrm{c},i}^{(k)}$.

3.2 Optimization Problem

The goal of this study is to find a joint compression-transmission policy $(\boldsymbol{\tau}, \mathbf{L}, \mathbf{P})$ that decides how much to compress the data and how much time and power to assign to each node in each frame. The objective is to minimize the distortion given a time horizon of n frames and some initial per-node energy allocation. Solving this problem as a function of the value of n allows one to determine the tradeoff between the system lifetime and the corresponding achievable distortion performance. More specifically, given the lifetime n, the problem to solve is the following:

$$D_{\mathrm{mean}}^{\star} \triangleq \min_{\boldsymbol{\tau},\, \mathbf{L},\, \mathbf{P}} \frac{1}{n} \sum_{k=1}^{n} \max_{i \in \mathcal{N}} \frac{D_i^{(k)}}{D_{th,i}^{(k)}}, \tag{3.6a}$$

subject to:

$$D_i^{(k)} \leq D_{th,i}^{(k)}, \qquad \forall i, \quad \forall k, \tag{3.6b}$$

$$L_i^{(k)} \leq \tau_i^{(k)} r_i^{(k)}, \qquad \forall i, \quad \forall k, \tag{3.6c}$$

$$P_{\min,i} \leq P_{tx,i}^{(k)} \leq P_{\max,i}, \quad \forall i, \quad \forall k, \tag{3.6d}$$

$$\sum_{i=1}^{N} \tau_i^{(k)} \leq T, \qquad \forall k, \tag{3.6e}$$

$$\sum_{j=1}^{n} E_{\text{used},i}^{(j)} \leq B_i^{(0)}, \qquad \forall i. \tag{3.6f}$$

The distortion was defined in (3.1), and Constraint (3.6b) guarantees that it does not exceed the given threshold for any user. Inequality (3.6c) ensures that the channel capacity given in (3.2) is not exceeded. Quantities $P_{\min,i}$ and $P_{\max,i}$ in Constraint (3.6d) specify the minimum and maximum allowed transmission power, respectively. Constraint (3.6e) limits the total time allocated to the users to the frame duration. Notice that this is the only constraint that considers the users jointly: without it, Problem (3.6) could be readily decomposed into N separate problems. Finally, Constraint (3.6f) specifies that the total energy assigned to node i during the network lifetime cannot exceed the initial content of its battery, $B_i^{(0)}$. This is the only constraint that considers multiple frames simultaneously.

The objective function (3.6a) is the distortion averaged over the lifetime n. Note that the function $\max_{i\in\mathcal{N}} D_i^{(k)} / D_{th,i}^{(k)}$ considers the "worst" user in frame k in order to guarantee fairness. To achieve that, it is necessary to decide how much each node should compress its packet, and with what power and for how long it should transmit. It is important to remark that Problem (3.6) does not need to be computed by the IoT nodes. Indeed, the policy can be *evaluated offline* by the BS, disseminated using the downlink channel, and then *used online* by the users through a look-up table, whose size depends on the value of n and the granularity used in the discretization of the channel gain. The only computationally intensive action performed locally by the IoT nodes is *compression*. However, this can be realized using lightweight algorithms (e.g., LTC or DCT-LPF, see Sec. 3.1.3), which have very low complexity.

3.2.1 *Optimal Problem Decomposition*

For the sake of a simpler mathematical analysis, the problem is decomposed into two interdependent optimization processes. To do so, it is first necessary to formulate the original problem in an equivalent form. Consider the following problem:

$$D_{\text{mean}}^{\star} \triangleq \min_{\mathbf{E},\boldsymbol{\tau},\mathbf{L},\mathbf{P}} \frac{1}{n} \sum_{k=1}^{n} \max_{i\in\mathcal{N}} \frac{D_i^{(k)}}{D_{th,i}^{(k)}}, \tag{3.7a}$$

subject to: Constraints (3.6b)-(3.6e) , (3.7b)

$$E_{\text{used},i}^{(k)} \leq E_i^{(k)}, \quad \forall i, \quad \forall k, \tag{3.7c}$$

$$\sum_{j=1}^{n} E_i^{(j)} \leq B_i^{(0)}, \quad \forall i, \tag{3.7d}$$

which introduces the new auxiliary optimization variables $\mathbf{E} = \{\mathbf{E}^{(1)}, \ldots, \mathbf{E}^{(n)}\}$. For any feasible solution of (3.6), there exists a matrix $\mathbf{E}$ such that (3.7) also has a feasible solution. On the other hand, since Constraints (3.7c) and (3.7d) imply Inequality (3.6f) and all the other constraints have not changed, any feasible solution of (3.7c) is also feasible for (3.6). Therefore, the optimal solutions of the two problems coincide.

The term $E_i^{(k)}$ can be interpreted as the energy allocated to user i in frame k. According to this interpretation, Constraint (3.7c) guarantees that the energy used in a certain frame cannot exceed the amount of energy assigned to that frame, while Constraint (3.7d) states the exclusivity of the energy allocation, i.e., the fact that the amount of energy allocated to a particular frame cannot be used in any other frame. Thanks to the new variables $\mathbf{E}$, there is no constraint in (3.7) that considers the variables $\boldsymbol{\tau}$, $\mathbf{L}$, and $\mathbf{P}$ over multiple frames, therefore we can decompose Problem (3.7) in two intertwined blocks, namely FOP and EAP, as follows.

Frame-Oriented Problem (FOP). It focuses on a single frame k, assuming that the energy vector $\mathbf{E}^{(k)}$ is given. FOP is completely unaware of the energy allocation in the other frames. It can be formally stated as follows

FOP: $$f_{\text{FOP}}^{(k)}(\mathbf{E}^{(k)}) \triangleq \min_{\boldsymbol{\tau}^{(k)}, \mathbf{L}^{(k)}, \mathbf{P}^{(k)}} \max_{i \in \mathcal{N}} \frac{D_i^{(k)}}{D_{\text{th},i}^{(k)}}, \tag{3.8a}$$

subject to:

$$D_i^{(k)} \leq D_{\text{th},i}^{(k)}, \quad \forall i, \quad \forall k, \tag{3.8b}$$

$$L_i^{(k)} \leq \tau_i^{(k)} r_i^{(k)}, \quad \forall i, \quad \forall k, \tag{3.8c}$$

$$P_{\min,i} \leq P_{\text{tx},i}^{(k)} \leq P_{\max,i}, \quad \forall i, \quad \forall k, \tag{3.8d}$$

$$\sum_{i=1}^{N} \tau_i^{(k)} \leq T, \quad \forall k, \tag{3.8e}$$

$$E_{\text{used},i}^{(k)} \leq E_i^{(k)}, \quad \forall i, \quad \forall k. \tag{3.8f}$$

Its goal is to determine the transmission durations and powers, and the compression ratios that minimize Eq. (3.8a), for a selected frame and with a given energy allocation. There exist two versions of FOP, which differ in the level of CSI available at the nodes and will be discussed later.

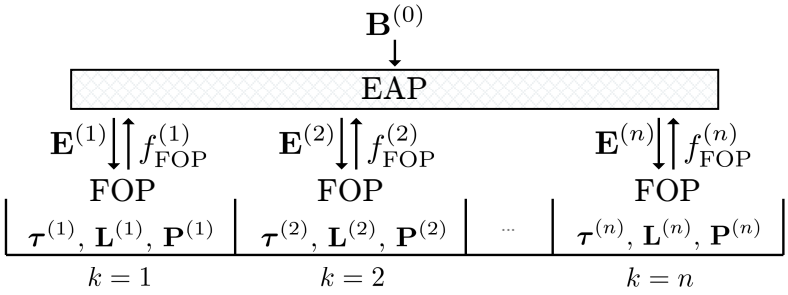


Figure 3.1: Structure of the problem for a fixed value of n: modules.

Energy Allocation Problem (EAP). EAP assumes the functions $f_{FOP}^{(k)}(\mathbf{E}^{(k)})$ to be known, and focuses on the optimization of the energy allocation over multiple frames. Formally,

$$\text{EAP:} \qquad D_{\text{mean}}^{\star} \triangleq \min_{\mathbf{E}} \frac{1}{n} \sum_{k=1}^{n} f_{\text{FOP}}^{(k)}(\mathbf{E}^{(k)}), \tag{3.9a}$$

$$\text{subject to:} \qquad \sum_{j=1}^{n} E_i^{(j)} \leq B_i^{(0)}, \quad \forall i, \tag{3.9b}$$

$$f_{\text{FOP}}^{(k)}(\mathbf{E}^{(k)}) \text{ is feasible}, \quad \forall k. \tag{3.9c}$$

Basically, EAP exploits FOP to obtain the distortion performance corresponding to a certain energy allocation $\mathbf{E}$ and determine the optimal energy allocation.

Fig. 3.1 shows the relation between the two problems, which are tightly coupled: EAP defines the energy allocation to use in every frame, which is used by FOP to determine (3.8a) and, on the other hand, the output of FOP influences EAP through (3.6a). FOP and EAP will be discussed in the following sections.

3.3 Frame-Oriented Problem (FOP)

The Frame-Oriented Problem does not consider the time evolution of the network and operates at the frame level, so that the amount of energy allocated to each node is fixed. The goal of FOP is to determine the optimal compression-transmission policy that minimizes the maximum normalized distortion experienced by the users in a single frame k. This section introduces FOP and proposes an equivalent formulation which is easier to be solved optimally. Secs. 3.4 and 3.5 discuss how to solve it under different assumptions on the knowledge available at the nodes.

For each user $i \in \mathcal{N}$, FOP determines:

1. The size of the data to transmit, $L_i^{(k)}$, which is strictly related to the distortion and to the energy consumed for processing as well as for transmitting.

2. The transmission power $P_{\mathrm{tx},i}^{(k)}$, which influences the transmission rate and energy consumption.
3. The transmission duration $\tau_i^{(k)}$, which relates $L_i^{(k)}$ to the channel rate $r_i^{(k)}$ and affects the consumed energy.

Problem (3.8) delineates the formal structure of FOP. Since FOP concerns single frames, for ease of notation the dependence on the frame index k throughout this section will be omitted. In this case, boldface letters refer to column vectors that span over the N users for the considered frame.

The objective function (3.8a) can be equivalently formulated by introducing an auxiliary optimization variable Γ:[1]

$$\text{FOP}_\Gamma: \qquad \min_{\Gamma,\boldsymbol{\tau},\mathbf{L},\mathbf{P}} \Gamma, \tag{3.10a}$$

$$\text{subject to:} \qquad \frac{D_i}{D_{\mathrm{th},i}} \leq \Gamma, \quad \forall i, \tag{3.10b}$$

$$\text{Constraints (3.8b)} - \text{(3.8f)}. \tag{3.10c}$$

Let $\Gamma^\star$ be the solution of FOP_Γ; since it depends on the energy allocated to each user, it can be explicitly written as $\Gamma^\star = f_{\text{FOP}}(\mathbf{E})$. Note that only distortions below threshold are acceptable (see (3.8b)), which means that $\Gamma^\star \leq 1$, otherwise FOP_Γ (and thus FOP) is infeasible. The optimal normalized distortion $\Gamma^\star$ can be determined exploiting the following result [61].

Lemma 1. *If* FOP_Γ *is feasible for* $\Gamma' \leq 1$, *then* FOP_Γ *is feasible for all* Γ'' *such that* $\Gamma' \leq \Gamma'' \leq 1$.

It follows that $\Gamma^\star$ can be found with a bisection search over the values of Γ in the interval $[0, 1]$. Thus, for every *fixed* Γ, it should be checked whether there exists a feasible solution (i.e., a solution that satisfies (3.10b) and (3.10c)) or not.

The optimization problem (3.10) and the concepts introduced hitherto have general validity and hold regardless of the knowledge of the channel status. Sec. 3.4 introduces the solution of (3.10b) without fading, which is simpler to determine and can be used as a building block for the other case (Sec. 3.5).

3.4 FOP with Full CSI

In this scenario, all nodes are assumed to have full CSI, i.e., the gain h_i in all frames is known exactly for all nodes a priori. For a fixed Γ, it is possible to determine the optimal

1 This is a standard approach to handle minimax problems. The new variable Γ represents an upper bound to $D_i/D_{\mathrm{th},i}$, $\forall i$, which is equivalent to an upper bound to $\max_i\{D_i/D_{\mathrm{th},i}\}$. Therefore, minimizing Γ or $\max_i\{D_i/D_{\mathrm{th},i}\}$ leads to the same solution.

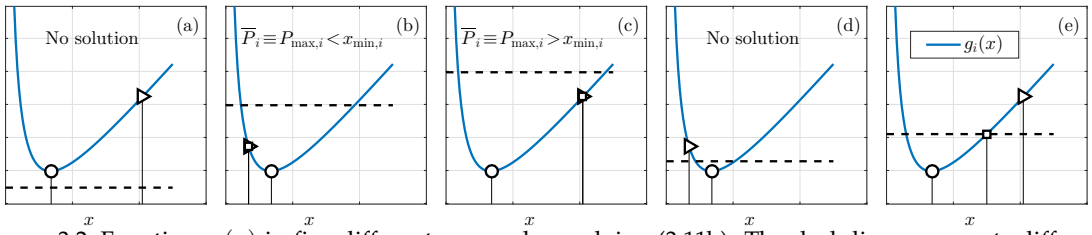


Figure 3.2: Function $g_i(x)$ in five different cases when solving (3.11b). The dash-line represents different values of $W/L_i(E_i - E_{\mathrm{p},i}(L_i) + \beta_i)$. The circle, triangle, and square markers represent $x_{\min,i}$, $P_{\max,i}$, and $P^{\star}_{\mathrm{tx},i} = \overline{P}_i$, respectively.

compression level, transmission power and duration for each user. Iterating over the values of Γ through a bisection search then yields the optimal $\Gamma^{\star}$.

This is possible thanks to the following key result.

Lemma 2. *There exists an optimal solution of* FOP_{Γ} *for which all users have equal relative distortion, i.e.,* (3.10b) *holds with equality* $\forall i$. *[61].*

This implies that, given Γ, the compression ratio is fixed for each user and, hence, $\mathbf{L}$ is also known deterministically and can be removed from the optimization variable.

The following result further simplifies the original Problem (3.10) and allows one to extract one optimal solution.

Lemma 3. *There exists at least one optimal solution where all nodes use the maximum available rate, i.e., Constraint* (3.8c) *is taken with equality* $\forall i$ *[61].*

Constraint (3.8c) is henced taken with equality, which is equivalent to choosing the smallest time possible τ_i when L_i and $P_{\mathrm{tx},i}$ are given. Lemmas 2 and 3 guarantee that, when Γ is fixed, the original Problem (3.10) can be reduced to a problem with a single optimization variable without loss of optimality. In particular, τ_i can be expressed as a function of $P_{\mathrm{tx},i}$, namely $\tau_i = L_i/(W \log_2(1 + h_i P_{\mathrm{tx},i}/N_0))$. The optimal solution uses the highest possible transmission power, since τ_i and $P_{\mathrm{tx},i}$ are inversely proportional to each other, and shorter transmission times are more likely to satisfy the frame duration constraint (3.8e). Accordingly, $P^{\star}_{\mathrm{tx},i}$ is chosen as the highest $P_{\mathrm{tx},i}$ that satisfies both the power (3.8d) and the energy (3.8f) constraints. Combining (3.8f) and Lemma 3 yields:

$$P^{\star}_{\mathrm{tx},i} \triangleq \max_{P_{\mathrm{tx},i} \in [P_{\min,i}, P_{\max,i}]} P_{\mathrm{tx},i}, \tag{3.11a}$$

subject to:
$$g_i(P_{\mathrm{tx},i}) \leq \frac{W}{L_i}(E_i - E_{\mathrm{p},i}(L_i) + \beta_i), \tag{3.11b}$$

with $g_i(x) \triangleq (x/\eta_{A,i} + \varepsilon_{c,i})/\log_2(1 + h_i\, x/N_0)$.

Note that all the terms on the right-hand side (RHS) of (3.11b) are fixed: E_i is given, L_i is derived from Γ through Lemma 2, and the remaining are system parameters. It can

Algorithm 1 Procedure to find $\overline{P}_i$

1: **if** $g(x_{\min,i}) > W/L_i(E_i - E_{\mathrm{p},i}(L_i) + \beta_i)$ **then**
2: no solution exists and $\overline{P}_i$ is undefined (case (a))
3: **else**
4: **if** $g(P_{\max,i}) \leq W/L_i(E_i - E_{\mathrm{p},i}(L_i) + \beta_i)$ **then**
5: set $\overline{P}_i = P_{\max,i}$ (cases (b) and (c))
6: **else**
7: **if** $x_{\min,i} \geq P_{\max,i}$ **then**
8: no solution exists and $\overline{P}_i$ is undefined (case (d))
9: **else**
10: find $\overline{P}_i$ with a bisection search in $[x_{\min,i}, P_{\max,i}]$ (case (e))

be shown that $g_i(x)$ is a decreasing-increasing function of x, and therefore admits only one minimum, as shown in Fig. 3.2 (and proved in [60]). Problem (3.11) can be solved by firstly using the golden-section search algorithm [65] to find the point of minimum $x_{\min,i}$ of $g_i(x)$, which is then used to determine the amount of power $\overline{P}_i$ that solves (3.11) when the constraint on $P_{\min,i}$ is neglected. The procedure is formally described in Algorithm 1 (see Fig. 3.2 for a graphical interpretation).

If $g(x_{\min,i}) > W/L_i(E_i - E_{\mathrm{p},i}(L_i) + \beta_i)$, there is no feasible point (Line 2). Otherwise, the algorithm checks whether the maximum solution, namely $P_{\max,\mathrm{i}}$, is feasible or not (Line 4). If this is not the case and $x_{\min,i} < P_{\max,i}$, a bisection search in $[x_{\min,i}, P_{\max,i}]$ is used to find $\overline{P}_i$ as the solution of $g(P_{\mathrm{tx},i}) = W/L_i(E_i - E_{\mathrm{p},i}(L_i) + \beta_i)$, i.e., the value of $P_{\mathrm{tx},i}$ that satisfies (3.11b) with equality (Line 10). If, using the previous procedure, $\overline{P}_i$ does not exist or $\overline{P}_i < P_{\min,i}$, then $P^{\star}_{\mathrm{tx},i}$ is not defined and the problem is infeasible for the given Γ; otherwise, $P^{\star}_{\mathrm{tx},i} = \overline{P}_i$. The packet size is straightforwardly determined from the selected Γ, and then the transmission duration is $\tau_i = L_i/(W \log_2(1 + h_i P^{\star}_{\mathrm{tx},i}/N_0))$. Then, it is possible to iterate over the values of Γ to find the optimal $\Gamma^{\star}$.

3.5 FOP with Statistical CSI

This scenario considers instantaneous CSI at the transmitter, i.e., exact knowledge of the channel gain only for the current slot, but only statistical knowledge of the future channel realizations, as typical in the presence of fading. Again, the dependence on the frame index will be omitted throughout this section, if not ambiguous.

The notion of "expected distortion" is introduced to account for the unknown channel status. A device transmits only if the channel gain is sufficiently high, since a deep fade would lead to unacceptable data quality. In particular, there is a fixed probability of performing a transmission, which yields the threshold above which the channel is "good enough". Then, the optimal $\Gamma^{\star}$ is determined using a bisection search as described in Sec. 3.3. For every fixed Γ, a dynamic policy adapts the compression ratio and the

transmission power to the channel status, and the transmission duration is determined as the smallest value that makes the problem feasible.

The channel coefficient can be decomposed as $h_i = h_{0,i}\,\theta_i$, where $h_{0,i}$ represents the average channel gain given by path loss and shadowing, and θ_i is the realization of a random variable Θ_i that models the fast-fading effects. Function $f_{\Theta_i}(\theta_i)$ represents the probability density function of Θ_i (e.g., $\Theta_i \sim \mathrm{Exp}(1)$ for Rayleigh fading). The terms $h_{0,i}$ are assumed to be known only for future frames, whereas only a statistical knowledge of θ_i is available. In this case, FOP defines different values of L_i and $P_{\mathrm{tx},i}$ as a function of θ_i, so that when a node executes the policy, it can dynamically adapt $L_i(\theta_i)$ and $P_{\mathrm{tx},i}(\theta_i)$ to the instantaneous channel conditions. The transmission duration τ_i, instead, is not dynamically adapted to θ_i but optimized regardless of the channel conditions to avoid coordination issues among the nodes.

3.5.1 *Distortion Definition with Fading*

In order to account for the statistical knowledge of the fading realization,the distortion function (3.1) needs to be redefined in terms of θ_i: $D_i(\theta_i) \triangleq D_i(L_i(\theta_i))$.

Then, given that the fading coefficient is above a certain threshold $\theta_{\mathrm{tx},i}$, it is possible to consider the conditional expectation of $D_i(\theta_i)$:

$$\overline{D}_i \triangleq \int_{\theta_{\mathrm{tx},i}}^{\infty} D_i(\theta_i) f_{\Theta_i}(\theta_i)\,\mathrm{d}\theta_i \Big/ \int_{\theta_{\mathrm{tx},i}}^{\infty} f_{\Theta_i}(\theta_i)\,\mathrm{d}\theta_i. \tag{3.12}$$

The parameter $\theta_{\mathrm{tx},i}$ is a fading gain threshold defined in order to avoid transmissions when the channel conditions are too bad (and would require excessive data compression and, consequently, distortion). In practice, nodes refrain from data transmission in a frame if the current fading gain is lower than $\theta_{tx,i}$. To guarantee a certain level of QoS, the transmission probability $\overline{\mathrm{Pr}}_{\mathrm{tx},i}$ is fixed a priori, and $\theta_{\mathrm{tx},i}$ is derived consequently. For example, with Rayleigh fading it is $\theta_{\mathrm{tx},i} = -\log(\overline{\mathrm{Pr}}_{\mathrm{tx},i})$.

Throughout this subsection, and when we deal with the statistical CSI case in general, D_i in (3.8a) is replaced by $\overline{D}_i$, and the optimization variables $\mathbf{L}(\boldsymbol{\theta}) = \{L_1(\theta_1), \ldots, L_N(\theta_N)\}$ and $\mathbf{P}(\boldsymbol{\theta}) = \{P_1(\theta_1), \ldots, P_N(\theta_N)\}$ depend on the channel status $\boldsymbol{\theta}$. The equivalent structure of FOP (see (3.8)) is:

$$\min_{\boldsymbol{\tau},\mathbf{L}(\boldsymbol{\theta}),\mathbf{P}(\boldsymbol{\theta})} \max_{i\in\mathcal{N}} \frac{\overline{D}_i}{D_{\mathrm{th},i}}, \tag{3.13a}$$

subject to:

$$D_i(\theta_i) \leq D_{th,i}, \qquad \forall i, \quad \forall \theta_i \geq \theta_{tx,i}, \tag{3.13b}$$

$$L_i(\theta_i) \leq \tau_i \, r_i(\theta_i), \qquad \forall i, \quad \forall \theta_i \geq \theta_{tx,i}, \tag{3.13c}$$

$$P_{\min,i} \leq P_{tx,i}(\theta_i) \leq P_{\max,i}, \quad \forall i, \quad \forall \theta_i \geq \theta_{tx,i}, \tag{3.13d}$$

$$\textstyle\sum_{i=1}^{N} \tau_i \leq T, \tag{3.13e}$$

$$E_{\text{used},i}(\theta_i) \leq E_i, \qquad \forall i, \quad \forall \theta_i \geq \theta_{tx,i}. \tag{3.13f}$$

The equivalent of FOP_Γ can be derived analogously.

To solve FOP_Γ, it is necessary to specify how to find the optimal packet size and transmission power as a function of θ_i, which together allow one to optimally find τ_i. Similar to Sec. 3.4, FOP_Γ is then solved by applying a bisection search.

3.5.2 *Packet Length and Transmission Power*

For the moment, consider the transmission duration vector $\boldsymbol{\tau}$ to be given (it will be optimized in Sec. 3.5.3). Thus, the Constraint (3.13e) can be disregarded, and the objective function (3.13a) can be rewritten as

$$\begin{aligned}\min_{\mathbf{L}(\boldsymbol{\theta}),\mathbf{P}(\boldsymbol{\theta}),\forall \boldsymbol{\theta} \succeq \boldsymbol{\theta}_{tx}} \max_{i \in \mathcal{N}} \frac{\overline{D}_i}{D_{th,i}} &= \max_{i \in \mathcal{N}} \min_{L_i(\theta_i),P_{tx,i}(\theta_i),\forall \theta_i \geq \theta_{tx,i}} \frac{\overline{D}_i}{D_{th,i}} \\ &= \max_{i \in \mathcal{N}} \frac{\int_{\theta_{tx,i}}^{\infty} f_{\Theta_i}(\theta_i) \min_{L_i(\theta_i),P_{tx,i}(\theta_i)} D_i(\theta_i) \, \mathrm{d}\theta_i}{D_{th,i} \int_{\theta_{tx,i}}^{\infty} f_{\Theta_i}(\theta_i) \, \mathrm{d}\theta_i},\end{aligned}$$

$\overline{D}_i$ depends only on $L_i(\cdot)$ and $P_{tx,i}(\cdot)$, thus, it is possible to focus on every min term in the numerator and study it independently for every fixed θ_i:

$$(L_i^\star(\theta_i), P_{tx,i}^\star(\theta_i)) \triangleq \operatorname*{argmin}_{L_i(\theta_i),P_{tx,i}(\theta_i)} D_i(\theta_i) = \operatorname*{argmax}_{L_i(\theta_i),P_{tx,i}(\theta_i)} L_i(\theta_i), \tag{3.14a}$$

subject to:

$$L_{th,i} \leq L_i(\theta_i) \leq \tau_i W \log_2\left(1 + h_0 \theta_i \frac{P_{tx,i}(\theta_i)}{N_0}\right) \tag{3.14b}$$

$$P_{\min,i} \leq P_{tx,i}(\theta_i) \leq P_{\max,i} \tag{3.14c}$$

$$E_{p,i}(L_i(\theta)) + \beta_i + \left(\frac{P_{tx,i}(\theta)}{\eta_{A,i}} + \varepsilon_{c,i}\right)\tau_i \leq E_i \tag{3.14d}$$

where $L_{th,i}$ is the value such that (3.13b) is satisfied with equality, Constraint (3.14b) coincides with (3.13b) and (3.13c), (3.14c) is equivalent to (3.13d), and (3.14d) is the energy constraint (3.13f). Moreover, since, by definition, the distortion decreases with the packet length, minimizing $D_i(\theta_i)$ is equivalent to maximizing $L_i(\theta_i)$ in (3.14a).

Notice that the left-hand side (LHS) of (3.14d) increases with $L_i(\theta_i)$. Thus, $L_i(\theta_i)$ is upper bounded by

$$L_i(\theta_i) \leq \min \left\{ \begin{array}{l} \tau_i W \log_2 \left(1 + h_0 \theta_i \frac{P_{\text{tx},i}(\theta_i)}{N_0}\right), \\ E_{\text{p},i}^{-1}(E_i - \beta_i - (P_{\text{tx},i}(\theta_i)/\eta_{A,i} + \varepsilon_{c,i})\tau_i) \end{array} \right\}, \tag{3.15}$$

where $E_{\text{p},i}^{-1}(\cdot)$ is the inverse function of $E_{\text{p},i}(\cdot)$. Since the goal is to maximize $L_i(\theta_i)$, the optimal solution must satisfy Eq. (3.15) with equality. Accordingly, three different cases can be considered.

Full Channel Capacity. In this case $L_i(\theta_i) = \tau W \log_2(1 + h_0 \theta_i \, P_{\text{tx},i}(\theta_i)/N_0)$, so that the full channel capacity is used. $L_i(\theta)$ is a function of $P_{\text{tx},i}(\theta)$, thus it is possible to focus on the optimization of the transmission power only:

$$\underset{P_{\text{tx},i}(\theta_i)}{\text{argmax}} \tau W \log_2 \left(1 + h_0 \theta_i \frac{P_{\text{tx},i}(\theta_i)}{N_0}\right) = \underset{P_{\text{tx},i}(\theta_i)}{\text{argmax}} P_{\text{tx},i}(\theta_i), \tag{3.16a}$$

subject to:

$$P_{\min,i} \leq P_{\text{tx},i}(\theta_i) \leq P_{\max,i} \tag{3.16b}$$

$$\begin{aligned} &E_{\text{p},i}\left(\tau_i W \log_2(1 + h_0 \theta_i P_{\text{tx},i}(\theta_i)/N_0)\right) + \beta_i \\ &\quad + \left(\frac{P_{\text{tx},i}(\theta)}{\eta_{A,i}} + \varepsilon_{c,i}\right)\tau_i \leq E_i. \end{aligned} \tag{3.16c}$$

The LHS of (3.16c) increases with $P_{\text{tx},i}$ and the RHS is a constant. Three subcases should be analyzed: i) Constraint (3.16c) cannot be satisfied even using $P_{\min,i}$, ii) Constraint (3.16c) can be satisfied using $P_{\max,i}$, and iii) neither of the previous two. In case i), no solution exists and the procedure falls back on the *Partial Channel Capacity* case, as using the whole channel capacity is infeasible, because the allocated energy E_i would not be sufficient. In case ii), the optimal solution is $P_{\max,i}$, as it maximizes the objective function (3.16a) while satisfying (3.16b) and (3.16c). In case iii), the optimal solution satisfies Constraint (3.16c) with equality and $P_{\text{tx},i}^{\star}(\theta_i)$ can be found with a bisection search. The optimal packet length is then given by $L_i^{\star}(\theta_i) = \tau_i W \log_2(1 + h_{0,i} \theta_i \, P_{\text{tx},i}^{\star}(\theta_i)/N_0)$.

Partial Channel Capacity. If the whole channel capacity cannot be used, then $L_i(\theta_i) = E_{\text{p},i}^{-1}(E_i - \beta_i - (P_{\text{tx},i}(\theta_i)/\eta_{A,i} + \varepsilon_{c,i})\tau_i)$ (see (3.15)), and Problem (3.14) can be expressed as

$$\underset{P_{\text{tx},i}(\theta_i)}{\text{argmax}} E_{\text{p},i}^{-1}(E_i - \beta_i - (P_{\text{tx},i}(\theta_i)/\eta_{A,i} + \varepsilon_{c,i})\tau_i) = \underset{P_{\text{tx},i}(\theta_i)}{\text{argmin}} P_{\text{tx},i}(\theta_i), \tag{3.17a}$$

subject to:

$$P_{\min,i} \leq P_{\mathrm{tx},i}(\theta_i) \leq P_{\max,i}, \tag{3.17b}$$

which yields

$$L_i^\star(\theta_i) = E_{\mathrm{p},i}^{-1}(E_i - \beta_i - (P_{\min,i}/\eta_{A,i} + \varepsilon_{c,i})\tau_i), \tag{3.18}$$

$$P_{\mathrm{tx},i}^\star(\theta_i) = P_{\min,i}. \tag{3.19}$$

No Solution. If with the previous two approaches $L_i^\star(\theta_i) < L_{\mathrm{th},i}$, the problem is infeasible.

3.5.3 *Optimal Transmission Duration*

The previous sections explained how to optimally set the transmission parameters for a given transmission duration; this section discusses how to optimize $\boldsymbol{\sigma}$. Consider FOP_Γ, assume that Γ is given; then, the objective is to find $\boldsymbol{\tau}$ such that the constraints on the tolerable distortion, channel capacity, transmission power, frame duration and energy consumption (i.e., (3.8b)-(3.8f)) hold, together with Constraint (3.10b) on Γ. Since $\overline{D}_i$ depends on τ_i, there may be many values of τ_i that satisfy $\overline{D}_i \leq \Gamma D_{\mathrm{th},i}$ (Constraint (3.10b)). Among all these values, we choose the *lowest*; by doing so for every i, the constraint on the frame duration (3.8e) is satisfied whenever possible. Therefore, it is possible to focus on each device i independently.

Given τ_i, $\overline{D}_i$ can be evaluated using $\theta_{\mathrm{tx},i}$, $L_i^\star(\theta_i)$, and $P_{\mathrm{tx},i}^\star(\theta_i)$ as described in Secs. 3.5.1 and 3.5.2. Since the optimal packet size increases with the channel gain, Problem (3.14) is feasible $\forall \theta_i \geq \theta_{\mathrm{tx},i}$ if

$$L_i^\star(\theta_{\mathrm{tx},i}) \geq L_{\mathrm{th},i}. \tag{3.20}$$

The following important property of the distortion function allows one to determine the transmission duration τ_i that satifies this condition.

Lemma 4. *For a given θ_i, the optimal distortion $D_i^\star(\theta_i)$ decreases with τ_i until a minimum distortion point at $\tau_i^{\min}(\theta_i)$, and increases for $\tau_i > \tau_i^{\min}(\theta_i)$ [61].*

If $\tau_i^{\min}(\theta_{\mathrm{tx},i})$ computed at $\theta_{\mathrm{tx},i}$ does not satisfy (3.20), FOP is infeasible. Otherwise, there exists an interval $[\tau_i^{\mathrm{low}}, \tau_i^{\mathrm{high}}]$, with $\tau_i^{\mathrm{low}} \leq \tau_i^{\min}(\theta_{\mathrm{tx},i}) \leq \tau_i^{\mathrm{high}}$, for which (3.20) holds. To solve FOP, consider the *smallest* value of $\tau_i \in [\tau_i^{\mathrm{low}}, \tau_i^{\mathrm{high}}]$ that satisfies $\overline{D}_i \leq \Gamma D_{\mathrm{th},i}$ in order to make all users fit in the frame duration and satisfy (3.8e). If $\overline{D}_i > \Gamma D_{\mathrm{th},i}$, $\forall \tau_i \in [\tau_i^{\mathrm{low}}, \tau_i^{\mathrm{high}}]$, then the constraints of (3.10) cannot be satisfied and FOP_Γ is infeasible.

This procedure holds for a fixed Γ; the optimal $\Gamma^\star$ is found by iterating over the values of Γ with a bisection search.

3.6 Dismission Policy

It may happen that the optimization over all N users fails and FOP turns out to be infeasible in a specific frame. This happens if at least one constraint of FOP is not satisfied, i.e., there exists no allocation of $\boldsymbol{\tau}$, $\mathbf{L}$, and $\mathbf{P}$ that allows all users to transmit their packets in the considered frame with the allocated energy. Notice that FOP is infeasible even if the time, capacity, power, and energy constraints can be met but at least one user exceeds its threshold distortion, thereby violating the QoS constraint (3.8b).

Since any constraint can be relaxed, the only strategies available when FOP is infeasible are to allocate a larger amount of energy or to dismiss some users. The former possibility is discussed here, whereas the energy allocation problem will be analyzed in Sec. 3.7.

Let $\underline{\tau}_i$ be the minimum transmission time user i needs to transmit its data in the considered frame. This value is obtained when $P_{\mathrm{tx},i} = P_{\max,i}$ and $D_i = D_{th,i}$, which means that the size of the packet to send, L_i, is the smallest possible. If the inequality $\sum_{i=1}^{N} \underline{\tau}_i \leq T$ is not verified, it is impossible to allow all users to transmit in the same frame and satisfy their requirements at the same time, whatever the available energy. In this case, users are assigned different importance levels to the users so that it is possible to develop a priority-based dismission policy.When a user with low priority is dismissed, the condition $\sum_{i=1}^{N} \underline{\tau}_i \leq T$ is rechecked, and the same dismission procedure is applied, if necessary. Note that this dismission policy is *independent* of the energy allocation and is performed before EAP.

3.7 Energy-Allocation Problem (EAP)

Secs. 3.3–3.6 discussed how to maximize the quality of the transmitted information assuming that a known amount of energy is assigned to each user. To solve Problem (3.6), it is necessary to distribute over time the total energy available to each user, $\mathbf{B}^{(0)}$. In general, there is a balance between lifetime and distortion that depends on the value of n. This section describes how to solve the Energy-Allocation Problem (3.9), where $f_{\mathrm{FOP}}^{(k)}$ is derived as described in the previous sections.

EAP is defined in (3.9) as the problem of optimally allocating energy to each frame, given the lifetime n. The objective function (3.9a) represents the average over the lifetime of the distortion metric $f_{\mathrm{FOP}}^{(k)}(\cdot)$ in every frame. Therefore FOP needs to be feasible for all frames, as required by (3.9c), otherwise EAP would be infeasible (see Sec. 3.6). Unfortunately, the solution of FOP cannot be expressed in closed form, making it very challenging to directly relate FOP and the energy allocation. In the following, $f_{\mathrm{FOP}}^{(k)}(\mathbf{E}^{(k)})$

is assumed to be convex.[2] Then, note that the constraints induce a convex feasibility set and thus EAP becomes a convex optimization problem.

3.7.1 *Solution of EAP*

Let EAP_ℓ be a "reduced" version of EAP, where the optimization is done only over the energy vector $\mathbf{E}_\ell = [E_\ell^{(1)}, \dots, E_\ell^{(n)}]$, $\ell \in \mathcal{N}$, and all the other variables in $\mathbf{E}$ are kept fixed:

$$\text{EAP}_\ell: \qquad \min_{\mathbf{E}_\ell} \sum_{k=1}^{n} f_{\text{FOP}}^{(k)}(\mathbf{E}^{(k)}), \tag{3.21a}$$

subject to:

$$\sum_{k=1}^{n} E_\ell^{(k)} \leq B_\ell^{(0)}, \tag{3.21b}$$

$$\underline{E}_\ell^{(k)} \leq E_\ell^{(k)} \leq \overline{E}_\ell^{(k)}, \quad \forall k. \tag{3.21c}$$

The objective (3.21a) and the energy constraint (3.21b) are derived straightforwardly from (3.9a) and (3.9b), respectively. Constraint (3.21c) ensures that the solution is feasible and that (3.9c) in the original formulation is satisfied. The lower bound $\underline{E}_\ell^{(k)}$ represents the minimum amount of energy that node ℓ needs to make FOP feasible in frame k, when the energy allocated to the other devices is $E_i^{(k)}$, $i \in \mathcal{N} \setminus \{\ell\}$ [61]. Similarly, $\overline{E}_\ell^{(k)}$ is the energy threshold such that any $E_\ell^{(k)} \geq \overline{E}_\ell^{(k)}$ does not make the objective function decrease further (even if more energy were allocated, FOP would yield the same solution because of, e.g., the constraint on the maximum transmission power or on the frame duration). Clearly, both $\underline{E}_\ell^{(k)}$ and $\overline{E}_\ell^{(k)}$ depend on the energy levels of the other nodes, $\{E_1^{(k)}, \dots, E_{\ell-1}^{(k)}, E_{\ell+1}^{(k)}, \dots, E_N^{(k)}\}$. The solution of FOP for frame k is decreasing in $E_\ell^{(k)}$ and, in particular, it turns out to be strictly decreasing in $(\underline{E}_\ell^{(k)}, \overline{E}_\ell^{(k)})$. Consequently, EAP_ℓ can be solved using the Lagrangian dual problem, where the objective becomes $\mathcal{L}(\lambda, \mathbf{E}_\ell) \triangleq \max_{\lambda,\, \mathbf{E}_\ell} \sum_{k=1}^{n} f_{\text{FOP}}^{(k)}(\mathbf{E}^{(k)}) - \lambda \left(\sum_{k=1}^{n} E_\ell^{(k)} - B_\ell^{(0)} \right)$. All solutions must satisfy the Karush-Kuhn-Tucker conditions

$$E_\ell^{(k)} = \max\{\underline{E}_\ell^{(k)}, \min\{\overline{E}_\ell^{(k)}, \psi^{-1}(\lambda)\}\}, \tag{3.22}$$

$$\psi(E_\ell^{(k)}) \triangleq \frac{\partial f_{\text{FOP}}^{(k)}(\mathbf{E}^{(k)})}{\partial E_\ell^{(k)}}, \tag{3.23}$$

2 Here we assume that the solution of FOP is either convex, or approximated as such. The solution of EAP is optimal under such assumption, whereas it is to be considered a heuristic approximation otherwise. See [60] and [61] for a deeper discussion on such approximation.

Algorithm 2 Random Alternate Convergence Algorithm

1: Initialize a feasible $\mathbf{E}$
2: $D_{\text{mean}} \leftarrow \infty$
3: **while** $D_i^{(k)}$, $\forall i, \forall k$ have not converged **do**
4: **for** $\ell = 1, \dots, N$ **do**
5: $\mathbf{E}_\ell \leftarrow$ solve $\text{EAP}_\ell(\mathbf{E})$
6: $v \leftarrow$ prob. vector of size $\sum_k \chi\{E_\ell^{(k)} = \overline{E}_\ell^{(k)}\}$
7: $S \leftarrow \sum_{k=1}^{n} E_\ell^{(k)}$ ▷ consumed energy
8: $v_{\text{ind}} \leftarrow 1$ ▷ index of frames with $E_\ell^{(k)} = \overline{E}_\ell^{(k)}$
9: **for** $k = 1, \dots, n$ **do**
10: **if** $E_\ell^{(k)} = \overline{E}_\ell^{(k)}$ **then**
11: $E_\ell^{(k)} \leftarrow E_\ell^{(k)} + v(v_{\text{ind}}) \cdot (B_\ell^{(0)} - S)$
12: $v_{\text{ind}} \leftarrow v_{\text{ind}} + 1$
13: $D_{\text{mean}} \leftarrow 1/n \ \sum_{k=1}^{n} f_{\text{FOP}}^{(k)}(\mathbf{E}^{(k)})$
14: $D_{\text{mean}}^{\star} \leftarrow D_{\text{mean}}$

where the weight λ is such that $\sum_{k=1}^{n} E_\ell^{(k)} = B_\ell^{(0)}$, which is obtained by imposing $\partial \mathcal{L}(\lambda, \mathbf{E}_\ell)/\partial \lambda = 0$. Eq. (3.22) can be interpreted as a water-filling solution with minimum and maximum levels and in which the water level (i.e., the allocated energy) $\psi^{-1}(\lambda)$ is put in every frame k, if possible.

Random Alternate Convergence Algorithm. EAP can be solved by using an alternate approach based on EAP_ℓ, that focuses on one user at a time. The procedure is described in Algorithm 2. The key idea is to perform the optimization of a single user at every iteration, until the distortion for every user in every frame, i.e., $D_i^{(k)}$ (or $\overline{D}_i^{(k)}$ with imperfect CSI), does not change further (convergence condition). The alternate optimization is done in Lines 4-12: matrix $\mathbf{E}$ is used in Line 5 to solve EAP_ℓ (that, under the considered assumptions, is a convex optimization problem) for a specific user ℓ and to update the ℓ-th row of the energy matrix. It may happen that part of the initial energy is not used, i.e., $\sum_{k=1}^{n} E_\ell^{(k)} < B_\ell^{(0)}$. Accordingly, Lines 6-12 randomly distribute the residual energy $B_\ell^{(0)} - \sum_{k=1}^{n} E_\ell^{(k)}$ among all the frames for which $E_\ell^{(k)}$ is equal to $\overline{E}_\ell^{(k)}$ ($\chi\{\cdot\}$ is the indicator function). Note that, because of how $\overline{E}_\ell^{(k)}$ is defined, this operation does not change the distortion level obtained by solving EAP_ℓ, but simply provides a new $\mathbf{E}_\ell$ that allows the alternate optimization to converge. In particular, the probability vector v (Line 6) assigns a random positive weight to all of these elements, which are then normalized such that $\sum_{v_{\text{ind}}} v(v_{\text{ind}}) = 1$. The residual energy is then distributed according to such weights (Line 11).

Note that, if $f_{\text{FOP}}^{(k)}(\mathbf{E}^{(k)})$ is convex, the alternate approach of Algorithm 2, in which all users are sequentially considered and optimized one by one, leads to the optimal solution. See [61].

Summary. Since Algorithm 2 solves EAP, it also solves the original problem (3.6). In particular, EAP allocates energy over time using an alternate procedure: at every iteration of Algorithm 2, EAP_ℓ is solved for the current node ℓ, and FOP is invoked multiple times to evaluate the derivative in (3.23), which relates the allocated energy to the corresponding distortion. Such distortion is used by EAP to evaluate Eq. (3.6a)≡(3.9a) and decide how to update the energy allocation matrix, till convergence. Every time that FOP is invoked, the optimization procedures of Secs. 3.4 or 3.5 are used. By solving Problem (3.6) as n varies, we can find the optimal distortion as a function of the lifetime, thereby characterizing the energy/distortion tradeoff for our system.

3.8 Numerical Evaluation

Problem (3.6) was solved using the decomposition in EAP and FOP and Algorithm 2 as described in the previous sections for different scenarios to numerically assess the influence of the system parameters on the distortion and lifetime of the network. All cases envisage three groups of nodes placed at different locations. The two policies (FOP with full CSI and statistical CSI) have been compared in a realistic setting, where the channel evolution is not known in advance. A possible way to use the full-CSI policy in this context is to impose $\theta_i = \theta_{\text{tx},i}$ and derive the policy accordingly. Indeed, if the solution is feasible at $\theta_{\text{tx},i}$, it is guaranteed to be feasible even for better channel conditions, but it is expected to provide worse performance than the optimal statistical-CSI policy.

The transmission parameters are taken from the datasheets of two real devices, namely the RN-131C 802.11 b/g Wireless LAN Module (a module with extremely low power consumption for Wi-Fi connections) and the RC2400HP RF Transceiver Module (an RF module based on ZigBee and IEEE 802.15.4). The former uses a central transmission frequency and bandwidth of 2.441 GHz and $W = 5$ MHz, respectively. The maximum power used for transmission is 237.7 mW, and, when a transmission is performed, the minimum and maximum consumed powers are 462 mW and 699.6 mW, respectively. When only the RF chain is considered, assuming a minimum transmission power $P_{\text{min},i} = 100$ mW and according to Sec. 3.1.3, it is $\eta_{A,i} = 0.58$ and $\varepsilon_{c,i} = 167.75$ mW (note that this is an approximation, and the values may slightly change depending on $P_{\text{min},i}$).

The RC2400HP module uses the same central frequency and bandwidth of the other module, but has different energy related parameters; in particular, $P_{\text{min},i} = 11.22$ mW, $P_{\text{max},i} = 107.15$ mW, $\varepsilon_{c,i} = 60.15$ mW and $\eta_{A,i} = 0.23$.

If not otherwise stated, the other parameters are common to all devices, and in particular: the constant energy term β_i (see (3.5)) is equal to 1 mJ in every frame; the energy consumption function due to the processing evolves linearly with $L_i^{(k)}$ as in Eq. (3.3), with a slope of $E_{0,i}\,\alpha_{p,i} = 50$ nJ/bit and a coefficient $\beta_{p,i} = 0$ bit^{-1}; the channel

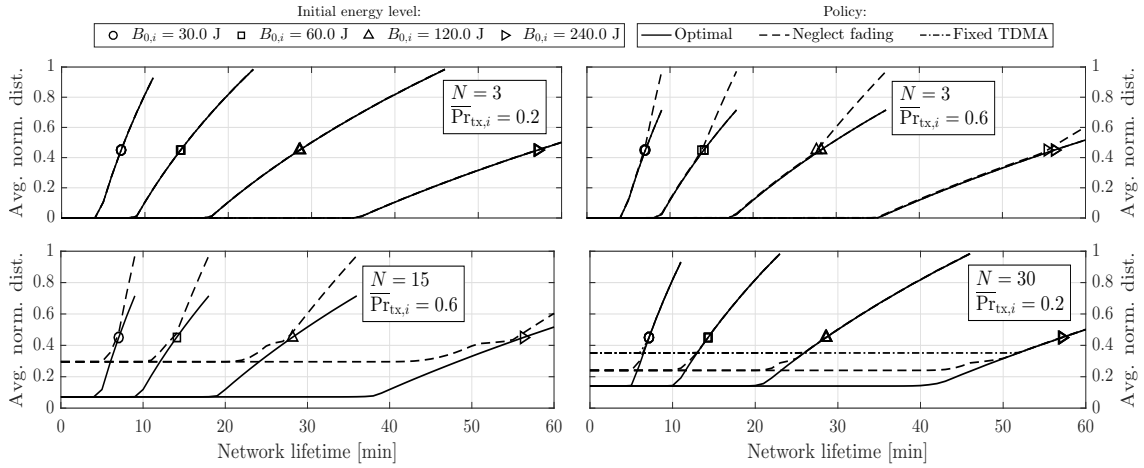


Figure 3.3: Average normalized distortion as a function of the lifetime n with fading. We do not explicitly represent the scenario $N = 15$ and $\overline{\Pr}_{tx} = 0.2$ because it is analogous to the case $N = 3$ and $\overline{\Pr}_{tx} = 0.2$. Moreover, the case $N = 30$ and $\overline{\Pr}_{tx} = 0.6$ has no solutions since too many users are considered and $\overline{\Pr}_{tx}$ is too high. The "Fixed-TDMA" policy coincides with the optimal one when $N = 3$ and cannot be derived when $N = 15$ and $\overline{\Pr}_{tx} = 0.6$.

gains are computed using the standard path loss model with a path-loss exponent equal to 3.5 (e.g., as in an urban scenario) and are affected by Rayleigh fading; the overall noise power spectral density is -167 dBm/Hz; the frame duration T is 1 s. The parameters of the distortion model (3.1) are $a_i = 0.35$ and $b_i = 19.9$, which have been derived empirically fitting the realistic rate-distortion curves of [43]. Moreover, nodes are divided in three groups, G_1, G_2 and G_3. Nodes in G_1 and G_2 use the parameters of the RN-131C module, whereas devices in G_3 use the RC2400HP module. The groups are heterogeneous in terms of distance from the base station, amount of data to send, and QoS requirements. These parameters are constant during the whole lifetime. The first group includes nodes that are located at a distance $d_{G_1} = 4$ m from the BS (very close to the base station), transmit packets with $L_{0,i}^{(k)} = 2$ Mbit and demand a distortion below $D_{th,i}^{(k)} = 8\%$. Nodes in G_2 have $d_{G_2} = 20$ m, $L_{0,i}^{(k)} = 1$ Mbit and looser distortion requirements $D_{th,i}^{(k)} = 15\%$. Finally, the third group consists of nodes far away from the BS ($d_{G_3} = 100$ m), which transmit fewer bits ($L_{0,i}^{(k)} = 10$ kbit), but require better QoS ($D_{th,i}^{(k)} = 4\%$). Although for simplicity the numerical evaluation only considers two cases of distortion thresholds, in a real scenario these values may be different at every node according to the network condition and to the application requirements. Note however that our model is able to handle the heterogeneity of the analysed scenario, which is a key feature for the IoT.

Distortion vs. lifetime. Fig. 3.3 shows the distortion vs. the lifetime obtained by solving the optimization problem (3.6a) for different values of n. We considered different values of the number of nodes ($N \in \{3, 15, 30\}$), uniformly distributed among the three groups, and a transmission probability $\overline{\Pr}_{tx} \in \{0.2, 0.6\}$. The continuous lines represent the

optimal solution described in Sec. 3.5, which explicitly accounts for the fading effects. The dashed lines are obtained using the full-CSI policy of Sec. 3.4 with $\theta_i = \theta_{\mathrm{tx},i}$; since it assumes the channel gains to be constant during the whole frame, this policy is suboptimal when the channels are actually affected by fading. Finally, the dash-dot lines represent a fixed-TDMA policy in which the time available to every user is at most $T^{(k)}/N$; this approach is suboptimal because it does not consider the network heterogeneity and gives the same amount of resources to every node.

The curves have been obtained by changing the value of the lifetime n, and the dismission procedure of Sec. 3.6 is not performed here. The lifetime is very short because very high traffic conditions were considered, with each node having data to transmit in every time frame. Although this scenario mat not always be realistic, it is significant for two reasons: 1) it gives information about the maximum performance achievable by the network, since with lower traffic the distortion cannot get worse, and 2) it can be easily remapped to a more general model with many more nodes and lower traffic patterns.

The distortion tends to increase with the lifetime, as expected, since smaller amounts of energy can be allocated in each frame and thus nodes must compress more to transmit their data. For small values of n, the curves are constant because the target lifetime objective is reached even without depleting the batteries. Clearly, in this case it is possible to set the working point to the right extreme of the constant regions, as it yields the same QoS with longer lifetime. Also, it can be noticed that the higher $\overline{\mathrm{Pr}}_{\mathrm{tx}}$, the higher the distortion (i.e., the worse the performance); this happens because larger transmission probabilities impose to transmit more often even when the channel is in bad conditions (which, in turn, does not allow the transmission of many bits).

As shown in Fig. 3.4, when $\overline{\mathrm{Pr}}_{\mathrm{tx}}$ is low and N is small, the optimal and suboptimal approaches almost coincide, since a transmission is performed only when the channel gain is very high. In this case, $\sum_i \tau_i^{(k)} \ll T^{(k)}$ and it is possible to serve all users even with a suboptimal approach. However, when $\overline{\mathrm{Pr}}_{\mathrm{tx}}$ increases, transmissions are more frequent, whereas when N is larger the sum $\sum_i \tau_i^{(k)}$ may saturate to $T^{(k)}$. In these cases the benefits of adopting the optimal scheme rather than the suboptimal policies are significant. The fixed-TDMA scheme may even be infeasible (e.g., when $N = 15$ and $\overline{\mathrm{Pr}}_{\mathrm{tx}} = 0.6$) because, by maintaining the slots fixed, some users may not be able to transmit in any frame as they would need more time to meet the QoS and power constraints.

The maximum lifetime is reached when the problem is close to becoming infeasible, i.e., any further increase of the lifetime would yield a violation of some of the constraints of FOP, irrespective of the available energy. Even in this limiting conditions, the normalized distortion does not always reach 1 (i.e., the distortion may be strictly lower than its maximum acceptable threshold). This happens because, when solving the optimization

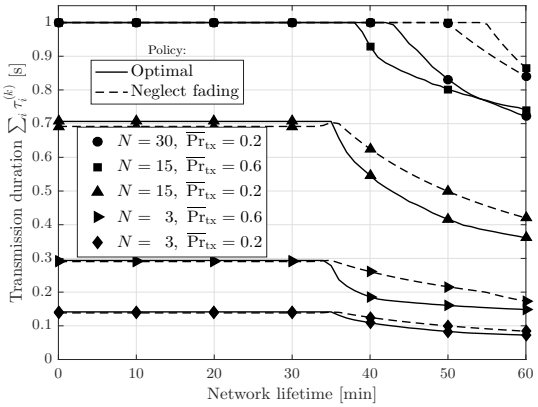


Figure 3.4: Transmission durations vs. lifetime n with fading when $B_{0,i} = 240$ J.

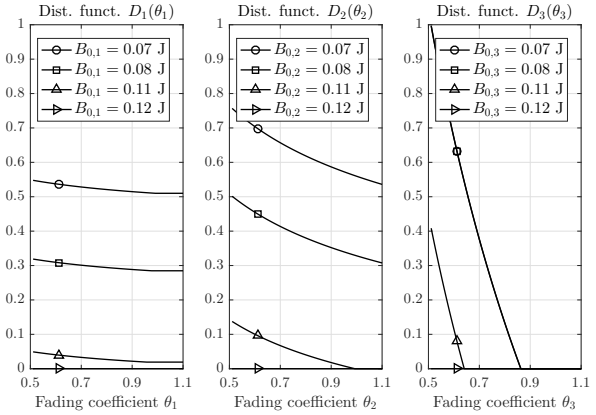


Figure 3.5: Optimal distortion functions $\delta_i(\cdot)$ vs. θ_i for $i = 1, 2, 3$ in a single frame ($\overline{\Pr}_{\text{tx},i} = 0.6$).

problem, the normalized distortion is required not to exceed 1 even when the channel conditions are unfavorable (but still acceptable), i.e., when the fading gain $\theta_i^{(k)}$ gets close to the minimum threshold $\theta_{\text{tx},i}^{(k)}$. Since with better channel conditions the distortion is lower, on average we get $\overline{D}_i^{(k)} < 1$.

Transmission duration. The overall transmission durations are represented in Fig. 3.4. As previously explained, when N is large (e.g., when $N = 30$ and $\overline{\Pr}_{\text{tx}} = 0.2$), or if larger transmission probabilities are required (e.g., when $N = 15$ and $\overline{\Pr}_{\text{tx}} = 0.6$), then $\sum_i \tau_i^{(k)}$ reaches $T^{(k)} = 1$ s. In all the other cases, every node can be considered independently of the others, since they always satisfy the time constraint. Note that $\sum_i \tau_i^{(k)}$ decreases with the lifetime, since less energy is available for transmission in every frame.

Distortion function. For every channel realization, $D(\theta_i)$ is evaluated using the packet length and the transmission power computed as described in Sec. 3.5.2. Its trend can be seen in Fig. 3.5 for the case $N = 3$. As expected, the distortion decreases as θ_i increases, since this corresponds to better channel conditions. However, the distortion does not reach 0 in all cases, and may instead saturate to a constant value. This happens because the energy constraints (given by the initial battery levels) are tight and packets need to be heavily compress to save energy. In this case, it is not possible to exploit the full channel capacity and the amount of data sent in the constant regions is given by Eq. (3.18).

Different packet sizes. In Fig. 3.6, the original packet size $L_{0,i}^{(k)}$ is modified according to a fixed pattern. Let L_{0,G_j} be the original packet length previously introduced for group $j = 1, 2, 3$. Then, in this example $L_{0,i}^{(k)} = \zeta^{(k)} L_{0,\text{G}_j}$, with $i \in \text{G}_j$ and a coefficient $\zeta^{(k)}$ that periodically increases and then decreases with k; in particular, $\zeta^{(k)}$ evolves as $\frac{1}{2}$, 1, 2, 1, $\frac{1}{2}$, 1, 2, etc. (e.g., this models sensors that track a periodic phenomenon).

In Fig. 3.6a, the initial energy levels of all nodes in G_2 and G_3 are infinite are assumed to be infinite, whereas the nodes in G_1 have a limited reserve of energy (similarly, in Fig. s 3.6b and 3.6c, group G_2 or G_3 is energy constrained, respectively). It is interesting to

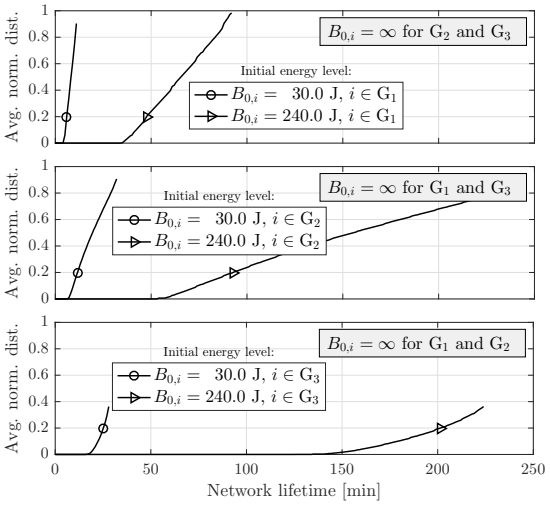


Figure 3.6: Average normalized distortion vs. lifetime with fading for $N = 3$ when $L_{0,i}^{(k)}$ evolves over time.

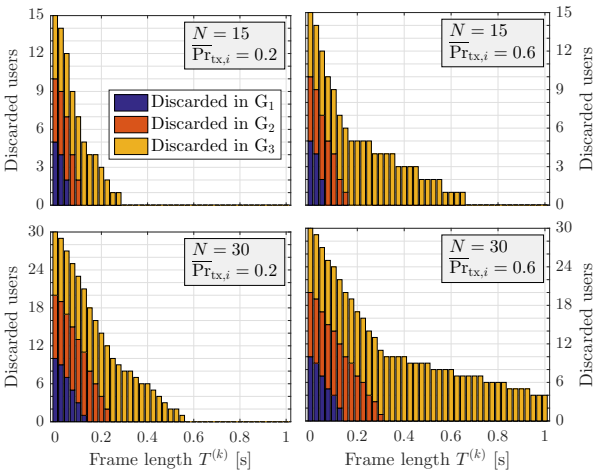


Figure 3.7: Number of discarded users as a function of the frame length.

note that, in order to guarantee fairness, even if most of the nodes have infinite resources, the network distortion may be greater than zero. Also, the lifetime strongly depends on which group has limited resources: G_1 transmits lots of data, hence the system quickly becomes infeasible when its initial energy level is low; instead, nodes in G_3 do not transmit large packets, and thus they consume less energy over time and the lifetime is much longer. Note that EAP plays a fundamental role here, since it assigns different amounts of energy according to the packet lengths $L_{0,i}^{(k)}$.

Dismission policy. So far, the dismission policy described in Sec. 3.6 has been neglected. However, when the problem is infeasible, it is possible to discard some users and allow the others to transmit. In Fig. 3.7, the priority of the users is given by the group they belong to ($G_1 \succ G_2 \succ G_3$). As can be seen, when $T^{(k)}$ is larger, fewer users are dismissed, because it is easier to make all users fit the frame duration (see (3.8e)). Note that in the case $N = 30$ and $\overline{\mathrm{Pr}}_{tx} = 0.6$, discarding some users is necessary even if $T^{(k)} = 1$ s. This explains why Fig. 3.3 this setup was not shown, since the dismission policy was not applied.

3.9 Lesson learned

IoT networks may include devices with different capabilities and requirements. Simple channel access schemes that do not take heterogeneity into account may lead to significant inefficiencies: the standard TDMA approach, e.g, can be impractical in some cases. The policies proposed in this study, instead, allocate channel resources differently to the various devices and perform power control. The thorough numerical evaluation based on the characteristics of realistic devices validates the analytical results and shows that the

proposed policies outperform simpler schemes. This study also shows the improvements that can be obtained when data compression is dynamically adapted to the transmission operations, so as to guarantee the desired level of QoS.

The proposed policies may serve as guidelines to develop and gauge more lightweight heuristics, which may, e.g., take into account also the latency in the coordination and control messages with the BS. Moreover, it would be interesting to analyze the effect of packet losses on the network performance, since they may have a strong impact on both the energy and the distortion metrics, and likely require a retransmission mechanism.

4

An energy-aware data processing and transmission strategy

This study considers a monitoring application where sensors periodically report their readings to a common receiver in a single-hop access network, similarly to the scenario of Ch. 3. Given the predictability of the reporting patterns, the devices access the channel according to a TDMA scheme, which, unlike the scheme of Ch. 3, assigns constant time resources to the various users. The ultimate goal is to combine an efficient energy utilization with QoS requirements, in terms of quality of the reported data, which is impacted by lossy compression and channel impairments.

Sensors are battery-powered and can harvest energy from the environment. To maximize the quality of the reported data, the packets transmitted contain newly generated data blocks together with up to $r-1$ previously unsuccessfully delivered ones, where r is a design parameter. These data blocks are compressed, concatenated and encoded with a channel code. The scheme applies lossy compression, such that the fidelity of the individual blocks is traded off with the reliability provided by the channel code. In fact, source compression affects data quality, but on the other hand reduces the volume of information bits to send over the channel, allowing the use of more redundancy to combat the channel impairments.

The selection of the optimal operating strategy in terms of the tradeoff between selection of compression rate and channel coding rate is constrained by the energy availability and statistics of the EH process, and, at the same time, driven by the target minimization of the distortion of the reported data.

The problem is formulated by means of an MDP, and the optimal policy is derived with a variant of the Value Iteration algorithm.

MDPs are often employed to derive energy management policies, as they represent an appealing solution to optimize some long-term utilities in the presence of stochastic EH [66]. Investigating the effects of packet losses on data distortion is not a new topic. However, many works limit their studies to Gaussian data sources or neglect the energy

limitations, see [67, 68]. A common approach is to use the distortion exponent as the performance metric [69, 70], but this is meaningful only for the high SNR regime, which is not the case for IoT networks. Another beaten path is that of layered transmission schemes, where the source is coded in superimposed layers, like in [71]. Each layer successively refines the data description and is transmitted with a larger coding rate, thus the transmission is less robust to failures. This practice is often used in multimedia applications [72], but is not very meaningful in other contexts.

Several works focus on the minimization of data distortion in the presence of energy limitations and/or EH, but they often neglect the effect of packet losses [73], or consider Gaussian and binary sources [74]. Two works that bear similarities with this study are [57] and [75]. In [57], the goal is to maximize the long-term average quality of the transmitted packets by adapting the degree of lossy compression and the transmission power. Power control is used to maintain the packet error probability below a chosen threshold, depending on the state of the channel. Like in this study, the optimal transmission strategy is determined through an MDP. In [75], the reconstruction of time-correlated sources in a point-to-point communication is formulated as a convex optimization problem and solved with an iterative algorithm. The node has to decide on the transmission power and rate and is subject to EH constraints.

It is instead difficult to find works that deal with retransmissions in conjunction with data processing outside of the literature on multimedia networks. Several studies investigate the performance of cooperative retransmissions, where devices collaborate with each other [76, 77], but the focus of the study proposed in this thesis is in scenarios where the devices are unaware of the presence of the others and of the resources they have available, and communicate solely with the gateway. Retransmissions in IoT constrained networks are considered, e.g., in [78], which addresses the problem of transmit power control in the presence of EH when a sensor node makes use of an Automatic Repeat Request (ARQ) protocol and retransmits the lost packets. Differently from this study, it is assumed that some CSI is available at the nodes, and no data processing is considered. In [79], sensor nodes implement an energy-aware hop-by-hop retransmission mechanism where only packets carrying critical information are ensured to be retransmitted. This differs from the scheme proposed here, where packets are not differentiated according to priorities, but can be compressed at the source.

This study has been presented in [80] and [64].

4.1 System model

The scenario considers a single-hop, star topology network, comprising a multitude of IoT devices that periodically monitor some phenomena of interest and report data to

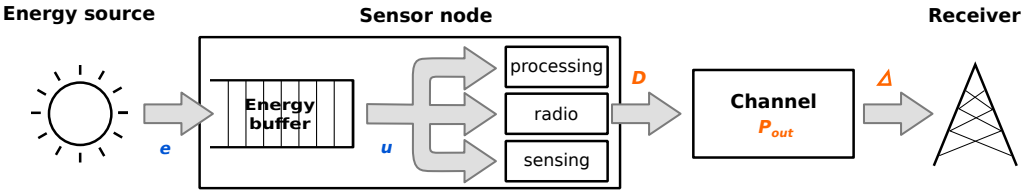


Figure 4.1: Block diagram of a device. The blue labels indicate the energy flows, while the orange labels are related to the data reporting.

a common receiver. This data collector is assumed to be connected to the energy grid, whereas the sensor nodes are battery-powered, but also endowed with EH capabilities.

The predictability of the traffic pattern makes TDMA a good choice for transmission scheduling among the devices. Since in this case the devices do not interfere with each other, it is possible to focus just on a single device, with the aim to find its optimal transmission strategy. The corresponding model is represented in Fig. 4.1. A device is endowed with a circuitry to scavenge energy from the environment; this energy is stored in a buffer of finite size and used by the node to generate, process, and send data to the common receiver through a Rayleigh-fading channel.

4.1.1 *Compression at the source*

A device is capable of compressing in a lossy fashion the time series generated through sensing the environment, as described next. Between two consecutive reporting events, a sensor node collects a block of readings[1], where each block has a constant size of L_0 bits and is independent of the previous ones. The device compresses the generated block by selecting a compression level k, where $k \in \{0, \dots, m\}$, and produces a compressed block of size L bits. When $k = 0$, the size of the compressed block is $L = 0$, i.e., no packet is transmitted, and when $k = m$, the size of the compressed block is $L = L_0$, i.e., no compression takes place. The *compression ratio* is defined as the ratio between the size of the compressed block and that of the original one: $L/L_0 = k/m \in [0, 1]$. Lossy compression makes it possible to trade some accuracy in the data representation using a lower compression ratio for additional error-correction redundancy, and consequently have an increased robustness to channel impairments, as will be explained in Sec. 4.1.2.

The compression ratio-distortion curve is signal- and algorithm-dependent. The distortion function used in this study is the same as that of Ch. 3 [10]:

$$D(k) = \begin{cases} b\left(\left(\frac{k}{m}\right)^{-a} - 1\right) & \text{if } k \geq 1 \\ D_{full} & \text{if } k = 0 \end{cases} \tag{4.1}$$

1 The blocks of readings will be referred to as data blocks, to distinguish them from the packets sent after processing.

where $b > 0$, $0 < a < 1$. The choice $k = 0$ entails that the packet is discarded (e.g., due to energy restrictions) and the corresponding distortion is equal to the maximum value $D_{full} \triangleq 1$. Eq. (4.1) shows that the distortion is a convex and non-increasing function of the compression ratio k/m. [2]

4.1.2 *Data transmission*

Each IoT node transmits during its dedicated time slot, whose duration T defines the maximum number of bits that can be sent $S = T/T_b$, with T_b being the (fixed) bit duration. It is reasonable to assume that $L_0 \leq S$, i.e., a device may avoid compressing a packet. After compression, there are $L \leq L_0$ information bits and the corresponding coding rate is $R = L/S$. Depending on R and the actual channel conditions, the packet may not be correctly received with an outage probability $P_{out}(R)$.

An acknowledgment mechanism ensures that the receiver sends feedback to the transmitter; however, no channel state estimation is performed, so the transmission power P_{tx} is kept constant. The communication channel is affected by block Rayleigh fading; when the channel is in a deep fade the packet is lost and an outage occurs.

The packets sent by IoT devices are likely to be short, and thus it is necessary to use the recent results of finite-length information theory that adapt the classical concepts of channel capacity to the case of short data packets [82]. In particular, the results of [83] justify the use of the quantity $\log_2(1+\gamma)$ to well approximate the maximum rate even in the finite-length regime, where γ is the SNR at the receiver, given by:

$$\gamma = \frac{|H|^2 P_{tx}}{N} \triangleq |\tilde{H}|^2 \bar{\gamma}. \tag{4.2}$$

H is the channel gain coefficient that represents fading (assumed to be constant over the packet duration in the quasi-static scenario) and path loss, which depends on the distance d from the receiver, P_{tx} is the transmission power, and N the noise power. The SNR can be decomposed into the expected SNR at the receiver $\bar{\gamma}$ (which includes the path loss, the transmission power and the noise power), and the random fading coefficient $\tilde{H}$. Thus, the outage probability is here modeled as:

$$P_{out}(R) = Pr\left(\log_2(1+\gamma) < R\right). \tag{4.3}$$

2 The monotonicity property does not hold in general, because it may be $D(1) > D(0)$. However, by using the parameters a and b estimated from the results of [81], the function is monotonic even for very fine granularity (values of m larger than 100); this property is always ensured in the numerical evaluation.

Since Rayleigh fading is considered, $\tilde{H}$ follows a complex Gaussian distribution with zero mean and unit variance. In this case, the outage probability becomes:

$$\mathrm{P}_{out}(R) = 1 - e^{-(2^R-1)/\bar{\gamma}}, \tag{4.4}$$

which is non-decreasing in R (and thus in k, since $L = k/m\, L_0$), and initially convex and then concave. Clearly, the farther the device from the receiver, the larger the outage probability, since $\bar{\gamma}$ decreases with distance. If a packet gets lost, the block of readings contained in it can be compressed and transmitted again along with the subsequently generated blocks, as described in Sec. 4.2. After r failed transmission attempts, the block is considered outdated and discarded.

4.1.3 *Energy dynamics*

In the considered scenario, the data collector does not have energy constraints, while the reporting devices are powered by finite batteries and have EH capabilities. The energy model has been discussed in Ch. 2. The expressions used in this study are reported here for clarity.

Data processing. The energy consumed by processing algorithms is modeled as in (2.2) for the case of the LTC algorithm (see (2.3)). The compression ratio η_{p} is given by k/m. Thus, the energy consumption can be expressed as

$$E_p(k) = \begin{cases} E_0 L_0(\alpha_p \frac{k}{m} + \beta_p) & \text{if } 1 \leq k \leq m-1 \\ 0 & \text{if } k = 0, m\,. \end{cases} \tag{4.5}$$

Notice that, if the packet is not compressed ($k = m$) or is discarded ($k = 0$), no energy is consumed.

The contribution of channel encoding is not considered, because it is typically negligible [50].

Transmission. As given in (2.4), the energy cost of a wireless transmission with power P_{tx} for a period of length T is modeled:

$$E_{\mathrm{tx}} = \frac{T\, P_{\mathrm{tx}}}{\eta_{\mathrm{A}}}, \tag{4.6}$$

with $\eta_{\mathrm{A}} \in (0, 1]$ representing the efficiency of the antenna's power amplifier.

Sensing and circuitry. The energy consumption due to sensing operations and circuitry is modeled as:

$$E_c(k) = \beta_s + \beta_c + \mathcal{E}_c\, T \cdot \chi_{\{k>0\}}, \tag{4.7}$$

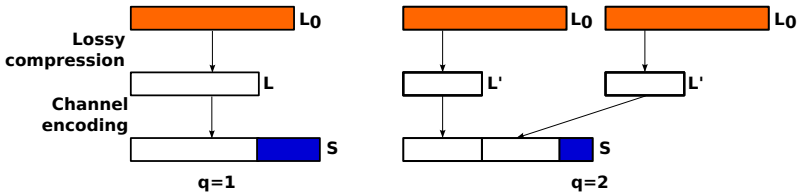


Figure 4.2: The retransmission mechanism: encoding procedure.

where $\chi_{\{k>0\}}$ is the indicator function equal to 1 if $k>0$ and to zero otherwise (recall that the packet is dropped if $k=0$). β_s is the energy drained by sensing in the window between two transmission slots (see (2.1), here the number of sensing operations is constant), while β_c includes the energy required for switching between idle and active mode and maintaining synchronization with the receiver.

Energy harvesting and battery dynamics. The sensor nodes are not connected to the energy grid, but are provided with a battery and some energy-scavenging circuitry and can collect energy from the environment. The energy supply is supposed to be time-correlated (e.g., solar power). In this case, as discussed in Ch. 2, the dynamics of the energy source can be tracked through an X-state MC [52, 51]: the source is in state $x \in \mathcal{X} = \{0, \dots, X-1\}$ and scavenges $e \in \{0, \dots, E\}$ quanta of energy from the environment, according to some p.m.f..

Let B be the finite size of a node's battery, and u the energy used in the current slot, which depends on processing, transmission and circuitry, as given in Eqs. (4.5), (4.6) and (4.7). If b and b' respectively represent the current and next battery level, the temporal evolution of the battery can be modeled as:

$$b' = \min\{b+e-u, B\} \triangleq (b+e-u)^{+}, \tag{4.8}$$

with $u \leq b$, as imposed by the energy causality principle. Notice that this model assumes a *harvest-store-use* protocol, according to which the energy e scavenged in slot n is not immediately available, but can only be used from slot $n+1$ onwards. Since we consider a finite size battery, an overly aggressive or conservative energy management could either deplete the battery ($b=0$) or fail to use the excess energy and waste it ($b=B$). These situations need to be prevented by designing a scheme that dynamically adapts to the randomness of the energy inflow, so as to ensure acceptable performance on a long-term horizon.

4.2 Energy-aware (re)transmission scheme

Here, the joint source-channel coding scheme described in Sec. 4.1 is extended to account for packet retransmissions.

The acknowledgment feedback allows the transmitter (i.e., the reporting device) to know whether or not a packet has been correctly received at its intended destination. When an outage occurs, the packet is lost and needs to be retransmitted. However, if a simple ARQ protocol is employed, the delay for sending newly generated data blocks will increase, depending on the number of the devices participating in the TDMA scheduling as well as on the number of allowed retransmissions, and can potentially become unbounded. Consequently, it may happen that the newly generated data blocks will become outdated, without having a chance to be transmitted. To prevent this, nodes employ a retransmission scheme in which they send their new data blocks together with the previously lost data blocks (if any) in the same time slot, as explained further.

Suppose that the transmission of a data block fails; in the next time slot, the device will try to send the new block together with the previously lost one. Thus, the device has to compress and transmit two blocks of original size L_0 each within the S channel uses available. If this transmission also fails, in the next slot the sensor node will process and transmit the new block and the two that were previously lost. In general, if the last q transmissions failed, the sensor node has to process $q+1$ blocks of L_0 bits each and send them together. A maximum number r of transmission attempts can be made for each piece of data, where the value of r is dictated by the application, e.g., because of latency or QoS considerations.

If multiple data blocks are sent in the same time slot, the information they contain is not fused or processed jointly, but the compression is done separately for each of them[3]. For the sake of fairness, all data blocks are treated in the same way, i.e., the same compression ratios are applied. This yields q compressed blocks of size L, where $q = r+1$, which are then joined in a single block of size qL, which is then encoded, producing a packet of size S, and transmitted. Thus, the same distortion is introduced at the source for all data blocks placed in the same packet, but then they are treated as a single entity which is sent over the channel and subject to a certain outage probability that depends on the coding rate $R = qL/S$. Fig. 4.2 shows the encoding mechanism when a single block (left) or two blocks (right) are transmitted in the same slot.

Fig. 4.3 shows the evolution of the data queue, which behaves like a success-runs MC. If the data queue in a certain slot has size q and the transmitted packet is lost, the queue state becomes $q' = \min\{q+1, r\}$, otherwise the new queue state is $q' = 1$. We denote as $\Delta_q(k)$ the distortion at the receiver when the queue state is q and the chosen compression ratio is k. Such distortion depends on the reception outcome: in case of failure (NACK),

3 Even in the case of statistically independent data assumed in this paper, the longer the compressed block, the better the compression ratio for the same average data fidelity. Thus, the separate compression approach is possibly suboptimal. However, processing the data blocks together requires to build an aggregate distortion function, which is beyond the scope of this work.

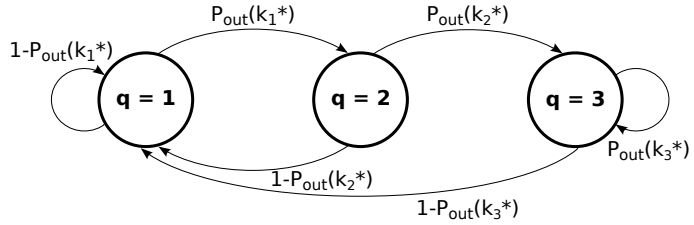


Figure 4.3: Structure of the MC that models the dynamics of the backlog state q for $r = 3$.

the distortion at the receiver is set to $\Delta_q(k) = D_{full}$, even though the data blocks could be retransmitted; instead in case of successful transmission (ACK), it is necessary to sum all the distortions of the q blocks that have been compressed with the same ratio k/m, but also to subtract all the penalties that were obtained for the previous packet losses that were accounted for, since the blocks were eventually successful. Thus, in this latter case, it is $\Delta_q(k) = D_{ack}(q,k)$ with:

$$D_{ack}(q,k) = q\,D(k) - (q-1)D_{full}. \tag{4.9}$$

Let $k_q^\star$ be the optimal value of k (i.e., the one that determines the optimal source-channel coding scheme to use) when q packets share the same time slot; Sec. 4.4.1 explains how to determine it. smit all q data blocks. However, the energy required by transmission and circuitry does not depend on q, because the duration of the transmission is always T, i.e., S bits, see Eqs. (4.6) and (4.7). Hence, the energy consumed by the node when there are q data blocks in the queue and it uses a compression ratio equal to k/m is:

$$q\,E_p(k) + E_{\text{tx}} \cdot \chi_{\{k>0\}} + E_c(k). \tag{4.10}$$

The optimization described in Sec. 4.4.1 determines the optimal source coding scheme, i.e., the value $k_q^\star$, and the corresponding energy consumption is obtained by substituting $k = k_q^\star$ in Eq. (4.10).

4.3 Problem formulation

This sections mathematically defines the objective of the optimization problem and describes the structure of the MDP, while the solution technique is explained in Sec. 4.4.

4.3.1 *The optimization objective*

The objective of this study is to maintain for each node an energy-neutral operation mode while minimizing the long-term average distortion at the receiver, which depends on the outcomes of the transmissions, as explained in the previous section. Considering

q data blocks of size L_0 that are compressed with the same compression ratio k/m and encoded jointly at rate $R = qL(k)/S$, the expected distortion at the receiver is:

$$\mathbb{E}[\Delta_q(k)] = qD(k)\left(1 - \mathrm{P}_{out}\left(q\frac{L(k)}{S}\right)\right) + qD_{full}\,\mathrm{P}_{out}\left(q\frac{L(k)}{S}\right), \tag{4.11}$$

where $\mathrm{P}_{out}(\cdot)$ is the outage probability as given in Eq. (4.3), and $qD(k)$ and qD_{full} are the distortions obtained when the packet is acknowledged or lost, respectively.

In other words, if the packet is successfully received, its distortion corresponds to that introduced at the source for all initial q blocks of measurements, otherwise the maximum distortion level is considered for all packets, as if they had not even been sent. We are interested in the expected distortion at the receiver, thus the two cases need to be weighed with their probabilities. Notice that the distortion at the source $D(k)$ decreases as k increases, whereas the outage probability decreases for smaller coding ratios, i.e., as k decreases. This implies a tradeoff between the distortion introduced by the lossy compression and the probability that the transmitted packet will be successfully received, through the choice of the value of k.

To minimize $\mathbb{E}[\Delta_q(k)]$ and guarantee self-sufficiency of the network, it is necessary to (i) decide on k, and (ii) in each slot, allocate the energy consumption based on the current battery level, the dynamics of the energy source, and the energy consumption profile, in such a way to prevent energy outages (that disrupt the communication) and battery overflows (that waste energy). The problem can be formulated by means of an MDP, where the actions correspond to the energy to use, while the costs are represented by the expected distortion at the receiver. By doing so, the energy self-sufficiency of the node is ensured and the QoS is optimized.

4.3.2 *The Markov Decision Process*

The MDP is defined by the tuple $\left(\mathcal{S}, \mathcal{U}, P, c(\cdot)\right)$, where $\mathcal{S}$ denotes the system state space, $\mathcal{U}$ is the action set space, P is the set of transition probabilities of the system state space, and $c(\cdot)$ is the associated cost function for taking an action.

System state space $\mathcal{S} \triangleq \mathcal{X} \times \mathcal{B} \times \mathcal{Q}$, where $\mathcal{X} = \{0, 1\}$ represents the set of energy source states, $\mathcal{B} = \{0, \ldots, B\}$ the set of energy buffer states, and $\mathcal{Q} = \{1, \ldots, r\}$ the set of backlog states, i.e., the number of waiting packets. Notice that it is necessary to keep track of the data queue size.

Action set space $\mathcal{U} \triangleq \{0, \ldots, B\}$. In each slot, the device observes the current system state $s \in \mathcal{S}$ and decides how much energy $u \in \mathcal{U}_s \subseteq \mathcal{U}$ to use to process and transmit the data it collected. In accordance with the energy causality principle, this quantity cannot exceed the battery level, i.e., $\mathcal{U}_s = \{0, \ldots, b\}$.

Transition probabilities P govern the system dynamics. The probability of going from state $s = (x,\, b,\, q)$ to state $s' = (x',\, b',\, q')$ with action u is:

$$\Pr(s'|s,u) = p_x(x'|x) \cdot p_e(e|x) \cdot p_q(q'|q,u \cdot \delta\left(b' - (b+e-u)^{\dagger}\right) \tag{4.12}$$

where $p_x(x'|x)$ is obtained from the transition probability matrix of the MC that models the source state, $p_e(e|x)$ is the mass distribution function of the energy inflow in state x (see Sec. 4.1.3), and $p_q(q'|q,u)$ represents the probability that the backlog size goes from q to q' when action u is taken. The last term $\delta(\cdot)$ is equal to 1 if its argument is zero, and zero otherwise, and ensures that the transitions between states are consistent with the dynamics of the battery level, see Eq. (4.8).

Cost function $c(\cdot)$. When the sensor node is in state $s = (x, b, q)$ and selects action u, it implicitly decides upon the source-channel coding scheme that minimizes $\mathbb{E}[\Delta_q(k)]$, i.e., it decides the optimal number $k_q^\star$ of information bits to send over the channel per data block, given the available energy u. The transition from state s to state s' when action u is taken entails a cost:

$$c(s,u,s') = \begin{cases} D_{\text{ack}}(q,k_q^\star) & \text{if } q' = 1 \\ D_{\text{full}} & \text{otherwise} \end{cases} \tag{4.13}$$

and therefore the cost of choosing action u in state s is:

$$\tilde{c}(s,u) = \sum_{s' \in \mathcal{S}} \Pr(s'|s,u)\, c(s,u,s'). \tag{4.14}$$

Notice that the cost only depends on the data queue component q of the state, while the battery level b affects the set of admissible actions.

When retransmissions are not allowed ($r = 1$), the backlog state is always $q = q' = 1$, whatever the outcome of the transmission. In this case, the MC of Fig. 4.3 reduces to a single state, and Eq. (4.13) is no longer meaningful, thus when $r = 1$ we set $c(s,u,s') \equiv \tilde{c}(s,u) \equiv \mathbb{E}[\Delta_1(k_1^\star)]$.

4.4 Optimal policy

The optimal policy is the one that, based on the statistics of the EH process and the current battery level, decides how much energy to use in order to guarantee the lowest average distortion at the receiver. This section first discusses how to determine the optimal source-channel coding scheme, hence $k_q^\star$ for each possible backlog state $q \leq r$, and then describes how to solve the MDP optimally.

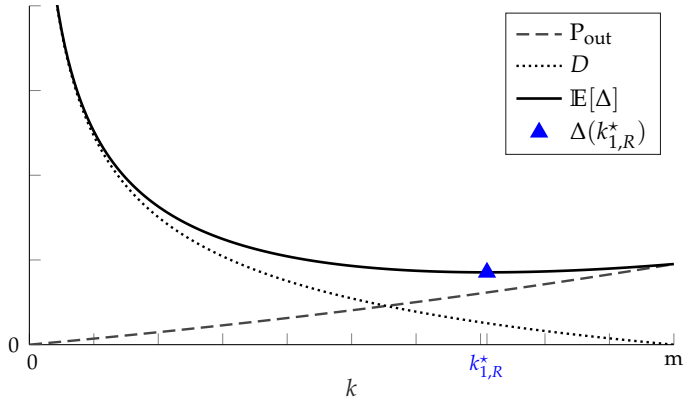


Figure 4.4: Example of location of $k_R^\star$, which is the point of minimum of Eq. (4.11), for $q = 1$.

4.4.1 *Rate-distortion tradeoff*

For the case $q = 1$, the expected distortion at the receiver given in Eq. (4.11) exhibits a unique point of minimum $k_{1,R}^\star$ when $D(k)$ and $P_{out}(k)$ are characterized as in Eqs. (4.1) and (4.4), respectively (see [80] for the proof). An example of this is shown in Fig. 4.4. This is still valid when retransmissions are introduced, as can be inferred by looking at Eq. (4.11). The q packets are compressed separately but with the same compression ratio, which cannot exceed m/q, hence $D(k)$ is truncated for $q > 1$. Instead, the outage probability maintains the same shape since the packets are encoded together. Hence, there exists an optimal point $k_{q,R}^\star$, whose value depends on the number of packets q that are sent in the same time slot and that minimizes the expected distortion at the receiver. If the device uses $k < k_{q,R}^\star$, the packet will go through the channel with a higher probability but its distortion will be larger; on the other hand, if $k > k_{q,R}^\star$, the distortion will be smaller, but it is more likely that the packet will be lost.

If the amount of energy allocated u allows it, the device will choose the optimal coding scheme corresponding to $k_{q,R}^\star$, otherwise it simply selects the maximum possible k dictated by the energy constraint, because $\mathbb{E}[\Delta_q(k)]$ is decreasing if $k \leq k_{q,R}^\star$, which is due to how $k_{q,R}^\star$ is defined. The energy consumption (see Eq. (4.10)) is non-increasing in k. Let $k_{q,E}^\star(u)$ be the largest value of k that solves $q\,E_p(k) + E_{\text{tx}} \cdot \chi_{\{k>0\}} + E_c(k) \leq u$ for a given q; then, when the backlog state is q, the device will choose the source-channel coding scheme corresponding to:

$$k_q^\star = \min\{k_{q,R}^\star, k_{q,E}^\star(u)\}. \tag{4.15}$$

Such value clearly depends on the energy u that the node decides to employ, but this is omitted in favor of a lighter notation. As proved in [64], the expected distortion at the receiver is convex for $k \leq k_{q,R}^\star$ and, thus, is a convex non-increasing function of u.

The choice of how much energy to use when in state $s = (x, b, q)$ uniquely determines the joint source-channel coding scheme (i.e., the number of transmitted information bits per data block) that leads to the smallest expected distortion at the receiver [64].

4.4.2 *Solving the MDP*

When the sensor device is in a certain state, it selects the action to take according to a policy $\pi : \mathcal{S} \to \mathcal{U}$. The corresponding long-term average cost is:

$$J^{\pi}(s) = \lim_{M \to +\infty} \frac{1}{M} \mathbb{E}_s \left[\sum_{m=0}^{M-1} \tilde{c}(s_m, u_m) \middle| s_0 = s \right], \tag{4.16}$$

where the initial state s_0 is given. Notice that each decision affects all subsequent decisions.

The goal is determining the optimal policy $\pi^{\star}$, i.e., the set of rules that map each system state into the optimal action with respect to the average cost criterion. The MDP defined in Sec. 4.3.2 has unichain structure and bounded costs, implying that Eq. (4.16) does not depend on the initial state:

$$J^{\pi}(s) \equiv J^{\pi}, \quad \forall s \in \mathcal{S}. \tag{4.17}$$

Hence, the search can be restricted to Markov policies only [84].

$\pi^{\star}$ can be determined using the Relative Value Iteration Algorithm (RVIA), which is a variant of the well-known Value Iteration algorithm for average long-term problems and provably converges [85]. To understand why it works, we first define the n-step value-function by induction as:

$$v_n(s) = \min_{u \in \mathcal{U}_s} \left\{ \tilde{c}(s, u) + \sum_{s' \in \mathcal{S}} \Pr(s'|s, u) v_{n-1}(s') \right\}, \tag{4.18}$$

where $v_0(s)$ is arbitrarily defined, e.g., $v_0(\cdot) = 0$. Function $v_n(s)$ represents the minimum expected n-step cost that can be achieved from an initial state s, because it sums the immediate cost $\tilde{c}(s, u)$ obtained in the initial state with the expected optimal cost obtained in the $n-1$ subsequent slots (through $v_{n-1}(\cdot)$ and then recursively). Based on this, the optimal policy $\pi^{\star}$ is such that $J^{\pi^{\star}}(s) = \lim_{n \to \infty} v_n(s)/n$, and as n grows the dependence on s fades ($J^{\pi^{\star}}(s) \equiv J^{\pi^{\star}}$).

RVIA leverages on this and defines two functions J and Q that are alternatively updated starting from an initial estimate $J_0(\cdot)$ until convergence:

$$Q_j(s,u) = \tilde{c}(s,u) + \sum_{s' \in \mathcal{S}} Pr(s'|s,u)\, J_{j-1}(s') \tag{4.19}$$

$$J_j(s) = \min_{u \in \mathcal{U}_s} Q_j(s,u), \tag{4.20}$$

with j being the iteration index. The convergence criterion is chosen as the span seminorm operator $sp(w) \triangleq \max(w) - \min(w)$, because it guarantees that (4.20) is a contraction mapping [85]. RVIA is stopped at iteration l, when $sp(J_{l+1}(s) - J_l(s)) \leq t$ for a chosen threshold t. The optimal policy dictates the action $u^\star(s)$ to take in each state s:

$$u^\star(s) = \underset{u \in \mathcal{U}_s}{\operatorname{argmin}} Q_l(s,u), \tag{4.21}$$

and entails the following average long-term cost:

$$C^\star = \sum_{s \in \mathcal{S}} \rho_s\, \tilde{c}(s, u^\star(s)) \tag{4.22}$$

where ρ_s is the steady state probability of state s induced by the optimal policy (the MDP reduces to a MC when the actions to take in each state are deterministic).

4.5 The effects of retransmissions

This section investigates the benefits of introducing retransmissions in the communication framework and how they are related to the value of r. We first analyze the behavior of the retransmission scheme when the energy constraints are neglected, then discuss what happens when these constraints are introduced, and, finally, explain how to choose r.

4.5.1 *Benefits of retransmissions (without energy constraints)*

When a transmission slot is used for multiple packets combined together, these packets will be more compressed and have a larger distortion than when a slot is reserved to a single packet. However, the convexity of $D(\cdot)$ suggests that combining packets improves the average long-term distortion. This is formalized in the following result, whose proof is reported in [64].

Theorem 1. *For any value r of the maximum number of transmission attempts that can be dedicated to a packet such that $k^\star_{r,R} > 0$, when the energy constraints are neglected, i.e., $k^\star_r = k^\star_{r,R}$ and $k^\star_1 = k^\star_{1,R}$, the retransmission mechanism described in Sec. 4.2 achieves an average long-term distortion that is no higher than that of the base case where packets can be sent only once.*

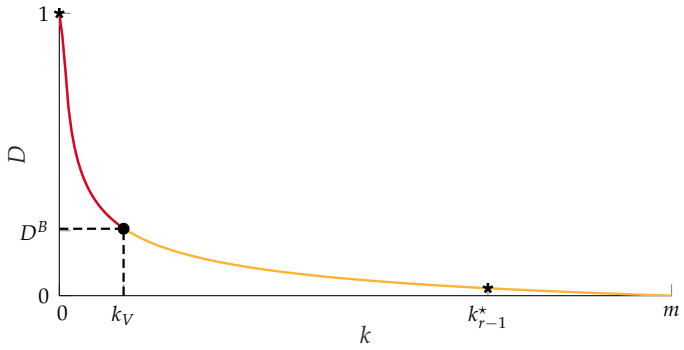


Figure 4.5: Determination of k_V, see Eq. (4.25). The red part of the curve represents distortion levels that are larger than D_B, whereas the orange part is related to distortion levels that are smaller. If $k_r^\star$ falls to the left of k_V, then choosing $r-1$ instead of r improves the performance.

Notice that, if $k_r^\star = 0$, there would clearly be no gain in using the retransmission scheme, as packets are simply discarded; this is unlikely to happen, especially if $r < m$.

The enhancements that can be obtained with the retransmission scheme are not proportional to the value of r, but depend on the particular shapes of the distortion and outage probability functions, and on how large r is. It is very hard to quantify and compare the performance corresponding to different values of r, because there is no closed-form expression for $k^\star_{r,R}$. Nevertheless, some insights can be given. Assume $r > 2$ and compare the performance obtained by the retransmission schemes with r (case A) and with $r-1$ (case B) transmissions allowed when $k_r^\star = k^\star_{r,R}$ and $k^\star_{r-1} = k^\star_{r-1,R}$. Considering r consecutive transmissions, the only difference between the two cases is obtained when a successful transmission follows $r-1$ failed attempts. The corresponding distortions normalized to r are, respectively:

$$D^A = D(k_r^\star) \tag{4.23}$$

$$D^B = \frac{r-1}{r} D(k^\star_{r-1}) + \frac{1}{r} D_{full}. \tag{4.24}$$

Assuming $k_r^\star > 0$ and considering that $k_r^\star \leq k^\star_{r-1}$, there are two possible cases to analyze. If $k_r^\star = k^\star_{r-1}$, then $D^A < D^B$, and choosing r instead of $r-1$ leads to improved performance. This is more likely to happen when r gets closer to m, which represents the maximum value that k can take. If $r > m$, it surely is $k_q^\star = k^\star_{q-1}$ for some $q \leq r$. The other case to consider is $k_r^\star < k^\star_{r-1}$. Let k_V be the point such that $D(k_V) = D^B$, as represented in Fig. 4.5. By applying the definition of distortion given in Eq. (4.1), it is:

$$k_V = \left(\varphi (k^\star_{r-1})^{-a} + (1-\varphi) \frac{b + D_{full}}{b\, m^a} \right)^{-1/a}, \tag{4.25}$$

where $\varphi \triangleq (r-1)/r$. The distortion is a decreasing function, hence $D(k) > D(k_V) = D^B$ for all $k < k_V$. This implies that $D^A > D^B$ if $k_r^\star < k_V$, $D^A = D^B$ if $k_r^\star = k_V$, and $D^A < D^B$ if $k_r^\star > k_V$. Basically, if $k_r^\star < k_V$, it is better to choose $r-1$ rather than r. However, determining when this happens is challenging and hampered by the fact that there is no closed-form expression for the relationship between r and $k_r^\star$ (see Sec. 4.4.1 and [64]). Surely, as r increases, there is a higher probability that D^A is larger than D^B. In particular, let $\mathcal{K}^- \triangleq \{k \in \mathbb{N} : 0 < k < k_V\}$ and $\mathcal{K} \triangleq \{k \in \mathbb{N} : 0 < k < k_{r-1}^\star\}$. As r increases, $\varphi \to 1$, and k_V gets closer to $k_{r-1}^\star$ (from the left). Hence, the ratio $|\mathcal{K}^-|/|\mathcal{K}| \to 1$ and the probability that $k_r^\star$ falls in $\mathcal{K}^-$ rather than in $\mathcal{K} \setminus \mathcal{K}^-$ increases. In practice, for small values of r, in general it holds that $k_V < k_r^\star < k_{r-1}^\star$, and increasing r leads to lower distortion at the receiver. But, as r increases, k_V gets very close to $k_{r-1}^\star$ and it is more unlikely that $k_r^\star$ falls in between these two values, thus increasing r brings no benefit.

4.5.2 *When energy comes into play*

Theorem 1 proves that the combined retransmission scheme leads to enhanced performance in terms of QoS in the absence of energy constraints. However, this may no longer hold when these constraints come into play. The maximum compression ratio allowed by the allocated energy is $k_E^\star(u)/m$, which depends on the energy allocation u and on the backlog state q through Eq. (4.10). Unless $k_q^\star = 0$, the only contribution to the energy consumption that depends on q is the energy due to processing. Since $qE_P(k)$ increases with q, see Eq. (4.5), it is $k_{q,E}^\star(u) \leq k_{1,E}^\star(u)$, i.e., given the amount of energy available, fitting more packets into a single slot is more expensive than processing a single packet. The actual relation between $k_q^\star$ and $k_1^\star$ is not clear, because it highly depends on the energy the node allocates and therefore also on the energy arrivals statistics. If $k_{q,E}^\star(u) \geq k_{q,R}^\star$, then $k_q^\star = k_{q,R}^\star$ and the improvement stated by Theorem 1 holds for any $k_1^\star$. If instead $k_{q,E}^\star(u) < k_{q,R}^\star$, i.e., $k_q^\star = k_{q,E}^\star$, there is a simple relation between the average distortion obtained with and without the retransmission scheme and, according to the actual values of $k_q^\star$ and $k_1^\star$, it may be better to use the retransmission scheme or the single-transmission one.[4] Anyway, $k_{q,E}^\star(u)$ depends on the energy that the node decides to use in the considered time slot, and it is up to the MDP to manage it in such a way to obtain the lowest average distortion which, as shown in Theorem 1, is smaller when the retransmission scheme is adopted. In practice, the proposed retransmission mechanism coupled with an intelligent energy management scheme achieves a better QoS, i.e., a lower distortion.

4 If $k_q^\star \geq k_1^\star/q$, then the retransmission scheme leads to some improvement, otherwise nothing can be inferred, in general, because the performance depends on the particular shape of $\mathbb{E}[\Delta]$ in the two cases. See [64] for details.

It is worth noticing that, since the energy consumption increases with q and the battery has finite size, there exists a maximum value $r_{\max}$ of data blocks that can fit the same time slot. This means that, if $q > r_{\max}$, there is not enough energy to process and send all q data blocks, i.e., $k_{r,E}(B) = 0$, $\forall q > r_{\max}$. So, after $q-1$ consecutive transmission failures, the MDP will get stuck in a state $(\cdot, \cdot, q)$, where the only possible action is to choose $k_q^\star \equiv k_{q,E}^\star(\cdot) = 0$, which means that packets are always discarded. As a consequence, since there is a non zero probability of going to an absorbing state $(\cdot, \cdot, q)$, the average expected distortion will be 1. The value $r_{\max}$ can be easily determined as the largest value of q for which the energy consumption of Eq. (4.10) with $k = 1$ (the smallest possible compression ratio) is not larger than B, which is the maximum amount of energy available to the device.

4.5.3 *How to choose r*

Although in general the retransmission mechanism is more efficient than the single transmission strategy, selecting the appropriate value of r is not trivial, because the attainable gain is not proportional to r, but also depends on some system parameters such as the discretization of the compression ratio (hence, m) and the energy availability. However, there exists a simple procedure to determine $\bar{r}$, the largest value of r for which choosing any $q < r$ would lead to poorer performance. It consists in the three steps.

1. Determine the maximum number $r_{\max}$ of data blocks that can fit a packet of size S when using the maximum available energy $u = B$.
2. Compute $k_{q,R}^\star$ for every $q \leq r_{\max}$.
3. $\bar{r}$ is the largest $q \in \{2, \ldots, r_{\max}\}$ for which $D(k_{q,R}^\star) \leq (q-1)/q\, D(k_{q-1,R}^\star)$.

This procedure requires little computation and allows one to determine the maximum value $\bar{r}$ of transmission attempts that can be dedicated to each data block so as to ensure the best performance. Finally, there are two additional factors that need to be taken into account in the design of the retransmission scheme.

- *Latency*: the information contained in a packet may lose significance as latency increases. In this case, sending a data block long after it has been originated may even be disadvantageous: receiving it brings no benefit to the final application, and, at the same time, reduces the quality of the other data blocks that are transmitted along with it, as it occupies part of the bits available for the transmission.
- *QoS*: the final application may dictate a minimum QoS threshold, i.e., a maximum distortion that can be tolerated on the received information. When data blocks are transmitted together, they are compressed more and have a larger distortion, which may violate the QoS constraints.

Thus, the choice of r should be guided by the specific application constraints and by the values of $k_r^\star$ and $D(k_r^\star)$.

Observation. As discussed previously, unless $r > r_{\max}$, the simple retransmission scheme proposed in this study lowers the distortion at the receiver, and this gain depends on the value of r and on the energy availability. Further improvements may be obtained by adopting a more flexible retransmission strategy. For example, if $q \leq r$ consecutive failures occurred, the device may dynamically decide the number of data blocks to transmit, i.e., retransmit $q' \leq q$ packets rather than exactly q, and retransmit the remaining $q - q'$ data blocks in successive slots. However, the related investigations are beyond the scope of this work.

4.6 Numerical evaluation

This sections shows how the average long-term cost $C^\star$ is affected by the system parameters and compare the performance obtained with the single-transmission scheme (i.e., $r = 1$), and the retransmission scheme with $r \geq 2$. The average distortion is evaluated also for the case of an energy-unaware greedy scheme is adopted. This *greedy* policy is myopic and does not optimize the energy consumption according to the expected future availability; i.e., when in state (x, b, q), the node uses all the energy it has in the battery, unless it is more than $u_q^\star$, i.e., the energy needed to achieve $k_{q,R}^\star$. Hence, $u = \min(b, u_q^\star)$.

4.6.1 *System parameters*

The role of the various system parameters has been investigated by running RVIA for the chosen system configurations. In all cases, the original packet size is $L_0 = 500$ bits and the nodes can decide among $m = 30$ different compression ratios. The parameters of the distortion curve of Eq. (4.1) have been derived from [81]; in particular, we set $a = 0.35$ and $b = 19.9$. Notice that with these choices of a, b and m, the granularity of the compression ratio (i.e., $1/m$) is such that $D(1) < D_{full}$, which guarantees that the distortion function is decreasing.

The considered transmission power is $P_{\text{tx}} = 25$ mW, the used bandwidth is $W = 125$ kHz, and the overall noise power spectral density $N_0 = -167$ dBm/Hz. The path loss exponent is 3.5 and Rayleigh fading is modeled as an exponential random variable with unit mean. The battery dynamics of Eq. (4.8) assumes that the energy is quantized. Consistently, all the terms of energy consumption, i.e., Eqs. (4.5), (4.6), (4.7), have been mapped into quanta (we ensured an appropriate granularity for this purpose). Also, the contribution of the processing operations to the overall energy consumption is significant (cf. [43]); in particular, $E_P(\cdot)$ represents from 2% up to 45% of the total energy

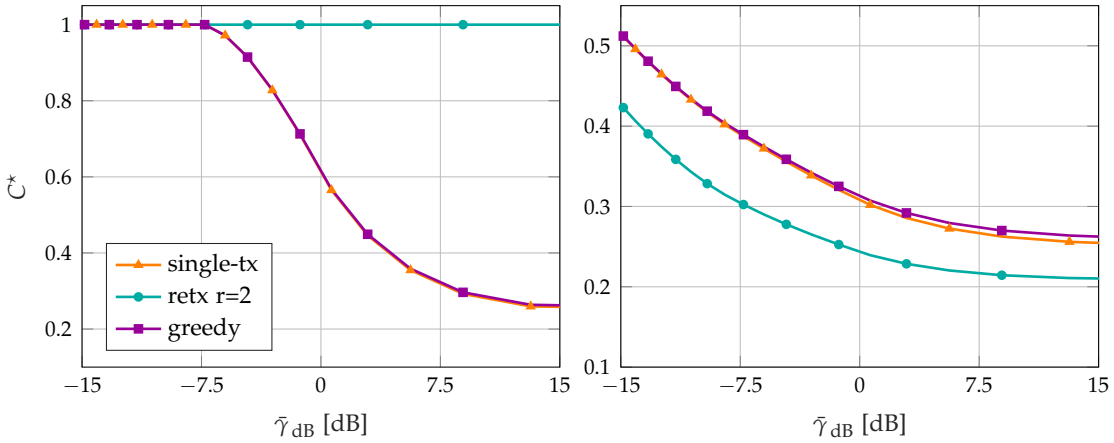


Figure 4.6: Average long-term distortion vs. SNR for $\bar{B} = 0.6$ (left) and $\bar{B} = 0.8$ (right) when $\bar{\mu} = 1$.

consumption for a single data block, according to the compression ratio. The circuitry contribution is smaller and represents about 5% of the overall energy consumption. It is to be remarked that, according to Eq. (4.10), a packet cannot be sent if the available energy is below a certain threshold.

In the EH process, the probability that the source goes from the bad to the good state is 3 times greater than that of the opposite transition. The proposed framework is general and can accommodate any assumptions about the number of states of the harvesting process and the harvesting statistics in each state. However, to obtain some representative results, we consider a 2-state MC ($X = 2$). In particular, $x = 0$ represents a low energy state (e.g., night) during which no energy can be harvested ($e = 0$ with probability 1); when $x = 1$, the source is in a high energy state (e.g., day) and the energy income follows a truncated discrete normal distribution, $e \sim \mathcal{N}(\mu, \sigma^2)$ in the discrete interval $\{1, \dots E\}$ (analogous to [57] and [52]). In the simulations, the variance is fixed as $\sigma^2 = 10$, whereas the average energy income μ varies.

Finally, let $e_{\max}$ be the maximum energy consumption demanded by the processing and transmission of a single data block; then, it is possible to introduce the normalized quantities $\overline{\mu} = \mu / e_{\max}$, and $\overline{B} = B / e_{\max}$. The focus of the simulations was on situations of energy scarcity, i.e., with small $\bar{B}$ or $\bar{\mu}$.

4.6.2 *Results*

Fig. 4.6 shows the average long-term cost as a function of the average SNR from the receiver when $\bar{\mu} = 1$ and for two different values of the battery size, namely $\bar{B} = 0.6$ and $\bar{B} = 0.8$. Clearly, when the node is farther from its receiver, the path loss component increases (low SNR), thereby leading to a larger outage probability. To guarantee that its measurements do not get lost in the transmission, the node will use a stronger coding

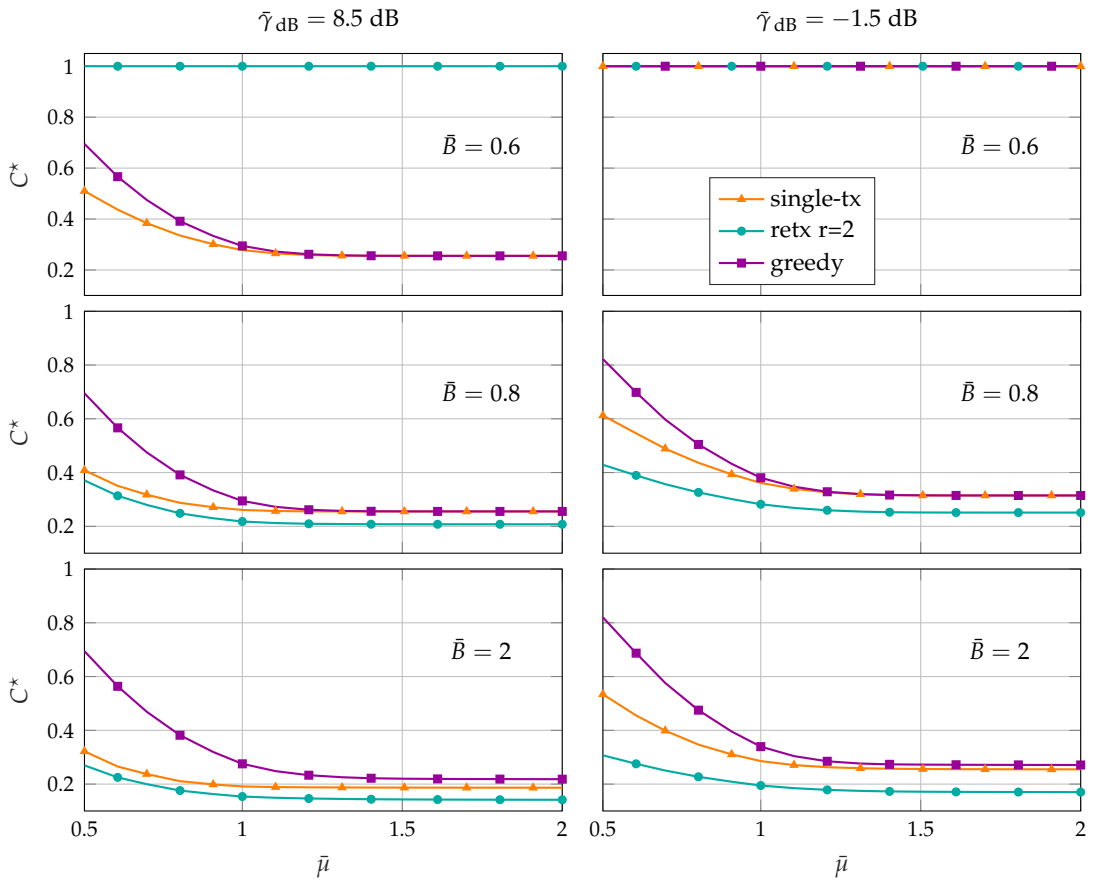


Figure 4.7: Average long-term distortion vs. normalized average energy income during the "good" state for $\bar{B}=0.6$ (top), $\bar{B}=0.8$, and $\bar{B}=2$ (bottom) and for $\bar{\gamma}_{\text{dB}}=8.5$ dB (left) and $\bar{\gamma}_{\text{dB}}=-1.5$ dB (right).

rate, thus the compression ratio needs to be smaller and a larger distortion is introduced at the source. In both cases, the battery size is very small and such that the node's readings have to be compressed because the energy stored is not enough to send them as they are ($\bar{B} < 1$). This clearly impinges on the distortion that can be achieved on average. It is interesting to note the behavior of the retransmission policy when $\bar{B} = 0.6$ (Fig. 4.6, top). As discussed in Sec. 4.5.2, the energy required to send two or more packets together is larger than that needed for a single one, since it requires to spend more energy for processing. In this case, the battery size is very small, and the node cannot store enough energy to transmit two packets together ($r_{\max} = 1$). So, after a failure, the device will keep trying to transmit two data blocks together but the only action it can choose is to discard them, obtaining full distortion. Also, because of the small battery size, the optimal single-transmission policy performs similarly to the greedy one. When instead B is large enough, the retransmission mechanism leads to improvements with respect to the single-transmission one, as depicted in Fig. 4.6, bottom.

Fig. 4.7 shows the performance as a function of the normalized average energy income during the good source state, μ. The average SNR is fixed to 8.5 dB for the figures on the

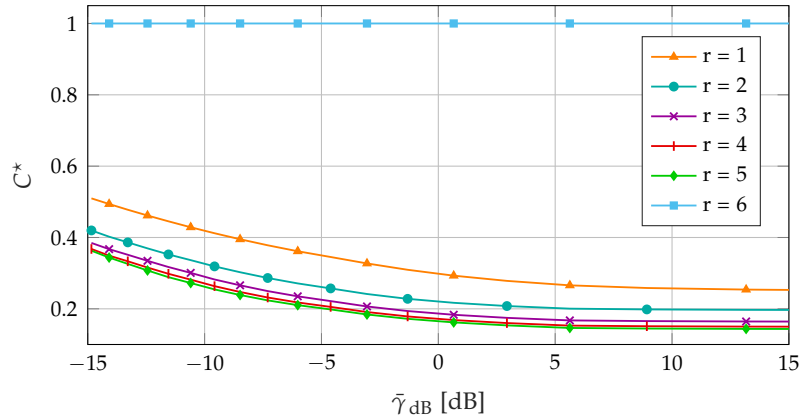


Figure 4.8: Average long-term distortion vs. SNR for different values of r when $\bar{B} = 1$ and $\bar{\mu} = 1$.

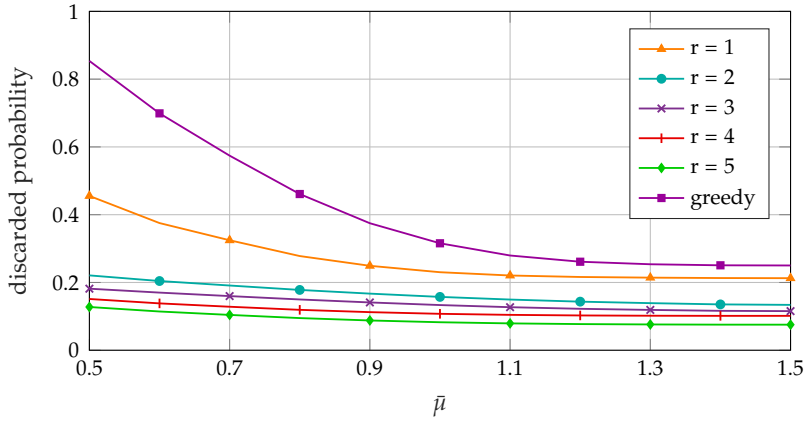


Figure 4.9: Packet loss probability vs. $\bar{\mu}$ for different values of r when $\bar{B} = 1$ and $\bar{\gamma}_{\text{dB}} = -1.5$ dB.

left, and -1.5 dB for those on the right. The normalized battery size is $\bar{B} = 0.6, 0.8$, and 2 for the figures on the top, middle, and bottom, respectively. Intuitively, the more the energy that can be harvested (large μ), the lower the average distortion, because the node can choose the optimal point in the rate distortion tradeoff (see Sec. 4.4.1) more often. However, after a certain value of μ, the distortion curve tends to remain constant because (i) the battery size is too small and part of the incoming energy needs to be discarded, and (ii) the optimal $k^\star$ is already achievable and the excess energy is not useful. As already seen in Fig. 4.6, if the battery size is too small compared to the minimum energy required to send $q = 1, \dots, r$ packets, the node will never transmit them, and this effect sharpens as the SNR decreases. Fig. 4.7 also shows that, when the SNR is large and there is enough energy available (second and third plots on the left), the single-transmission optimal policy behaves similarly to the retransmission one, as the outage probability is low; otherwise, its performance becomes closer to that of the greedy policy and the improvement obtained with the retransmission scheme becomes more significant (see plots on the right).

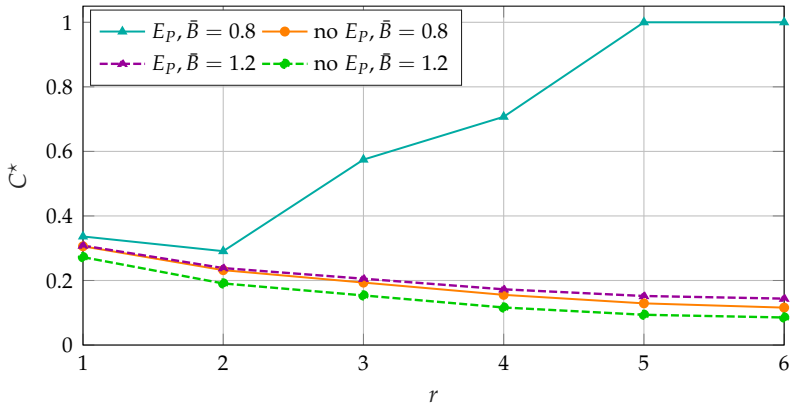


Figure 4.10: Average long-term distortion as a function of r for $\bar{\gamma}_{\text{dB}} = -1.5$ dB.

The effect of the maximum number of retransmissions on the distortion as a function of the average SNR is shown in Fig. 4.8 for $\bar{B} = 1$ and $\bar{\mu} = 1$. As r increases, the gain obtained becomes smaller: if many packets are sent together in a single slot, the distortion introduced by the lossy compression is large, and this reduces the benefit of retransmitting. If r becomes too large ($r > r_{\max}$), there never is enough energy to process and transmit r data blocks and the distortion becomes maximal, as explained in Sec. 4.5.

Fig. 4.9 shows the packet loss probability as a function of $\bar{\mu}$ for different values of r and $\bar{B} = 1$, where this probability accounts for both the packets lost in the communication and those discarded at the source. Again, it can be observed how the benefits of the retransmission scheme fade as r increases. The gain obtained over the greedy policy by the Value Iteration algorithm is more significant when energy is scarce (small μ), for both the single-transmission scheme ($r = 1$) and the retransmission mechanism ($r > 1$).

Finally, we gauged the impact of the energy consumption due to processing. In all previous results, this contribution was proportional to the compression ratio, as per Eqs. (4.5), so that the energy availability strongly influenced the processing operations, see Eq. (4.15). Fig. 4.10 shows the impact that the processing energy has on the performance and compares the average distortion obtained when the compression energy is considered and neglected as r increases, for two values of $\bar{B}$. The average energy income is $\bar{\mu} = 1$, and the average SNR is $\bar{\gamma}_{\text{dB}} = -1.5$ dB. Clearly, when compression has no energy cost, the average distortion is lower. However, when there is enough energy available ($\bar{B} = 1.2$, dashed lines), the performance obtained in the two cases is similar because the energy abundance allows one to have $k_q^\star \equiv k_{q,R}^\star$ very often, even when the compression operations consume energy. On the other hand, if the available energy is scarce ($\bar{B} = 0.8$, solid lines), there is an evident mismatch between the two approaches, as processing consumes part of the scarce energy available and $k_q^\star \equiv k_{q,E}^\star$ in many states. This behavior is exacerbated as r grows, since the processing energy is proportional to the number

of data blocks that share the time slot. In fact, the average distortion saturates to 1 as $r > r_{\max}$ (which is equal to 4 for the configuration of Fig. 4.10, solid cyan line). If $E_P(k) = 0 \, \forall k$, then there exists no $r_{\max}$ (see Sec. 4.5.2). The results of Fig. 4.10 highlight the importance of using an accurate model for the energy consumption, which is central to obtain a reliable and meaningful numerical evaluation.

4.7 Lesson learned

The goal of this study is similar to that of Ch. 3 and in both cases the devices access the channel in a TDMA fashion. However, the two scenarios are different, as the study in this chapter includes energy harvesting and channel coding, which are not considered in Ch. 3. Moreover, the two strategies tackle the problem differently. In Ch. 3, the network resources are assigned to the devices based on their requirements, thus the optimization considers all users jointly; data compression allows one to trade the accuracy of the transmitted information with the energy consumption. The scheme proposed in this chapter, instead, optimizes the processing and transmission operations for each device separately, given the resources it is provided. This latter approach is easier to deploy in a distributed fashion and requires less information to be exchanged with the receiver. The channel encoding introduces an additional dimension in the tradeoff between data compression and energy utilization.

Both the analytical investigation and the numeral evaluation showed that the retransmission scheme ensures an improved average quality of the received information with respect to the simpler single-transmission scheme, unless the energy available to the node is very scarce (because either the battery size is too small or the energy inflow is insufficient). In this case, the retransmission scheme demands too much energy and the node cannot afford it. When the energy is scarce, a more flexible scheme that decides whether or not to retransmit a packet may improve the performance. For instance, if a packet is lost and the battery charge is low, then the node should opt for transmitting only the new readings and give up on those that were lost, so as to preserve some energy. Similarly, a device could also choose how many packets to retransmit rather than considering all those that were lost. The gain introduced by the retransmission scheme becomes more significant as the probability of outage increases. As the maximum number of transmission attempts that can be dedicated to a packet increases, the performance improvement tends to fade, while also affecting latency.

Another interesting extension consists in allowing for some shared time slots, where the devices can send their lost data in a random fashion, i.e., contending for the channel. This may bring improvements with respect to the proposed simpler schemes where resources are not shared.

5

Energy-efficient random access schemes

Chs. 3 and 4 propose TDMA-based access schemes for IoT networks, where each device has its own dedicated time resources that thus can be optimally exploited to guarantee the desired QoS. However, a dedicated channel use may be inefficient in dense scenarios and when communication with the BS is not periodic or is infrequent. In these cases, contention-based schemes may yield improved performance since they are more flexible than coordinated ones and have no synchronization overhead.

This chapter investigates the QoS/energy tradeoff when devices access the channel randomly. Similarly to the scenarios in Chs. 3 and 4, multiple sensor nodes track some phenomena of interest and report data to a common Fusion Center (FC). Unlike the TDMA approaches discussed in the previous chapters, the study proposed here leverages the time correlation characteristics of the monitored signals to adapt the sampling and transmission rates based on the target estimation accuracy and on the probability of packet losses caused by the interference from other sensors.

In principle, the phenomena of interest should be constantly monitored, so as not to miss any critical event. However, there are some issues that need to be taken into account and that affect both the sensing and the reporting regimes. First, wireless sensors need to operate with little to no maintenance and without wired connections to any infrastructure, in particular to the electricity grid. To prolong their operating life, energy must be carefully rationed between the sensing and the communication apparatuses. Second, in the presence of a massive number of sensing devices, channel access needs to be carefully managed in order to limit the mutual interference among the transmitters and, consequently, the fraction of transmission losses due to packet collisions, which may decrease the QoS at the FC. Most of the work that deals with signal compression and data monitoring do not consider the effect of channel errors and interference, which instead may have a significant impact on the accuracy of the monitoring operation. Notice that packet losses also waste energy. A third issue that should be considered is related to the large dynamics of the sensed signals, which causes issues in their estimation. While many signals may appear stationary, or even almost constant, on a small time scale, their

behavior often changes if observed for a sufficiently long time. Also, intervals when data has a large variance, and therefore are difficult to predict, is often of greater interest than intervals where data is almost static. This is because the former is typical of anomalous conditions in the monitored system and, therefore, must trigger the warnings leading to an appropriate intervention.

The use of compression has the potential to reduce communication and sampling energy cost, thus increasing network lifetime. Unfortunately, conventional compression algorithms are not directly applicable to WSNs [86] because they minimize space occupation instead of energy expenditure for data transmission. Also, exceedingly complex algorithms are not implementable in constrained sensing devices. Instead, compression techniques specifically designed for sensor networks have proved to substantially increase network lifetime. These techniques can operate on three different levels [86]. *Sampling compression* leverages data correlation in space and/or time to reduce the sampling activity of devices; the FC is then in charge of reconstructing the complete process by exploiting the correlation properties [87, 88, 89]. Similarly, compressive sensing techniques exploit suitable projections into a sparse space, which makes it possible to recover the original signal with excellent accuracy from just a few samples [90, 91]. *Data compression* processes the sampled measures in order to limit the length of messages directed to the FC [92], e.g., by quantizing the difference between consecutive samples with an appropriate modulation scheme [93]. Finally, *communication compression* aims at reducing the number of data transmissions and their time-on-air, in order to reduce the energy consumed by the transceiver module. For example, the sensors and the FC may agree on an appropriate model for the tracked parameter, so that only the measurements that deviate significantly from the model need to be sent [94, 95]. The model parameters may be updated in real-time when the prediction error starts diverging. Note that the sensor readings need to be acquired with the desired temporal accuracy even if most of them will not be transmitted. Hence this approach may consume much more energy than sampling compression techniques, especially if the energy required by sensing is large, as for some types of sensors [96, 86].

Clearly, the use of any compression technique reduces the accuracy of the signal reconstructed by the FC, trading it with increased network lifetime. Therefore, it is of interest to apply a combination of these techniques and optimize their parameters in order to provide the optimal balance between energy consumption, wireless channel occupation, and sensing accuracy.

This chapter introduces three different random channel access schemes with the common goal of minimizing the devices' energy consumption, while guaranteeing that

the error in the signal estimate at the FC does not exceed a chosen threshold. However, the three approaches differ in target application and channel access mechanism.

The first one [97] considers an event-triggering system, where the interest is on the identification of some events that may trigger actions, e.g., excessively high pressure in a pipe that requires the release of a valve. Responsiveness is crucial to guarantee a smooth operation of the controlled system; the sensor nodes report their readings to the FC, that is able to notify changes of conditions to human operators and, possibly, to act on their behalf. The error metric is calculated independently for every sample, since past values are not of interest when new data is available. The proposed strategy jointly determines the quantization granularity at the source and the optimal reporting window based on the expected collision probability and the estimation error at the FC; sampling and transmission operations are further decoupled for a reduced energy consumption.

The second scheme [J8] targets the same scenario of the first one but considers a probabilistic channel access scheme, where the transmission probability depends on the lag since the last successful transmission and is computed based on the correlation profile of the monitored signal and the expected interference. Differently from the previous scheme, here no data compression is performed.

Finally, the third scheme [98] considers a different application, where the process of interest is the time integral of measurements provided by some sensors, e.g., the volume of fluid crossing a pipe, the total amount of fertilizer/insecticide dispersed onto a certain cultivation area, and so on. The aim is to collect data to enable its statistical analysis and thereby perform trend analysis or predict future values. While the two former cases analyze the time series as a whole, including its past history, this application only considers its current realization. This translates into a different target performance metric, which considers the cumulative signal estimate error up to the current time, possibly using a weight function to smooth out the impact of the past errors.

All the three schemes use a stochastic geometry approach to characterize the interference caused by the other devices. This allows for a realistic wireless channel model, where successful reception is determined based on a Signal-to-Interference-and-Noise Ratio (SINR) threshold model, which has been proven to accurately represent real channels [99]. Notice that most of the studies in the literature do not consider the impact of packet losses or use a simplified wireless channel where a message is lost if it collides with any other message [91, 100, 101, 102].

Finally, we would like to mention that these schemes have been realized in collaboration with Daniel Zucchetto.

5.1 System model

Devices are organized in a static wireless network with a star topology; the sensor nodes track some phenomena of interest and report their measurements to a central FC via single-hop wireless communication. The objective is to guarantee a limited reconstruction error at the receiver for each signal, while maximizing the network lifetime. The network is asymmetric: the FC has no energy, computational, and storage restrictions, whereas the nodes are battery-powered and have lower computational capabilities, so that simple energy-efficient transmission policies are sought.

5.1.1 *Monitored process model*

Time is slotted and slots have fixed duration. Each sensor node tracks a temporal signal of interest $\{x_k\}$, where $k \in \mathbb{N}$ is the slot index; the signal exhibits temporal correlation. Different devices may measure different signals, but the node index is omitted in favor of a lighter notation. The sensory data is measured only in correspondence of a transmission attempt to the FC, otherwise the node is in an energy-preserving sleep mode.

To keep the analysis simple, we consider a straightforward but popular and significant signal model, consisting in an autoregressive (AR) model with degree 1. Therefore, the time series evolves as

$$x_k = \alpha x_{k-1} + u_k \quad k > 0\,, \tag{5.1}$$

with α a non zero constant and $u_k \sim \mathcal{N}(0, \sigma^2)$ a zero mean Gaussian innovation term, with variance σ^2 and assumed to be independent and identically distributed (i.i.d.) over time. It is assumed that $|\alpha| < 1$, so that $\{x_k\}$ is a stable process.

Although the AR model is very simple, it can provide a good representation of real-world time-correlated time series. However, the procedures proposed in the following sections can be readily adapted to different signal models. The core step consists in characterizing their correlation profile, so as to have a mathematical expression for the expected estimation error at lag k.

5.1.2 *Data transmission.*

If in slot k a device chooses to transmit x_k, it can also decide how much information to send, so that it may transmit a distorted version $\tilde{x}_k$ of x_k to the receiver. The resulting packet size L can take values in a finite set $\mathcal{L}$, and the smaller the packet, the larger the distortion of the compressed measurement. Therefore, the current data sent by a node can be modeled as $\tilde{x}_n = x_k + v(L)$, where $v(L)$ represents the error due to the lossy compression of the original signal x_k, whose statistical distribution depends on the type of data processing performed by the node. For the sake of simplicity, in this study $v(L)$

is assumed to have zero-mean normal distribution, $v(L) \sim \mathcal{N}(0, \omega^2(L))$, with variance $\omega^2(L)$ that increases for higher compression ratios, i.e., for smaller values of L. The distortion function used here is the same of Chs. 3 and 4 [10]

$$\omega^2(L) = a \left(\frac{L - L_0}{L_{\max} - L_0} \right)^{-b} - 1 \tag{5.2}$$

where L_0 is the size of the fixed part of the packet (header and non-compressed data), $L_{\max} > L_0$ is the maximum packet length, and a and b are parameters that depend on the compression algorithm.

The devices access the channel randomly, as this avoids the burden of generating and maintaining a synchronized schedule with the FC. The sensor nodes perform power control to counteract the path loss. Depending on the channel conditions and the interference caused by the other nodes, the packet may be lost. A successful transmission is immediately acknowledged by a downlink packet, which is considered to be always successful. If the transmission is successful, the receiver is able to perfectly reconstruct $\tilde{x}_k$. Otherwise, the FC maintains an estimate $\hat{x}_k$ of x_k based on the last received data and the time-correlation characteristics of the signal model. When a device is neither transmitting nor sensing, it switches to a sleep mode in order to save energy. Meanwhile, the FC keeps estimating the process.

5.1.3 *Channel model*

The sensor nodes are at known distances from the FC and transmit wirelessly over Rayleigh fading channels. They access the channel according to a Slotted ALOHA (S-ALOHA) scheme, which avoids the need of central coordination and is more flexible to changes in the network topology and node density than scheduled access schemes. The price to pay for such a simplicity is the risk of destructive interference caused by simultaneous transmissions from different devices. A transmission is successful if the average SINR at the receiver is larger than a predefined threshold; this model has been shown to be more realistic than the simple collision model [99].

Nodes adapt their transmission power in order to counteract the path loss, so that the average received power $\overline{P}_{\text{rx}}$ is the same for all devices. The power control assumption makes the statistics of the SINR the same for all the transmitting devices, and, in particular, independent of their location. The target average received power $\overline{P}_{\text{rx}}$ is chosen such that $\overline{P}_{\text{rx}} \leq P_{\text{tx,max}}/\ell(R)$, where $P_{\text{tx,max}}$ is the maximum transmission power, $\ell(\cdot)$ is the path loss function, and R is the desired coverage radius.

It is possible to adopt a stochastic geometry reasoning and model the sensing devices that transmit in a given slot as a Poisson Point Process (PPP) $\Psi(x, t)$, defined in the

space-time domain $\mathbb{R}^2 \times \mathbb{N}$, with spatial density $\lambda_s(x, \mathtt{P})$, where $x \in \mathbb{R}^2$ is the position of an active node and $\mathtt{P}$ is the *persistency constant*, i.e., the per-slot transmission probability of a node. Thanks to Slivnyak's theorem, the conditional distribution of the interferers given the position of the tagged node is still modeled by Ψ [103].

Each Poisson point is associated with a fading coefficient F, which is potentially different for each device. The interference at the FC is the sum of signal powers received from all active nodes, except the target transmitter j:[1]

$$I = \sum_{i \in \Psi, i \neq j} \overline{P}_{\mathrm{rx}} F_i \,. \tag{5.3}$$

Denoting by $\tilde{F}$ the fading coefficient of the target receiver and N_s the noise power, the corresponding SINR is then $\gamma(I) = \overline{P}_{\mathrm{rx}} \tilde{F} / (N_s + \sum_{i \in \Psi} \overline{P}_{\mathrm{rx}} F_i)$.

Using Shannon's bound as an approximation, the SINR threshold for a packet of size L can be expressed as

$$\Gamma^{\circ}(L) = \beta \left(2^{L/(TB_W)} - 1 \right) , \tag{5.4}$$

where B_W is the transmission bandwidth, T is the time slot duration, and $\beta > 1$ is a coefficient that accounts for the gap between the spectrum efficiency of practical modulation schemes and Shannon's capacity bound.

The transmission success probability can be expressed as

$$\begin{aligned} p_s(L) &= \Pr\left(\gamma(I) > \Gamma^{\circ}(L)\right) \\ &= \Pr\left(F_0 > \left(\frac{N_s}{\overline{P}_{\mathrm{rx}}} + \sum_{i \in \Psi} F_i \right) \Gamma^{\circ}(L) \right) = e^{-N_s \Gamma^{\circ}(L)/\overline{P}_{\mathrm{rx}}} \, \mathrm{E}\left[e^{-\Gamma^{\circ}(L) \sum_{i \in \Psi} F_i} \right] , \end{aligned} \tag{5.5}$$

where the expectation is computed with respect to the interference distribution, conditional to the presence of the target transmitter.

The fading coefficients $\{F_i\}$ can be seen as marks of this PPP, making it possible to apply Campbell's theorem for marked processes [103]. Then, it is

$$\mathrm{E}_{\Psi}\left[e^{-\Gamma^{\circ}(L) \sum_{i \in \Psi} F_i} \right] = \exp\left(- \int_{\mathbb{R}^2} \int_0^{\infty} \left(1 - e^{-\Gamma^{\circ}(L) \varphi} \right) \lambda_s(x, \mathtt{P}) e^{-\varphi} \, \mathrm{d}\varphi, \mathrm{d}x \right) , \tag{5.6}$$

where the expectation is taken with respect to the marked PPP, i.e., considering both the spatial position of the nodes and the fading coefficients. Now, assuming uniform distribution of the nodes within the cell radius and neglecting the "arrivals" of the PPP

1 The interference expressed in this term is conditioned on the presence of the target transmitter, however the Slivnyak's theorem allows one to remove the conditioning [103].

outside the cell, i.e., assuming $\lambda_s(x, \mathsf{P}) \equiv \lambda_s(\mathsf{P})$ for all $|x| \leq R$ and $\lambda_s(x, \mathsf{P}) \equiv 0$ otherwise, it is

$$\begin{aligned} \mathrm{E}_\Psi \left[e^{-\Gamma^\circ(L) \sum_{i \in \Psi} F_i} \right] &= \exp\left(-\lambda_s(\mathsf{P}) \pi R^2 \int_0^\infty \left(1 - e^{-\Gamma^\circ(L)\varphi} \right) e^{-\varphi} \, \mathrm{d}\varphi \right) \\ &= \exp\left(-\lambda_s(\mathsf{P}) \pi R^2 \frac{\Gamma^\circ(L)}{\Gamma^\circ(L)+1} \right) . \end{aligned} \tag{5.7}$$

Replacing this result into (5.5) yields

$$p_s(L) = \exp\left(-\Gamma^\circ(L) \left(\frac{N_s}{\overline{P}_{rx}} + \frac{\lambda_s(\mathsf{P}) \pi R^2}{\Gamma^\circ(L)+1} \right) \right) . \tag{5.8}$$

Notice that the success probability depends on the adaptive transmission strategy through two parameters, namely the packet size L, and the persistency constant P.

5.1.4 *Energy consumption*

Each sensor is powered by a battery with finite initial charge. The energy consumption of a device is mainly due to data sampling, data transmission, and circuitry.[2] According to the model of Ch. 2, each sensing operation requires a fixed amount of energy. The energy for transmission depends on the transmission time, equal to the fixed slot duration, and the transmission power, which is assumed to be constant across the transmission attempts. Each transmission, therefore, consumes the same amount of energy, including that required by circuitry for switching among the node's operating mode.

5.2 Event-triggering system: deterministic channel access

The first proposed scheme targets an event-triggering system where the goal is to identify when the monitored phenomena exceeds some predefined threshold. The reconstruction accuracy at the FC is therefore measured by means of the squared error $|x_k - \hat{x}_k|^2$ for each slot k. Note that this error is zero only if the device transmitted in slot k (no estimation error at the FC, $\hat{x}_k = \tilde{x}_k$), and $L = L_{\max}$ (no distortion introduced at the source, $\tilde{x}_k = x_k$).

5.2.1 *Optimization problem*

The network is assumed to be homogeneous, so that nodes operate based on the same strategy. Moreover, the network dynamics is assumed to be slowly varying, implying that the transmission scheme needs to be updated only seldom. Note that, even though the channel status may vary because of the fading component, the transmission strategy

2 Receiving an acknowledgment from the FC also requires to spend some energy, which however is typically smaller than the transmission energy and, hence, is neglected in this study.

is based on the expected channel conditions, which are static. Therefore, the proposed policy will not depend on the slot index k. The objective is to determine the optimal duration $S^\star$ of the sleeping phase of a device and the optimal packet size $L^\star$ such that (i) the probability that the squared error at the receiver exceeds a predefined threshold b is bounded, and (ii) the sensor's lifetime is maximized.

As explained in Sec. 5.1.4, both sensing and transmission operations have a constant cost. Consequently, maximizing the lifetime of a device is equivalent to minimizing the number of transmission attempts and sensing operations, and thus, to maximizing the duration S of the sleeping phase (while not violating the QoS constraint). Basically, the objective is determining

$$S^\star \triangleq \max_{L,\,S} S(L)\,, \tag{5.9}$$

where $0 \leq L \leq L_{\max}$, subject to the QoS constraint

$$\mathrm{P}\left(|x_k - \hat{x}_k|^2 > b\right) < p_{th} \quad k = 0, 1, \ldots\,, \tag{5.10}$$

i.e., the probability that the squared reconstruction error exceeds b is required not to exceed a predefined threshold p_{th} at any time. The optimal packet size $L^\star$ is the one that maximizes (5.9) under the constraint (5.10).

Before delving into the analysis of the transmission strategy, it is worth to highlight some tradeoffs in the choice of S and L that are induced by the lifetime and QoS requirements. Intuitively, a device should choose a large sleeping window to save energy and limit the interference, since the larger S, the less frequent the transmissions. However, S cannot be too large, in order to respect the QoS constraint (5.10). Similarly, decreasing L increases the signal reconstruction error, but also the success probability, because transmissions will use more robust modulations (see (5.4)).

Therefore, determining the optimal transmission strategy is not trivial. We propose an iterative approach, where, for a given value of L, the sleeping phase duration and the corresponding probability of successful transmission are alternatively optimized, until convergence; this allows one to derive $S_L(L)$, i.e., the optimal value of S for the chosen L. Then, an outer optimization process determines the value $L^\star = \mathrm{argmax} S_L(L)$ that yields $S^\star = S_L(L^\star)$.

This procedure is summarized in Algorithm 3. For a fixed L (Line 2), the duration $S_0(L)$ of the sleeping phase is firstly computed as if there were no interference, i.e., $S_0(L)$ only depends on the QoS requirement (5.10) (Line 3). The interference caused by the other nodes makes transmissions prone to losses. Let $S(L)$ be the number of time slots a node waits from the last successful transmission. Then, assuming that in case of packet

Algorithm 3 Alternate optimization

1: $\mathbf{S}_L \leftarrow$ vector of size $L_{\max}$ ▷ Contains $S_L(L)\ \forall L$
2: **for** $L = 1, \ldots, L_{\max}$ **do**
3: Determine $S_0(L)$ ▷ Sleeping duration if $p_s = 1$
4: Initialize $p_s = 1$, $S(L) = 1$
5: **while** $S(L)$ has not converged **do**
6: $S(L) \leftarrow \lfloor S_0(L) + 1 - 1/p_s \rfloor$
7: $p_s \leftarrow$ Eq. (5.8) with $\lambda_s(\mathtt{P}) = \lambda / S(L)$
8: $\mathbf{S}_L(L) \leftarrow S(L)$
9: $S^\star = \max \mathbf{S}_L$, $L^\star = \text{argmax}\mathbf{S}_L$

loss a device transmits again in the next slot, the expected time elapsed between two consecutive successful transmissions is

$$\sum_{k=0}^{+\infty} (S(L) + k)(1 - p_s(L))^k p_s(L) = S(L) + \frac{1}{p_s(L)} - 1, \tag{5.11}$$

which should not be larger than $S_0(L)$ to satisfy the QoS constraint (5.10). Therefore

$$S(L) = \lfloor S_0(L) + 1 - 1/p_s(L) \rfloor . \tag{5.12}$$

At the first iteration, no packet losses are assumed ($p_s = 1$), and thus $S(L) \equiv S_0(L)$. Then (Line 7), the success probability p_s is updated using Eq. (5.8), where the number of interferers depends on the frequency of transmissions which is influenced by $S(L)$; in particular, it is $\lambda_s(\mathtt{P}) \equiv \lambda / S(L)$, where λ is the density of all devices (regardless they are active or not) . This alternate optimization is repeated until $S(L)$ converges. Then, $L^\star$ and $S^\star$ are those that maximize the value of $S(L)$ as in Line 9.

5.2.2 *Sleeping phase duration*

Here, the optimal duration of the sleeping phase $S_0(L)$ for a given L is derived in the case there is no interference. Exploiting Eq. (5.1), the signal in slot $k+n$ can be related to that in slot k as follows

$$x_{k+n} = \alpha^n x_k + \sum_{\ell=1}^{n} \alpha^{n-\ell} u_{k+\ell} = \alpha^n x_k + w_n = \alpha^n \tilde{x}_k - \alpha^n v(L) + w_n = \alpha^n \tilde{x}_k + e_n(L) \ ; \tag{5.13}$$

where $e_n(L) \triangleq -\alpha^n v(L) + w_n$ is the error due to the distortion introduced at the source and the estimation process.

The estimation error w_n is the linear combination of n i.i.d. zero-mean Gaussian random variables (r.v.s) and, therefore, is itself a zero-mean normal r.v., $w_n \sim \mathcal{N}(0, \sigma_n^2)$, with variance

$$\sigma_n^2 = \sum_{\ell=1}^{n} \alpha^{2(n-\ell)}\sigma^2 = \sum_{\ell=0}^{n-1} \alpha^{2\ell}\sigma^2 = \frac{1-\alpha^{2n}}{1-\alpha^2}\sigma^2 . \tag{5.14}$$

Notice that σ_n^2 is a concave and non-decreasing function of the lag n (number of slots elapsed since the last successful transmission) with a horizontal asymptote at $1/(1-\alpha^2)$ as $n \to +\infty$. This implies that the variance of the estimation error is larger for $|\alpha|$ closer to 1. The reason behind this behavior is that $|\alpha|$ closer to 0 corresponds to a weakly correlated process, which makes the estimation error almost independent of the value of n and practically bounded in the range $[-3\sigma, 3\sigma]$.

Since the compression should not introduce a bias in the measurement, the error $v(L)$ can be assumed to have zero mean, hence the best estimate that the FC can make is $\tilde{x}_{k+n} = \alpha^n \tilde{x}_k$, so that the squared reconstruction error after k slots from the last received data is $|e_n(L)|^2$. Based on its definition (see Eq. (5.13)), $e_n(L)$ is non-decreasing in n, therefore, to guarantee the QoS constraint (5.10), it is sufficient to evaluate the Cumulative Distribution Function (CDF) of the squared error at b only for $n \equiv S$. For analytical tractability, in the following it is assumed that $v(L) \sim \mathcal{N}(0, \omega^2(L))$, although the proposed framework is general and can accommodate any distribution of $v(L)$. Also, $\omega^2(L)$ is decreasing in L, since the smaller the amount of information bits, the larger the distortion. In this case, $e_S(L)$ can be modeled as the weighed sum of two independent normal r.v.s and, thus, is also normally distributed:

$$e_S(L) \sim \mathcal{N}\left(0,\, \alpha^{2S}\omega^2(L) + \sigma_S^2\right). \tag{5.15}$$

The squared error follows a Gamma distribution, $|e_S(L)|^2 \sim \text{Gamma}(k_S, \theta_S(L))$, where the shape and scale parameters are $k_S = 1/2$ and $\theta_S(L) = 2\left(\alpha^{2S}\omega^2(L) + \sigma_S^2\right)$, respectively. Accordingly, the largest value S_0 that satisfies the QoS constraint for the given L is

$$S_0(L) = \max\left\{S : \frac{\gamma\left(k_S,\, b/\theta_S(L)\right)}{\Gamma(k_S)} \geq p_{th}\right\}, \tag{5.16}$$

where $\gamma(\cdot)$ is the lower incomplete gamma function, and $\Gamma(\cdot)$ represents the gamma function. Eq (5.16) gives the maximum duration of the sleeping phase of a sensor when transmission is always successful ($p_s = 1$) such that (5.10) holds true. If the chosen p_{th} is too large, since S is discrete, the set in (5.16) could be empty, which means that the required QoS constraint can not be satisfied even when no interferers are present. In this case, we set $S_0 = 1$.

5.2.3 *Transmission strategy*

With the procedure explained in Algorithm 3, after a successful transmission a device remains silent for $S^\star$ slots before attempting to transmit again, where $S^\star$ yields an expected time between two consecutive successful transmissions such that the QoS requirement (5.10) is satisfied. However, based on how signal $\{x_k\}$ actually evolves, the squared error after $S^\star$ slots may be larger or smaller than threshold b. In the latter case, the device could extend the sleeping phase by $S'(x_{k+S^\star})$ additional slots to further save energy. How to determine $S'(x_{k+S^\star})$ is explained in the following.

Let $\tilde{x}_k = x_k + v(L^\star)$ be the last data received by the FC from a certain user. After slot k, the device sleeps for $S^\star$ time slots and then wakes up to sense the environment. Based on the new measurement $x_{k+S^\star}$ and its knowledge of the estimate $\hat{x}_{k+S^\star} = \alpha^{S^\star}\tilde{x}_n$ performed by the FC, the device chooses whether to immediately transmit the new data (if $|x_{k+S^\star} - \alpha^{S^\star}\tilde{x}_n|^2 > b$) or keep sleeping for $S'(x_{k+S^\star})$ additional time slots. In the latter case, the prediction error $\epsilon_k(L^\star)$ in slot $k \geq k + S^\star$ depends on the estimate of the *FC*, which is based on the last received data $\tilde{x}_k$, and on the expected evolution of the time series, which is based on the last sensed data $x_{k+S^\star}$:

$$\begin{aligned}\epsilon_k(L^\star) = |x_{k+S^\star+k} - \hat{x}_{k+S^\star+k}| &= \left|\alpha^k x_{k+S^\star} + w_k - \alpha^{S^\star+k}\tilde{x}_k\right| \\ &= \left|\alpha^k x_{k+S^\star} + w_k - \alpha^{S^\star+k}(x_k + v(L^\star))\right| \quad (5.17)\end{aligned}$$

where $w_k \sim \mathcal{N}(0, \sigma_k^2)$ was defined in (5.13). The other terms are known by the transmitter, because $x_{k+S^\star}$ and x_k are the new and old sensed data, and the processing error $v(L^\star)$ depends on $L^\star$, already obtained with Algorithm 3. Then, the error is normally distributed: $\epsilon_k(L^\star) \sim \mathcal{N}\left(\mu_k(L^\star),\, \sigma_k^2\right)$, with mean

$$\mu_k(L^\star) = \alpha^k \left(x_{k+S^\star} - \alpha^{S^\star} x_n - \alpha^S v(L^\star)\right) . \quad (5.18)$$

Since $\epsilon_k(L)$ has non-zero mean, its square is proportional to a non-central χ^2 random variable with one degree of freedom and non-centrality parameter equal to the ratio between the squared mean and the variance of $\epsilon_k(L)$, i.e.,

$$\frac{1}{\sigma_k^2}\epsilon_k^2(L^\star) \sim \chi_1^2\left(\frac{\left(\mu_k(L^\star)\right)^2}{\sigma_k^2}\right) . \quad (5.19)$$

It is then straightforward to compute the CDF of $\epsilon_k^2(L^\star)$ for a given value of b and derive S' as the largest k for which such value is not lower than p_{th}, as done in Sec. 5.2.2 to find $S_0(L)$.

Table 5.1: Simulation parameters.

Interference and communication parameters		
Time slot duration	T	100 ms
Density of sensor devices	λ	0.001 nodes/m^2
Cell radius	R	500 m
Received power[3]	$\overline{P}_{\text{rx}}$	4 nW
Transmission bandwidth	B_W	125 kHz
Noise power	N_s	$2.5 \cdot 10^{-15}$ W
Signal model and QoS parameters		
Autoregressive model parameters	α	0.99
	σ^2	0.001
Initial value	x_0	0.8
Maximum message length	$L_{\max}$	24 bit
Threshold on $P[\lvert x_n - \hat{x}_n \rvert^2 \leq b]$	p_{th}	0.8
Kulau et al. strategy [87]		
Maximum sleep time	$t_{\max}$	50 slots
Weighting exponent	ϕ_{bb}	2
Sliding window size	n_s	30

Besides allowing the sensing devices to save energy, this dynamic adaptation of the sleeping phase also reduces the interference on the channel. However, for mathematical tractability, in the optimization routine the interference is calculated in the pessimistic case, i.e., considering sender devices to sense and transmit messages exactly every S slots, ignoring the dynamic sampling rate adaptation carried out by each device.

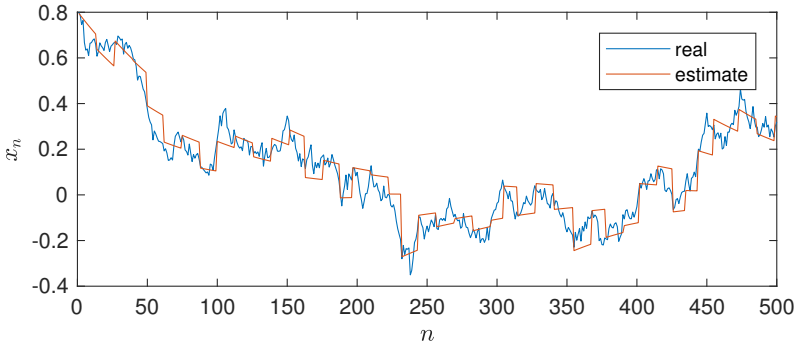
5.2.4 *Numerical evaluation*

The performance gain of the proposed system was analyzed by means of simulations, with the parameters set as in Tab. 5.1.

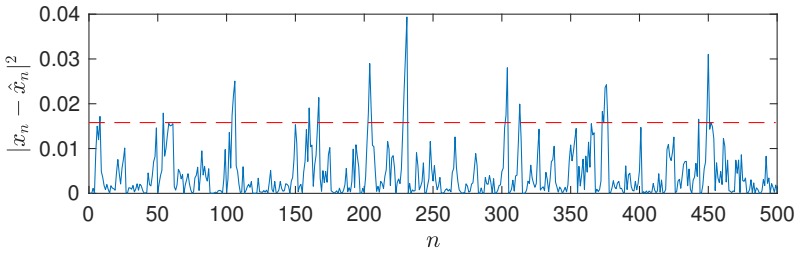
Fig.5.1 shows an example of the original time series and the corresponding estimate with the technique described previously. The match is very good, and the squared error is almost always below the given threshold b. Supported by this first result, we compare our strategy (named *dynamic*) against other two techniques. The *static* technique is the same as the strategy proposed in this study, but without the dynamic extension of the sleeping phase described in Sec. 5.2.3. The other one is the sample rate adaptation technique described by Kulau et al. [87], which uses Bollinger bands to dynamically estimate the next sleeping time based on the variability of the previously seen data. In particular, the time between two sample acquisitions is calculated as $t_{\text{wait}}(n) = t_{\max}/(1 + (b_{\text{bb}}\,\sigma_{\text{bb}}(n))^{\phi_{\text{bb}}})$ where $\sigma_{\text{BB}}(n)$ is the standard deviation of the last n_s acquired samples, and $t_{\max}$ is the maximum sleeping duration.

Fig. 5.2 shows the probability that the squared error at the FC stays within threshold b, as b increases. To guarantee a fair comparison, b_{bb} was set so that the probability

3 Note that this value allows devices at the cell edge to use the ETSI imposed limit of 25 mW on the transmission power for the 868 MHz band.



(a) Time series



(b) Squared error. The red line indicates the QoS threshold b.

Figure 5.1: Example of a time series and its estimate with our dynamic technique (with $b = 0.0158$).

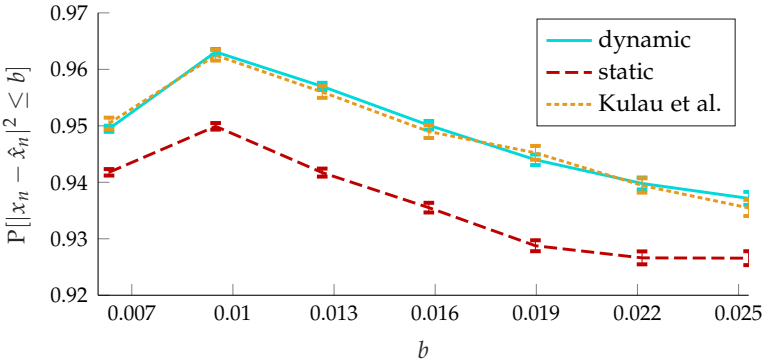


Figure 5.2: Probability that the squared error stays within threshold b, with 95% confidence intervals.

that the squared error is lower than b is the same as for our strategy.[4] It can be seen that the proposed technique respects the QoS constraint with a large margin. Fig. 5.2 actually shows that the proposed dynamic policy is overly conservative, since the interference level considered is obtained by the static optimization of S, but the dynamic adjustment of the sleeping interval lowers the actual interference on the channel. Also the static policy is conservative, because, for analytical tractability, the optimization routine of Algorithm 3 considers every message to be repeated the expected number of transmissions needed to get a successful reception (see Eq. (5.12)). Instead, a more

4 Namely, $b_{\text{bb}} = [55.0, 48.1, 32.9, 22.8, 16.4, 12.0, 9.5]$.

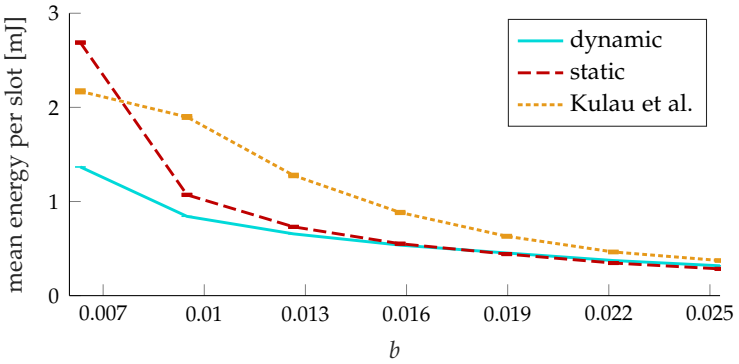


Figure 5.3: Mean energy consumed per slot, with 95% confidence intervals.

precise optimization could be performed by considering the CDF of the number of time slots a node waits from the last successful transmission.

To evaluate the energy consumption, the sum of the circuit and transmission power was set to 40.5 mW, and the sensing energy to be 495 µJ.[5] As Fig. 5.3 shows, the proposed strategy is between 15% and 50% more efficient than the technique described in [87], especially when a lower error is required. Also, note that the dynamic policy, compared to the static one, saves energy by postponing transmissions when the estimation values are still sufficiently accurate. However, if the value of b is too large, the transmission can be often postponed, but the device may have to perform frequent sensing operations. In this situation, as shown in Fig. 5.3, the energy efficiency loss can be quite significant and the static strategy may outperform the dynamic one.

5.3 Event-triggering system: probabilistic channel access

This scenario is similar to that considered in 5.2, but does not include data compression and assumes a probabilistic S-ALOHA channel access. Each device has a probability $p_{\text{tx}}(n;\boldsymbol{\varphi})$ of waking up to transmit a packet, which depends on the number of slots n since the last successful transmission and a number of parameters $\boldsymbol{\varphi} = \{\varphi_1, \varphi_2, \ldots\}$, which are the optimization variables. Notice that frequent transmissions can potentially improve the reconstruction accuracy because they reduce the estimation error, but deplete the battery faster and also generate more interference, which may cause packet losses. The objective is to determine the transmission probabilities $\{p_{\text{tx}}(\cdot;\boldsymbol{\varphi})\}$ that guarantee a desired level of accuracy in the tracking process and, at the same time, optimize the energy usage.

It is assumed that each sensing and transmission operation requires the same amount of energy (see Sec. 5.1). Consequently, maximizing the lifetime of a device is equivalent

5 These values have been determined by considering the use of the Atmel AT86RF212B radio transceiver and the Infineon KP275 digital pressure sensor.

to minimizing the number of transmission attempts and sensing operations, and, thus, to maximizing the mean time interval τ_{tx} between two consecutive transmission attempts. Such a maximization must be performed while guaranteeing a certain QoS, which is here defined in terms of a threshold p_{th} on the average outage probability. The outage probability after n slots since the last received data is defined as the probability that the squared signal prediction error exceeds a threshold b, i.e.,

$$p_{out}(n) = \Pr\left[|x_{k+n} - \hat{x}_{k+n}|^2 > b\right], \tag{5.20}$$

where k is the last slot where a sample was correctly received by the FC, while x_{k+n} and $\hat{x}_{k+n}$ are the actual and the estimated signal in slot $k+n$, respectively. Note that the outage probability depends only on the lag n and not on the absolute time k of the last correct reception because the prediction error is reset to zero any time a new measurement is correctly delivered to the FC. This yields $p_{out}(0) = 0$. So, formally, the optimization problem is

$$\boldsymbol{\varphi}^\star = \underset{\boldsymbol{\varphi}}{\operatorname{argmax}}\ \mathrm{E}\left[\tau_{tx} | p_{tx}(\,\cdot\,;\boldsymbol{\varphi})\right] \tag{5.21a}$$

$$\text{s.t. } \mathrm{E}_n\left[p_{out}(n)\right] < p_{th} \tag{5.21b}$$

where $\mathrm{E}[\cdot]$ denotes the expectation operator, that, with the subscript n, is intended to be applied to the distribution of the random variable n. The optimal transmission probability function is then given by $p_{tx}(\,\cdot\,,\boldsymbol{\varphi}^*)$. Although the numerical evaluation is performed with the AR model of Sec. 5.1.1, the framework proposed here has general validity and could be used with different time evolution of the signal $\{x_k\}$. The QoS constraint (5.21b) can be expressed as

$$\overline{p}_{out} = \mathrm{E}_n\left[p_{out}(n)\right] = \sum_n p_{out}(n)\pi_n < p_{th} \tag{5.22}$$

where π_n is the probability that, at any given time, the last successful transmission happened n slots before, and will be computed in Sec. 5.3.1.

Solving Problem (5.21) is not trivial. First, we determine the expression of the QoS constraint as a function of the transmission probability function $p_{tx}(\,\cdot\,;\boldsymbol{\varphi})$. To this end, in Sec. 5.3.1, the expression of the number of slots τ needed to *successfully* deliver a message (hence, $\tau \geq \tau_{tx}$) is derived in terms of the transmission probabilities and the probability p_s of successful transmission. Then, in Sec. 5.3.2, p_s is expressed as a function of the mean transmission probability, which, in turn, depends on τ. An alternate optimization

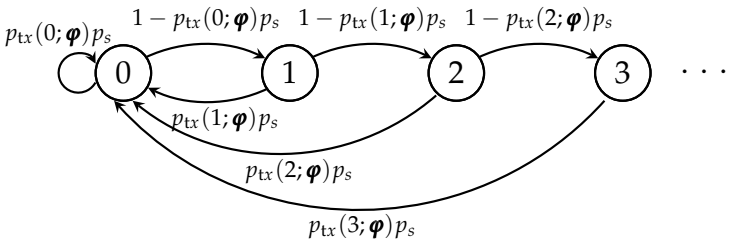


Figure 5.4: Markov chain for the node state.

allows one to derive both p_s and τ, which are interrelated. Finally, in Sec. 5.3.3, the knowledge about the success probability is leveraged to obtain the objective function, i.e., the mean time between two successful transmissions, given the transmission probability function $p_{tx}(\,\cdot\,;\boldsymbol{\varphi})$. Sec. 5.3.4 provides a guideline that summarizes the whole procedure and highlights its main results.

5.3.1 *Distribution of the lag τ*

The number of slots since the last successful message delivery (i.e., the lag τ) can be modeled as the state of the MC in Fig. 5.4. Starting from state i, the next state of the MC is 1 if the device successfully transmits a message, and $i+1$ otherwise. These transitions happen with probability $p_{tx}(i;\boldsymbol{\varphi})p_s$ and $1-p_{tx}(i;\boldsymbol{\varphi})p_s$, respectively (where $p_{tx}(i;\boldsymbol{\varphi})$ is the probability that a device transmits after a lag of i slots). The success probability p_s depends on the channel gain and the interference produced by the other nodes, which are assumed to be independent and stationary in time. The expression of p_s will be derived in Sec. 5.3.2 resorting to a stochastic geometry argument.

Assuming $p_{tx}(n;\boldsymbol{\varphi})$ and p_s are given, the probability mass distribution of τ equals the steady-state probability vector of the MC in Fig. 5.4. The equilibrium equations yield

$$\begin{aligned}
\pi_1 &= \pi_0(1-p_{tx}(0;\boldsymbol{\varphi})p_s)\\
\pi_2 &= \pi_1(1-p_{tx}(1;\boldsymbol{\varphi})p_s) = \pi_0\prod_{k=0}^{1}(1-p_{tx}(k;\boldsymbol{\varphi})p_s)\\
&\vdots\\
pi_i &= \pi_{i-1}(1-p_{tx}(i-1;\boldsymbol{\varphi})p_s) = \pi_0\prod_{k=0}^{i-1}(1-p_{tx}(k;\boldsymbol{\varphi})p_s)
\end{aligned} \tag{5.23}$$

with the additional normalization constraint

$$\sum_{i=0}^{\infty}\pi_i = 1. \tag{5.24}$$

Combining equations (5.23) and (5.24) gives

$$\pi_0 = \left\{ 1 + \sum_{i=2}^{\infty} \prod_{k=1}^{i-1} (1 - p_{\text{tx}}(k; \boldsymbol{\varphi}) p_s) \right\}^{-1} . \tag{5.25}$$

while π_i for any $i > 0$ is given by (5.23).

Observation. The sum in (5.25) has an infinite number of terms, which in practice is difficult to evaluate for an arbitrary choice of the transmission probability function $p_{\text{tx}}(\,\cdot\,; \boldsymbol{\varphi})$. However, when the system is stable, the sum converges and can be approximated with the desired level of accuracy by considering a finite number of terms.

5.3.2 *Success probability p_s*

The success probability is given in Eq. (5.8). In this scenario, the persistency constant $\mathtt{P}$ is given by the transmission probability: $\lambda(\mathtt{P}) \equiv \lambda \mathrm{E}_n [p_{\text{tx}}(n; \boldsymbol{\varphi})]$, where λ is the density of all devices in the network, independently of whether they are active or not.

Therefore, p_s depends on the mean transmission probability

$$\mathrm{E}_n [p_{\text{tx}}(n; \boldsymbol{\varphi})] = \sum_{i=0}^{\infty} p_{\text{tx}}(i; \boldsymbol{\varphi}) \pi_i \, . \tag{5.26}$$

The steady-state probabilities $\boldsymbol{\pi} = [\pi_0, \pi_1, \ldots]$ are computed as described in Sec. 5.3.1. Notice, however, that $\boldsymbol{\pi}$ and p_s are strictly intertwined, as one is needed in order to derive the other and vice versa. To deal with this issue, it is possible to use a fixed-point approach: the success probability is initially set to 1, then the corresponding steady-state probabilities are computed as in (5.23), and used to update the probability of successful transmission as in (5.8), and so forth until convergence. The proof of convergence for such iterative method can be found in [J8]. This allows us to calculate $\boldsymbol{\pi}$ and p_s given $\boldsymbol{\varphi}$. However, the complete solution to the optimization problem, which means finding the optimal value of $\boldsymbol{\varphi}$, requires an external optimization routine. To this end, we now analyze the objective function.

5.3.3 *Mean time between transmissions*

The objective function (5.21a) is the expected time between two consecutive transmission attempts (regardless of their outcome). In order to derive its expression in terms of the transmission probabilities $\{p_{\text{tx}}(n; \boldsymbol{\varphi})\}$, an MC equivalent to that of Sec. 5.3.1 is introduced, where each original state $i > 0$ (which represents a lag of i slots since the last *successful* transmission) is split into two distinct states: i_{fail} and i_{sleep}, corresponding to an unsuccessful transmission and a sleep phase in the last slot, respectively. State 0

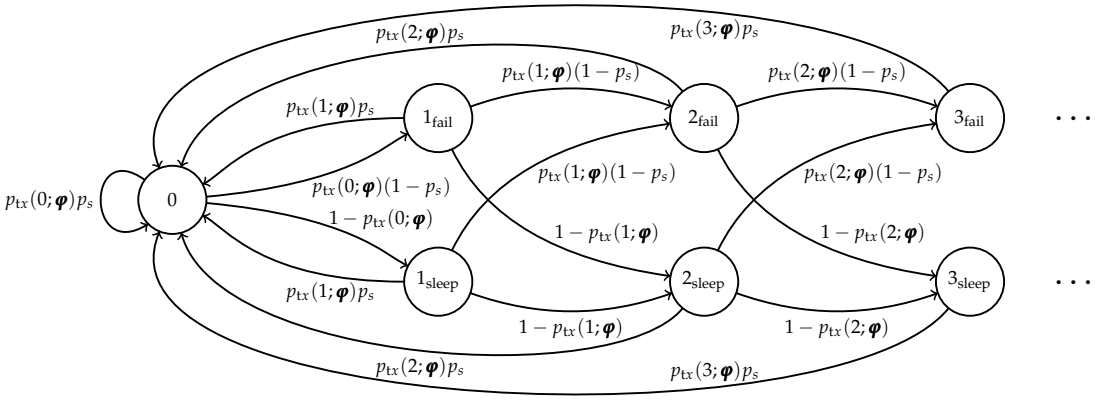


Figure 5.5: Markov chain for the node state with explicit indication of *failure* or *sleep* status.

remains unchanged. As shown in Fig. 5.5, starting from state i_{fail} or i_{sleep} it is possible to transition to three different states:

- State 0 in case of successful transmission in the current slot, which resets the lag. This happens with probability $p_{tx}(i;\boldsymbol{\varphi})p_s$.
- State $(i+1)_{\text{fail}}$ if the device transmits in the current slot, so that the lag i increases by 1, but the packet is lost. This happens with probability $p_{tx}(i;\boldsymbol{\varphi})(1-p_s)$.
- State $(i+1)_{\text{sleep}}$ if the device sleeps in the current slot. This happens with probability $1-p_{tx}(i;\boldsymbol{\varphi})$.

The same transitions happen from state 0. The steady-state probabilities $\tilde{\boldsymbol{\pi}}$ of the expanded MC can be directly computed from those of the original MC ($\boldsymbol{\pi}$) as follows

$$\begin{cases} \tilde{\pi}_0 = \pi_0 \\ \tilde{\pi}_{i_{\text{fail}}} = p_{tx}(i-1;\boldsymbol{\varphi})(1-p_s)\,\pi_{i-1} \qquad i>0 \\ \tilde{\pi}_{i_{\text{sleep}}} = (1-p_{tx}(i-1;\boldsymbol{\varphi}))\,\pi_{i-1} \qquad i>0\,. \end{cases} \tag{5.27}$$

Note that $\tilde{\pi}_{i_{\text{fail}}} + \tilde{\pi}_{i_{\text{sleep}}} = (1-p_{tx}(i-1;\boldsymbol{\varphi})p_s)\,\pi_{i-1} = \pi_i$ for $i>0$.

The expected time $\mathrm{E}[\tau_{tx}]$ between two transmission attempts is computed by introducing $T_{tx}(i)$, which defines the number of slots until the next transmission, given that the MC is in state i_{fail}, with $i>0$, or in $i=0$. Since state i_{fail} corresponds to a failed transmission attempt, the time till the next transmission is at least h slots if the device sleeps in slots $i, i+1, \ldots, i+h-1$, i.e.,

$$\Pr[T_{tx}(i) \geq h] = \prod_{j=0}^{h-1}(1-p_{tx}(i+j)), \tag{5.28}$$

This yields

$$\mathrm{E}\left[T_{\mathrm{tx}}(i)\right] = \sum_{h=1}^{+\infty} \Pr\left[T_{\mathrm{tx}}(i) \geq h\right] = \sum_{h=1}^{+\infty} \prod_{j=0}^{h-1} (1 - p_{\mathrm{tx}}(i+j)). \tag{5.29}$$

Averaging over the starting state i_{fail}, it is possible to calculate the expected time between two transmission attempts as follows

$$\mathrm{E}\left[\tau_{\mathrm{tx}}\right] = A \left(\sum_{i=1}^{+\infty} \mathrm{E}\left[T_{\mathrm{tx}}(i)\right] \tilde{\pi}_{i_{\mathrm{fail}}} + \mathrm{E}\left[T_{\mathrm{tx}}(0)\right] \tilde{\pi}_0 \right) \tag{5.30}$$

where A is a normalization factor required by the definition of T_{tx}, which considers only paths starting from states i_{fail} or $i = 0$. It follows that

$$A = \frac{1}{\sum_{i=1}^{+\infty} \tilde{\pi}_{i_{\mathrm{fail}}} + \pi_0} \tag{5.31}$$

In this way, the objective function (5.21a) is completely defined.

5.3.4 *Summary of relations*

The entire procedure described so far is needed to explicitly write the optimization problem described in (5.21) and obtain a numerical evaluation of the optimal policy.

First, an MC is introduced to model the number of slots since the last successful transmission. Its steady-state probabilities $\boldsymbol{\pi}$ are reported in (5.23) and (5.25), and depend on the expected outcome of a transmission. The success probability p_s can be derived with a stochastic geometry reasoning; it is reported in (5.8) and depends on the mean transmission probability $\mathrm{E}_n\left[p_{\mathrm{tx}}(n;\boldsymbol{\varphi})\right]$. In turn, this quantity depends on the steady-state probabilities of the lag from the last successful transmission. Since a mutual relation between $\boldsymbol{\pi}$ and p_s is induced, a fixed-point iteration approach can be used to derive them jointly. The expected QoS outage probability is calculated from the steady-state probabilities of the lag from the last successful transmission, as for (5.22).

The objective function is obtained by introducing a second MC, equivalent to the first one, but where the two conditions of failed transmission and sleep mode are separated into two distinct states for each possible lag. This makes it possible to compute the expected time between two consecutive transmission attempts, as for (5.30) and (5.31).

5.3.5 *Proposed scenario*

The proposed framework is rather general and can accommodate different scenarios. In particular, it is possible to employ arbitrary signal models and transmission probability functions.

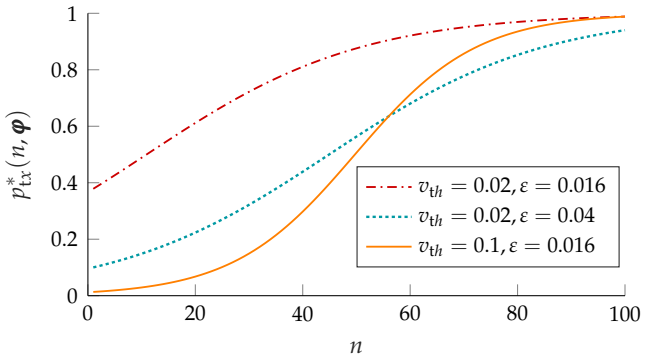


Figure 5.6: Transmission probability function resulting from the optimization procedure ($\lambda = 0.1$ devices/m^2).

Signal model. The numerical evaluation is done using the same signal model of the previous scheme (Sec. 5.2), i.e., the AR signal described in Sec. 5.1.1. It is necessary to derive the corresponding outage probability, which is needed to define the QoS constraint (5.21b).

The signal in slot $k+n$ can be expressed in terms of the signal in slot k as per Eq. (5.13). In this scenario, however, the data is not compressed before transmission, so that $x_{k+n} = \alpha^n x_k + w_n$, with the estimation error $w_n \sim \mathcal{N}(0, \sigma_n^2)$, where σ_n^2 is given in Eq. 5.14. The squared error after n steps from the last known value, $|x_{k+n} - \hat{x}_{k+n}|^2$, follows a Gamma distribution, $w_n^2 \sim \text{Gamma}(K_n, \theta_n)$, where the shape and scale parameters are $K_n = 1/2$ and $\theta_n = 2\sigma_n^2$, respectively. The outage probability (5.20) becomes

$$p_{out}(n) = 1 - F_{w_n^2}(b), \tag{5.32}$$

where $F_{w_n^2}(\cdot)$ is the CDF of the squared estimation error w_n^2 at lag n.

Transmission probability function. The proposed model assumes that the transmission probability function can be defined by a set of parameters $\boldsymbol{\varphi}$. Intuitively, the transmission probability should not decrease with the lag n in order to limit the estimation error that tends to grow with n. Furthermore, a non-decreasing probability function guarantees the convergence of the iterative process to determine the success probability p_s and the steady-state probability distribution of the MC of Fig. 5.4.

In this study, the transmission probability function is defined as a generalized sigmoid

$$p_{\text{tx}}(n; \boldsymbol{\varphi}) = \frac{1}{1 + e^{-\varphi_1(n-\varphi_2)}}, \quad n \geq 0; \tag{5.33}$$

where φ_1 defines the steepness of the curve, while φ_2 represents the horizontal shift. Notice that $\varphi_2 \leq 0$ yields a concave function. By tuning the parameters φ_1 and φ_2, the generalized sigmoid function can well approximate a number of cumulative probability

Table 5.2: Simulation parameters.

Interference and communication parameters		
Time slot duration	T	100 ms
Cell radius	R	100 m
Received power[6]	$\overline{P}_{\text{rx}}$	1 nW
Transmission bandwidth	B_W	125 kHz
Noise power	N_s	$1.25 \cdot 10^{-15}$ W
Signal model and QoS parameters		
Autoregressive model parameters	α	0.99
	σ^2	0.001
Initial value	x_0	0.8
Threshold on $\mathrm{E}_\tau\left[p_{out}(\tau)\right]$	p_{th}	0.1
Error threshold	b	0.08
Kulau et al. strategy [87]		
Maximum sleep time	$t_{\max}$	50 slots
Weighting exponent	ϕ_{bb}	2
Sliding window size	n_s	30
EDSAS [104]		
EWMA coefficient – long	ρ_{long}	0.2
EWMA coefficient – short	ρ_{short}	0.8
EWMA reset threshold	η	1

distributions, thus being particularly suitable for the considered purpose. However, we remark that the proposed framework can be applied to any other parametric probability distribution function. Fig. 5.6 shows some examples of the curve $p_{\text{tx}}(n;\boldsymbol{\varphi})$ for different QoS constraints when the node density is $\lambda = 0.1$ devices/m^2.

Solution. From (5.14) and (5.32), it is apparent that, after each successful transmission, the outage probability steadily grows in time, till the next successful transmission, which occurs after τ steps. However, the relation between τ and the optimization variable $\boldsymbol{\varphi}$ is quite complex. Also, the optimization problem (5.21) is in general not convex, so that analytical solutions cannot be found. Consequently, the solution to the problem can be found with numerical methods. In this study, we used the routines available in the MATLAB Optimization Toolbox.

5.3.6 *Numerical evaluation*

The proposed strategy was evaluated strategy by means of simulations to prove its scalability and to show the improvements compared to the state of the art. In particular, we studied the performance in terms of QoS, i.e., outage probability, and energy efficiency. The values of the parameters used in the simulations are reported in Tab. 5.2. Also, energy calculations are normalized to the cost of each joint sensing and transmission operation, implying that the normalized energy can be seen as the fraction of slots where a device is awake.

6 Note that this value allows devices at the cell edge to respect the limit of 25 mW imposed by ETSI on the transmission power for the 868 MHz band.

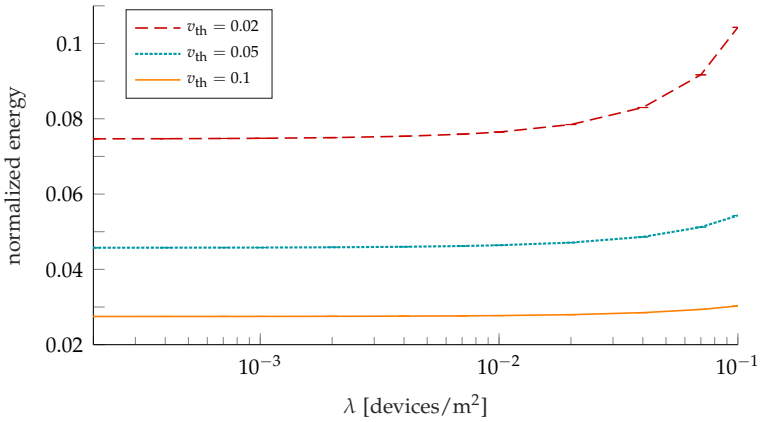


Figure 5.7: Energy consumed for increasing device density, with 95% confidence intervals.

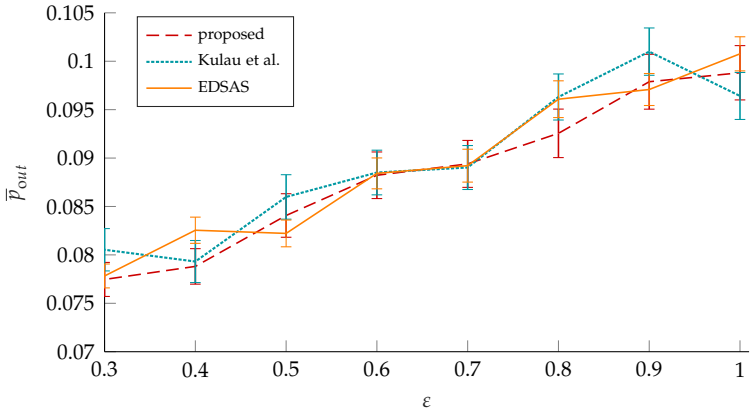


Figure 5.8: Outage probability of the considered schemes for $\lambda = 10^{-3}$ devices/m^2, with 95% confidence intervals.

Scalability on device density. To test the strategy in a massive access scenario, the outage probability was evaluated for increasing values of the devices density λ. The parameters of the AR signal used for the simulation are given in Tab. 5.2. The outage probability obtained with the simulation matches exactly the imposed threshold v_{th}, even for strict constraints. This proves that that the proposed strategy, when used with AR signals, is able to cope with the increasing device density while maintaining a QoS close to the desired value.

Fig. 5.7 shows the consumption for different values of the the threshold v_{th}. Interestingly, the amount of energy used is almost constant for the different device densities. This proves that the proposed strategy is able to scale well, since it proactively tunes the transmission probabilities in response to network congestion, thus avoiding the negative effects of collisions on the channel.

Comparison with previous strategies. The proposed strategy is compared with two other techniques in the literature that address directly our use case, and that represent

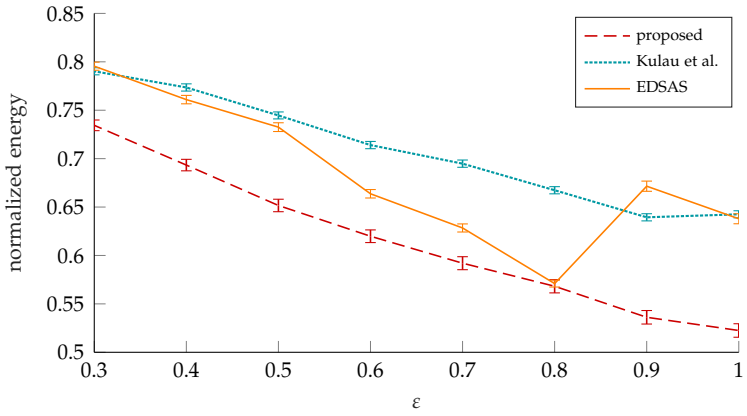


Figure 5.9: Energy consumed by the considered schemes for $\lambda = 10^{-3}$ devices/m^2, with 95% confidence intervals.

the typical approaches to this problem. The first technique is proposed by Kulau et al. [87] and has been used also to gauge the performance of the first scheme, in Sec. 5.2.4. The second strategy from the literature, named Exponential Double Smoothing-based Adaptive Sampling (EDSAS) [104], uses irregular data prediction to dynamically change the sampling rate (up to a maximum sampling interval $S_{\max}$), while maintaining the error below a threshold ε. EDSAS starts with a 1-step prediction and, as long as the prediction error stays below ε, the sampling interval k is increased by 1 (until $S_{\max}$); when the error exceeds ε, k is decremented by 1. In more detail, the strategy uses Wright's extension to the Exponential Double Sampling technique, where a k_n-step prediction at time n is calculated as $\hat{x}_{n+k_n} = L_n + k_n M_n$. The coefficients L_n and M_n are, respectively, the estimate and the trend of the signal at time n, and they are given by

$$L_n = (1 - V_n)(L_{n'} + k_{n'} M_{n'}) + V_n x_n \,; \tag{5.34}$$

$$M_n = (1 - U_n) M_{n'} + U_n (L_n - L_{n'}) / k_{n'} \,, \tag{5.35}$$

where n' is the instant when the previous sample was taken (i.e., $n = n' + k_{n'}$). Also, the normalizing factors V_n and U_n are given by

$$V_n = V_{n'} / (b_n + V_{n'}) \,; \qquad b_n = (1 - \alpha_{\mathrm{E}})^{k_{n'}} \,; \tag{5.36}$$

$$U_n = U_{n'} / (d_n + U_{n'}) \,; \qquad d_n = (1 - \beta_{\mathrm{E}})^{k_{n'}} \,, \tag{5.37}$$

and depend on the hyperparameters α_{E} and b_{E} .

The algorithm inputs an adjustment feedback based on exponentially weighted moving averages (EWMA) to minimize errors due to unpredictable events that suddenly change the estimated measurements. A long term moving average (S_{long}) and a short term moving average (S_{short}) are calculated using a standard moving average technique

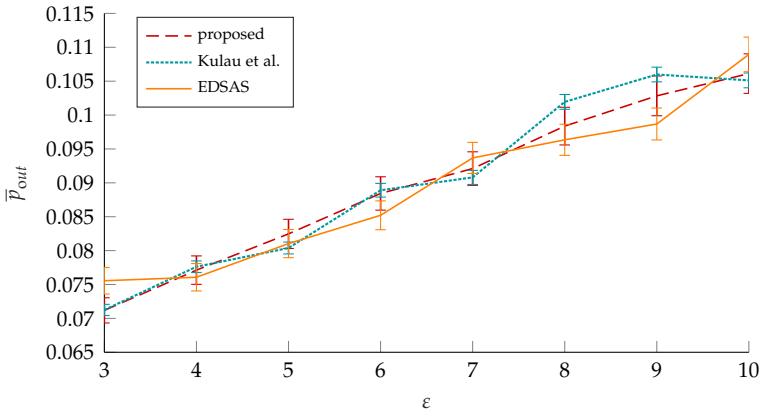


Figure 5.10: Outage probability of the considered schemes for $\lambda = 10^{-2}$ devices/m^2, with 95% confidence intervals.

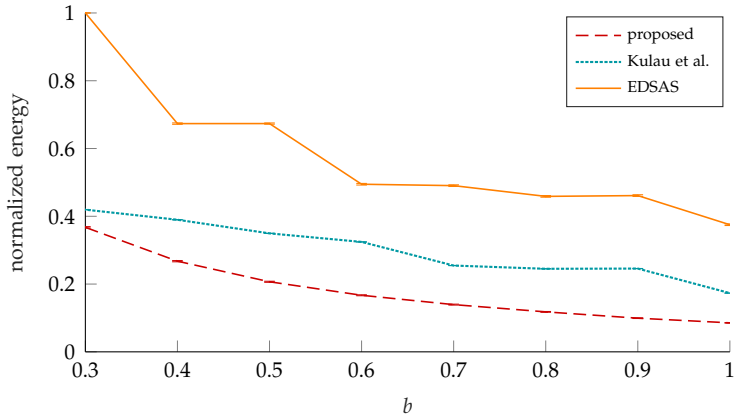


Figure 5.11: Energy consumed by the considered schemes for $\lambda = 10^{-2}$ devices/m^2, with 95% confidence intervals.

($S_n = \rho x_{n'} + (1-\rho)S_{n'}$) with the coefficient ρ being equal to ρ_{long} or ρ_{short}, respectively. The ratio $\eta = S_{long}/S_{short}$ exceeding a predefined threshold indicates a sudden change in the data, requiring the sampling interval to be reset to 1.

Fig. 5.8 shows the outage probability as b increases for $\lambda = 10^{-3}$ devices/m^2. The signal used in the numerical evaluation varies between -20 and 50, therefore the considered values of ε correspond to a relative error in the $0.4\% - 1.4\%$ range. To guarantee a fair comparison, we set b_{bb} (for the Kulau et al. strategy), α_{E}, β_{E}, and $S_{\max}$ (for EDSAS) so that the outage probability is almost the same as for our strategy. The detailed values of these parameters are reported in Tab. 5.3. Moreover, since the two above mentioned techniques are not tailored to AR signals, these simulations were performed with real world time series.[7] Note that, to use our proposed strategy with

7 Taken from the public dataset available online at `https://www.ncdc.noaa.gov/crn/qcdatasets.html`. See H. J. Diamond et al., *U.S. Climate Reference Network after one decade of operations: status and assessment*, Bull. Amer. Meteor. Soc., 94, 489-498, 2013

Table 5.3: Parameters for Kulau et al. and EDSAS strategies to yield the same error as the proposed strategy for $\lambda = 10^{-3}$ devices/m^2.

	b	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Kulau et al.	b_{bb}	9.1	8.1	7.217	6.5	4.1	4	3.94	2.78
EDSAS	α_{E}	0.006	0.5	0.54	0.5	0.5	0.52	0.55	0.4
	β_{E}	0.006	0.5	0.54	0.5	0.5	0.52	0.55	0.4
	$S_{\max}$	3	3	4	3	3	4	4	4

Table 5.4: Parameters for Kulau et al. and EDSAS strategies to yield the same error as the proposed strategy for $\lambda = 10^{-2}$ devices/m^2.

	b	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Kulau et al.	b_{bb}	9	8	7.215	6.5	4.1	4	3.94	2.78
EDSAS	α_{E}	0.999	0.98	0.45	0.5	0.5	0.2	0.15	0.996
	β_{E}	0.999	0.98	0.45	0.5	0.5	0.2	0.15	0.996
	$S_{\max}$	1	2	2	3	3	3	3	4

the real signal, it is sufficient to fit the data series to an AR model, i.e., determine the parameters α and σ^2, and then use such approximation to feed the algorithm.

Fig. 5.9 shows that the proposed strategy is able to provide the desired QoS with an energy expenditure that is significantly lower than those of the other approaches. This is because of two reasons. First, the proposed approach is *proactive*, instead of *reactive*, which means that, unlike EDSAS, we try to derive the error that the estimate will have the next time the device wakes up instead of simply increasing or decreasing the sleeping time based on the past. Secondly, we explicitly take into account the effect of interferers on the ability of the FC to estimate the time series. While the other approaches neglect the effect of collisions, so that a higher transmission rate will always result in a reduction of the estimation error (at the cost of higher energy consumption), our strategy keeps into account that a lower transmission rate yields a longer time-on-air and therefore may increase the number of collisions and, hence, eventually increase the estimation error, in particular in massive access scenarios. The oscillating behavior of EDSAS is due to the fact that [104] does not specify how to set the parameters, and thus it is necessary to manually tune the algorithm so that the outage probability (Fig. 5.8) matches that of the proposed algorithm.

5.4 Channel access scheme for integral measurements

Unlike the two previous scenarios, in this case the FC is interested in tracing the time integral of each single process x_k. For example, the integral measure may refer to the volume of fluid processed by an industrial pump, the distance covered by a fork lift

in an automated warehouse, the amount of water used to irrigate a cultivation, and so on. The proposed scheme assumes a duty-cycled operation mode, where transmissions are performed after each sampling. As in the first considered scenario, data can be compressed at the sensor nodes.

5.4.1 *Process measurement model*

In this case, besides the monitored signal x_k, it is necessary to define the integral process $y_k = \sum_{\ell=1}^{k} x_\ell$. Notice that, since $|\alpha| < 1$ (see Sec. 5.1.1), y_k is asymptotically stationary. As explained in Sec. 5.1.1, the sensor nodes can sample the monitored process x_k with a certain, maximum accuracy, and then perform a lossy compression of their measurements to reduce the size of the transmitted messages. Moreover, a device can also transmits the measurement of the integral process y_k with maximum accuracy. The correct reception of a data packet, therefore, makes it possible to completely nullify the estimate error of the integral measure y_k at the receiver, while the current value of the process will be known with an error that depends on the compression level adopted by the transmitter.

The objective is to guarantee that the cumulative error at the FC, which is affected by both the compression of the transmitted data and the estimation of the missing samples, does not exceed a given threshold. In the considered application, it makes sense to evaluate the absolute value of the cumulative error $\mathcal{E}_n$ after n slots since the last successful transmission, i.e.,

$$\mathcal{E}_n = y_n - \hat{y}_n = \sum_{k=1}^{n} (x_k - \hat{x}_k) \ , \tag{5.38}$$

with $\mathcal{E}n = 0$ for $n = 0$ (i.e., in case of consecutive successful transmissions).

Note that, under the considered assumptions, the error (5.38) has zero mean, but its variance grows with n, because of both the lack of new measurements from the sensor and the distortion that affects the last received measurement. Therefore, the probability that the error exceeds a given threshold becomes progressively higher in time, until a new packet will be correctly received, renewing the estimate process. Even if the focus is on y_n, the current measure x_k is nonetheless needed for the estimation, which is fundamental to reduce the sampling and transmission rates and, by that, save energy.

Since the error $\mathcal{E}_n$ is reset at every successful transmission, without loss of generality index 0 can be used to indicate the slot when the last message from a given sensor was received and n can be considered as the number of slots elapsed since the last successful reception. Then, by leveraging on the temporal correlation profile, it is:

$$x_n = \alpha^n x_0 + \sum_{k=1}^{n} \alpha^{n-k} u_k = \alpha^n \tilde{x}_0 - \alpha^n v(L) + \sum_{k=1}^{n} \alpha^{n-k} u_k, \tag{5.39}$$

where $\tilde{x}_0$ is the latest compressed data sample available at the FC. Considering that u_k has zero mean, the estimate with minimum Mean Squared Error (MSE) in slot n is $\hat{x}_n = \alpha^n \tilde{x}_0$. This makes the reconstruction error of that measurement equal to the sum of the distortion $\alpha^n v(L)$ and the estimation error $\sum_{k=1}^{n} \alpha^{n-k} u_k$. It follows that

$$\mathcal{E}_n = \sum_{k=1}^{n} (x_k - \hat{x}_k) = \sum_{k=1}^{n} \left[-\alpha^k v(L) + \sum_{\ell=1}^{k} \alpha^{k-\ell} u_k \right] == \sum_{k=1}^{n} -\alpha^k v(L) + \sum_{k=1}^{n} \sum_{\ell=1}^{k} \alpha^{k-\ell} u_\ell \,. \tag{5.40}$$

The first error term in (5.40) is associated to the distortion due to data compression and can be expressed as

$$\mathcal{E}'_n = \sum_{k=1}^{n} -\alpha^k v(L) = \frac{\alpha^{n+1} - \alpha}{\alpha - 1} v(L), \tag{5.41}$$

which means that $\mathcal{E}'_n \sim \mathcal{N}\left(0, \left(\frac{\alpha^{n+1}-\alpha}{\alpha-1}\right)^2 \omega^2(L)\right)$. Similarly, the second sum in (5.40) becomes

$$\mathcal{E}''_n = \sum_{k=1}^{n} \sum_{\ell=1}^{k} \alpha^{k-\ell} u_k = \sum_{\ell=1}^{n} u_\ell \sum_{k=\ell}^{n} \alpha^{k-\ell} = \sum_{\ell=1}^{n} u_\ell \frac{1 - \alpha^{n-\ell+1}}{1 - \alpha} \,. \tag{5.42}$$

The terms $\{u_\ell\}$ are zero-mean i.i.d. Gaussian r.v.s, so that $\mathcal{E}''_n$ is a zero-mean gaussian r.v. with variance

$$\sigma_e^2(n) = \sum_{\ell=1}^{n} \sigma^2 \left(\frac{1 - \alpha^{n-\ell+1}}{1 - \alpha} \right)^2 = \frac{\sigma^2}{(1-\alpha)^2} \left(n - 2\frac{\alpha(\alpha^n - 1)}{\alpha - 1} + \frac{\alpha^2(\alpha^{2n} - 1)}{\alpha^2 - 1} \right). \tag{5.43}$$

In conclusion, the cumulative error over a window of size k is $\mathcal{E}_n = \sum_{n=1}^{n} (x_n - \hat{x}_n) = \mathcal{E}'_n + \mathcal{E}''_n$, and follows a normal distribution $\mathcal{N}(0, \sigma_t^2(n))$, where

$$\sigma_t^2(n) = \sigma_e^2(n) + \left(\frac{\alpha^{n+1} - \alpha}{\alpha - 1} \right)^2 \omega^2(L), \tag{5.44}$$

for $n = 1, 2, \ldots$, and $\sigma_t^2(0) = 0$. Note that, as expected, $\sigma_t^2(n)$ is increasing with the window size n.

5.4.2 *Transmission strategy*

As done for the other schemes, the transmission strategy is derived by considering the perspective of a single node and assuming that the other devices follow the same strategy.

Sensors are off the grid and are powered by batteries with a finite initial charge. In order to limit the energy consumption, both the sampling and transmission rates should be reduced. To this purpose, it is necessary to quantify their energy demand. We assume that each sampling operation requires a constant amount of energy E_s. Moreover, since

devices are static and perform power control, the transmission power is the same for all the transmissions of a device. Since the time-on-air of each transmission is also constant, the energy consumed by a device for packet transmissions is the same for each attempt. As a consequence, minimizing the energy consumption of a node is equivalent to maximizing the duration S of the sleeping phase (under the QoS constraint).

In particular, capitalizing on both the sampling and data compression approaches described in the introduction, the goal is to determine i) the mean duration $S^\star$ of the sleeping window,[8] and ii) the size $L^\star$ of the compressed packet that maximizes the lifetime while satisfying the QoS constraint. Clearly, both decisions i) and ii) induce some tradeoffs between energy efficiency and accuracy of the monitoring service at the FC. A larger sleeping window corresponds to fewer transmissions and therefore less energy consumption and interference but, on the other hand, leads to higher reconstruction errors of the monitored phenomena as most of the data need to be estimated by the FC. Vice versa, a larger packet size L reduces the reconstruction error because data is less compressed, but reduces the success probability since it requires a larger SINR threshold. The two tradeoffs are intertwined: since a larger L results in a reduced success probability, more transmissions are needed for a given QoS, and the sleeping window needs to be smaller.

As mentioned, the reconstruction error $\mathcal{E}_n$ is a normal r.v., therefore its magnitude, $|\mathcal{E}_n|$, is half-normally distributed with scale parameter $\sigma_t(n)$. As discussed in Sec. 5.1.1, the reconstruction error is reset at every successful transmission, since the device also sends the integral measurement. As a consequence, the parameter $\sigma_t(n)$ follows a sawtooth pattern that renews itself at each successful transmission, i.e., every W slots (the time between two consecutive successful transmissions, which is stochastic).

This means that the QoS constraint can be defined by focusing on the error in a window of length W. More specifically, let consider the error at the end of a window, $\mathcal{E}_W$; the QoS can be defined as an upper threshold p_{th} on the mean probability that $|\mathcal{E}_W|$ exceeds a given value b.

The optimization problem can then be formulated as follows

$$S^\star \triangleq \max_{L \in \mathcal{L}} S(L)\,, \tag{5.45a}$$

$$\text{subject to:} \qquad \mathrm{E}\left[\Pr\left(|\mathcal{E}_{W-1}| > b\right)\right] < p_{th}\,, \tag{5.45b}$$

where the expectation is taken over the statistical distribution of W, while $S(L)$ is the mean sleeping period when the selected packet size is L. The sleeping periods are

8 If a device has an additional sensor that provides the integrated measure, it can avoid sensing the environment during the sleeping phase, otherwise it needs to keep sensing even during this phase. This does not impact the optimization procedure, since the sensing energy is a constant.

Algorithm 4 Transmission strategy

1: Initialize $\mathbf{S} \leftarrow$ vector of size $|\mathcal{L}|$ ▷ Contains $S(L)\ \forall\, L$
2: **for** $L \in \mathcal{L}$ **do**
3: Set $p_s(L) = 1$
4: **while** S has not converged **do**
5: $S \leftarrow \max\{S(L) : \text{cond. (5.10) holds true}\}$
6: $p_s(L) \leftarrow$ eq. (5.8) with $\mathtt{P} = 1/S$
7: $\mathbf{S}(L) \leftarrow S$
8: $S^\star = \max \mathbf{S}(L)$, $L^\star = \text{argmax}\mathbf{S}(L)$

assumed to be i.i.d. geometric r.v.s with parameter $1/S(L)$. Moreover, considering that the number of trials before success is also geometrically distributed with parameter $p_s(L)$, the distribution of W turns out to be geometric, with parameter $p_{tx} = p_s(L)/S(L)$. Therefore, the condition (5.45b) can be expressed as

$$\sum_{w=1}^{\infty} p_{tx} \left(1 - p_{tx}\right)^{w-1} Q_{\text{hf}}\left(b; \sigma_t(w-1)\right) < p_{th}\,; \tag{5.46}$$

where $Q_{\text{hf}}(\cdot)$ is the complementary cumulative function of the half-normal distribution and $\sigma_t(\cdot)$ is the square root of the variance given by (5.44), with $\sigma_t(0) = 0$.

$S^\star$ (and the associated $L^\star$) can then be determined using an iterative approach, which is described in Algorithm 4. For each possible $L \in \mathcal{L}$, the corresponding optimal mean sleeping duration $S(L)$ is computed through an alternated optimization of the duty cycle and its corresponding success transmission probability, until convergence. Then, $L^\star = \underset{S}{\text{argmax}}(L)$, which yields $S^\star = S(L^\star)$ (Line 8). The iterative procedure to derive $S(L)$ for a given L corresponds to the instructions in the `while` cycle in Algorithm 4. The success probability is initially set equal to 1, as if there were no interference; then the corresponding mean sleeping duration S, i.e., the one that satisfies the QoS requirement (5.10) when $p_s(L) = 1$ (Line 3), is determined. By adopting a mean sleeping period S, however, the success probability will actually be lower than 1 because of the interference caused by the different nodes, so that the QoS constraint will not be satisfied. The value of $p_s(L)$ for the current value of the mean sleeping period S is then updated by evaluating (5.8) with $\mathtt{P} = 1/S$ (Line 6), so that $\lambda_s = \lambda/S$, where λ is the density of all nodes (active and not) . The procedure is repeated iteratively until convergence (Lines 4-6).

5.4.3 *Numerical evaluation*

Fig. 5.12 shows the optimal value of the mean sleeping period, $S^\star$, when varying the value of b, and for two values of the node density λ_s. It can be observed that the

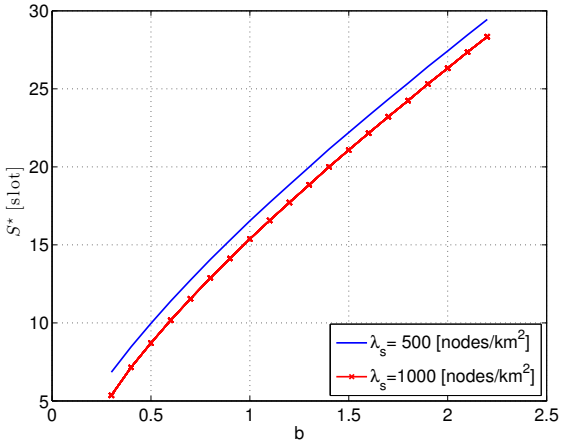


Figure 5.12: Optimal mean sleeping period $S^\star$ vs. b, for different node densities λ_s.

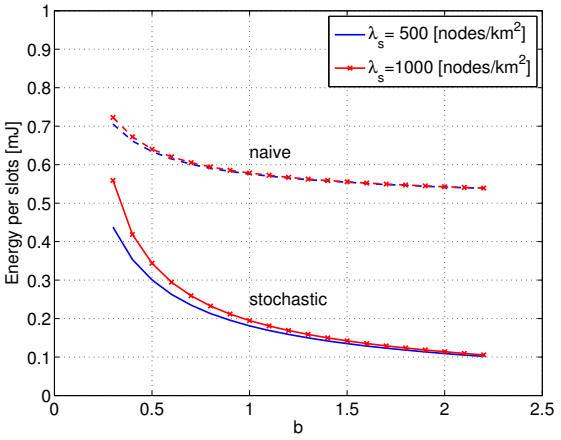


Figure 5.13: Average energy used per slot vs. b for different node densities λ_s.

Table 5.5: Simulation parameters

Interference and communication parameters		
Slot duration	T	0.1 s
Density of sensor devices	λ	$\{500, 1000\}$ devices/km^2
Cell radius	R	500 m
Received power[9]	$\overline{P}_{\text{rx}}$	4 nW
Transmission bandwidth	B_W	125 kHz
Noise power	N_s	$2.5 \cdot 10^{-15}$ W
Sensing energy	E_s	495 μJ
Signal model and QoS parameters		
Autoregressive model parameters	α	0.99
	σ^2	0.001
Initial value	x_0	0.8
Minimum packet size	L_0	24 bits
Maximum packet size	$L_{\max}$	48 bit
Compression error power	$\omega(L)$	$a = b = 0.05$
Shannon gap coefficient	β	1
Threshold on E $\left[\Pr\left(\mathcal{E}_n > b \,\middle|\, S(L)\right)\right]$	p_{th}	0.3

sleep period grows with b, as expected since the QoS constraint becomes progressively less strict, thus allowing for less frequent transmissions. Furthermore, the mean sleep duration decreases for higher node densities, in order to counteract the larger packet collision probability. We observe that, by further increasing the node density, the QoS constraint can no longer be guaranteed for smaller values of b.

To better assess the performance of the proposed strategy, we compare it to a *naive* approach where the device senses the data at each time slot (consuming a certain amount of energy E_s) and transmits it if the absolute error $|\mathcal{E}_n|$ is larger than a given threshold $\rho(b)$. In order to get similar results between the proposed and the naive strategies in

9 Note that this value allows devices at the cell edge to use the ETSI imposed limit of 25 mW on the transmission power for the 868 MHz band.

terms of QoS (i.e., equal $\Pr(|\mathcal{E}_n| > b)$), $\rho(b)$ grows from 1.7 to 1.95 as b is varied from 0.3 to 2.2. The simulation parameters are reported in Tab. 5.5.

Given that both strategies satisfy the QoS constraint, both of them can be used in the described scenario. However, because of the energy constraints, the performance must also be assessed in terms of energy efficiency. We define the energy efficiency b as the overall average energy consumption rate, i.e., the mean energy spent by the nodes in one slot. The energy efficiency of the proposed method can be easily computed as $\epsilon = \frac{\mathrm{P_{tx}}T+E_s}{S^\star}$ The energy efficiency of the naive protocol, instead, cannot be easily determined in mathematical form, and is evaluated only through simulations. The comparison is shown in Fig. 5.13, where it can be seen that the naive strategy requires a much larger amount of energy, mainly due to the continuous sensing. This is avoided by the proposed strategy, which samples the signal more sporadically, thus saving energy, while guaranteeing the same QoS level of the naive protocol.

5.5 Lesson learned

Random access schemes can yield a better resource utilization than scheduled ones in massive access scenarios, but require to address novel challenges. Transmissions are prone to interference from other users and collisions waste energy, thus designing an efficient access strategy is of utmost importance. In this study, we proposed three different random access schemes aimed at providing an accurate estimate of the monitored signal at the FC while maximizing the energy efficiency, under three different scenarios. The proposed strategies make use of a combination of the three types of compression typically adopted in standard approaches.

Unlike most state-of-the-art techniques, we also consider the role played by interference, as collisions affect both the QoS and the energy consumption, especially in dense scenarios. This allows one to obtain better performance than state-of-the-art algorithms, as shown by the numerical evaluations. It is evident that this issue plays a major role in the system performance and needs to be included in the protocol design.

The proposed model can serve as starting point for more realistic frameworks that consider more complex signal models and possibly include spatial correlation among devices, which may further reduce the number of transmissions needed to provide an accurate estimate at the FC.

6

Energy-depleting jamming attacks

IoT devices might not only have to deal with the environment they are placed in, but also with intelligent attackers who want to disrupt their functionality or disable them outright. The broadcast nature of wireless transmissions makes them highly vulnerable to jamming attacks, which may become a serious threat in IoT networks of battery-powered nodes, as attackers can disrupt packet delivery and significantly reduce the lifetime of the nodes. This chapter studies an active defense scenario in which an energy-limited node uses power control to defend itself from a malicious attacker, whose energy constraints may not be known to the defender. The interaction between the two nodes is modeled as an asymmetric Bayesian game where the victim has incomplete information about the attacker. The optimal Bayesian strategies are derived for both the defender and the attacker; they may serve as guidelines to develop efficient heuristics that are less computationally expensive than the optimal strategies. For example, this work proposes a neural network-based learning method that allows the node to effectively defend itself from the jamming with a significantly reduced computational load. The results of the model highlight the trade-off between node lifetime and communication reliability and the importance of an intelligent defense from jamming attacks.

This work has been partly presented in [105] and realized in collaboration with Federico Chiariotti.

6.1 The jamming problem in IoT networks

An attacker can interfere with a sensor network's communication by jamming the wireless channel, creating interference and forcing retransmissions. Jamming includes a wide range of attacks, which may differ in their spectral pattern (single-tone, multi-tone or partial jamming), and timing pattern, since jammers may continuously transmit, randomly alternate sleeping and jamming phases, or send a jam signal only when some traffic is sensed over the channel [106]. In this study, the focus is on the latter case, which is commonly denoted as *reactive jamming*. The energy-limited nature of IoT nodes is also

an important issue from a security standpoint: for battery-powered IoT devices, jamming is not just a Denial of Service (DoS) attack, but it also decreases the network lifetime, as the nodes are forced to deplete their batteries either in unsuccessful transmissions or trying to actively fight the interference.

Jamming is a simple though powerful attack and has been widely studied in the literature. The idea of using it to deplete wireless nodes' batteries is more than a decade old, as duty-cycle based protocols have been known to be vulnerable to this kind of attacks since the early 2000s. Researchers have found several highly effective jamming strategies for well-known WSNs protocols [107, 108, 109]. Jamming attacks on IEEE 802.15.4-based IoT networks have been thoroughly investigated [110, 111], and many attacks on that technology aiming at depleting the devices' batteries have been proved to be practicable with little effort [112]. A comprehensive survey of jamming and defensive protocols in WSNs can be found in [113]. However, these attacks are highly dependent on the specific protocol that the nodes use: a more general mathematical analysis of jamming attacks can be performed using a game theoretical approach, which models an adversarial situation in which independent agents with conflicting objectives try to outsmart each other and maximize their own payoff.

Non-cooperative games have been widely used to model the relation between the jammer and the victim [114, 115]. Typically, the payoff functions of the two players concern throughput [116], Signal-to-Noise Ratio (SNR) [117], or packet-drop ratio [118], and only seldom take into account energy. However, energy constraints are a significant vulnerability in IoT and WSNs and cannot be neglected. The traditional approach in the literature is to model a passive victim running a duty cycling algorithm to save energy [119], but there are also works that study active defense strategies[118, 120]. In [121], the two players have temporal energy limitations that model the maximum amount of energy that can be used over a period of time without causing overheating; thus, energy consumption plays a role in the short term, but the total device lifetime is not considered. In [122], the authors develop optimal steady-state strategies for the legitimate network to detect the attack, and the detection accuracy is balanced with the energy costs of monitoring. In [123] the total energy constraint on the jammer is captured by a limit on the number of transmissions that it can perform. A finite energy availability for both nodes is considered in [124], where jamming is modeled as a zero-sum finite-horizon stochastic game with deterministic transitions and is solved by means of dynamic programming. The energy consumption depends on the choices made by the players, which can decide whether to transmit or to sleep, the transmission power, and which channel(s) to use. One of the main differences between [124] and our

paper is that it considers channel hopping as a defense mechanism, while our model accounts for packet retransmissions.

Several works in the literature assume that both players have full knowledge about their opponent, because this allows one to optimally solve the game, e.g., by means of dynamic programming. Obtaining the optimal solution under incomplete knowledge assumptions requires much more effort, as the complexity of the problem increases significantly. However, some works in the literature include uncertainty when modeling jamming attacks, especially in the context of cognitive radios [125, 126]. In [127], an OFDM network does not know whether it will be under jamming attack, and the authors investigate how this uncertainty affects the anti-jamming strategy. In [128], the optimization function is represented by the SINR, and the authors find the optimal strategies in closed-form for two different levels of knowledge available to the legitimate transmitter.

Finally, some works consider distributed systems and network-level attacks: [129] considers a wireless network where nodes can be of two types, namely selfish and malicious, and have only incomplete information about the other transmitters' types. Further, each type is characterized by a specific utility function that combines throughput and energy consumption. A similar model is studied in [130], where the users of a distributed wireless network do not have complete information about the other nodes' identities and a Bayesian game is studied to quantify the expected performance under jamming attacks with different network uncertainties.

6.1.1 *The proposed model*

This study exploits game theory to model a smart IoT node defending itself from a similarly energy-constrained jammer. Both nodes are assumed to be rational actors and their interaction can be modeled as a zero-sum game (i.e., a completely adversarial game in which each gain for one player is balanced by a loss for the other), whose mathematical properties can be exploited to efficiently find a solution. The utility functions balance lifetime and throughput in a weighed sum. The problem is solved firstly under the assumption of full information available to both players and then for two different incomplete information scenarios. In fact, the development of cognitive radio techniques enables smart jammers able to adapt their attack strategies to the victims' communication pattern [131]. It thus makes sense to consider also asymmetric scenarios where the jammer has full information while the legitimate transmitter has only limited knowledge about the energy availability and strategies of the attacker. In this case, the jamming attack is modeled as a Bayesian game where the transmitter has a belief on the attacker's state and updates it based on the feedback information.

The drawback of the Bayesian approach is its computational load, so that it makes the optimal Bayesian strategies to be more suitable as guidelines to develop and gauge the performance of more lightweight heuristics and suboptimal approaches, rather than to be run by the energy constrained devices. In this sense, this study also investigates the use of a neural network through which the transmitter can approximate the iterated best response algorithm, choosing the best strategy to play against the jammer when it has no knowledge about the jammer's energy availability without having to run through the whole strategy iteration. The pre-trained network can be implemented on a low-power node, as computing the output of a neural network without any back-propagation is not computationally expensive. This development is critical to allow low-power nodes to behave intelligently without consuming more power computing optimal strategies than they save by using them.

Although the approaches used to solve the game are well-known and validated, their use in IoT scenarios is still mostly limited to toy examples because of the complexity of finding Nash Equilibria. The scenario considered in here includes constraints, retransmissions, power control, and uncertainty about the attacker's capabilities. As low-power nodes become more pervasive, jamming attacks and sabotages can be expected to become more common, and implementing an optimal defense strategy against a smart attacker can significantly increase the resilience of the IoT to this type of threats. To the best of our knowledge, this model is more realistic than those in the relevant literature, and the good performance of the learning algorithm makes it possible to actually use the results in a real network.

6.2 Game-theoretic model

This study considers a monitoring application where a sensor node, namely the *communicator (C)*, periodically reports some data to another user in the network, and a malicious node, the *jammer (J)*, tries to disrupt this communication. The interaction between the two nodes is modeled as a non-cooperative game G between rational players. The model is flexible and can accommodate disparate scenarios by properly setting its parameters and payoff functions.

Both nodes are battery-powered, and the initial charge of their battery $B_{i,0} \in \mathbb{N}$, $i \in \{C, J\}$ is the only available energy, which is discretized into integer energy quanta. The battery level of a node thus takes values in the set $\mathcal{B}_i = \{0, 1, \ldots, B_{i,0}\}$, $i \in \{C, J\}$. The game G starts with both players having maximum battery charge and consists of a series of rounds $m = 0, 1, \ldots, M$: in each subgame G_m, the legitimate transmitter C spends $E_{C,m}$ energy units to send the packet, while the attacker J consumes $E_{J,m}$ energy quanta to disrupt the communication. The whole game ends when C has an empty battery, $B_{C,M} =$

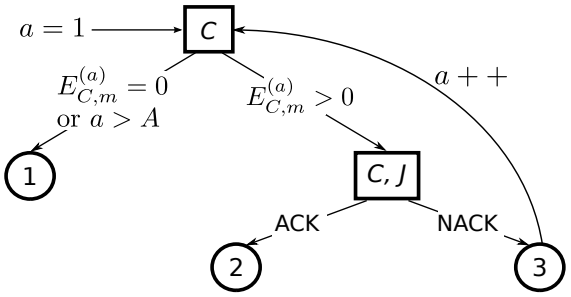


Figure 6.1: Representation of the m-th subgame in dynamic form.

0. If the battery of the jammer empties first, C keeps 'playing' against the communication channel. The system state at stage m is hence $S_m \triangleq (B_{C,m}, B_{J,m}) \in \mathcal{B}_C \times \mathcal{B}_J$.

Notation. The realization of a generic r.v. x is denotes as x, its estimate as $\hat{\mathrm{x}}$, and the realization of the estimate as $\hat{x}$. Furthermore, for writing convenience, the probabilities $\Pr(\mathrm{x} = x)$ and $\Pr(\mathrm{x} = x|\mathrm{y} = y)$ for any pair of r.v.s x and y will be compactly written as $p(x)$ and $p(x|y)$, respectively (whenever the particular r.v.s to which they refer can be clearly identified from the context).

6.2.1 *Structure of a subgame*

Each subgame models the attempt made by C to transmit a packet. Its temporal duration is limited by the length of the reporting window, which defines the maximum number of attempts A that can be made for each packet. Accordingly, if the packet is not correctly delivered within A transmission attempts, it is discarded and a new subgame starts. Fig. 6.1 shows the dynamics of a subgame, which is a Stackelberg entry game [132] with a random component.

Index $a \in \{1, \ldots, A\}$ represents the current transmission attempt for the considered subgame. At each attempt, node C decides how much energy $E^{(a)}_{C,m}$ to spend for sending the packet. This energy might be spent by using power control to increase the transmission power and, consequently, the SINR, or by using a different modulation and coding scheme to increase the robustness to noise.

Each attempt can have three possible outcomes, depending on the choices made by C and J, and the conditions of the channel, which is modeled stochastically.

- *Drop (case 1)* This happens either because C sets $E^{(a)}_{C,m} = 0$, which implies that C gives up on that packet and chooses to drop it, or the maximum number A of attempts has been reached. In both cases, the subgame ends.
- *ACK (case 2)* The energy $E^{(a)}_{C,m} > 0$ was enough to contrast the channel impairments and the jamming attack, and the packet is correctly received. This ends the subgame.

- *NACK (case 3)* In this case, the energy chosen by C was not enough and the packet is lost. The attempt counter a increases by one and C makes its choice again, as long as the maximum number of retransmission attempts A is not exceeded and the energy level in the battery is positive, otherwise we fall under case 1.

Let $a_m \leq A$ be the final value of a, i.e., the number of transmission attempts that C made at the end of the m-th subgame. In order to reduce computational complexity and save power, the strategy for both nodes is determined at the beginning of a subgame and is not recalculated after each attempt.

The actions available to the players, i.e., the transmission energy levels $E_{i,m}^{(a)}$, $i \in \{C, J\}$, $a \in \{1, \ldots, a_m\}$, are limited by the current battery level and the maximum energy that can be used for transmission, $E_{i,\max}$, which depends on the maximum available transmission power and overheating considerations. Hence, $E_{i,m}^{(a)} \in \mathcal{E}_{i,m} \triangleq \{0, 1, \ldots, \min(B_{i,m}, E_{i,\max})\}$. Also, the action strategy of the subgame is feasible solely if the following condition is satisfied:

$$E_{i,m} = \sum_{a=1}^{a_m} E_{i,m}^{(a)} \leq B_{i,m}, \quad i \in \{C, J\}, \tag{6.1}$$

which means that the total energy used by node i during the m-th subgame cannot exceed the available energy.

As outlined in Sec. 6.1.1, the players have a twofold goal and their payoffs are convex combinations of monotonic functions of the energy consumption of node C and the Packet Delivery Ratio (PDR), respectively. The main objective of the players can be shifted between saving energy and delivering more packets by tuning the weights $\alpha_i \in [0, 1]$, $i \in \{C, J\}$. The payoffs of the two players in a single subgame are:

$$u_{C,m} = (1 - \alpha_C) f_C(a_m)\, \chi_{C,m} + \alpha_C \sum_{a=1}^{a_m} g_C\left(E_{C,m}^{(a)}\right) \tag{6.2a}$$

$$u_{J,m} = (1 - \alpha_J)\, f_J(a_m) \;+\; \alpha_J \sum_{a=1}^{a_m} g_J\left(E_{C,m}^{(a)}\right) \tag{6.2b}$$

The first term of both equations concerns the *outcome of the communication*. In particular, the indicator term $\chi_{C,m}$ is equal to one if the subgame m ends with a successful transmission, so that the function f_C only rewards the communicator for successfully transmitting the packet. Vice versa, the function f_J rewards the jammer for disturbing the transmission. In the general formulation in (6.2b), the term $\chi_{C,m}$ related to the transmission outcome is not explicitly present, because the jammer may have a reward also for simply delaying the packet delivery (i.e., C succeeds in the communication but not at the first attempt); the particular shape of f_J depends on the application scenario.

However, the shape of f_J affects the complexity of the solution: in this work, the solution is limited to constant-sum games, which have particular properties that simplify the solution.

The second part of the payoff equations is related to *energy*. The function g_C gives C a penalty for consuming energy, while g_J rewards J for the energy spent by C to contrast the attack. Notice that the jammer's energy is not directly considered in the utility function in order to keep the game symmetric. However, it still plays a role in the whole game evolution, as will be clarified in the following section.

It is worth to highlight that the definition of the functions does not affect the validity of the solution methods proposed in the next sections: while the solution proposed here requires the game to be equivalent to a zero-sum game, any definition of $u_{C,m}$ and $u_{J,m}$ that satisfies this property is solvable without modifications. However, in order for the solution to be easily computable, the jammer's objective must be assumed as purely adversarial against the communicator, i.e., $\alpha_J = \alpha_C \triangleq \alpha$. Nonetheless, this is a realistic assumption, as the objective of the jammer will be to disrupt the network and cause as much damage as possible to the communicator, so that the choice $\alpha_J = \alpha_C$ is reasonable.

Finally, this model does not consider detection to be an issue, as the jammer only needs to block the communicator's transmissions. A future version of the work might include secrecy as one of the jammer's goals.

6.2.2 *The full jamming game*

The general solution of the whole game G maximizes a long-term payoff function with a given time horizon T. The payoffs in the multistage game G at stage m are given by:

$$U_i(m) = \sum_{\tau=m}^{m+T-1} \lambda^{\tau-m} u_{i,\tau}, \quad i \in \{C, J\}, \tag{6.3}$$

where $\lambda \in [0, 1]$ is a future exponential discounting factor, as often considered in the literature on game theory [133], and T is the length of the payoff horizon, i.e., the number of future subgames to take into account in the reward calculations. If T is finite, λ can be safely set to 1 with no convergence issues, while the infinite horizon case requires $\lambda < 1$. The payoff functions define the goals of the players, and affect the strategy they use to achieve them, and, by that, the actual duration of the game. Notice that the optimal solution of the whole game G does not necessarily coincide with the greedy strategy that maximizes the payoff for each subgame independently, unless the time horizon window is $T = 1$.

Table 6.1: Parameters of the Neural Network.

Parameter	Meaning
J	Jammer node
C	Communicator node
$E_{i,\max}$	Maximum transmission energy for player i
$B_{i,\max}$	Maximum battery size for player i
a_m	Index of the last attempt in the m-th subgame
$\chi_{C,m}$	Indicator variable of the success of the transmission in the m-th subgame
$E_{i,m}^{(a)}$	Move for player i in the a-th attempt of the m-th subgame
$B_{i,m}$	Battery level of player i at the beginning of the m-th subgame
$u_{i,m}$	Payoff for player i at the end of the m-th subgame
$U_{i,m}$	Long-term reward for player i in the m-th subgame
$s_{i,m}^*$	Optimal (pure) strategy for player i in the m-th subgame
$\mathbf{\Phi}_{i,m}^*$	Optimal (mixed) strategy for player i in the m-th subgame
$\hat{B}_{J,m}$	Estimated battery level of the jammer at the beginning of the m-th subgame
d_m	Feedback to the communicator in the m-th subgame

6.3 Complete information case

First of all, a full information game is considered, where both players are aware of the type $B_{i,m}$ and moves $E_{i,m}^{(a)}$ of the opponent $i \in \{C, J\}$ at all times m. This assumption, although unrealistic, allows one to obtain closed-form results, which play a fundamental role in the derivation of the strategies for the incomplete information scenario and in the assessment of their performance.

6.3.1 *System parameters*

The game model described in Sec. 6.2 is flexible and can be readily adapted to different scenarios, by properly choosing the payoff functions (6.2) and (6.3). In the following, a possible configuration is specified. The cost functions in (6.2) are set as:

$$f_C(a) = \gamma(a), \qquad f_J(a) = 1 - \gamma(a) + \bar{\chi}_C, \tag{6.4}$$

$$g_C(E_{C,m}^{(a)}) = -\frac{E_{C,m}^{(a)}}{E_{C,\max}+1}, \qquad g_J(E_{C,m}^{(a)}) = \frac{E_{C,m}^{(a)}}{E_{C,\max}+1}, \tag{6.5}$$

where $\gamma(a_m)$ is an arbitrary monotonically non-increasing function with output in $[0, 1]$, which accounts for the delay introduced by retransmissions (it is $a \in \{1, \ldots, A\}$). In particular, if $\gamma(a)$ is strictly decreasing, C gets a penalty for the latency induced by each retransmission even if the communication succeeds, and J receives a reward for delaying the packet delivery. The term $\bar{\chi}_C$ is equal to 1 if J disrupts C's communication, and to 0 otherwise. Notice that, according to (6.2), the rewards obtained by the two players are computed only at the end of each round, i.e., for $a = a_m$. Function $g_C(\cdot)$ penalizes C for consuming energy; the amount of energy actually used is normalized to the maximum

amount of energy that can be used. The additional term 1 in the denominator is arbitrary and ensures that the absolute values of $g_C(\cdot)$ and $g_J(\cdot)$ are always smaller than 1: without it, any strategy, including a transmission with the maximum energy, would be dominated by not transmitting, effectively reducing the strategy space. The reward given to node J as defined in $g_J(\cdot)$ is exactly the opposite. Notice that the energy spent by node J, although not explicitly present in the payoff definition for the single subgame, plays a major role in the complete game, because it has a direct impact on C's strategy and on its packets' error probability. Therefore, J's moves affect the depletion speed of C's battery and the game duration.

Another important function that needs to be defined is the packet error probability, P_e, which gives the probability that a packet sent by C is not correctly received by the intended destination. The packet error function affects the numerical results, but does not impact the validity of the model. The function used in the simulated scenario is $P_e(E_C, E_J) = e^{-\frac{E_C}{E_J+1}}$ for a given moveset (E_C, E_J). The uncertainty given by the random nature of the channel is modeled as a Bernoulli random variable $\chi_{C,m} \sim \mathcal{B}(1 - P_e)$, which represents the correct reception of the packet. The expression chosen for the error probability is consistent with real fading models, as shown in [134]. However, the proposed model is still valid if using any function $P_e : \{1, \ldots, E_{C,\max}\} \times \{0, \ldots, E_{J,\max}\} \longrightarrow [0, 1]$ such that $P_e(E_c, E_j)$ is monotonically decreasing and convex in E_c and increasing and concave in E_j, vanishing as $E_c \to \infty$ and approaching 1 as $E_j \to \infty$.

6.3.2 *Dynamic programming solution*

Under the assumption of full information, the expected payoffs of future subgames can be calculated exactly with dynamic programming. In particular, the state value $U_i(B_C, B_J)$ depends on the state value of all possible past subgames. Since the battery charge cannot increase, the possible past subgames are the states $(a, b) \in \mathcal{B}_C \times \mathcal{B}_J$ such that either $(a < B_C) \wedge (b \leq B_J)$ or $(a, b) = (B_C, B_J) \wedge a \neq 0$; $i \in \{C, J\}$, as J is a reactive jammer and does not consume energy if the communicator drops the current packet (i.e., $a = 0$). An illustrative example for $B_{C,0} = B_{J,0} = 2$ is given in Fig. 6.2. Notice that transitions between states are allowed from bottom to top and from right to left.

In each state, C and J play the subgame described in Sec. 6.2.1, and choose how much energy to use at each attempt to transmit or jam the legitimate packet, respectively. The action chosen by player $i \in \{C, J\}$ is called *strategy* and denoted as s_i; the strategy space is $\{0, E_{i,\max}\}^A$. For each state it is possible determine the Nash Equilibrium (NE), which is the combination of strategies such that neither of the players has an incentive to deviate unilaterally from its choice. The NE is given by the pair of strategies (s_C^*, s_J^*), which are mutual best responses [135]. The payoffs of the full game can be computed exploiting

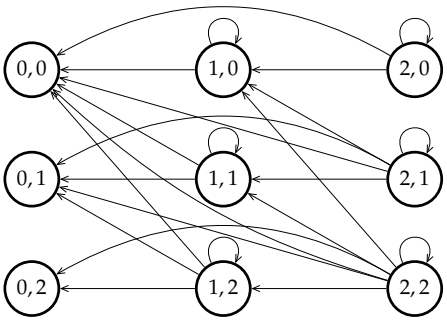


Figure 6.2: Example of state transitions for the multistage game.

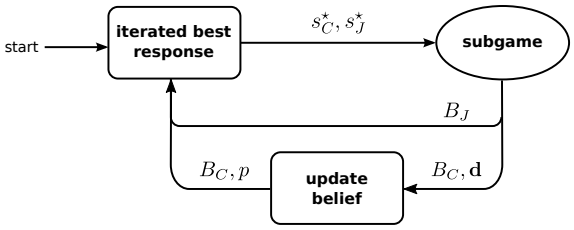


Figure 6.3: Block diagram of the Bayesian game.

dynamic programming, as given by. (6.3). The NE can be "pure", which means that the strategies (s_C^*, s_J^*) are deterministic, or "mixed", when the optimal strategy of at least one player is comprised of multiple pure strategies with an associated probability. The Lemke-Howson algorithm [136] allows one to compute the mixed strategies. The game dynamics are obtained deriving the optimal strategies for every possible state, which corresponds to determining the path from the initial state (at the right-hand side of Fig. 6.2) to an ending state of type $(0, \cdot)$. A more detailed explanation of the dynamic programming solution of the game can be found in [105].

6.4 The Bayesian jamming game

Here the full information assumption is relaxed, so that the jamming attack is modeled as an incomplete information game, in which the battery state of the jammer is unknown to the communicator. Only the case of time window $T = 1$ is considered, as it showed the best results in terms of delivered packets in the complete information case (see [105]).

Fig. 6.3 represents the scheme of the Bayesian game. At each subgame, the optimal strategies $(s_C^\star, s_J^\star)$ of the two rational players are derived by means of iterated best response. The outcome of the subgame (see Sec. 6.2.1) depends on such strategies and the stochastic channel conditions. When playing the subgame, C obtains some information, denoted as $\mathbf{d}$, about the jammer. This feedback is used by C to update its belief about the current battery level of J. This is repeated until the battery of the communicator is depleted.

To ensure a fair comparison with the results of the full information scenario, the parameters are the same as in Sec. 6.3.1.

6.4.1 *Solution of the subgame*

In the incomplete information game, player C has a prior belief $p(\hat{B}_{J,m})$ on J's battery state, for each possible battery level between 0 and $B_{J,\max}$.[1] The belief is updated after every subgame iteration, according to the feedback $\mathbf{d}$ that C receives. The two players need to choose an energy level for any possible attempt in the subgame, hence their strategy spaces are $\{0, E_{C,\max}\}^A$ and $\{0, E_{J,\max}\}^A$, respectively. The Bayesian Nash Equilibrium (BNE) [137], i.e., the pair of optimal strategies for the considered subgame, is reached when both players maximize their expected payoff and thus have no incentive to choose another possible action. The best response strategy for both players in the m-th subgame are defined as:

$$s^*_{C,m}(s_{J,m}, B_{C,m}, p(\hat{B}_{J,m})) = \underset{s_{C,m}}{\operatorname{argmax}} \sum_{\hat{B}_J=0}^{B_{J,\max}} p(s_{J,m}|\hat{B}_J)p(\hat{B}_J)\mathrm{E}\left[U_C(m)|s_{C,m}, s_{J,m}, B_{C,m}, \hat{B}_J\right] \tag{6.6}$$

$$s^*_{J,m}(s_{C,m}, B_{C,m}, B_{J,m}) = \underset{s_{J,m}}{\operatorname{argmax}}\, \mathrm{E}\left[U_J(m)\middle|s_{C,m}, s_{J,m}, B_{C,m}, B_{J,m}\right] \tag{6.7}$$

The jammer obviously has an advantage: its complete knowledge of the state allows it to be aware of C's optimal strategy and play accordingly, while C has to act on incomplete information. In fact, the communicator chooses the strategy that maximizes its expected payoff $\mathrm{E}\left[U_C(m)|s_{C,m}, s_{J,m}\right]$ weighed by the probability that J chooses action $s_{J,m}$ when its battery level is $B_{J,m} = \hat{B}_{J,m}$, and the probability distribution of J's battery level is given by C's current belief. As the game goes on, C's belief on J's battery state will become more accurate, reducing J's advantage.

The structure of the payoff functions is such that each subgame can be represented as the sum of a dummy game (i.e., a game whose result is not affected by the players' strategies) and a zero-sum game. Thanks to the Minimax theorem [138], all zero-sum games can be solved by using the iterated best response algorithm, which is far less computationally demanding than other strategies to compute Nash equilibria. The algorithm itself is very intuitive: starting from any strategy $s_{C,m}$, the best response $s^*_{J,m}(s_{C,m})$ is computed. Then, it is J's strategy that is fixed, and C computes its best response. If a pure BNE exists, the iterated best response algorithm will converge to it with probability 1. However, the game might not have a pure BNE; mixed strategies are

1 The value $B_{J,\max}$ is chosen by C, and may be infinite if it believes the jammer not to be energy constrained.

not a single set of moves, but a distribution over the strategy space $\{0, E_{C,\max}\}^A$. In this case, the algorithm can be generalized using

$$\boldsymbol{\Phi}^*_{C,m}(\boldsymbol{\Phi}_{J,m}, B_{C,m}, p(\hat{B}_{J,m})) = \operatorname*{argmax}_{\boldsymbol{\Phi}_{C,m}} \sum_{\hat{B}_J=0}^{B_{J,\max}} p(s_{J,m}|\hat{B}_J) \sum_{s_{C,m}} \sum_{s_{J,m}} \Phi_{C,m}(s_{J,m})\Phi_{C,m}(s_{J,m})p(\hat{B}_J)\mathrm{E}\big[U_C(m)|s_{C,m}, s_{J,m}, B_{C,m}, \hat{B}_J\big] \tag{6.8}$$

$$\boldsymbol{\Phi}^*_{J,m}(\boldsymbol{\Phi}_{C,m}, B_{C,m}, B_{J,m}) = \operatorname*{argmax}_{\boldsymbol{\Phi}_{J,m}} \sum_{s_{C,m}} \sum_{s_{J,m}} \Phi_{C,m}(s_{C,m})\Phi_{J,m}(s_{J,m})\mathrm{E}\big[U_J(m)\big|s_{C,m}, s_{J,m}, B_{C,m}, B_{J,m}\big] \tag{6.9}$$

The strategies the general version of the iterated best response algorithm converges to are always a BNE by definition, since the convergence criterion is for $\boldsymbol{\Phi}_{C,m}$ and $\boldsymbol{\Phi}_{J,m}$ to be mutual best responses. In zero-sum games, the algorithm always converges in a finite time [136].

6.4.2 *Updating beliefs*

Two different scenarios are considered, making two kinds of information available to C.

- *Full-feedback scenario (FF)*: in this scenario, the communicator observes both the success or failure of the transmission and the moves played by the jammer.
- *Low-information scenario (LI)*: in this case, C only observes the feedback from its intended destination, i.e., it only knows whether the packet was correctly received, but does not know the move played by J.

It should be noted that the full-feedback scenario is still an incomplete information game: the full knowledge of the jammer's moves does not give the communicator full knowledge of J's battery state; however, as the game evolves, C updates its beliefs in a Bayesian fashion, getting closer to the real value of J's battery state.

In the following, the posterior probability for J's battery level is derived by conditioning on the feedback available to C in each of the two scenarios and then repeatedly applying Bayes' theorem.

In a given subgame m, the jammer J starts the round with battery level $B_{J,m}$ and chooses a strategy $\boldsymbol{\Phi}_{J,m}$. The communicator C receives a feedback $\boldsymbol{d}_m$, which is a vector containing a feedback for each transmission attempt made in round m. In the full-feedback scenario, $\boldsymbol{d}_m$ contains all the observed moves of J, while in the low-information one it only contains the sequence of transmission outcomes (which always ends with

either an ACK or a packet drop). According to Bayes' theorem, the posterior probability of the estimate that J plays a certain strategy $\hat{\mathbf{\Phi}}_{J,m}$ given the feedback $\boldsymbol{d}_m$, is:

$$p(\hat{\mathbf{\Phi}}_{J,m}|\boldsymbol{d}_m) = \frac{p(\boldsymbol{d}_m|\hat{\mathbf{\Phi}}_{J,m}) \sum_{\hat{B}_J} p(\hat{B}_J)p(\hat{\mathbf{\Phi}}_{J,m}|\hat{B}_J)}{\sum_{\mathbf{\Phi}} p(\boldsymbol{d}_m|\mathbf{\Phi}) \sum_{\hat{B}_J} p(\hat{B}_J)p(\mathbf{\Phi}|\hat{B}_J)}, \tag{6.10}$$

where $p(\hat{B}_{J,m})$ is the communicator's prior belief distribution on the jammer's battery level in round m. The posterior probability of $\hat{B}_{J,m}$ in the m-th stage is given by:

$$p(\hat{B}_{J,m}|\boldsymbol{d}_m) = \sum_{\hat{\mathbf{\Phi}}_J} p(\hat{B}_{J,m}|\hat{\mathbf{\Phi}}_J)p(\hat{\mathbf{\Phi}}_J|\boldsymbol{d}_m) = \sum_{\hat{\mathbf{\Phi}}_J} \frac{p(\boldsymbol{d}_m|\hat{\mathbf{\Phi}}_J)p(\hat{\mathbf{\Phi}}_J|\hat{B}_{J,m})p(\hat{B}_{J,m})}{\sum_{\mathbf{\Phi}} p(\boldsymbol{d}_m|\mathbf{\Phi}) \sum_{\hat{B}_J} p(\hat{B}_J)p(\mathbf{\Phi}|\hat{B}_J)} \tag{6.11}$$

where the second equality is derived by applying Bayes' rule again and substituting the result of (6.10). C receives feedback $\boldsymbol{d}_m$ when J plays strategy $\mathbf{\Phi}_{J,m}$ with probability:

$$p(\boldsymbol{d}_m|\mathbf{\Phi}_{J,m}) = \sum_{\hat{\mathbf{E}}_J} p(\mathbf{d}|\hat{\mathbf{E}}_J)p(\hat{\mathbf{E}}_J|\mathbf{\Phi}_{J,m}), \tag{6.12}$$

where $\hat{\mathbf{E}}_J$ is a vector of moves of length a_m.

Full-feedback scenario. In this case, J's move is part of the feedback $d_m(a)$ for the a-th transmission attempt, which is given by $d_m(a) = \left(E_{J,m}^{(a)}, a, a_m, \chi_{C,m}\right)$, where $E_{J,m}^{(a)}$ is the action that J chooses in the a-th transmission attempt of the m-th round. Accordingly, the complete feedback vector for the m-th round is $\boldsymbol{d}_m = (d_m(1), \ldots, d_m(a_m))$. Since the feedback is obtained at the end of each round, hence C knows the value of a_m. As in (6.2), $\chi_{C,m}$ accounts for the success of the transmission.

Then, the first factor in the right-hand side of (6.12) is equal to $p\left(d_m(a)|E_{J,m}^{(a)}\right)$ if $\hat{E}_J = E_{J,m}^{(a)}$, and to zero otherwise: since C knows J's move as part of the feedback, that feedback cannot be the outcome of a different move. Naturally, since $E_{J,m}^{(a)}$ is contained in the feedback, any feedback element $d_m(a)$ which does not contain the correct E_J becomes impossible, and the sum in (6.12) reduces to a single term. If the attempt is not the last, the packet was not received correctly, and the feedback should reflect that.

The probability of feedback $d_m(a)$ when the energy used by J in the a-th attempt is $E_{J,m}^{(a)}$ is computed as:

$$p(d_m(a)|E_{J,m}^{(a)}) = P_e(E_{C,m}^{(a)}, E_{J,m}^{(a)}) \tag{6.13}$$

$$p(d_m(a_m)|E_{J,m}^{(a_m)}) = \chi_{C,m} + (2\chi_{C,m} - 1)P_e(E_{C,m}^{(a_m)}, E_{J,m}^{(a_m)}). \tag{6.14}$$

This allows us to solve (6.12), since the second factor $p(\hat{E}_J|\hat{\mathbf{\Phi}}_J, a)$ can be directly computed from the strategy itself and $E_{C,m}^{(a)}$ is known to C. Eq. (6.12) can then be simplified to:

$$p(\boldsymbol{d}_m|\mathbf{\Phi}_{J,m}) = \prod_{a=1}^{a_m} \sum_{\hat{E}_J=0}^{E_{J,\max}} p(d_m(a)|\hat{E}_J) p(\hat{E}_J|\mathbf{\Phi}_{J,m}), \tag{6.15}$$

since each attempt is independent from each other.

Low-information scenario. In this case, the feedback available to C is limited to $d_m(a) = (a, a_m, \chi_{C,m})$. The probability $p(d_m(a)|\hat{E}_{J,m}^{(a)})$ can be found by using (6.13) and (6.14) as in the previous case. In this case, the solution to (6.12) needs to take into account all possible move vectors, since C does not know the moves that were actually played by J. Note that the sum in (6.12) is over all possible $\hat{E}_J$, unlike in the full-feedback case.

Battery depletion. In both scenarios, since J's battery is depleted by the moves it plays in round m, this new information is included in the belief on the future battery state. The communicator believes that the total energy consumed by J in round m is $\hat{E}_{J,m} = \sum_{a=1}^{a_m} \hat{E}_{J,m}^{(a)}$. In the full-feedback scenario, the estimate $\hat{E}_{J,m}$ is always correct and deterministic, since $E_{J,m}^{(a)}$ is part of the feedback. In the low-information case, the probability distribution of $\hat{E}_{J,m}$ is:

$$p(\hat{E}_{J,m}|\boldsymbol{d}_m) = \sum_{\hat{\mathbf{E}}_{J,m} \in \mathcal{E}_{J,m}} \prod_{a=1}^{a_m} p(\hat{E}_{J,m}^{(a)}|\boldsymbol{d}_m), \tag{6.16}$$

where $\mathcal{E}_{J,m}$ represents the set of acceptable move vectors $\left\{\hat{\mathbf{E}}_{J,m} : \sum_{a=1}^{a_m} \hat{E}_{J,m}^{(a)} = \hat{E}_{J,m}\right\}$. We assume that the transmission attempts are spaced apart in time enough that the attempts can be considered independent. The probability $p(\hat{E}_{J,m}(a)|\boldsymbol{d}_m)$ can be computed by applying Bayes' rule:

$$p(\hat{E}_{J,m}^{(a)}|\boldsymbol{d}_m) = \sum_{\hat{\mathbf{\Phi}}_J} p(\hat{\mathbf{\Phi}}_J|\boldsymbol{d}_m) \frac{p(d_m(a)|\hat{E}_{J,m}^{(a)})\, p(\hat{E}_{J,m}^{(a)}|\hat{\mathbf{\Phi}}_J)}{\sum_{\hat{E}_J=1}^{E_{J,\max}} p(\hat{E}_J|\hat{\mathbf{\Phi}}_J)\, p(d_m(a)|\hat{E}_J)}. \tag{6.17}$$

The factors $p(\hat{\mathbf{\Phi}}_J|\mathbf{d})$ and $p(d_m(a)|\hat{E}_J)$ can be calculated using (6.10) and (6.12), respectively. In both scenarios, the updated posterior probability of J's battery level after round m becomes:

$$p(\hat{B}_{J,m+1}|\hat{B}_{J,m}, \boldsymbol{d}_m) = \sum_{\hat{E}_{J,m}=0}^{\hat{B}_{J,m}} p(\hat{E}_{J,m}|\boldsymbol{d}_m) \delta(\hat{B}_{J,m+1}, \hat{B}_{J,m} - \hat{E}_{J,m}), \tag{6.18}$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta. The probability distribution of $\hat{B}_{J,m+1}$ given the feedback can be calculated as:

$$p(\hat{B}_{J,m+1}|\boldsymbol{d}_m) = \sum_{\hat{B}_{J,m}=0}^{B_{J,\max}} p(\hat{B}_{J,m}|\boldsymbol{d}_m)p(\hat{B}_{J,m+1}|\hat{B}_{J,m},\boldsymbol{d}_m). \tag{6.19}$$

The belief update lets the communicator use the new information from the subgame rationally, continuing the overall game until its battery is depleted. All the equations above are equally valid for pure and mixed BNEs.

6.5 Simulation results

The performance of the optimal strategies are evaluated by studying the energy consumption and the Packet Delivery Ratio (PDR) of the communicator. Such performance are compared under different levels of information available at C, namely the complete information (*CI*) scenario of Sec. 6.3, the full-feedback (*FF*) and low-information (*LI*) scenarios of Sec. 6.4. The results have been obtained through Montecarlo simulation by running 10^6 independent games.

As discussed in Sec. 6.4, only a payoff horizon $T = 1$ is considered and a corresponding discount factor $\lambda = 1$. The weight α in the multiobjective optimization (see (6.2)) is varied between 0 and 0.8. Higher values of α were not considered, as the optimal action for the communicator would be to always drop packets, making the game trivial. The maximum number of quanta that can be used for a transmission are $E_{C,\max} = E_{J,\max} = 8$, and the communicator can retransmit the same packet at most once, i.e., $A = 2$. Larger values of A are computationally heavy due to the exponential growth of the action space and require suboptimal but lighter approaches. In the simulation we arbitrarily assumed that no penalty/reward is obtained if the first $A - 1$ transmission attempts of C fail, which corresponds to $\gamma(a) = 1\,, \forall a \leq A$ in (6.4); this may represent a scenario with non critical latency requirements where the role of A may only be to limit the amount of resources dedicated to each packet. The initial battery level of C is always $B_{C,0} = 50$, while that of J's battery varies: $B_{J,0} = \{0, 20, 50\}$. Notice that $B_{J,0} = 0$ means that there is no jammer and the communicator is playing only against the channel. At the beginning of the game, C has no information at all about the energy available to its attacker and thus its prior belief follows a uniform distribution: $p(B_{J,0} = B) = 1/(B_{J,\max} + 1)$ for all possible battery values $B = \{0, \ldots, B_{J,\max}\}$, with $B_{J,\max} = 50$.

Fig. 6.4 shows the number of packets sent by C as a function of the objective weight α for different values of $B_{J,0}$. Notice that this value is independent of the transmission outcomes and corresponds to the number of subgames played, which is 0 if C always

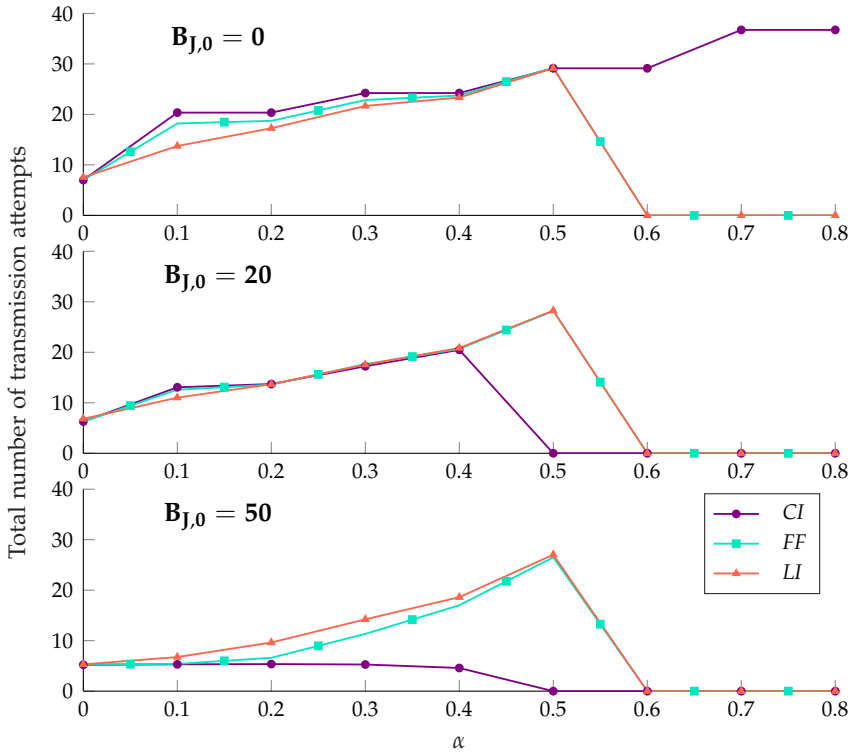


Figure 6.4: Total number of transmission attempts as a function of the weight α for different values of the initial battery charge of J and different levels of information available to C.

chooses to discard the packets to avoid wasting energy. As expected, the higher the energy available to the jammer, the fewer the packets sent by the communicator, because either it depletes its energy faster to fight the attack, or it decides to avoid sending packets as a passive response to the jamming, never sending any packets until the threat is over. When α is low, the main goal for C and J is to successfully transmit the packet and disrupt the communication, respectively, and there is no significant difference among the three scenarios. This happens because, by playing, C can infer the jammer's strategy and, consequently, its battery level. This represents a key result: when the communicator has limited information about the jammer (FF and LI scenarios), after only a few subgames of uncertainty, it gains knowledge about J's energy status and thus acts similarly to the CI case. The similarity between the two scenarios is an important result: the C node does not need to measure the jamming power to react optimally to the jammer's actions, as the feedback given by the packet acknowledgments is sufficient.

However, the node behaves differently in the full information case: when $B_{J,0} = 0$, the communicator knows that it can transmit without fear in the CI case, while it needs to learn it from the feedback in the two incomplete information scenarios, wasting some energy protecting itself against a jammer that is not there. The opposite happens when $B_{J,0} = 50$: in the full information case, C knows that the jammer is strong and behaves

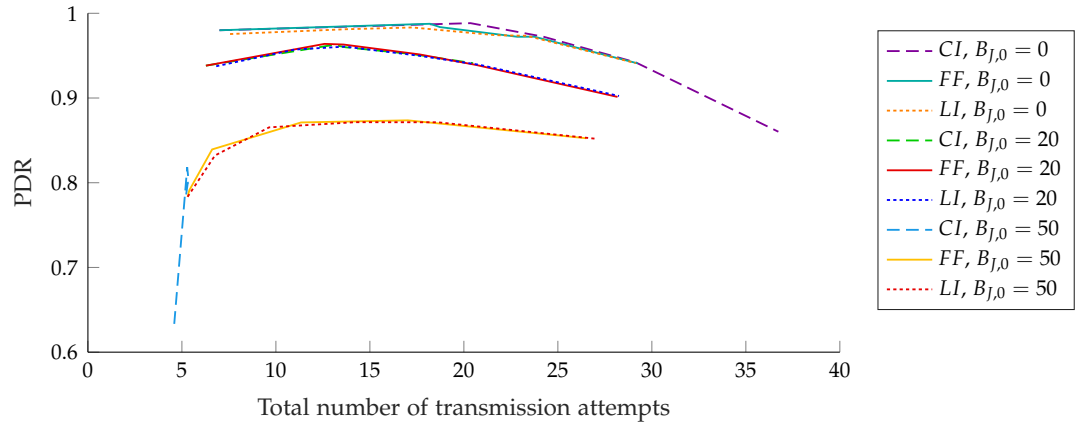


Figure 6.5: PDR vs. number of transmission attempts made by C for different values of the initial battery charge of J and levels of information available to C.

far more conservatively, often going to sleep mode to avoid wasting energy, while in the incomplete information scenarios C is more inclined to devote resources for transmission.

As α increases, the main goal shifts towards prolonging (or shortening, for J) the lifetime of the communicator. Therefore, C will dedicate less energy to each transmission attempt, thus the number of subgames increases. However, after a certain threshold that depends on the information gained by C and the energy availability of J, the optimal strategy for C is to always sleep and not even try to send its packets, because this would result in a waste of energy. When C is playing only against the channel ($B_{J,0} = 0$, top plot of Fig. 6.4), the more complete the information available to C, the longer its lifetime (which coincides with the number of subgames). This is not true when $B_{J,0} = 20$ and especially when $B_{J,0} = 50$ (middle and lower plots, respectively). When the jammer has a lot of energy available to attack and C knows it (CI scenario), C behaves very cautiously: in the first rounds, it is extremely conservative and only transmits rarely (or even sleeps if $\alpha \geq 0.5$). In this case, since the payoff horizon is set to 1, the reward on a single subgame is not a good metric to maximize the lifetime or PDR of C, which would be less conservative with a longer time horizon.

Fig. 6.5 shows the PDR against the total number of played subgames (i.e., the lifetime of node C). The first noticeable thing is that the curves obtained for the three scenarios are almost identical, confirming the fact that C quickly achieves enough knowledge about J's battery level even with a limited feedback. Further, as expected, as the initial battery charge of the jammer increases, the fraction of successful transmissions decreases because of the prolonged attack. It is interesting to note that the PDR curve is rather smooth. Finally, the performance obtained in the CI case when $B_{J,0} = 50$ are much worse than those of the FF and LI scenarios in the same energy case. As explained previously,

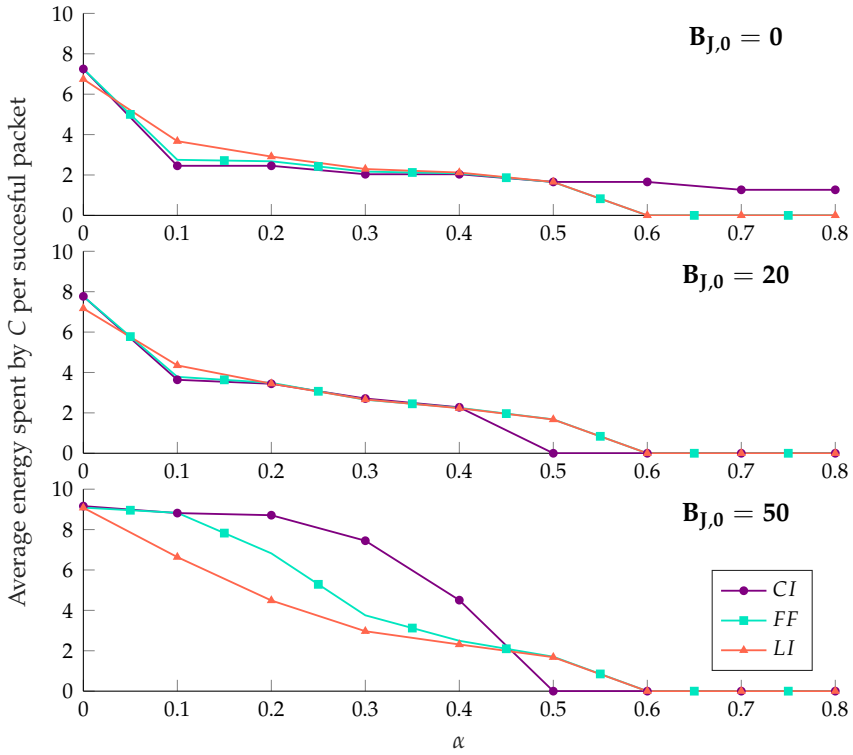


Figure 6.6: Average amount of energy spent by C per successful transmission vs. the objective weight α for different values of the initial battery charge of J and levels of information available to C.

this is due to an excessively conservative behavior of the communicator in the presence of a powerful jammer, that could be mitigated by widening the payoff horizon.

Finally, Fig. 6.6 shows the average amount of energy spent by C for a successful transmission and confirms and completes the previous results. As energy efficiency becomes more important (α closer to 1), C assigns less and less energy to each packet, until the extreme case of continuous sleep. Again, the three scenarios provide similar performance, with the exception of the case $B_{J,0} = 50$ (lower plot), whose different behavior, as already discussed, reflects the strategy adopted by C, which devotes a significant amount of energy to the packets it decides to transmit in order to be sure to succeed in the communication.

In conclusion, these results show that a partial knowledge of the jammer's state does not affect the ability of the transmitting node to defend itself against the attack; an analysis of the results of its moves reveals the jammer's strategy very quickly, and the communication performance and lifetime of the communicator are not significantly affected by the initial lack of information.

6.6 Learning to play

Sec. 6.3 analyses the jamming game in the case of complete information available at both players, when the optimal strategies are derived via dynamic programming. In Sec. 6.4 the assumption on the knowledge available to the communicator is relaxed and the dynamic programming solution is exploited to determine the optimal strategies through the iterated best response algorithm. The drawback of this approach is the computational load required to run the best response algorithm iteratively till convergence to the BNE. This section describes how to use a Neural Network (NN) to make the nodes learn the strategy to play, rather than deriving the BNE every time, and exploit the solution derived in the previous sections to evaluate the performance of the machine learning approach.

As mentioned, in the proposed scenario the jammer has complete knowledge of the communicator's moves and battery charge, so that the learning process concerns C only. C aims at choosing the optimal strategy to play in each round, given its current belief on J's battery level, and its own (known) battery level. Notice that C is unaware of both the initial and actual charge of J's battery. This problem lies in the area of *supervised learning*, because the goal is to infer the mapping between the input and a known output (derived as explained in Sec. 6.4). In particular, we decided to implement a NN using the open source Keras library in Python.[2]

NNs [139] model the relation between some input data and the corresponding output through interconnections among *neurons* (non-linear multi-input single-output functions) that are organized in multiple layers. NNs are an extremely powerful tool because they have the possibility of learning: the structure of the network is defined by the weights of the interconnections among neurons, which can be tuned during a *training* phase so that a *loss function* is minimized. Once the network is trained, the output of a given input is readily derived by simply doing some multiplications.

This approach has some limitations, as it will always yield a pure strategy, which will be an approximation of the optimal strategy and therefore might not be a BNE strategy. However, if performance is satisfactory, the reduction of computational complexity makes the optimality gap of the learned strategy acceptable. While back-propagation is computationally complex and might be too energy-intensive for a battery-limited node, simply calculating the output of a pre-trained neural network is simple, and our proposal is to train the NN offline and pre-load the weights into the node, so that it can calculate strategies efficiently.

2 `https://keras.io/`

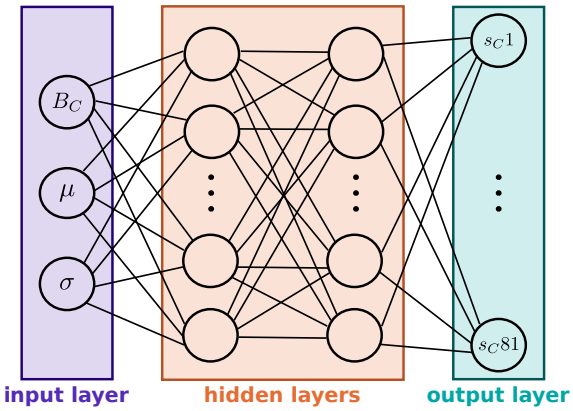


Figure 6.7: Structure of the NN. The network is fed with the current battery level of C, the mean μ and standard deviation σ of its belief on J's battery charge; the output is one among 81 strategies.

6.6.1 *Structure of the neural network*

The design of the NN has been influenced by the two following choices.

1. The optimal solution of the Bayesian jamming game also includes mixed NEs, where the optimal strategy of a player is a probability distribution over multiple pure strategies. Due to simplicity considerations, C only learns pure strategies; in the case of mixed NEs, C has to learn only the strategy with the largest probability to be played.
2. The goal of the two players does not change during the game; so, a different NN is implemented for each value of α, rather than giving it as input to the NN.

Thanks to choice 1), the learning reduces to a *multiclass classification* problem, where each strategy represents a separate class (and classes are mutually exclusive).

For both levels of information available to C (FF and LI scenarios), the training data were obtained by simulating more than 10^6 possible inputs (battery charge of C and statistics of its belief of the opponent's energy level), with the weight $\alpha \in \{0, 0.1, \ldots, 0.8\}$ and J's initial charge $B_{J,0} \in \{0, 20, 50\}$ (the parameters are the same as in Sec. 6.4). Instead of feeding the network with the probability distribution of C's belief of J's battery level, the inputs are its statistical mean and variance. Hence the input layer of the NN has 3 neurons: two for the mean and variance of the belief probability and one for C's own current battery charge, while the output layer has a neuron for each possible strategy. The number of possible strategy is $(E_{C,\max} + 1)^A$, as it depends on how many energy quanta the node can use in each transmission attempt and on the maximum number of transmission attempts for each subgame. With the parameters of Sec. 6.4, C can choose one between $(8 + 1)^2 = 81$ strategies. Fig. 6.7 shows the structure of the NN, which was trained with a value $B_{J,\max} = 50$.

Table 6.2: Parameters of the Neural Network.

Parameter	Value
# of layers	4
# of neurons	$[3, 40, 60, 81]$
Activation function	[relu[1], relu, relu, softmax]
Loss function	cross-entropy
Optimizer	Adam
Metrics	accuracy

[1] Relu stands for *REctified Linear Unit*.

Each NNs has the same structure, regardless of the value of α. In particular, the two hidden layers have 40 and 60 nodes, respectively, so that the NN has 4 layers in total. For the first 3 layers, the activation function, i.e., the function that dictates how a neuron's weighted input is mapped into its output, is the rectifier function, whereas for the output layer it is a softmax function, as usually done in classification problems. Thus, the network will output a separate probability for each one of the possible classes (with such probabilities all summing to 1) and chooses the most probable class. The learning process for updating the weights of the interconnections between the neurons is defined by a loss and an optimization function. The former represents the objective function that is used to evaluate a set of weights and that the training phase aims at minimizing; in this case, the multinomial cross-entropy between the predicted and the labeled real output has been used. The latter is the function that determines how weights are updated; the proposed NNs are trained with the Adam optimizer [140] with a Nesterov momentum [141], which in our case proved to guarantee faster convergence than the commonly used stochastic gradient descent algorithm. These choices are summarized in Table 6.2.

As commonly done in classification problems, the efficiency of the NN is evaluated by means of accuracy, i.e., the percentage of correct predictions. The computational cost of running the trained neural network is 7380 sum operations and 181 Relu functions, which amounts to less than 10 thousand total floating point operations. This is extremely light when compared to the iterated best response algorithm, which requires the calculation of the payoffs for every possible response strategy and every possible battery level. In the Bayesian case, just one round of IBR requires more than 8000 floating point operations, as payoffs for the communicator must be calculated for every possible pure response strategy and for every possible battery level of the jammer, then weighted by the probability of the given battery level and summed.

6.6.2 *Results*

The networks have been trained with more than 10^5 data samples uniformly distributed among the possible values of α only for very few iterations (about 15 training epochs

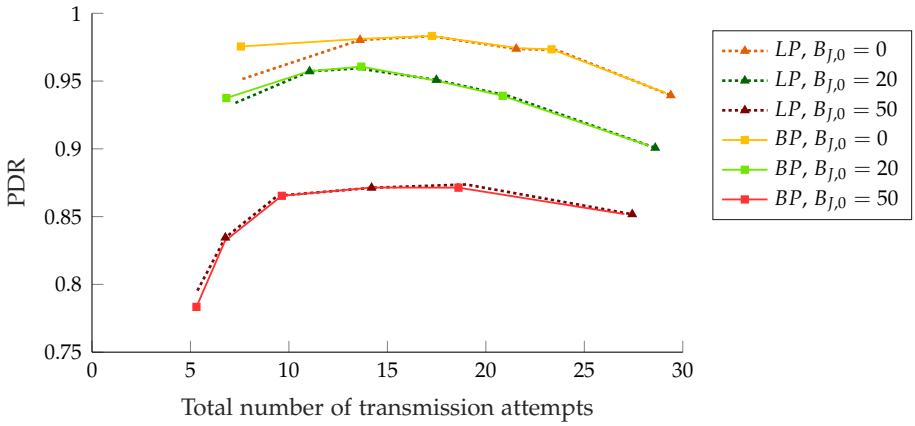


Figure 6.8: PDR vs. number of transmission attempts made by C for different values of $B_{J,0}$ in the *Low Information* scenario. Comparison between Bayesian (BP) and learning (LP) policies.

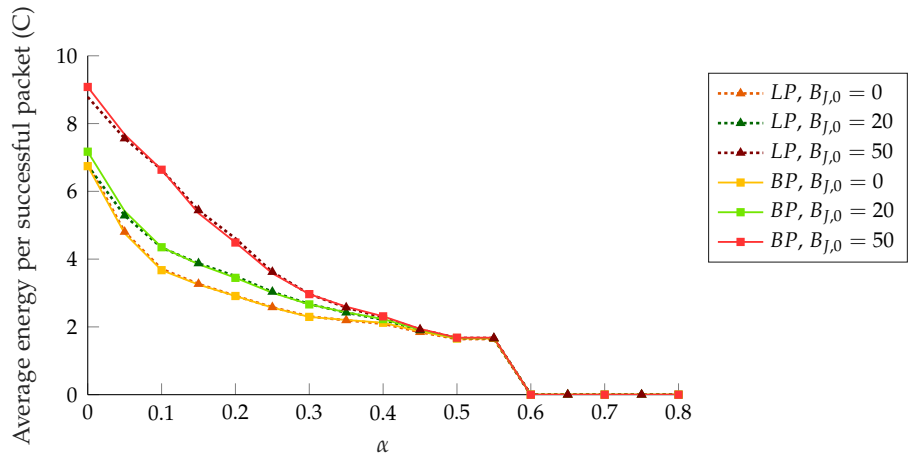


Figure 6.9: Average amount of energy spent by C per successful transmission vs. α for different values of $B_{J,0}$ in the *Low Information* case. Comparison between Bayesian (BP) and learning (LP) policies.

for network). 15% of the data was used for testing, i.e., to infer the performance of the network with unseen data, and obtained an accuracy larger than 95% for all values of α; even, values of $\alpha > 0.5$, when the strategy of C is trivial and consists in sleeping most of the time, yielded an accuracy larger than 99%.

The results obtained for the accuracy are extremely good, and indicate that it is possible to train a device to fight against a jammer with unknown energy availability. However, to determine how well the NN performs, we also repeated the Montecarlo simulations of Sec. 6.5 using the trained communicator against the omniscient jammer. We also compared the performance of the Bayesian policy (BP) with that obtained with the learning algorithm (LP).

Fig. 6.8 shows the PDR as a function of the lifetime (or, equivalently, the number of transmission attempts) obtained with the iterated best response algorithm as explained in Sec. 6.4 and with the trained communicator in the LI scenario. It is evident that C has

efficiently learned how to play against J, even without knowing the value of $B_{J,0}$. Fig. 6.9 also confirms that the communicator is perfectly able to learn the optimal strategy. The plot shows the average amount of energy spent by C per successful transmission and, again, the curves obtained with the two approaches basically coincide. The plots of the Full Feedback scenario are omitted, but the performance obtained with the learning system is almost identical to the optimal one also in this case.

6.7 Lesson learned

Jamming attacks can be a serious problem in IoT networks, where the constrained devices may not be able to support heavy defense mechanisms. In such a scenario, it is important to take into account the energy restrictions of both the jammer and the attacked node, as they can have a strong impact on the strategies played by the two devices.

Interestingly, the knowledge that the victim has on the jammer has a marginal impact, because as the game goes on, it gains an increasing knowledge about the jammer's energy condition, so that the performance obtained under incomplete information approaches that of the complete information scenarios. The analytical investigation proposed here sheds light on how the game evolves differently according to the level of awareness that the legitimate transmitter has about its attacker, making realistic assumptions on the nodes' capabilities.

Finally, this work proves that using a simple machine learning algorithm it is possible to obtain performance that is very close to those of the optimal Bayesian strategies, with a much lower computational load.

The model and the results discussed here may serve a starting point to develop and gauge lightweight attack and defense strategies, possibly extended to a whole network of jammed devices and with the inclusion of energy harvesting capabilities.

Part II

STUDIES ON EXISTING TECHNOLOGIES

7

Improved channel access in LTE

The RACH procedure in LTE serves as a synchronization mechanism between the Base Station (eNB) and the User Equipments (UEs): the UEs compete for resources by randomly choosing a preamble and getting assigned data resources by the eNB, provided that they do not collide with each other in the preamble-based contention phase. The collision resolution mechanism, however, is known to represent a bottleneck in case of MTC. Its main drawbacks are seen in the facts that eNBs typically cannot infer the number of collided UEs and that collided UEs learn about the collision only implicitly, through the lack of the feedback in the later stage of the RACH procedure. The collided UEs then restart the procedure, thereby increasing the RACH load and making the system more prone to collisions.

The current system was designed to handle about 128 attempts/s [142], however forecasts show that traffic could reach up to 370 attempts/s in the near future [143], mainly due to the expected increase of machine-type traffic in cellular access. In case of synchronized alarms, simultaneous accesses to the channel may yield up to a tenfold increase with respect to normal traffic, almost guaranteeing that all UEs will collide [144, 145]. RACH overload represents a serious issue that can significantly degrade the network performance, and in the last years a variety of approaches to mitigate collisions were proposed. Studies generally focus either on improving the preamble detection or on efficient usage of the random access-related resources.

This study ([C9] and avaialble in [146]) proposes a new machine learning based detection scheme that requires no modification of the current protocol stack, and that can be fully implemented at the eNB. It leverages machine learning techniques to design a system that outperforms the state-of-the-art schemes in preamble detection for the LTE RACH procedure. Most importantly, the proposed scheme can also estimate the collision multiplicity, and thus gather information about how many devices chose the same preamble. This data can be used by the eNB to resolve collisions, increase the supported system load and reduce transmission latency. The presented approach is applicable to novel 3GPP standards that target massive IoT, e.g., LTE-M and NB-IoT.

7.1 LTE RACH

The random access procedure in LTE is performed in the Physical Random Access Channel (PRACH), a dedicated physical channel with an overall bandwidth of 1.08 MHz and duration between 1 and 4 LTE subframes. This section describes how PRACH preambles are generated and how they are used in the multiple channel access scheme, and finally gives an overview of the state-of-the-art algorithms for preamble detection.

7.1.1 *Preamble generation*

LTE uses Zadoff-Chu (ZC) sequences, complex-valued sequences that satisfy the Constant Amplitude Zero Autocorrelation (CAZAC) property, as a basis to create RACH preambles. A ZC sequence of odd length N_{ZC} is defined as:

$$z_r(n) = \exp\left\{-j2\pi r\frac{n(n+1)}{N_{ZC}}\right\}, \quad n = 0, 1, \ldots, N_{ZC}-1 \tag{7.1}$$

where $r \in \{1, \ldots, N_{ZC}-1\}$ is the sequence root index; the LTE standard uses $N_{ZC} = 839$.[1] Given a root r, it is possible to generate multiple versions of the base sequence z_r through a circular shift, thereby obtaining orthogonal sequences that exhibit a zero correlation with one another at the receiver. The cyclic correlation of a ZC sequence with its root is a delta function with a peak corresponding to the circular shift that was applied to the root sequence. It follows that circular correlation of a root against a superposition of different sequences obtained from that root results in multiple peaks that correspond to the individual shifts.

According to the cell size, the PRACH preamble can have four different formats, with duration from 1 to 4 subframes. This study focuses only on preamble format 0, which is typically used in cells with a radius up to 14 km, and consists in a normal 1 ms random access burst with preamble sequences of duration 800μs (an exhaustive description of the different preamble formats and their structure can be found in [142]) The full PRACH preamble is obtained by prepending the so called Cyclic Prefix (CP) to the ZC sequence chosen by the UE. The CP is a replica of the last few symbols of the sequence, that helps counteracting the multi-path reflection delay spread, and whose length is specified according to the chosen preamble format. Additionally, the cyclic prefix makes it so that if the preamble signal is delayed in time (like in the case of a UE at the cell edge), there is an additional shift in the correlation peak. While this feature enables the eNB to estimate the channel delay experienced by a device, it also means that different preambles arriving with different delays can yield a peak in the same location of the

1 Except for preamble format 4, which is not treated in this study.

correlation signal. In order to maintain the separation of different preamble correlation peaks in the presence of delays, LTE UEs are allowed to only pick sequences whose shift is a multiple of a base quantity N_{CS}, which depends on the cell radius (and hence on the maximum delay) and on the propagation profile. N_{CS} is in fact chosen in such a way that each cyclic shift, when viewed within the time domain of the signal, is greater than the combined maximum round trip propagation time and multi-path delay-spread. This guarantees that the eNB can identify different preambles by applying a correlator and a peak detector to the received signal.

The cyclic cross-correlation between any two ZC sequences generated from different roots is instead a constant value. Hence, in order to reduce noise at the correlation, it is preferable to generate all preambles using as few root indices as possible. Each root allows one to obtain $\lfloor N_{ZC}/N_{CS} \rfloor$ different preambles, and the LTE standard contemplates the use of $N_{prb} = 64$ preambles in each cell. By assigning orthogonal ZC sequences to adjacent eNBs, the inter-cell interference is highly reduced. This study only considers the case $N_{CS} = 13$, which is the smallest possible cyclic shift dimension that allows one to obtain exactly N_{prb} preambles, so that a single root is used to generate all the LTE preambles of the cell. This configuration can be used in all cells covering a radius smaller than 0.79 km [142], and is thus well suited to represent densely deployed urban scenarios.

7.1.2 *Procedure*

The RACH procedure consists of four phases (see Fig. 7.1):

1. *Random Access Preamble*. The UEs that intend to transmit data randomly choose one among N_{prb} available preambles and send it to the eNB.
2. *Random Access Response (RAR)*. The eNB processes the received signal, consisting in the superposition of all the transmitted preambles, and detects which preambles were chosen. For each detected preamble, it then sends a RAR message to assign the uplink resources to the corresponding UE.
3. *L2/L3 message*. The UEs use the newly assigned data channel resources to communicate their connection request, and a unique identifier.
4. *Contention Resolution Message*. The eNB responds to the UEs using the identifiers they communicated in their L2/L3 message, granting the requested resources.

UEs can access the channel either in a *contention-free* mode, where the eNB forces the UEs to use a particular signature for the preamble generation, thus avoiding collisions, or in a *contention-based* mode. While contention-free access is reserved for handover and other special delay-sensitive cases, contention-based access represents the default method for UEs to access the channel, and is thus taken as the focus of this paper. In

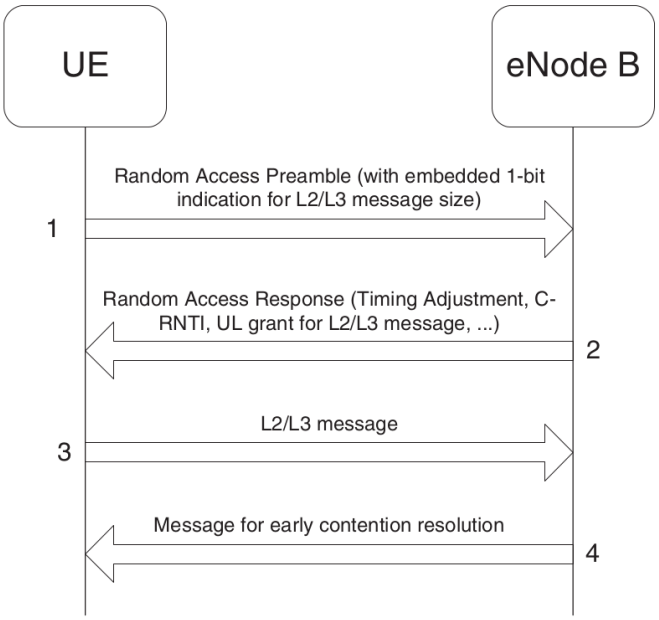


Figure 7.1: The LTE RACH procedure [142].

this procedure, which is illustrated in Fig. 7.1, if multiple UEs pick the same preamble, a collision happens, and may be resolved at different stages of the procedure. If the colliding preambles are received with high enough SNR, and are sufficiently spaced apart in time,[2] the eNB can detect both of them and recognize the collision, avoiding to send the RAR message to the involved UEs. Otherwise, the eNB will not detect the collision but a single preamble, and all colliding UEs will receive a RAR message and try to access the same resource simultaneously, colliding in the L2/L3 message phase. The eNB will therefore be unable to decode the received message, and will send no contention resolution message; the collided UEs can then try again in a new PRACH phase. Collisions go undetected especially in urban environments, where smaller cells are employed and the distance between different UEs is too small to allow a differentiation of multiple copies of the same preamble through their delay [145].

7.1.3 *State of the art in preamble detection*

The conventional approach to preamble detection in LTE RACH is provided in [142]. This category of detectors (which will be referred to as *threshold-based*) work by comparing the circular correlation of the received signal with its base sequence against a previously set threshold. A preamble is detected when the corresponding bin in the correlation signal contains values above the threshold. Such threshold is typically a function of the

2 More specifically, the energy peaks corresponding to the preambles should be separated, in time, by at least the maximum delay spread of the cell.

estimated noise level: this provides direct control over the false alarm probability, which can be made arbitrarily low (at a price in missed detection performance).

The threshold mechanism described previously is extended in [147] to take into account quantization and discretization steps that can improve computational performance, at the expense of missed detection probability. Another approach to increase the detection performance for LTE frequency division duplex systems is presented in [148], where noise is smoothed through an additional preprocessing phase prior to computing the correlation signal. This improves the performance in an ideal Additive White Gaussian Noise (AWGN) channel and with low SNR, but may be impractical for fading channels.

Multiple root sequences are considered in [149], which proposes a preamble detector able to identify non-orthogonal preambles and suppress the noise rise. The key idea is that the eNB can detect preambles in an almost interference-free environment by eliminating the interfering signals from the original received signal. The Power Delay Profile (PDP) allows one to obtain the channel profile, used by the eNB to reconstruct the preamble signal.

The problem of performance degradation due to time dispersion of the channel is investigated in [150]. Under the assumption of known PDP of Rayleigh fading that is independent across antennas and multiple paths, the paper derives an optimal statistic for preamble detection for frequency selective channel, as an alternative to increasing the eNB target PRACH received power to counter the time dispersion.

7.1.4 *Interest in multiplicity detection*

The benefits of the multiplicity detection are briefly commented in the following paragraphs. A straightforward application is to avoid sending a RAR message when multiple UEs activated the same preamble, so that the involved UEs will be implicitly informed about the collision. This enables to avoid subsequent collisions of L2/L3 messages and to shorten the RACH procedure. Further, multiplicity detection may allow one to infer the current load of the LTE RACH, as well as trends regarding its changes. This could help, e.g., a proper dimensioning of the resources dedicated to the PRACH, or adjust the operation of the RACH procedure. Examples are the dynamic allocation algorithm [151], and the dynamic access class barring [152], respectively.

Another line of works where the multiplicity knowledge could be beneficial are the ones which assume reengineering of LTE RACH procedure, e.g., grouping LTE preambles in codewords that could convey information to the eNB [153, 154] or use of advanced collision resolution algorithms RACH [155, 156], which are shown to lower the latency and increase the reliability and throughput of LTE RACH.

As a final remark, note that the focus of the paper is on the multiplicity detection, while its coupling with advanced LTE RACH algorithms is left for further work.

7.2 Machine learning approaches

The main contribution of this work is the application of Machine Learning (ML) to preamble detection (i.e., determining whether a certain preamble was sent or not) and preamble multiplicity detection (i.e., determining how many devices sent the same preamble) in LTE. The necessary premise for every ML algorithm is to have data available, which can be used to train these systems and make them "learn". Thus, this section firstly gives some details on the dataset used, and then discusses the employed ML techniques, namely Logistic Regression (LR) and NN.

7.2.1 *Dataset generation*

Research about multiplicity collision in the RACH procedure is quite modest, and, to the best of our knowledge, there is no publicly available dataset that relates the signal received at the eNB with the information of the preambles chosen by each station. Hence, we generated our own dataset, using the MATLAB LTE module.[3] The LTE System Toolbox™ in fact provides standard-compliant functions for the design, simulation, and verification of both the LTE and LTE-Advanced communications systems, with models up to Release 12 of the standard (at the time this study was performed.).

7.2.1.1 *Data labeling*

To run the ML algorithms, it is necessary to have a map between (i) the signal received at the eNB at the end of Phase 1) of the RACH procedure, and (ii) the number of UEs that selected each preamble. These will represent the input and output data of the algorithms, respectively. Accordingly, each simulation consists of simultaneous random access attempts from a certain number of UEs, each choosing one of the N_{prb} available preambles. Using the MATLAB LTE module, it is possible to extrapolate the corresponding *correlation* signal at the eNB, i.e., the output of the cyclic correlation of the signal received at the eNB with respect to the root index, as described in Sec. 7.1.1.

As explained in Sec. 7.1.1, each preamble is uniquely mapped to a correlation window of predefined size, according to the value of N_{CS}. Such window will be denoted as *bin* from now on. Since the goal is determining how many devices chose each preamble, we decided to consider each bin separately, rather than the correlation signal as a whole. Fig. 7.2 shows an example output of the MATLAB LTE module. Here, the correlation

3 https://www.mathworks.com/products/lte-system.html

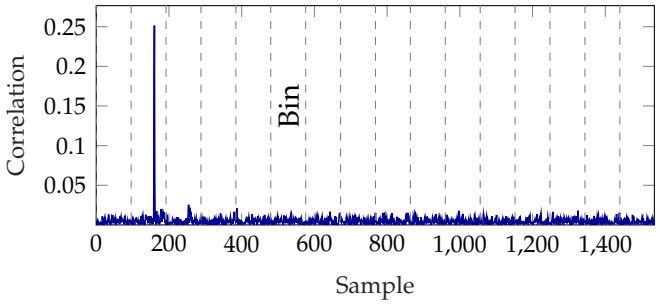


Figure 7.2: Example of correlation signal with 16 bins.

signal was divided in 16 bins for clarity; however the actual schemes work by dividing the correlation signal into $N_{\text{prb}} = 64$ bins, each one corresponding to a preamble.

7.2.1.2 *Applications*

Two distinct applications have been investigated:

- *Preamble detection*, to compare the performance of the proposed ML systems with respect to the state-of-the-art. For this application, the goal is to identify *whether* a preamble was sent, and *which* one. Three distinct datasets were generated to train and evaluate detection performance:
  - A *noise-only* set of correlations, used to evaluate false alarm probabilities according to the eNB testing specification [157];
  - A set of correlation signals obtained when only a *single UE* is transmitting, used to test missed detection;
  - A dataset containing both *noise-only and single UE* correlations, used to perform training of the ML systems.
- *Multiplicity detection*, to derive the accuracy obtained when estimating the number of UEs that chose the same preamble. For this purpose, dataset with bins containing from 0 to 5 preambles was generated. This dataset was then split in two, for training and testing.

For both applications, all datasets have been generated considering multiple scenarios, which differ for:

- *Noise level*. The signal received at the eNB is the superposition of all the transmitted preambles, corrupted by noise, expressed in terms of SNR. Based on performance from the tested ML systems, we decided to focus on an SNR range between -18 dB and -12 dB.
- *Channel model*. We considered an AWGN channel and an ETU70, i.e., an Extended Typical Urban channel with 70 Hz Doppler affected by Rayleigh fading.

Over 10^5 independent simulations were run for each scenario, deriving the correlation signal in each bin and the number of UEs that selected the corresponding preamble.

7.2.1.3 *Traffic intensity*

While the preamble detection problem requires the transmission of at most one preamble per RACH attempt, the number of competing devices is a critical parameter that influences the performance of the multiplicity detection procedure. In this study, the performance evaluation focused on high traffic scenarios corresponding, e.g., to alarm events that trigger transmission from a large number of UEs.

In [158], 3GPP proposes a model for highly correlated traffic arrivals, where the number of UEs in each random access opportunity inside the considered time frame follows a Beta distribution. The standard [158] defines a possible massive access scenario with a maximum of 30000 devices accessing the channel "synchronously", i.e., over a time interval of about 10 s. Considering RACH opportunities to happen every 20 ms, this corresponds to a maximum expected number of devices that participate in the same RACH opportunity of about 120. The dataset was therefore generated with 120 UEs that randomly choose among the N_{prb} available preambles. Due to the binomial distribution nature of the preamble selection mechanism, 99% of the bins generated in this way have at most 5 devices choosing that preamble.

7.2.2 *Logistic regression*

LR is a statistical method predicting the probability that a given input data belongs or not to a certain class. The rationale behind LR is that the input space can be separated into two complementary regions, one for each class, by a linear boundary. It differs from linear regression because the dependent, output variable is binary rather than continuous, and this method is in fact intended to model classification problems. The probability that the output belongs to one or the other class is expressed by means of a sigmoid $\sigma(\cdot)$, also called logistic function, from which LR takes its name.

In order to make a prediction, it is first necessary to *train* LR, i.e., use known labeled data to determine the weights in $\sigma(\cdot)$ that minimize a predefined cost function; such cost function measures the distance between the desired output and the predicted one.

Given an SNR value, a different LR model was trained for each type of channel considered (AWGN and Rayleigh), and for both the investigated problems (preamble detection and collision multiplicity); the implementation was done using the open source scikit-learn library in Python[4]. The dataset used is the one described in Sec. 7.2.1, so that the logistic regressor is fed with the correlation signal obtained at the receiver in a bin,

4 https://scikit-learn.org

while the output data is the number of UEs that picked the preamble corresponding to the considered bin. For preamble detection, there are two possible classes, i.e., 0 or at least one UE, while the number of possible classes in the collision multiplicity problem is $N_{\max} + 1$, where $N_{\max}$ was set to 5 and represents the maximum number of colliding UEs we are interested to estimate (see Sec. 7.2.1.3). Choosing among $N_{\max} + 1$ alternatives can be modeled as a set of $N_{\max}$ independent binary choices (a pivot alternative is compared to the remaining $N_{\max}$ ones). This makes the training complexity linear in $N_{\max}$.

As common in classification problems, the performance of the LR predictors were gauged through the *accuracy* metric, which represents the fraction of correct predictions.

7.2.3 *Neural network*

Artificial NNs [139] are a class of mathematical models that are considered universal approximators [159], as they are able to represent, up to any accuracy, any non linear function. The basic units of NNs are called *neurons* and are organized in multiple layers. Any NN has an input layer fed with the data to process, an output layer that represents the corresponding output determined by the network, and possibly one or more hidden layers. Neurons represent non-linear multi-input single-output functions; such functions are characterized by some weights and biases, which represent the interconnections among the neurons. Similarly to LR, a NN learns the relation between an input data and its corresponding output through a training phase, during which all the weights and biases are progressively tuned to produce the desired output. NNs are an extremely powerful tool because they can infer even very complex functions, that analytical models or simpler approaches such as LR fail to describe.

This study uses a simple feedforward NN, i.e., a fully-connected network where any neuron in layer i is connected to any neuron in the next layer $i + 1$. For each considered SNR value, channel type and problem to solve (preamble detection or collision multiplicity), a different NN was trained using the Keras[5] library. Since the ultimate goal consists in determining the number of users that picked a certain preamble, the input data is the correlation signal obtained at the eNB for the corresponding bin. The output layer has $N_{\max} + 1$ nodes, where $N_{\max}$ is the maximum number of colliding UEs we are interested in estimating ($N_{\max} = 1$ for the preamble detection, $N_{\max} = 5$ for the collision multiplicity).

In both scenarios, for all layers but the last one, the activation function (i.e., the function that dictates how a neuron's weighted input is mapped into its output) is the rectifier function, whereas for the output layer we decided to use a softmax function, as usually done in classification problems. This implies that the NN outputs a separate probability

5 https://keras.io

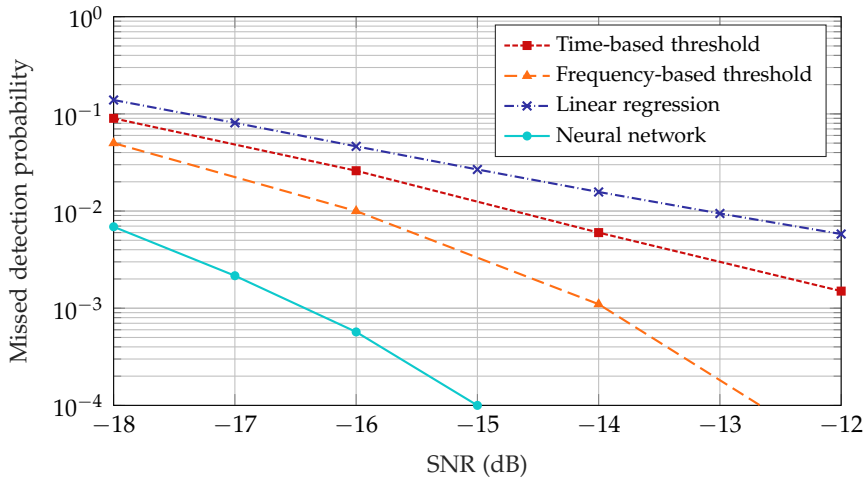


Figure 7.3: Detection performance comparison in an AWGN channel.

for each of the possible classes and chooses the most probable class. The efficiency of the neural network was evaluated by means of accuracy, as for LR.

7.3 Results

The performance of the LR and NN methods were evaluated for preamble detection and collision multiplicity estimation. This section describes the obtained results and discusses the new opportunities opened up by the ML tools, as well as their limitations.

7.3.1 *Preamble detection performance*

To guarantee a fair comparison with the state-of-the-art threshold based detection, the simulations were conducted according to the LTE specifications on eNB conformance testing [157], measuring the *missed detection probability*, i.e., the probability that a transmitted preamble remains undetected at the eNB. In this case, the test dataset (see Sec. 7.2.1) contains bins belonging to correlation signals in which exactly one device was transmitting. The dataset used for training of the ML systems, instead, consists of bins coming from 10^5 correlation signals containing either 0 or 1 preambles.

The results for both the LR and the NN schemes in the case of an AWGN channel are shown in Fig. 7.3, together with the performance of the algorithms proposed in [147] (see Sec. 7.1.3). The scheme based on LR yields worse performance than those leveraging thresholds, because of its extreme simplicity; the mediocre performance obtained with LR suggests that the correlation signal at the eNB and the transmission of defined preambles are data that are not completely separable, but are rather related in a more complex way. This is supported by the outstanding performance of the NN, which provides a gain in the range of 2 to 3 dB with respect to the best threshold-based detector for all

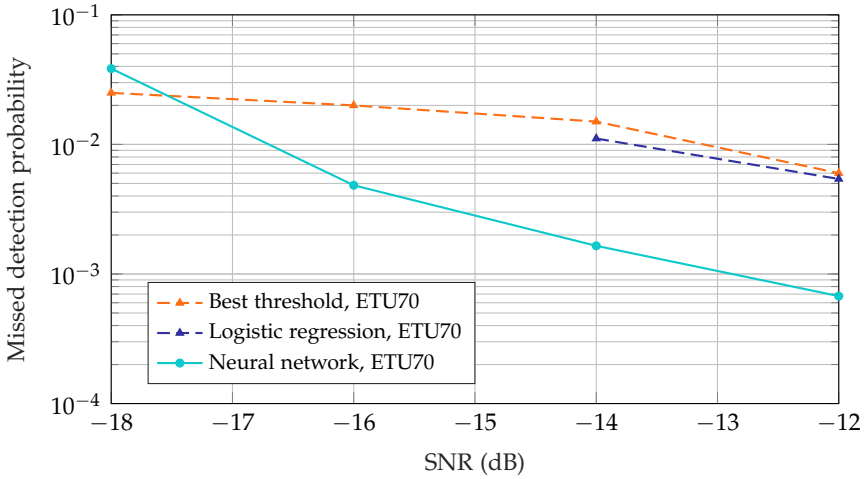


Figure 7.4: NN and best threshold system detection performance comparison in an ETU70 channel.

values of SNR, yielding a performance which conforms to the standard requirements. Rigorously, missed detection probability should also include the cases of a wrong UE delay estimation. However, in the case of ML based systems, which are not capable of estimating this parameter, delay detection cannot be taken into account.

Fig. 7.4 shows a comparison between the detection performance of the NN in the case of an ETU70 channel, and compares it with the performance of the best threshold-based detection scheme. Also in this case, the NN achieves a significant improvement in detection performance, upwards of 4 dB with respect to the threshold based schemes described in [147]. LR performance is only shown at SNR values in which the false alarm probability requirement is satisfied.

A metric which is complementary to the missed detection probability is the *false alarm probability*, i.e., the probability that the eNB wrongly detects a transmission in an unused bin. The minimum requirement for this metric is set by the standard at 0.1% [157]. ML approaches do not provide the same direct control over the false alarm probability that threshold-based schemes offer. With the LR scheme, the false alarm probability requirement is respected for SNR larger than -16 dB. On the other hand, the false alarm probability obtained with the NN was consistently under the required 0.1% threshold for all the analyzed SNR values, for both AWGN and ETU70 channels.

7.3.2 *Multiplicity detection*

The reliability of the proposed schemes in the case of multiplicity detection was assessed by measuring the frequency of errors in the estimation. Fig. 7.5 shows the probabilities for both LR and NN approaches to give a guess that is wrong by some amount, denoted as offset, for an AWGN channel. Offset 0 represents the probability of guessing exactly

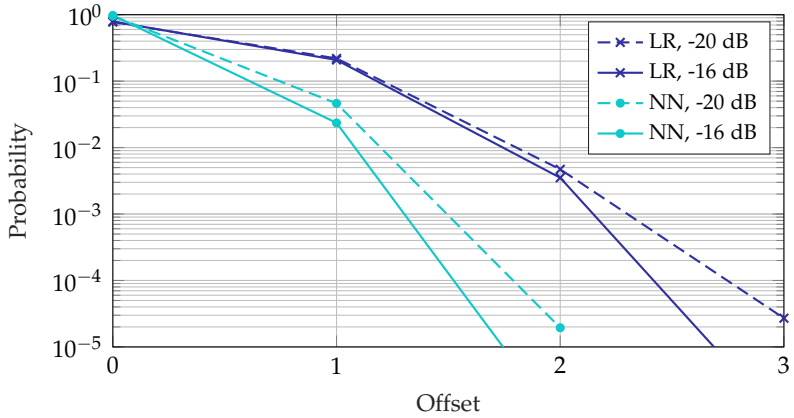


Figure 7.5: Probability of getting the multiplicity wrong by different offsets in an AWGN channel.

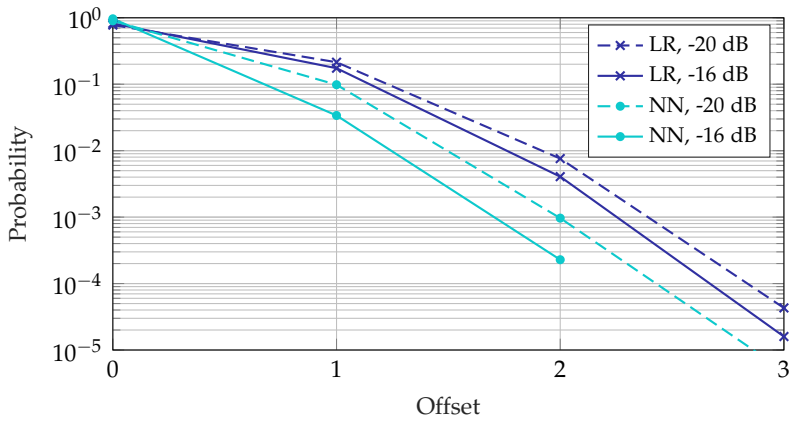


Figure 7.6: Probability of getting the multiplicity wrong by different offsets in an ETU70 channel.

the number of preambles in the bin, offset 1 represents the probability that the difference between the estimated and real number of transmitted preambles is ± 1, and so on.

As expected, lower SNR values give higher error probabilities. The NN clearly outperforms the LR approach, thanks to its higher complexity, which allows one to infer even very complicated relations between its input and output data. Although LR could not compete with the NN in the preamble detection (see Fig. 7.3), a linear approach is still able to extract the information needed to identify the number of transmitters in each bin from the signal received at the eNB with a probability of around 0.8, and gets the multiplicity wrong by at most 1 with a probability of 0.99. The NN scheme, instead, guarantees with a probability of about 0.999 to have an estimate that is either correct or only off by one, even for very low SNR values, and yields exact guesses with a probability of 0.9.

Similar considerations can be made for a ETU70 channel, as shown in Fig. 7.6. In this case, however, the improvement obtained with the NN scheme over the LR one is smaller.

Table 7.1 contains the normalized confusion matrices for the NN and LR systems, when SNR $= -16$ dB: the rows represent the actual multiplicity value, while the columns represent the value estimated by the ML scheme. Hence, the confusion matrix for an ideal system would have ones along the diagonal, signifying that for each input the labeling is correct with probability 1. It can be seen that the NN consistently outperforms the LR, except for the case of no preambles being sent, where LR achieves a perfect score. Despite this perfect performance in the correct detection of no preambles in the bin, the LR scheme is also heavily affected by a wrong detection rate in case of 1 preamble being sent, thus motivating the bad overall detection performance seen in Fig. 7.3. Furthermore, the confusion matrices also show that the LR scheme tends to overestimate the number of preambles in a bin, while the NN yields a more symmetrical error.

Table 7.1: Confusion matrices; SNR $= -16$ dB, AWGN channel.

Neural Network

	0	1	2	3	4	5
0	**0.996**	0.003	0	0	0	0
1	0	**0.972**	0.027	0	0	0
2	0	0.005	**0.987**	0.007	0	0
3	0	0	0.013	**0.978**	0.007	0
4	0	0	0	0.018	**0.971**	0.009
5	0	0	0	0	0.033	**0.966**

Logistic regression

	0	1	2	3	4	5
0	**1**	0	0	0	0	0
1	0.018	**0.735**	0.239	0.007	0	0
2	0	0.048	**0.604**	0.334	0.011	0
3	0	0	0.062	**0.659**	0.278	0
4	0	0	0	0.086	**0.813**	0.099
5	0	0	0	0	0.088	**0.912**

7.3.3 *Machine learning limitations*

Although they have been shown to provide good results in preamble detection and collision multiplicity estimation, ML approaches entail some disadvantages, which are described next.

Complexity. Even though the training phase of ML based detection is performed offline, such approaches are more complex than threshold based detection, and inevitably require a larger computational power. In fact, while state-of-the-art schemes only need to compare each correlation sample with a predefined threshold, a detector based on LR needs to perform a multiplication for each sample, and then sum all the obtained results to get to a decision. To perform multiplicity detection, such operations need to be performed for each classifier as described in Sec. 7.2.2. Neural networks also need to perform a number of operations that is proportional to the network complexity, and could be larger than the operations required by a LR scheme in the case of detection. When compared to a LR scheme in the case of multiplicity estimation, however, a NN needs to perform roughly the same number of operations as in the detection scheme, since it already outputs a classification. In fairness to ML, it is to note that the complexity burden of the proposed algorithms is on the eNB, whose hardware and software capabilities are

constantly improving and typically have very little computational and energy restrictions, such that the higher complexity demanded by ML schemes may not be a limiting factor.

Dataset collection. Both in the case of LR and NNs, a dataset of sufficient size must be available to perform effective training. In this work, the ML based approaches were trained on computer generated signals using a state-of-the-art simulator in order to have a representation as close as possible to what the eNB will actually see in a real deployment. This allowed us to exactly know the number of UEs choosing each preamble. For real deployments, this dataset could be integrated with some samples taken by the eNB. In particular, a bin associated to a certain preamble can be labeled according to the outcome of the RACH procedure: label as 0 if no device sends a *msg3*, as 1 if exactly 1 device answers with a *msg3*, and as 2 if a collision happens during *msg3*. Unfortunately, this does not allow for a precise estimation of the number of devices sending a specific preamble. Such an estimation may instead be possible if, once the eNB detects a collision (i.e., if it labels the considered bin as 2), another ad-hoc contention resolution phase were to take place iteratively among the colliding nodes, until resolution of all collisions.

7.4 Lesson learned

With RACH overload becoming an increasingly serious issue in massive access LTE scenarios, it is necessary to develop techniques to better manage the contention of resources by multiple users. This study investigates the use of machine learning techniques for preamble detection, and shows that they can improve the performance of state-of-the-art threshold-based schemes. Moreover, such techniques, and in particular neural networks, have proven to be capable of inferring the number of users that pick a selected preamble, so as to estimate the collision multiplicity.

The results described in this study may serve as a first step to improve the current RACH procedure, and are applicable to novel massive IoT 3GPP standards, e.g., LTE-M and NB-IoT. In particular, the possibility to immediately identify the number of devices that chose the same preamble allows more efficient collision management than the current approach, which envisages a repetition of the RACH procedure until all UEs uniquely pick a preamble. This would have a positive impact on the latency, the device power consumption, and the supported system load. Moreover, collision multiplicity detection may also allow one to promptly identify a switch in the data reporting regime, e.g., in case of an alarm that triggers synchronous transmissions from multiple devices.

8

mmWave communication: contention-based channel access in IEEE802.11ad

Millimiter waves commonly denote the Extremely High Frequency (EHF) band, i.e., the portion of spectrum between 30 and 300 GHz. They have been gaining a lot of momentum in telecommunications thanks to the large band of spectrum available, which has the potential to eliminate many of the issues of the overcrowded sub-6-GHz bands and allows for channels with larger bandwidth and, thus, higher capacity.

The propagation environment in the mmWave spectrum is significantly different from that at sub-6-GHz frequencies, and is characterized by a high propagation loss and a significant sensitivity to blockage. The coverage range can however be increased through beamforming, by focusing the power (both in transmission and in reception) towards the chosen direction, yielding a so-called directional link. This can be obtained by properly steering the elements of the antenna arrays, which, thanks to the short wavelengths, can be extremely compact and easily embedded into sensors and handsets [160].

Directional transmission opens up new possibilities: with all the power focused in a specific direction, the gain in the other directions is low, significantly reducing interference among concurrent transmissions. MmWave networks may even operate in a noise-limited rather than interference-limited regime [161], and the consequent high potential for spatial reuse can boost the network performance. Notice, however, that beamforming is a delicate process. First, beamforming training is necessary to establish a directional link (avoiding the use of inefficient quasi-omnidirectional communication), then beam tracking is needed to maintain the communication link. Beam misalignment may prevent communication, resulting in the so-called *deafness* issue, and poorly trained beams lead to extreme throughput drops (of up to 6.5 Gbps in 802.11ad [162]).

Because of the peculiar characteristics of the signal propagation at EHF, protocols designed for lower frequencies cannot simply be transposed to the mmWave band, but major design changes are required for both PHY and MAC layers. This pushed a standardization effort from several international organizations, including ECMA, the

IEEE 802.15.3 Task Group 3c (TG3c), the Wireless Gigabit Alliance (WiGig), the 802.11ad and 802.11ay standardization task groups, and the WirelessHD consortium. This study focuses on the IEEE 802.11ad standard, which operates in the 60 GHz unlicensed band [163].

While extensive research is ongoing to develop mechanisms for efficient beamforming training and beam tracking [164, 165], it is also necessary to understand how to access the wireless medium and use the beamformed links efficiently. The MAC layer of 802.11ad presents several features which yield an outstanding scheduling flexibility: it is possible to have contention-based and contention-free allocations, and an additional mechanism built on top of the schedule allows to dynamically allocate channel time in quasi real-time. However, the standard only provides very general guidelines on how to exploit this hybrid MAC layer and, to the best of our knowledge, efficient scheduling schemes that match each traffic pattern to the most appropriate allocation still need to be developed. To realize an adaptive scheduler able to optimally allocate the channel time resources and accommodate disparate traffic requirements, it is first necessary to assess the performance that can be obtained with the mechanisms available in 802.11ad.

The goal of this study is to serve as a first step in the modeling of contention-based allocations, assess the achievable throughput and delay and understand the role played by various scheduling parameters in the system performance. In particular, we propose a variation of Bianchi's seminal model for the Distributed Coordination Function (DCF) mechanism in legacy WiFi networks [166]. Such variation addresses all the novel features of the 802.11ad standard and, unlike most of the works proposed in the literature, takes into account the deafness and hidden node problems, which are exacerbated by directional transmissions. The proposed model is based on a division of the area around a considered station (STA) into regions, similarly to what done in [167]: STAs are clustered based on whether they can overhear the messages sent by the STA to and/or received from the Access Point (AP). However, [167] does not specify how to determine such regions; we instead explain how to compute the area of the regions mathematically, providing also the formulation for its expectation over the location of the considered station.

This study has been realized in collaboration with the National Institute of Standards and Technology (NIST) (Gaithersburg, MD, US) during a six-month-internship.

8.1 802.11ad

Ratified in December 2012, the 802.11ad amendment to the IEEE 802.11 standard targets short range mmWave communication in local area networks [163]. Since it can also be used in PBSS (i.e., network architectures for ad hoc modes), the central coordinator of

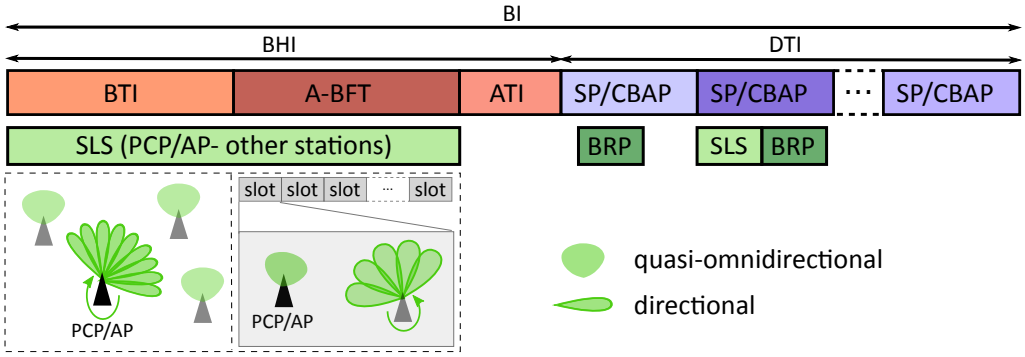


Figure 8.1: Structure of a BI. Green boxes correspond to beamforming training operations. The BHI is used for SLS with the PCP/AP: during the BTI, the PCP/AP trains its transmitting antenna pattern; during the A-BFT the other stations train their transmitting or receiving antenna patterns in dedicated slots. During the DTI, stations can perform both SLS and BRP phases with the PCP/AP and with other stations.

802.11ad networks can be either a PBSS Control Point (PCP) or an AP; accordingly, it is generally denoted as PCP/AP to include both infrastructures.

The nominal channel bandwidth in 802.11ad is 2.16 GHz, and there are up to 4 channels in the ISM band around 60 GHz, although channel availability varies from region to region. Only one channel at a time can be used for communication. There are 32 different Modulation and Coding Schemes (MCSs) available, grouped into three different PHY layers, which differ for robustness, complexity, and achievable data rates (up to 6.75 Gbps).

8.1.1 *Beamforming training*

802.11ad introduces the concept of antenna sectors, which correspond to a discretization of the antenna space and reduces the number of possible beam directions to try. Beamforming training is realized in two subsequent stages: the Sector-Level Sweep (SLS) phase and the Beam Refinement Protocol (BRP) phase. The SLS is necessary to set up a link between the involved stations, with one station sequentially trying different antenna sector configurations while the other station has its antennas configured in a quasi-omnidirectional pattern. Notice that both the transmitting and the receiving antenna patterns can be trained. This determines the best coarse-grained antenna sector configuration. The BRP is then used to fine-tune this configuration using narrower beams and, possibly, to optimize the antenna weight vectors in case of phased antenna arrays. Since a directional link has already been established previously during the SLS phase, in BRP stations can avoid using quasi-omnidirectional patterns and use a more efficient modulation and coding scheme, achieving higher throughput and better communication range.

8.1.2 *Beacon Intervals*

Medium access time is divided into Beacon Intervals (BIs), which are used to establish directional communication links through beamforming training, and for data transmission. The maximum duration of a BI is 1 s, but it is typically chosen around 100 ms. Each BI consists of two parts: the Beacon Header Interval (BHI) and the Data Transmission Interval (DTI), as shown in Fig. 8.1. The BHI replaces the single beacon frame of legacy WiFi networks and includes up to three access periods, all of them optional:

- The Beacon Transmission Interval (BTI) is used for beamforming training of the PCP/AP's antennas and network announcement. The PCP/AP broadcasts beacon frames through different sectors, performing the first part of the SLS phase with the other stations. The other devices have their receiving antennas configured in a quasi-omnidirectional pattern.
- The Association-Beamforming Training (A-BFT) is divided into slots during which stations separately train their antenna sectors for communication with the PCP/AP, and provide feedback to the PCP/AP about the sector to use for transmitting to them. This completes the SLS phase started in the BTI to establish a link with the PCP/AP.
- The Announcement Transmission Interval (ATI), used to exchange management information between the PCP/AP and associated and beamtrained stations, such as resource requests and allocation information for the DTI.

The DTI is used for data transmission. Prior to any additional frame exchange between stations, it is necessary to establish a link for directional communication; the DTI can also be used for beamforming training between stations (both SLS and BRP phases) and to perform the BRP phase with the PCP/AP. Since the BRP phase follows the SLS one, a reliable frame exchange is ensured and a station may transmit BRP packets along with other packets.

8.1.3 *Data transmission*

The DTI is made up of contention-free Service Periods (SPs) for exclusive communication between a dedicated pair of nodes and Contention Based Access Periods (CBAPs) where stations compete for access. SPs and CBAPs can be in any number and combination, and their scheduling is advertised by the PCP/AP through beacons in either or both the BTI and the ATI. An allocation is defined by several fields, including the type of allocation (SP or CBAP), the addresses of the source and destination STAs involved in the allocation (which can be unicast, multicast or broadcast), its total duration and starting time and the number of blocks it is made of, beamforming training information if

needed, and whether the allocation is pseudostatic, meaning that it recurs in subsequent BIs [163]. Note that this schedule is set up prior to the beginning of the DTI. Besides it, a dynamic channel time allocation mechanism allows STAs to reserve channel time in almost real-time over both SPs and CBAPs.

Contention-based access. CBAPs follow the Enhanced Distributed Channel Access (EDCA) mechanism, which is an enhanced DCF that includes mechanisms to handle traffic categories with different priorities, frame aggregation and block acknowledgments. Stations compete for access and can obtain Transmission Opportunities (TXOPs) (contention-free periods) by winning an instance of EDCA contention or by receiving a Grant frame.

The DCF is based on CSMA with Collision Avoidance (CSMA/CA): before transmitting the channel needs to be sensed idle for a minimum amount of time, namely a Distributed Interframe Space (DIFS). When the channel is sensed busy, the transmission is postponed: the STA picks a backoff counter uniformly distributed in $\{0, \ldots, W_i - 1\}$, where W_i is the size of the contention window at the i-th retransmission attempt. The contention window has a minimum value and doubles at each collision, until it saturates to a maximum value. The backoff time counter is decremented as long as the channel is sensed idle, frozen when a transmission is detected on the channel or the CBAP operation is suspended, and reactivated when the channel is sensed idle again for at least a DIFS. When the backoff counter expires, the STA accesses the channel.

In 802.11ad, the channel status is determined through combined physical and virtual carrier sensing; the latter is realized through Network Allocation Vectors (NAVs), which are counters based on the transmission duration information announced in Request-To-Send (RTS) and Clear-To-Send (CTS) frames prior to the actual exchange of data and maintain a prediction of future traffic on the medium.

The directional nature of communication at mmWaves makes the carrier sensing operation problematic [162] because there may be possible interference even though the medium was considered to be idle.

Contention-free access. SPs are contention-free periods assigned by the PCP/AP for exclusive communication between a pair of STAs. The directional communication enables the possibility of spatial sharing, i.e., simultaneous SPs involving different STAs can be scheduled, provided that they do not interfere with each other; this requires a preliminary interference assessment phase, which is coordinated by the PCP/AP. Note that building and updating the interference map may result in huge overhead in case of mobility.

SPs may be truncated and extended through the dynamic allocation mechanism.

Dynamic allocation mechanism. This mechanism is built on scheduled SPs and CBAPs with specific configuration and enables near-real-time reservation of channel time; the

dynamic allocations do not persist beyond a BI. Stations can be polled by the PCP/AP and ask for channel time, which will be granted back to back.

The dynamic mechanism also includes the possibility of truncating and extending SPs, to exploit unused channel time and finalize the ongoing communication without additional delay and scheduling, respectively. When an SP is truncated, either the relinquished channel time is used as a CBAP or the PCP/AP polls STAs so that they can ask for channel time.

8.2 The need for a mathematical model for data scheduling

The 802.11ad standard [163] provides rules for the channel access during the DTI, as described in Sec. 8.1. Performance analysis is however needed to understand how to schedule the DTI based on the traffic specifications, so as to use the type of allocation that best matches the QoS requirements. In particular, there are three main knobs available to the protocol designer:

- *Contention-based or contention free allocation.* This is the most meaningful choice since it impacts the way the medium is accessed and thus plays a direct role on the performance. SP allocations grant dedicated resources to few stations and the obtained performance only depends on the channel status, being therefore more predictable than when interference comes into play. Clearly, setting up the scheduled sessions introduces overhead and some latency, but the beam steering process is simplified since it is given by the schedule, and the STAs not involved in the SP can go to sleep and save power. On the other hand, CBAPs are distributed and robust and good for unpredictable bursty traffic. Nonetheless, carrier sensing may be problematic due to the use of directional antennas. Also, there is no power saving. SP allocations grant the use of dedicated resources and are particularly suitable for periodic reporting with QoS demands, but CBAPs may be preferable in case of less stringent QoS requirements because channel resources are available to more stations.
- *Pseudo static allocation.* In this way it is possible to decide whether the allocation will recur in successive BIs. This is very useful for predictable traffic patterns as it avoids the need to schedule the allocation every time and limits the signaling overhead.
- *Dynamic allocation.* It may prompt fast reaction to unexpected latency critical messages and accommodate bursty downstream traffic, since it allows quasi-real-time channel use. However, it has a polling overhead and the scheduled allocation over which it is applied needs to satisfy certain conditions. This feature can be

useful for unpredictable transmissions that need to be delivered with specific QoS requirements. The dynamic truncation of SP allows the exploitation of unused channel time.

Evidently, there are many elements that need to be taken into account; a mathematical model allows to understand the tradeoffs between the various system parameters and how they affect the network performance. Building a complete model is extremely challenging because there are several components to consider. In addition to modeling data transmission in both CBAP and SP allocations, it is necessary to understand in which cases the dynamic allocation mechanism yields better performance than the predefined schedule. Another aspect that should be taken into consideration is power consumption: the presence of energy constrained devices may require changes in the scheduling, e.g., in the allocation order or by assigning more SP allocations. A critical issue is represented by the beamforming training (see Sec. 8.5.1) which introduces overhead and may degrade the network performance.

This study only focuses on the performance that can be obtained in CBAP allocations, taking into account the presence of SPs allocations too. This is intended to represent a first step in the process of understanding and characterizing the various types of allocations that can be used in 802.11ad with the ultimate goal of designing an efficient allocation scheduler able to cope with heterogeneous traffic patterns and requirements.

The remainder of this section briefly describes the state of the art.

8.2.1 *Related work*

The seminal work of [166] introduces a useful MC model of the IEEE 802.11 DCF. Although several variations on such model have been proposed to account for, e.g., finite number of retransmissions [168], heterogeneous QoS [169] and hidden node problem [170], none of them can be readily applied to the hybrid MAC layer of IEEE 802.11ad, as different changes are needed to account for its peculiar features.

Some works in the literature propose adaptations of Bianchi's model for 802.11ad. Most of them, however, do not model the effect of directional communication properly, as they neglect the deafness and hidden node problems. For example, [171] uses a 3-dimensional MC model to analyze the channel utilization and the average MAC layer delay that can be obtained in CBAPs. This model accounts for the presence of allocations other than CBAP and for the fact that backoff counters are frozen when CSMA/CA operation is suspended. However, it does not introduce the maximum contention window size, so that the contention window keeps doubling at each retransmission stage. Moreover, the model assumes that CBAPs are allocated to sectors, so that two STAs belonging to different sectors cannot compete for the channel time in the same allocation. According to

the standard [163], this is not necessarily true, since any subset of stations can participate in a CBAP, with potential deafness and hidden node issues. The model also erroneously assumes that all STAs in the same sector can overhear the messages that other nodes exchange with the AP. Thus, the assumption made in [171] strongly affects the analysis of the delay and the impact of the number of sectors used by the PCP/AP on the system performance, as the role of directional transmissions and deafness is neglected.

Similar assumptions concerning the deafness and hidden node problem have been made in [172], which models CBAPs with a 2-dimensional MC for unsaturated sources considering also the contention-free allocations of 802.11ad. However, besides neglecting the deafness problem, the model assumes that the DTI is made of SP allocations followed by a single CBAP allocation at the end of the DTI, while the standard [163] envisages SP and CBAP allocations in any number and order. This assumption may strongly affect the delay, as different configurations of the DTI may yield different performance. Also [173] uses a 2-dimensional MC to analyze the saturation throughput in CBAP but neglects the deafness issues and assumes the same specific configuration of the DTI as in [172].

A correct approach to directional communication in WiFi networks is presented in [174], which however is not designed for 802.11ad so that it does not consider backoff counter freezing and the presence of SP allocations. The model considers an accurate model for directional transmission, with the presence of side lobes with small antenna gain and corresponding regions with different levels of interference. Also [167] takes into account deafness and hidden node problems, and subdivides the area around a STA based on the interference level; CBAPs are then modeled using a 3-dimensional MC.

Other works in the literature consider different aspects of the DTI. For example, [175] derives the theoretical maximum throughput for CBAPs when two-level MAC frame aggregation is used. [176] proposed a directional MAC protocol to be used on top of 802.11ad: it allows the use of sequential directional RTS messages that a STA sends in all directions and that can therefore be overheard by all other STAs. The beamforming issue is considered in [177], which proposes a joint optimization of beamwidth selection and scheduling to maximize the effective network throughput.

For what concerns SPs, an accurate mathematical model for their preliminary allocation is presented in [178]. It considers the presence of quasi-periodic structures with multiple blocks within the same allocation, the erroneous nature of the wireless medium, and the possibility of multiple consecutive transmissions within the same allocation. A 3-dimensional MC is used to model a Variable Bit Rate (VBR) flow with packets arriving in batches of random size at regular intervals and can be used to derive the optimal SP allocation that satisfies the QoS requirement.

8.3 System model

We denote as T_{BI} the duration of a BI and as T_{BHI}, T_{CBAP} and T_{SP} the time dedicated to BHI, CBAPs and SPs during a BI, respectively. The total time T_{CBAP} dedicated for contention-based access in a BI is distributed among N_{CBAP} allocations with same duration, while T_{SP} is distributed among N_{SP} allocations with same duration.

The model is based on two assumptions: i) all STAs in the network implement a single access category, hence service differentiation is not considered, and ii) the beamforming training has already been performed, so that the STAs already know how to steer their antennas to communicate with the AP. Also, it only focuses on the classic WiFi network where a certain number of STAs communicate solely with the AP; the RTS/CTS mechanism is used.

To gauge the performance that can be obtained in a CBAP, we leverage on Bianchi's seminal work [166] and adapt it to model the features of 802.11ad's CBAPs. First of all we explain how directionality affects the communication during the contention-based channel access, we then describe the proposed model, and finally we discuss the performance metrics used in the numerical evaluation.

8.3.1 *Directional communication in CBAPs*

Besides the need of beamforming training and beam tracking mechanisms, the directional nature of communication in 802.11ad implies substantial changes also from a data transmission perspective. As explained in Sec 8.1.3, CBAPs are based on the EDCA; however, the traditional approaches used in the literature need to be adapted to take directionality into account, since the consequent deafness and hidden node problems may significantly affect the system performance.

The most widely used approach in the literature to model the DCF and EDCA mechanisms is Bianchi's model [166]. It takes the perspective of a target node and models the backoff process as a two dimensional MC, where state (i, k) refers to the i^{th} backoff stage with the backoff counter $k \in \{0, \ldots, W_i - 1\}$, where W_i is the duration of the contention window at the i^{th} retransmission attempt. The counter is decremented with probability 1 whenever the channel is sensed idle; when it reaches 0, the STA attempts to transmit. The time spent in each state depends on what happens in the channel meanwhile, as it may be idle, used for a successful transmission, or used simultaneously by colliding STAs. The original model was proposed for omnidirectional communication, so that each STA is aware of ongoing transmissions and can defer its own when it senses the channel to be idle. Collisions only happen when multiple STAs access the channel

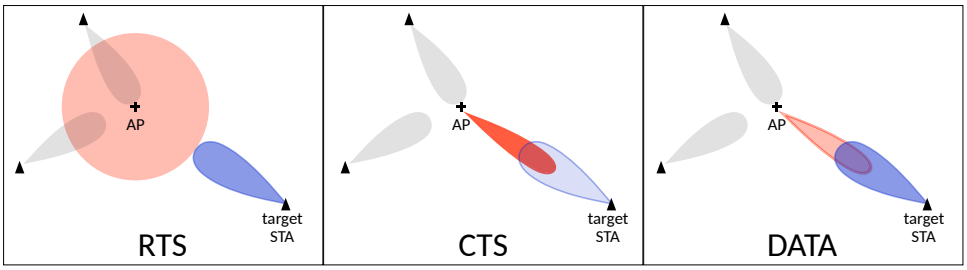


Figure 8.2: Communication phases between the AP and a target STA and two other STAs that listen in direction of the AP. Darker beams indicate a transmission, while lighter ones indicate that the device is listening. The target STA directionally transmits the RTS while the AP is listening in QO mode; then the AP steers its transmitting antennas towards the target STA and sends the CTS, which then replies transmitting a data message.

simultaneously because their backoff counters expired (at least two STAs are in a state $(\cdot, 0)$).

In the case of directional communication, however, STAs may not hear ongoing transmissions, resulting in a much higher collision probability. In this work, we assume that the RTS/CTS mechanism is used. Since a STA communicates only with the AP, it always has both its transmitting and receiving antenna patterns configured towards it. The AP instead listens to the channel in a quasi-omnidirectional (QO) mode, and, upon the reception of the RTS, it switches its antenna configuration to point towards the STA that sent it. Fig. 8.2 illustrates the direction of the various phases of the communication between a STA and the AP. Note that the messages can be heard only by a limited number of other STAs. The received power P_{rx} at a STA is in fact

$$P_{\mathrm{rx}} = P_{\mathrm{tx}} \frac{g_{\mathrm{tx}}(\theta_{\mathrm{tx}}, \varphi_{\mathrm{rx}}) g_{\mathrm{rx}}(\theta_{\mathrm{tx}}, \varphi_{\mathrm{rx}})}{A d^{\eta}} \tag{8.1}$$

where P_{tx} is the power used to transmit, d is the distance from the transmitter, η is the path-loss coefficient, A is a normalizing path-loss term, and g_{tx} and g_{rx} are the antenna gains of the transmitter and receiver, respectively. They both depend on the direction of the antennas with respect to the line of sight between the two STAs, thus on the angles θ_{tx} and φ_{rx}. If the gains are very small, P_{rx} may be too low and nothing is heard at the receiver.

Consider a network consisting of n STAs and a target STA that communicates with the AP, so that the STA and the AP point to each other, and the antenna gains in the other directions are minimal. It is possible to cluster the other $n-1$ STAs into four groups:

- $n_{I,1}$: STAs that can overhear the messages sent from the target STA to the AP but not those sent from the AP to the STA.
- $n_{I,2}$: STAs that can overhear the messages from the AP to the target STA but not those from the STA to the AP.

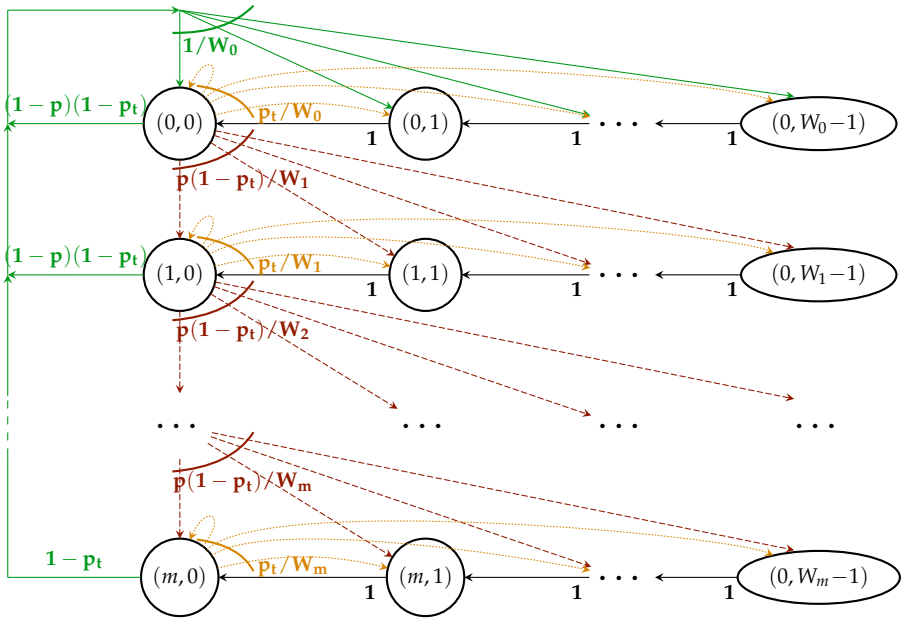


Figure 8.3: Macro Markov chain (adaptation of Bianchi's model [166]).

- $n_{I,3}$: STAs that can overhear the whole communication between the AP and the target STA.
- $n_{I,4}$: STAs that cannot overhear any messages exchanged between the AP and the target STA.

Analogously, from the perspective of a STA that listens to the channel, the other STAs can be divided into four groups $n_{O,1}$ (STAs of which it can hear the messages to the AP but not the messages that the AP sends to them), $n_{O,2}$ (STAs of which it cannot hear the messages to the AP but can hear those that the AP sends to them), $n_{O,3}$ (STAs of which it can hear all the messages exchanged with the AP), and $n_{O,4}$ (STAs whose messages exchanged with the AP cannot be heard).

Consequently, collisions can happen at three different stages of the uplink communication from a target STA to the AP.

1. The target STA accesses the channel to transmit its RTS, but collides for sure. This can happen for three different reasons: i) if any other STA accesses the channel at the same time, as in the legacy WiFi, ii) if a STA belonging to groups $n_{O,2}$ or $n_{O,4}$ is transmitting the RTS to the AP, or iii) if a STA in group $n_{O,4}$ has already sent the RTS and is going on with the communication with the AP. Notice that, in the latter case, the transmission of the target STA fails, because the AP is listening in the direction of that STA[1], but the ongoing transmission may still be successful, as

1 Different considerations can be made when considering Multiple-Input Multiple-Output (MIMO) systems, but this out of the scope of this work.

the directionality highly attenuates the interference. In this work we assume that, except for errors in the channel, the ongoing data transmission is successful.

2. If none of the previous conditions happened, the transmission of the RTS may still be vulnerable to interference. This happens when a STA in groups $n_{I,2}$ or $n_{I,4}$ accesses the channel meanwhile. The two packets will then collide.
3. If the transmission of the RTS was successful, the AP sends the CTS and the target STA can proceed with the data transmission. However, a STA in group $n_{I,4}$ is unaware of the ongoing communication and may try to access the channel. As assumed in case 1iii), the outcome of the ongoing transmission only depends on channel errors, while the STA that accesses the channel will register a destructive collision.

In the remainder of this section, we propose an adaptation of Bianchi's model that accounts for the directionality of transmission, assuming that the regions corresponding to the four groups of nodes are known; Sec. 8.5 introduces an analytical model to compute such regions when the antenna gain outside the main lobe is zero.

8.3.2 *Rethinking Bianchi's model*

Bianchi's model [166] needs three major adaptations in order to be suitable for 802.11ad, which are caused by the following features.

1. CBAPs can be interrupted because there is a scheduled SP or the BHI of the next BI. In this case, all backoff counters have to freeze [163]; they will be restored in the next CBAP. This affects the time that a STA spends in a state (i,k), $k \in \{1,\dots,W_i-1\}$ before decrementing its backoff counter and transitioning to $(i,k-1)$; the transition probability from (i,k) to $(i,k-1)$ is 1 as in Bianchi's original model. We denote the freezing probability as p_f.
2. The finite duration of CBAPs may also cause transmissions deferral. In fact, if the backoff counter of a STA reaches 0 but there is not enough time to complete a transmission, that STA should refrain from transmission. The standard [163] however does not specify how to handle the backoff counter in this case. A possibile strategy consists in freezing it and then restoring it in the next CBAP, as in case 1. However, such approach may yield a high number of collisions if this situation happened for multiple STAs (all STAs whose backoff counter expires during the time needed for a complete transmission at the end of the current CBAP cumulate), which therefore would all access the channel simultaneously in the next CBAP. To avoid this, we propose to use the same approach used in [172], so that a new backoff counter is randomly chosen from the current window (no collision happened). This causes the addition of new transitions from state $(i,0)$ to (i,k), $k \in \{0,\dots,W_i-1\}$.

We denote the probability of insufficient time in the current CBAP as timeout probability p_t.

3. As discussed in Sec. 8.3.1, the directional nature of mmWave communication has a huge impact on the operation of the DCF mode because of deafness and the increased hidden node problem. This modifies the collision probability and the time spent in each state, which depend on the behavior of STAs whose transmissions can be detected by the target STA.

Fig. 8.3 represents the MC we propose to model the behavior of a STA during a CBAP. As in Bianchi's model, a state (i,k), $i \in \{0,\dots m\}$, $k \in \{0,\dots,W_i-1\}$ refers to the i^{th} backoff stage with the backoff counter being equal to k. Here, m is the maximum number of retransmissions. The contention window in stage i is $W_i = \min\{2^i W_0, 2^{m'} W_0\}$, where the initial window W_0 and the maximum window $2^{m'} W_0$ are defined in the standard.

From a state (i,k), $k>0$, the backoff counter is decremented with probability 1 (solid black transitions in Fig. 8.3), but the time needed to transition to the next state $(i,k-1)$ is variable, depending on how the channel is being used. When it reaches a state $(i,0)$, the STA might be constrained to defer its transmission (case 2). The residual time in the current CBAP is uniformly distributed in $[0, T_{\text{CBAP}}/N_{\text{CBAP}}]$, where T_{CBAP}/N_{CBAP} is the average duration of a CBAP allocation in the BI. Then, the probability that there is no sufficient time to complete a transmission of duration T_L can be approximated as

$$p_t = \frac{T_L}{T_{\text{CBAP}}/N_{\text{CBAP}}}. \tag{8.2}$$

Thus, from each state $(i,0), i \in \{0,\dots m\}$, the MC transitions to a state (i,k), $k \in \{0,\dots,W_i-1\}$ with probability p_t/W_i (dotted orange transitions in Fig. 8.3), while with probability $1-p_t$ the STA accesses the channel. We identify such latter condition as being in a *transmission state* (the MC is in a state $(i,0)$ and attempts to transmit); when the other condition applies (transmission deferral) or the MC is in a state (i,k), $k>0$, we say that the STA is in a non-transmission state. As in [176], each transmission state is itself a MC, which will be described in Sec. 8.3.3; thus, in order not to generate confusion, we will refer to the MC of Fig. 8.3 as *macro MC*.

Let p be the failure probability, which includes the collision probability (see the discussion in Sec. 8.3.1) and the error probability due to the wireless channel, as explained later. Then, from state $(k,0)$ the MC goes to a state $(0,i), i \in \{0,\dots,W_0-1\}$ (successful transmission of a new packet; solid green transitions in Fig. 8.3) with probability $(1-p_t)(1-p)/W_0$, or to any state $(k+1,i), i \in \{0,\dots,W_{k+1}-1\}$ with probability $(1-p_t)p/W_{k+1}$ (dashed red transitions in Fig. 8.3). If it reaches the maximum number of

retransmission attempts ($k = m$), the MC goes from state $(m, 0)$ to a state $(0, i)$, $i \in \{0, \ldots, W_0\}$ with probability $(1 - p_t)/W_0$.

It is then possible to compute the steady-state probabilities $\{b_{i,k} : i \in \{0, \ldots, m\}, k \in \{0, \ldots W_i - 1\}\}$ of the macro MC, using the same approach as used in [166]. Assuming $p_t < 1$, it is

$$b_{i,k} = \frac{W_i - k}{W_i} p\, b_{0,0}\,, \tag{8.3}$$

where $b_{0,0}$ is

$$b_{0,0} = \begin{cases} \dfrac{2(1-2p)(1-p)}{W_0(1-(2p)^{m+1}(1-p) + (1-p^{m+1})(1-2p)} & \text{if } m \leq m' \\ \dfrac{2(1-2p)(1-p)}{W_0(1-(2p)^{m'+1})(1-p) + (2^{m'}W_0(p^{m'} - p^m)p + (1-p^{m+1}))(1-2p)} & \text{if } m > m' \end{cases} \tag{8.4}$$

Notice that $b_{0,0}$ does not depend on p_t, which, nonetheless, has an impact on the delay.

The probability of being in a transmission state is then

$$\tau = \sum_{i=0}^{m} b_{i,0}(1 - p_t) = \frac{1 - p^{m+1}}{1 - p}(1 - p_t) b_{0,0}\,. \tag{8.5}$$

The time spent, on average, in a transmission or non transmission state is denoted as $\mathrm{E}[T_{tx}]$ and $\mathrm{E}[T_{ntx}]$, respectively, which depend on the probabilities $\{b_{i,k}\}$ as explained next. It is then possible to define the probability π_{tx} that, in an arbitrary time instant, the macro MC is in a transmission state:

$$\pi_{tx} = \frac{\tau \mathrm{E}[T_{tx}]}{\tau \mathrm{E}[T_{tx}] + (1 - \tau)\mathrm{E}[T_{ntx}]}\,. \tag{8.6}$$

With probability $1 - \pi_{tx}$, in an arbitrary time instant, the MC will be in a non transmission state.

To derive $\mathrm{E}[T_{tx}]$ and $\mathrm{E}[T_{ntx}]$ it is first necessary to understand what happens in a transmission state.

8.3.3 *A transmission state*

Whenever a STA is in a transmission state $(i, 0)$, $i \in \{0, \ldots m\}$, it attempts to transmit with probability $1 - p_t$, otherwise it picks a new backoff counter from the same window W_i (see (8.2)) and enters a so-defined transmission state.

To model such behavior, each transmission state forms its own MC, similarly to the model proposed in [176]. The MC is made of 6 states: the *access* state A, the *collision RTS* state R_c, the *vulnerable RTS* state R_v, the *ongoing transmission* state O, the *failure* state F,

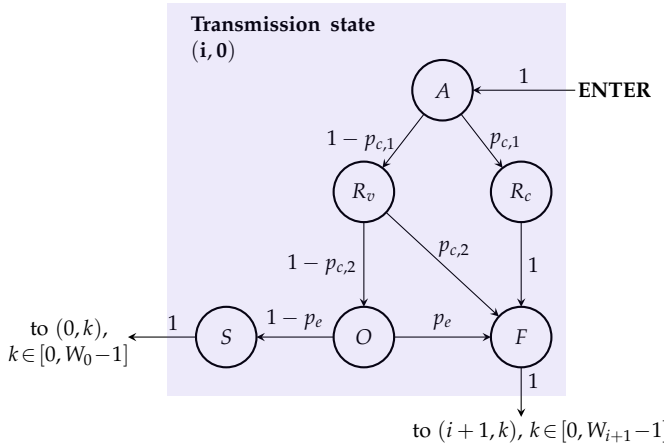


Figure 8.4: MC that models a transmission state. It is entered from i) state $(i, 1)$ with probability $1 - p_t$, ii) state $(i - 1, 0)$ with probability $p(1 - p_t)/W_0$ or iii) state $(i, 0)$ itself with probability p_t/W_0.

and the failure state S. A STA goes into state A when it accesses the channel and goes from the macro MC into the transmission state MC. It transmits the RTS to the AP, and, based on the discussion in Sec. 8.3.1, two cases can occur.

- As soon as the STA accesses the channel, it may immediately collide (case 1 in Sec. 8.3.1). This happens with probability $p_{c,1}$; in this case the STA transitions to state R_c where it transmits the RTS and then, with probability 1, goes to the failure state F.
- Otherwise, the transmission of the RTS is still vulnerable to interference. If it collides (case 2 of Sec. 8.3.1), the MC transitions to the failure state F; this happens with probability $p_{c,2}$. Otherwise, the STA goes to state O, where it receives the CTS from the AP and then sends its data. In turn, the data transmission may fail because of channel errors (but not because of interference, as assumed in Sec. 8.3.1) and therefore, with probability p_e, the next state in the MC is F. Otherwise, the transmission is successful and the next state in the MC is S. Then, from either F or S, the STA exits the transmission state.

The resulting MC is represented in Fig. 8.4. Let b_j be the steady-state probability that the MC is in state $j \in \mathcal{J}_{\text{tx}} \triangleq \{A, R_c, R_v, O, F, S\}$. Since the transmission state itself forms a repetitive MC, it yields: $b_A = 1/b_{\text{tx}}$, $b_{R_c} = (1 - p_{c,1})/b_{\text{tx}}$, $b_{R_v} = p_{c,1}/b_{\text{tx}}$, $b_O = (1 - p_{c,1})(1 - p_{c,2})/b_{\text{tx}}$, $b_F = (1 - (1 - p_{c,1})(1 - p_{c,2})(1 - p_e))/b_{\text{tx}}$, $b_S = (1 - p_e)(1 - p_{c,1})(1 - p_{c,2})/b_{\text{tx}}$, where $b_{\text{tx}} = 3 + p_e(1 - p_{c,1})(1 - p_{c,2})$.

Similarly to what done for the macro MC it is possible to define the probabilities π_j that, in an arbitrary time instant, given that the MC is in a transmission state, the MC is in a state $j \in \mathcal{J}_{\text{tx}}$:

$$\pi_j = \frac{T_j b_j}{\sum_{\ell \in \mathcal{J}_{\text{tx}}} T_\ell b_\ell} \quad j \in \mathcal{J}_{\text{tx}}\,, \tag{8.7}$$

where T_j is the time spent in state j. Through b_j, the probabilities $\pi_j, j \in \mathcal{J}_{\text{tx}}$ depend on the collision and error probabilities $p_{c,1}$, $p_{c2,}$ and p_e. The collision probabilities in turn depend on how many and which other STAs access the channel while the target STA is in the transmission state, and thus on π_j.

Considering the model in Sec. 8.3.1 where STAs can be grouped based on what they can hear of a communication between the AP and another STA, $p_{c,1}$ is given by the probability that none of these three cases verifies: i) any of the other STAs accesses the channel simultaneously, ii) at least a STA in group $n_{O,2}$ is transmitting an RTS to the AP, or iii) at least a STA in group $n_{O,4}$ is using the channel. We analyze the probabilities of these events to happen.

The first case verifies if at least another STA is accessing the channel, given that the target STA is accessing the channel. The probability of this to happen can be expressed as:

$$q_1 = \frac{1-(1-p_{\text{acc}})^n - np_{\text{acc}}(1-p_{\text{acc}})^{n-1}}{p_{\text{acc}}}, \tag{8.8}$$

where n is the total number of STAs in the network and $p_{\text{acc}} = \pi_A \pi_{\text{tx}}$ is the probability that a STA is accessing the channel.

Case ii) happens if at least a STA in group $n_{O,2}$ is either in state R_v or in R_c, given that the target STA is accessing the channel. Thanks to Bayes'rule, the probability of this to verify can be equivalently expressed as a function of the probability that the target STA accesses the channel given that at least a STA in group $n_{O,2}$ is either in state R_v or in R_c. This is certainly larger than the probability of the target STA to access the channel without the condition on the other STAs, but is not trivial to compute, because it requires an analytical expression for the relations between the coverage areas of multiple STAs. In fact, if a STA in group $n_{O,2}$ entered a transmission state, all STAs that can hear it refrain from transmitting, so that the number of STA that compete for the channel is reduced and it is more likely that the target STA senses the channel as idle and attempts transmitting. However, we do not consider such relations, which are extremely challenging to model, but only account for the fact that, if some STAs are in a transmission state, the ECDA operation is not frozen, so that the access probability is increased by a factor $1/(CBAP/BI)$. We thus express the probability of case ii) as

$$q_2 = \frac{1-(1-\pi_{\text{tx}}(\pi_{R_v}+\pi_{R_c}))^{n_{O,2}}}{CBAP/BI}. \tag{8.9}$$

As the numerical evaluation of Sec. 8.6 shows, this approximation affects the validity of the model only for highly dense scenarios, with more than 100 STAs.

Finally, case iii) can be treated analogously to case ii) and thus happens with probability

$$q_3 = \frac{1-(1-\pi_{tx}(\pi_{R_v}+\pi_{R_c}+\pi_O))^{n_{O,A}}}{CBAP/BI}. \tag{8.10}$$

Then, it is

$$p_{c,1} = 1-(1-q_1)(1-q_2)(1-q_3). \tag{8.11}$$

since there is a collision if at least any among cases i), ii), iii) verifies.

When the target STA did not collide while attempting to transmit and is thus in state R_v, it collides if at least a STA that cannot hear the uplink messages sent by the target STA accesses the channel during the whole duration of R_v. Following the same reasoning as per (8.9) and (8.10), this happens with probability

$$p_{c,2} = \frac{1-((1-p_{\text{acc}})^{n_{I,2}+n_{I,A}})^{T_{R_v}/T_A}}{CBAP/BI}. \tag{8.12}$$

Eqs. (8.7), (8.11), (8.12) form a nonlinear system in the unknowns $\pi_j, p_{c,1}$ and $p_{c,2}$, which can be solved using numerical techniques, as in the Bianchi's original model. The error probability p_e instead depends on the SNR and the MCS used.

8.4 Performance metrics

We evaluate the performance achievable in a CBAP in terms of throughput and delay. Before delving into their description, it is useful to derive the time spent in a transmission and non transmission state.

8.4.1 *Average time spent in a transmission state*

A STA that accesses a transmission state can follow 4 different paths, depending on collision and errors. The average time spent in a transmission state is thus the sum of the time associated to each of this paths, weighed for the probability of that path to happen:

$$\begin{aligned}\mathrm{E}[T_{tx}] = {} & (T_A+T_{R_c}+T_F)p_{c,1} \\ & +(T_A+T_{R_v}+T_F)(1-p_{c,1})p_{c,2} \\ & +(T_A+T_{R_v}+T_O+T_F)(1-p_{c,1})(1-p_{c,2})p_e \\ & +(T_A+T_{R_v}+T_O+T_S)(1-p_{c,1})(1-p_{c,2})(1-p_e).\end{aligned} \tag{8.13}$$

This can be easily seen in Fig. 8.4.

The probabilities π_j are defined in (8.7) and the times T_j are as follows: $T_A = \delta$, $T_{R_c} = RTS$, $T_{R_v} = RTS$, $T_O = CTS + \mathrm{E}[T_L] + ACK + 3SIFS + 3\delta, T_F = DIFS, T_S = DIFS$, where δ is the propagation delay, RTS and CTS represent the time needed to send an

RTS and CTS message, respectively, $\mathrm{E}[T_L]$ is the average time needed to transmit a data packet, ACK is the time to send an ACK, and $SIFS$ and $DIFS$ represent the Short Interframe Space (SIFS) and DIFS durations, respectively [163].

8.4.2 *Average time spent in a non-transmission state*

The time spent in a non-transmission state depends on what happens meanwhile: the CBAP may freeze, the target STA may hear a transmission or sense the channel as idle. The EDCA mechanism assumes that the backoff counter is decremented only after the channel is sensed idle for a time slot of duration σ (which is defined in the standard and depends on the PHY layer). Before that, the CBAP may freeze or be busy. We can interpret the freezing condition as a self-loop on a state (i,k), $k>0$ with probability

$$p_f = 1 - \frac{T_{\mathrm{CBAP}}}{T_{\mathrm{BI}}}, \tag{8.14}$$

so that $1/(1-p_f)$ iterations over (i,k) are expected before a transition to $(i,k-1)$.

The target STA senses the channel as idle when none of the STAs in groups $n_{I,1}$ and $n_{I,3}$ is using the channel (i.e., is in any of the states A, R_c, R_v, O) and none of the STAs is using the channel *and* has already received a feedback from the AP (i.e., it is in state O):

$$p_i = (1-\pi_{\mathrm{tx}}(\pi_A+\pi_{R_c}+\pi_{R_v}+\pi_O))^{n_{I,1}+n_{I,3}}(1-\pi_{\mathrm{tx}}\pi_O)^{n_{I,2}}, \tag{8.15}$$

The channel is sensed as busy with probability $1-p_i$ for an average duration of E_{tx}. Thus, the average time spent in a non-transmission state can be expressed as

$$\mathrm{E}[T_{\mathrm{ntx}}] = \sigma + \frac{(1-p_i)E_{\mathrm{tx}}}{1-p_f}. \tag{8.16}$$

8.4.3 *Throughput*

The normalized system throughput S is defined as the fraction of time that the channel is used to successfully transmit information. The average payload size is $\mathrm{E}[L]$ and a transmission is successful with probability $\pi_{\mathrm{tx}}(1-p)$. Thus the aggregate throughput is

$$S = n\frac{\pi_{\mathrm{tx}}(1-p)\mathrm{E}[L]}{\pi_{\mathrm{tx}}\mathrm{E}[T_{\mathrm{tx}}]+(1-\pi_{\mathrm{tx}})\mathrm{E}[T_{\mathrm{ntx}}]} \tag{8.17}$$

where the denominator represents the average duration of a time slot and n is the number of STAs in the network.

8.4.4 *Delay*

The delay experienced by a (successfully transmitted) packet is the time elapsed from when it arrived at the MAC layer until it is received. Let $\mathrm{E}[D_i]$ denote the expected delay that a packet experiences when it is successfully transmitted at stage i, TX(i) the event of transmission at stage i, and success the event of a successful transmission. Then

$$\mathrm{E}[D] = \sum_{i=0}^{m} Pr(\mathrm{TX}(i)|\text{success})\mathrm{E}[D_i] \tag{8.18}$$

where $Pr(\mathrm{TX}(i)|\text{success})$ represents the probability that, given that a successful transmission happened, it was at stage i. The event success happens when the packet is not discarded after m backoff stages, i.e., with probability p^{m+1}. Thus $Pr(\mathrm{TX}(i)|\text{success}) = (1-p)p^i/(1-p^{m+1})$ since the packet was discarded at stages $0, 1, \ldots, i-1$ and then successfully transmitted at stage i. The term $\mathrm{E}[D_i]$ is the sum of the average backoff process delay in stages $0, 1, \ldots, i$, the collision delay experienced in stages $0, 1, \ldots, i-1$ and the time needed for the successful transmission at stage i. The first state k in the j^{th} backoff stage is uniformly distributed between 0 and $W_j - 1$; the counter is decremented until state $(j, 0)$ ($k+1$ states are crossed) and then, with probability p_t there is a transition back to a random state at stage j. Therefore, it is:

$$\begin{aligned}
\mathrm{E}[D_i] &= iT_c + T_s + \mathrm{E}[T_{\mathrm{ntx}}] \sum_{j=0}^{i} \sum_{\ell=0}^{+\infty} p_t^{\ell} \sum_{k=0}^{W_j-1} \frac{k+1}{W_j} \\
&= iT_c + T_s + \frac{\mathrm{E}[T_{\mathrm{ntx}}]}{1-p_t} \sum_{j=0}^{i} \frac{W_j+1}{2} \\
&= iT_c + T_s + \frac{\mathrm{E}[T_{\mathrm{ntx}}]}{2(1-p_t)} (2^{\min(i,m')+1} - 1 + \max(i-m', 0)2^{m'}) W_0
\end{aligned} \tag{8.19}$$

where $\mathrm{E}[T_{\mathrm{ntx}}]$ is the time spent in a backoff state and T_c and T_s are the durations of a successful transmission and a collision, respectively:

$$T_s = RTS + CTS + \mathrm{E}[T_L] + ACK + 3SIFS + 4\delta\,, \tag{8.20}$$

$$T_c = RTS + DIFS + \delta\,. \tag{8.21}$$

8.5 A model for directional communication

The model of Sec. 8.3.1 assumes to know the number of STAs that can overhear the uplink and downlink messages exchanged between the AP and a target STA, which is equivalent to characterize the regions around the target STA corresponding to groups $n_{I,1}$, $n_{I,2}$, $n_{I,3}$ and $n_{I,4}$. This is not trivial to compute as the power received at a STA

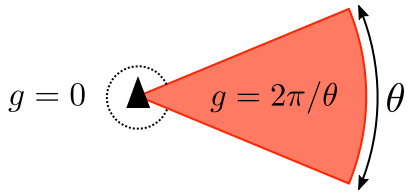


Figure 8.5: Pizza-slice beam of width θ. The antenna gain is $2\pi/\theta$ within the beam, 0 outside.

depends on the gains of the transmitting and receiving antennas, as per (8.1), which vary according to the considered direction (angles θ_{tx} and φ_{rx} in (8.1)). In the following, we describe the model we use for the beam shapes and then provide a mathematical approach to compute the areas corresponding to each group of STAs.

8.5.1 *Beam shapes*

The directivity of an antenna depends on the shape of a beam. There exists a multitude of mathematical models for antenna beams, such as the Gaussian beam shape, the sinc beam shape and the sampled beam shape, which, however, are very challenging to be used in mathematical models. A simpler approach is given by the constant gain beam shape (sometimes called pizza-slice beam shape), which is quite popular in complex analytical papers due to its simplicity. The space around the device is divided into N_b beams with constant beamwidth $W_b = 2\pi/N_b$; a beam has constant gain in the main lobe and there are no side lobes. From the expression of the directivity of an antenna [179], the antenna gain for a beam centered at φ is $g(\theta) = N_b$ if $\theta \in \left[\varphi - \frac{W_b}{2}, \varphi + \frac{W_b}{2}\right]$, and 0 otherwise.

We assume homogeneous STAs with the same antenna gains; however it makes sense to consider the AP to be more powerful than the STAs and with narrower beams. Thus, we denote as N_{AP} and N_S the number of sectors for the AP and a STA, respectively. We assume that $N_{AP} \geq 2$ and $N_S \geq 2$.

As explained in Sec. 8.3.1, since the STAs only communicate with the AP, they always have their transmitting and receiving antennas directed towards it.[2] The AP instead listens in a QO mode and switches to directional mode when engaged in a communication with a STA. In this work, we assume that the QO mode coincides with omnidirectionality, and leave to future work the investigation of smaller widths.

We also assume full transmitter/receiver reciprocity, meaning that the a STA uses the same sector to transmit to and receive from the AP, and vice versa. The antenna gains of the AP and STA in directional mode computed with the model in 8.5.1 are N_{AP} and N_S in the main lobes, respectively, and zero outside (see Fig. 8.5).

2 Except of course during the beamforming training, which is however out of the scope of this paper.

As a final remark, we highlight that, as in most of the literature, we consider only 2D directivity, which highly simplifies the problem, although antennas clearly have a 3D radiation pattern.

8.5.2 *Coverage area and power regulations*

Since there are no side lobes, two STAs can hear each other only if they are in each other's main lobe, respectively. Given this and considering the average, it is possible to derive a maximum transmission range by means of a threshold γ_{th} on the SNR $\gamma = P_{rx}/N$, where N is the noise power. Then, using (8.1), the distance d between two devices should be

$$d \leq \left(\frac{P_{tx} g_{tx}(\theta_{tx}, \varphi_{rx}) g_{rx}(\theta_{tx}, \varphi_{rx})}{\gamma_{th} A N} \right)^{1/\eta} . \tag{8.22}$$

The threshold γ_{th} can be computed by imposing a maximum tolerable Bit Error Rate (BER) and deriving the corresponding SNR (note that this depends on the MCS used). The antenna gains of the AP and STA are computed as described in 8.5.1.

Interestingly, if the AP and the STA have different transmission powers, there is an SNR asymmetry between downlink and uplink when considering the same noise level at receiver and transmitter (see (8.1) and (8.22)). We assume the coverage radius R to be bounded by the most stringent limit (8.22) between uplink (P_{tx} and g_{tx} are those of the STA, g_{rx} is that of the AP which can be listening either in QO or directional mode) and downlink communication (P_{tx} and g_{tx} are those of the AP, g_{rx} is that of the STA). Then, we consider an area $\mathcal{R} = \pi R^2$ around the AP and the STAs uniformly distributed according to a PPP of intensity λ.

8.5.3 *Stations that overhear uplink messages*

We consider a Cartesian plane whose origin coincides with the center of area $\mathcal{R}$, so that the AP is in $(0, 0)$.Without loss of generality, we assume that the target STA is in $(d_t, 0), d_t \in [0, R]$. Considering the beam model of Sec. 8.5.1, the interferer can overhear uplink communication from the target STA to the AP if it is in the main lobe of the target STA and vice versa, otherwise the received power is 0 as per (8.1). Consider an interferer STA at distance $d_i \in [0, R]$ from the AP. It can overhear the uplink communication if and only if the phase of its polar coordinates is in the range $[\varphi_{\text{lim}}(d_i), 2\pi - \varphi_{\text{lim}}(d_i)]$, where

$$\varphi_{\text{lim}}(d_i) = \begin{cases} \pi - \frac{\theta_S}{2} - \arcsin\left(\frac{d_i}{d_t} \sin\left(\frac{\theta_S}{2} \right) \right) & \text{if } d_i \leq d_t \\ \pi - \frac{\theta_S}{2} - \arcsin\left(\frac{d_t}{d_i} \sin\left(\frac{\theta_S}{2} \right) \right) & \text{if } d_i > d_t \end{cases} \tag{8.23}$$

The proof of this result is provided in Appendix 8.7.

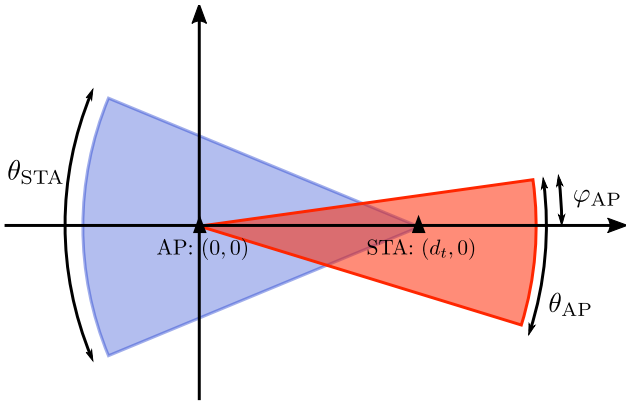


Figure 8.6: Location of the AP and the target STA in the Cartesian plane.

Considering all possible distances d_i, we obtain the expected area of STAs that can overhear uplink messages given the position of the target node $(d_t, 0)$ as

$$\begin{aligned}\mathcal{R}_R(d_t) &= \int_0^R \int_{\varphi_{\text{lim}}(d_i)}^{2\pi-\varphi_{\text{lim}}(d_i)} r\partial\theta r\,\partial r = \pi R^2 - 2\int_0^R \varphi_{\text{lim}}(r) r\,\partial r \\ &= \pi R^2 - 2\int_0^{d_t} \left(\pi - \frac{\theta_S}{2} - \arcsin\left(\frac{r}{d_t}\sin\frac{\theta_S}{2}\right)\right) r\,\partial r \\ &\quad - 2\int_{d_t}^R \left(\pi - \frac{\theta_S}{2} - \arcsin\left(\frac{d_t}{r}\sin\frac{\theta_S}{2}\right)\right) r\,\partial r\end{aligned} \tag{8.24}$$

which can be solved in closed form.

The expected area of STAs that can overhear uplink messages is obtained by averaging (8.24) over d_t:

$$\mathrm{E}\left[\mathcal{R}_R\right] = \int_0^R \mathcal{R}_R(d_t)\frac{2d_t}{R^2}\partial d_t \tag{8.25}$$

which also can be solved in closed form and only depends on the beam width θ_S.

8.5.4 *Stations that overhear downlink messages*

Without loss of generality we keep assuming that the target STA is in $(d_t, 0)$, and consider it to be in a random angular position within the AP sector that covers it, which has width θ_{AP}. We thus denote as $\varphi_{\text{AP}} \in [0, \theta_{\text{AP}}]$ the angular phase of such sector, so that it spans the angles in the Cartesian plane in the range $[\varphi_{\text{AP}} - \theta_{\text{AP}}, \varphi_{\text{AP}}]$, as shown in Fig. 8.6. The covered area is

$$\mathcal{R}_C = \int_0^R \int_{\varphi_{\text{AP}}-\theta_{\text{AP}}}^{\varphi_{\text{AP}}} r\partial\theta r\,\partial r = \pi R^2 = \frac{\theta_{\text{AP}}}{2} R^2\,. \tag{8.26}$$

All STAs in that sector can overhear downlink communication from the AP to the target STA, and, in particularly, the CTS. Note that $\mathrm{E}\left[\mathcal{R}_C\right] \equiv \mathcal{R}_C$.

8.5.5 *Stations that overhear both uplink and downlink messages*

In this case, we have to consider the area that satisfies the requirements of both Secs. 8.5.3 and 8.5.4. Thus

$$\mathcal{R}_{R,C}(d_t, \varphi_{\mathrm{AP}}) = \int_0^R \left(\int_{\varphi_{\lim}(r)}^{\varphi_{\mathrm{AP}}} \partial\theta + \int_{\varphi_{\mathrm{AP}}-\theta_{\mathrm{AP}}}^{2\pi-\varphi_{\lim}(r)} \partial\theta \right) r\, \partial r \tag{8.27}$$

which is not trivial to compute since $\varphi_{\lim}(\cdot)$ depends on the distance of the interferer from the AP. However, it is possible to obtain a closed form expression for (8.27), as explained in Appendix 8.7.

The corresponding expected area of STAs that can overhear both uplink and downlink messages is obtained by averaging (8.27) over $d_t \in [0, R]$ and $\varphi_{\mathrm{AP}} \in [0, \theta_{\mathrm{AP}}]$:

$$\begin{aligned} \mathrm{E}\left[\mathcal{R}_{R,C}\right] = \int_0^R \frac{2d_t}{R^2} \Bigg(& \int_0^{\theta_{\mathrm{AP}}} \frac{1}{\theta_{\mathrm{AP}}} \mathcal{R}_{R,C}(d_t, \varphi_{\mathrm{AP}}) \partial\phi_{\mathrm{AP}} \\ & + \int_0^{\theta_{\mathrm{AP}}} \frac{1}{\theta_{\mathrm{AP}}} \mathcal{R}_{R,C}(d_t, \varphi_{\mathrm{AP}}) \partial(\theta_{\mathrm{AP}} - \phi_{\mathrm{AP}}) \Bigg) \partial d_t \end{aligned} \tag{8.28}$$

Again, the calculation is not trivial as the integrals in (8.27) yield zero for some positions of the target STA and the interferers. Appendix 8.7 explains how to compute (8.28).

8.5.6 *Classification of the stations*

Given Eqs. (8.24)–(8.28), it is possible to quantify the regions $\mathcal{R}_\ell, \ell = \{1, 2, 3, 4\}$ corresponding to the groups of nodes $n_{I,\ell}$ introduced in Sec. 8.3.1:

$$\mathcal{R}_1 = \mathrm{E}\left[\mathcal{R}_R\right] - \mathrm{E}\left[\mathcal{R}_{R,C}\right] \tag{8.29}$$

$$\mathcal{R}_2 = \mathrm{E}\left[\mathcal{R}_C\right] - \mathrm{E}\left[\mathcal{R}_{R,C}\right] \tag{8.30}$$

$$\mathcal{R}_3 = \mathrm{E}\left[\mathcal{R}_{R,C}\right] \tag{8.31}$$

$$\mathcal{R}_4 = \mathcal{R} - \mathcal{R}_1 - \mathcal{R}_2 - \mathcal{R}_3\,. \tag{8.32}$$

In this work, we assume that the STAs are distributed according to a PPP. Notice that, given the symmetry of the coverage areas, it is $n_{O,\ell} \equiv n_{I,\ell}\ \forall \ell$.

8.6 Numerical evaluation

We validated the proposed model by comparing its performance in terms of throughput and delay with that of realistic Monte Carlo simulations for different system configurations.

Table 8.1: Simulation parameters.

BI structure		
BI duration	BI	100 ms
BHI duration	BHI	2 ms
EDCA parameters		
Minimum contention window size	W_0	16
Maximum contention window size	$2^{m'}W_0$	1024
Maximum # retransmission attempts	m	6
Slot duration	σ	5 μs
SIFS	$SIFS$	3 μs
DIFS	$DIFS$	13 μs
Propagation delay	δ	100 ns
Packets size		
MAC header	H_{MAC}	320 b
PHY header	H_{PHY}	64 b
RTS size	L_{RTS}	$20 * 8$ b
CTS size	L_{CTS}	$20 * 8$ b
ACK size	L_{ACK}	$14 * 8$ b
Data size	$\mathrm{E}[L]$	7995*8 b $-H_{MAC}$
Noise		
Noise figure	F_{dB}	10 dB
Bandwidth	W	2.16 GHz
Path loss exponent	η	3

The system parameters, most of which have been taken from the standard [163], are summarized in Table 8.1. The time required to send a message is computed as its size in bits (see Table 8.1) divided by the rate of the MCS used; RTS, CTS and ACK messages are sent using the control modulation, which corresponds to a rate of 27.5 Mb/s [163], while we assume to use the Single Carrier PHY layer with $mcs = 5$ for data transmission, which yields a data rate of 1251.25 Mb/s [163].

We also set a maximum BER of 10^{-6} and mapped such requirement onto a threshold γ_{th} on the SNR[3]. This allows one to estimate the area covered by the AP as per Sec. 8.5.1, where the antenna gains are derived from the number of antenna sectors, and the noise power is $N = kT_0FW$, where k is the Boltzmann constant, $T_0 = 290K$, and the noise figure F, the path loss exponent η and the bandwidth W are given in Table 8.1.

Figs. 8.7 and 8.8 show the throughput and delay, respectively, as functions of the STAs density λ for different values of $\nu \triangleq CBAP/(BI - BHI)$, which represents the fraction of DTI devoted to CBAP. Clearly, the throughput increases with ν, since there is more time for EDCA operation. Interestingly, the STA density does not have an impact on

3 We built an SNR-BER map using the WLAN Toolbox™of MATLAB software, which provides functions for modeling 802.11ad PHY.

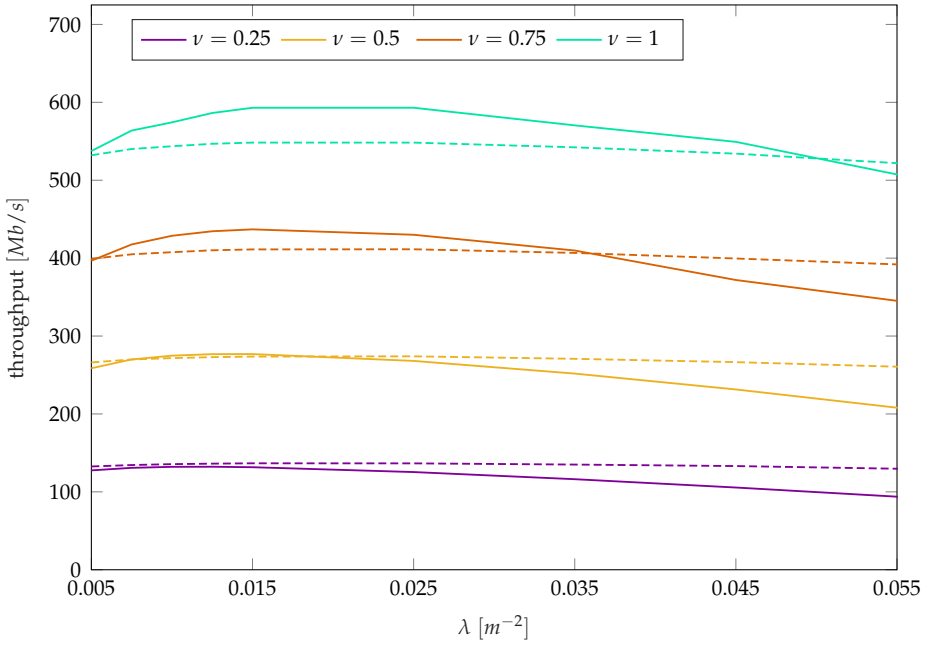


Figure 8.7: Throughput vs STAs density for different values of ν. Analytical model (solid lines) vs. simulation (dashed lines) for $n_{CBAP} = n_{SP} = 3$.

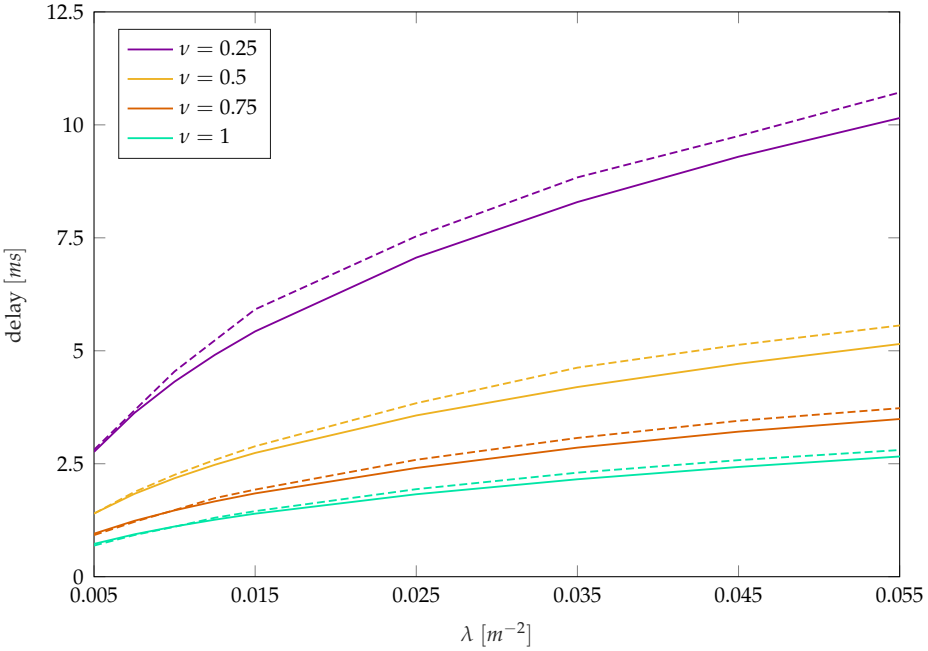


Figure 8.8: Delay vs STAs density for different values of ν. Analytical model (solid lines) vs. simulation (dashed lines) for $n_{CBAP} = n_{SP} = 3$.

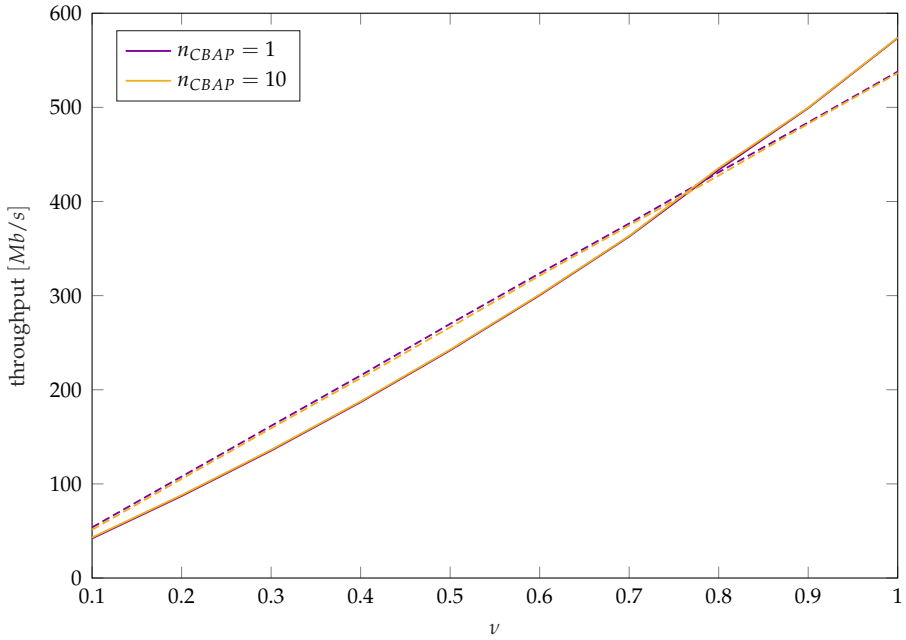


Figure 8.9: Throughput vs ν for two values of n_{CBAP}. Analytical model (solid lines) vs. simulation (dashed lines) for $\lambda = 0.04$ m^{-2} and $n_{SP} = 10$.

the aggregate throughput: although the success rate of each STA considered separately decreases for larger values of λ, the number of STA increases, yielding an almost constant value of S. The analytical model works better for smaller values of ν, while it tends to deviate from the simulated throughput as ν increases. Figs. 8.7 and 8.8 show the performance for up to about 100 STAs in the network. As λ increases, the model tends to underestimate the aggregate throughput. This happens because the collision probability is modeled by assuming that the STAs access the channel independently, while, as discussed in Sec. 8.3.3, this is not true. When the STAs density increases, the dependence among STAs become stronger but the model does not capture it and evaluates a poorer performance than that obtained in practice. The same considerations can be made for the delay (see Fig. 8.8). The delay increases with the STAs density because the higher collision rate leads to a larger number of retransmissions and decreases with ν since the EDCA operation is less likely to be frozen.

In Figs. 8.9 and 8.10 we evaluated the impact of ν and of the number of CBAP allocations. We fixed $n_{SP} = 10$ and varied ν for different values of n_{CBAP} with $\lambda = 0.04$ m^{-2}. Interestingly, both the throughput and the delay do not depend on n_{CBAP} (the plots show only two values of n_{CBAP}, but we obtained almost the same performance for each value of n_{CBAP} between 1 and $n_{SP} + 1$). The configuration of the CBAP allocation within a BI, thus, clearly plays a role on the transmission of a single packet, but does not have an impact on the aggregate performance. As already discussed, increasing ν improves the performance obtained during CBAP allocations.

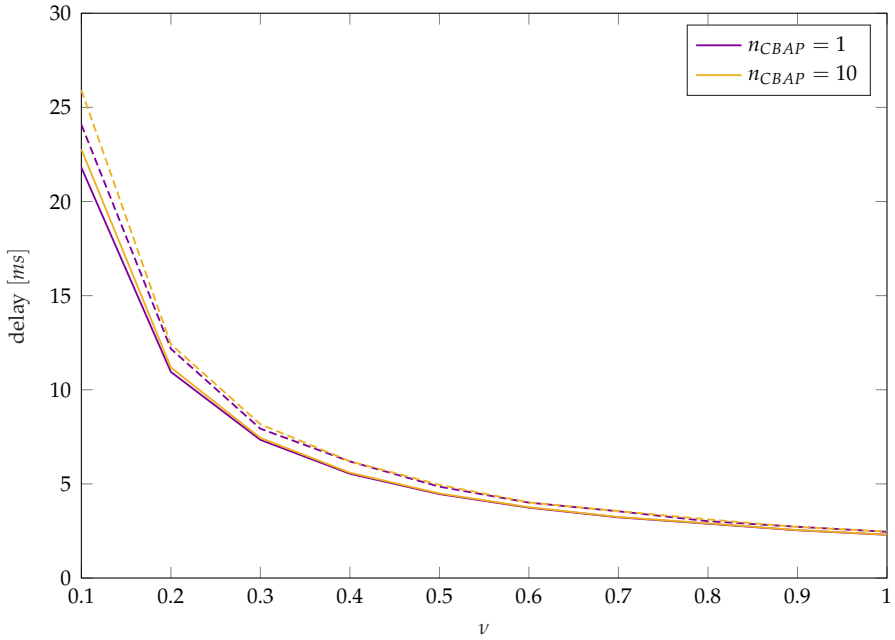


Figure 8.10: Delay vs ν for two values of n_{CBAP}. Analytical model (solid lines) vs. simulation (dashed lines) for $\lambda = 0.04$ m^{-2} and $n_{SP} = 10$.

8.7 Lesson learned

WiFi protocols have been object of studies for decades, and there exist many efficient models to evaluate and improve their performance. Nonetheless, the recent IEEE 802.11ad standard introduces novel features that are not readily addressed by the current models. A main characteristic of the WiFi protocol for mmWave networks is the use of contention-free scheduled SP in addition to the legacy EDCA operation and the use of a polling mechanism to allocate channel time almost in real time. It is however still unclear how to exploit such different allocation mechanisms in order to efficiently schedule the DTI and accommodate heterogeneous requirements. To this aim, it is first necessary to study the performance that can be obtained with SP and CBAP allocations depending on the DTI configuration and the network condition.

In this study, we focused on CBAP allocations and proposed an adaptation of Bianchi's original model that takes into account all the features introduced by the IEEE 802.11ad standard. Although the model is based on some assumptions, it is able to capure the system behavior during the ECDA operation and provides good estimates of throughput and delay. Moreover, the model is highly parameterized, making it possible to easily study the effect of various configurations, which is helpful for the design of an adaptive scheduling algorithm.

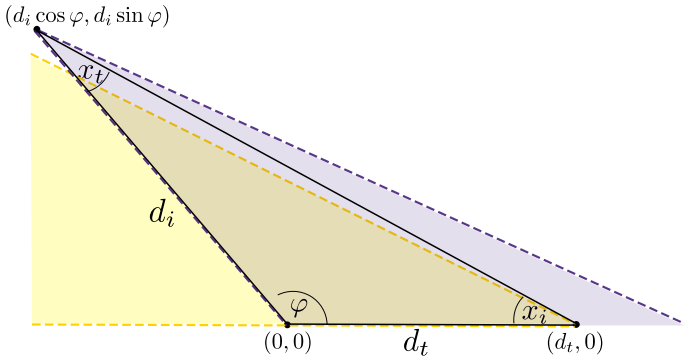


Figure 8.11: Target STA and potential interferer in the Cartesian plane. The blue area represents the right half of the beam of the potential interference directed towards the AP in $(0,0)$; it has width $\theta_S/2$ in correspondence of the potential interferer. Analogously, the yellow area is the upper half of the beam of the target STA. Notice that in this example the two STAs cannot hear each other, since the antenna gain of the target STA in $(d_i \cos\varphi, d_i \sin\varphi)$ is zero; it means that $\varphi < \varphi_{\text{lim}}(d_i)$.

Appendix A

Here we prove (8.23) and that STAs in group $n_{I,1}$ have the phase of their polar coordinates in the range $[\varphi_{\text{lim}}(d_i), 2\pi - \varphi_{\text{lim}}(d_i)]$.

Consider a possible interferer at distance $d_i \in [0, R]$ from the AP and with angular phase $\varphi \in [0, \pi]$; this means that we are only focusing on the upper half of the circular area around the AP, being the scenario symmetric. Consider the triangle whose vertices are the AP $(0,0)$, the target STA $(d_t, 0)$ and the interferer $(d_i \cos\varphi, d_i \sin\varphi)$, as in Fig. 8.11. The two edges that form the vertex coinciding with the AP have length d_i and d_t, and the angle they form has width φ. Denote the other two angles as x_i and x_t, as in Fig. 8.11.

We are interested in the angles φ such that the target STA is in the beam of the interferer, i.e., $x_i \leq \theta_S/2$, *and* the interferer is in the beam of the target STA, i.e., $x_t \leq \theta_S/2$ (the beams are symmetric with respect to the AP and have width θ_S). Moreover, the angles must satisfy the two following equations

$$x_i + x_t + \varphi = \pi, \tag{8.33}$$

$$\frac{d_i}{\sin x_i} = \frac{d_t}{\sin x_t}, \tag{8.34}$$

where (8.23) comes from the law of sines. If $d_i \leq d_t$, then $x_i \geq x_t$, and thus the limit condition is obtained for $x_i = \theta_S/2$. Considering (8.33) and (8.34), we then obtain that we are interested in all angles $\varphi \geq \varphi_{\text{lim}} = \pi - \theta_S/2 - \arcsin(d_i/d_t \sin\theta_S/2)$. Similarly, when $d_i > d_t$, the limit condition is obtained for $x_t = \theta_S/2$, yielding $\varphi_{\text{lim}} = \pi - \theta_S/2 - \arcsin(d_t/d_i \sin\theta_S/2)$. This proves Eq. (8.23). Taking into account also STAs in the lower half of the area aroud the AP, we finally obtain that the other STAs can overhear the messages sent from the target STA if and only if their phase is in $[\phi_{\text{lim}}(d_i), 2\pi - \phi_{\text{lim}}(d_i)]$, with d_i being their distance from the AP.

Appendix B

Here we explain how to express (8.27) and (8.28) so as to compute them in closed form. We focus only on the first integral in (8.27), since analogous considerations can be made for the second one, and refer to it as I_1. I_1 is zero if $\varphi_{\text{lim}}(r) > \varphi_{\text{AP}}$ for the considered $r \in [0, R]$ and position d_t of the target node (see (8.23)). In particular $I_1 = 0$ if

$$\arcsin\left(c_1 \sin\frac{\theta_S}{2}\right) < \pi - \frac{\theta_S}{2} - \varphi_{\text{AP}} \tag{8.35}$$

with $c_1 = r/d_t$ if $r \leq d_t$ and $c_1 = d_t/r$ otherwise (as per (8.23)). We recall that $0 \leq \theta_S \leq \pi$ and $0 \leq \varphi_{\text{AP}} \leq \theta_{\text{AP}} \leq \pi$ by assumption. Therefore, denoting the sum $\pi - \theta_S/2 - \varphi_{\text{AP}}$ as c_2, it is $-\pi/2 \leq c_2 \leq \pi$. Note also that the $0 \leq c_1 \sin\theta_S/2 \leq 1$, yielding $0 \leq \arcsin(c_1 \sin\theta_S/2) \leq \pi/2$.

- If $c_2 \in [\pi/2, \pi]$, then the conditions in (8.35) are certainly true, yielding $I_1 = 0$. This happens if $\varphi_{\text{AP}} < \pi/2 - \theta_S/2$, whatever the value of d_t.
- Otherwise $c_2 \in [-\pi/2, \pi/2]$. In this range the sine function is monotonically increasing, so that it is possible to apply it to both terms in (8.35) without additional adjustment. This gives the condition $c_1 \sin\theta_S/2 < \sin(\pi - \theta_S/2 - \varphi_{\text{AP}})$, which can be expressed as $c_1 < \sin\left(\pi - \frac{\theta_S}{2} - \varphi_{\text{AP}}\right) / \sin\frac{\theta_S}{2} \triangleq c_3$. It follows that $I_1 = 0$ if $r < d_t c_3$ in the case $r \leq d_t$ and if $r > d_t/c_3$ in the case $r > d_t$. This happens only if $c_3 \leq 1$, i.e., $\phi_{\text{AP}} < \pi - \theta_S$.

Summing up, it is

$$\begin{aligned} I_1 = &\left(\int_{\max(d_t c_3, 0)}^{d_t} \left(-c_2 + \arcsin\left(\frac{r}{d_t}\sin\frac{\theta_S}{2}\right)\right) r\, \partial r \right. \\ &\left. + \int_{d_t}^{\min(\frac{d_t}{c_3}, R)} \left(-c_2 + \arcsin\left(\frac{d_t}{r}\sin\frac{\theta_S}{2}\right)\right) r\, \partial r \right) \mathbb{1}_{\varphi_{\text{AP}} \geq \pi - \theta_S} \end{aligned} \tag{8.36}$$

with $\mathbb{1}_X$ being the indicator function, equal to 1 if the condition X is true, and to 0 otherwise. The min and max operators in the integral limits ensure that the range $[0, R]$ is not exceeded. Eq. (8.36) can be solved in closed form. The same procedure can be used to compute the second integral I_2 in (8.27) using $\theta_{\text{AP}} - \varphi_{\text{AP}}$ rather than φ_{AP}.

This expression of $\mathcal{R}_{R,C}(d_t, \varphi_{\text{AP}})$ can be used to compute its expectation as in (8.28). We can focus only on the first double integral, since analogous considerations can be made on the second one. Such first integral is made over I_1 and I_2. We consider only the integral over I_1 and denote it as J_1; the rest of the terms in (8.28) can then be derived following the same approach. It is necessary to characterize d_t/c_3 and $d_t c_3$ based on d_t

$$
\begin{aligned}
J_1 &= \frac{2}{R^2\theta_{\mathrm{AP}}} \int_{\pi-\theta_{\mathrm{S}}}^{\theta_{\mathrm{AP}}} \int_0^R d_t I_1 \, \partial d_t \partial \phi_{\mathrm{AP}} \\
&= \frac{2}{R^2\theta_{\mathrm{AP}}} \int_{\pi-\theta_{\mathrm{S}}}^{\theta_{\mathrm{AP}}} \int_0^R d_t \Bigg(\int_{\max(d_t c_3, 0)}^{d_t} \left(-c_2 + \arcsin\left(\frac{r}{d_t} \sin\frac{\theta_{\mathrm{S}}}{2} \right) \right) r \, \partial r \\
&+ \int_{d_t}^{\min(\frac{d_t}{c_3}, R)} \left(-c_2 + \arcsin\left(\frac{d_t}{r} \sin\frac{\theta_{\mathrm{S}}}{2} \right) \right) r \, \partial r \Bigg) \partial d_t \partial \phi_{\mathrm{AP}} \\
&= \frac{2}{R^2\theta_{\mathrm{AP}}} \int_0^R d_t \Bigg(\int_{\pi-\theta_{\mathrm{S}}}^{\pi-\theta_{\mathrm{S}}/2} \int_{d_t c_3}^{d_t} \left(-c_2 + \arcsin\left(\frac{r}{d_t} \sin\frac{\theta_{\mathrm{S}}}{2} \right) \right) r \, \partial r \partial \phi_{\mathrm{AP}} \\
&+ \int_{\pi-\theta_{\mathrm{S}}/2}^{\theta_{\mathrm{AP}}} \int_0^{d_t} \left(-c_2 + \arcsin\left(\frac{r}{d_t} \sin\frac{\theta_{\mathrm{S}}}{2} \right) \right) r \, \partial r \partial \phi_{\mathrm{AP}} \Bigg) \partial d_t \\
&+ \frac{2}{R^2\theta_{\mathrm{AP}}} \int_{\pi-\theta_{\mathrm{S}}}^{\theta_{\mathrm{AP}}} \Bigg(\int_0^{Rc_3} d_t \int_{d_t}^{d_t/c_3} \left(-c_2 + \arcsin\left(\frac{d_t}{r} \sin\frac{\theta_{\mathrm{S}}}{2} \right) \right) r \, \partial r \partial d_t \\
&+ \int_{Rc_3}^R \int_{d_t}^R \left(-c_2 + \arcsin\left(\frac{d_t}{r} \sin\frac{\theta_{\mathrm{S}}}{2} \right) \right) r \, \partial r \partial d_t \Bigg) \partial \phi_{\mathrm{AP}} \, .
\end{aligned}
\tag{8.37}
$$

and φ_{AP}, so as to remove the min and max operators in the terms in (8.36). It is $d_t c_3 > 0$ if $\varphi_{\mathrm{AP}} \leq \pi - c$, and $d_t / c_3 < R$ if $d_t \leq R c_3$.

This result and the considerations made previously yield Eq. (8.37).

Rigorously from (8.36), it is $J_1 = 0$ if $\phi_{\mathrm{AP}} < \pi - \theta_{\mathrm{S}}$. This can be checked beforehand as it only depends on system parameters. It is now easy to compute the integral in closed form and, repeating the same procedure for the other terms in (8.28), calculate $\mathrm{E}\left[\mathcal{R}_{R,C}\right]$.

9

Conclusions

This thesis concerns the channel access issue in novel technologies, namely IoT, LTE and IEEE802.11ad, and proposes several MAC layer schemes and models based on a mathematical analysis.

The first part of the thesis studies the challenge of designing energy-aware channel access policies for IoT networks. Ch. 2 discusses the energy efficiency goal in battery-powered sensor networks: it explains the main sources of energy consumption at the MAC layer and how they influence contention-based and contention-free access mechanisms. We also proposed a mathematical model to characterize the energy consumed by a device for the main operations, which is extremely helpful to design an energy-aware MAC layer: an incorrect characterization of the energy consumption in the planning phase may lead to the premature death of the network in real deployments and to unreliable performance analysis. We also discuss the use of data processing techniques to trade off some accuracy in the data representation for a reduced energy consumption.

Ch. 3 proposes an adaptive TDMA-based channel access policy for a network made up of devices with heterogeneous capabilities and requirements in the two cases of complete and incomplete CSI at the transmitter. The goal is to maximize the network lifetime while guaranteeing the desired level of QoS. The network resources are assigned to the devices based on their requirements, thus the optimization considers all users jointly. The proposed policy allocates channel resources differently to the various devices, dynamically adapts the data compression to the transmission operations, and performs power control to counteract the channel fading; these behaviors allow to outperform simpler schemes, as shown by both the numerical evaluations and the analytical results.

Similarly, also Ch. 4 presents a TDMA access policy with the goal of optimally using the available energy so as to achieve the target QoS. Here, however, the devices are endowed with energy harvesting capabilities, so that the goal is not to minimize the energy consumption but rather to adapt the consumption to the availability. Source and channel coding are optimized jointly, and they have a role on the communication robustness and the quality of the transmitted data. A retransmission mechanism allows

to further improve the QoS, at the cost of a higher energy consumption. Unlike in Ch. 3, the processing and transmission operations are optimized separately for each device, given the resources it is provided. This approach is easier to deploy in a distributed fashion and requires less information to be exchanged with the receiver.

Unlike Chs. 3 and 4, Ch. 5 studies random access schemes, which, compared to contention-free approaches, require no coordination among the devices but are prone to interference and collisions. The scenario is similar to the previous ones, with battery-powered sensors that track some time-series, which needs to be known at the receiver with a predefined accuracy. We considered three different schemes, which combine different compression techniques commonly used in signal estimation applications. Differently from several state-of-the art works about time-series monitoring, we explicitly consider the impact of interference on the QoS and energy consumption, and this allows us to obtain better performance.

Ch. 6 considers the energy efficiency issue from a different perspective, namely that of security, since energy restrictions may be a significant limit on a device's capabilities, making it more vulnerable to security attacks. An energy-depleting jamming attack is modeled as a zero-sum game and analyzed for different levels of information available to the legitimate transmitter. The proposed model sheds light on the attack and defense mechanisms that can be played by energy-constrained devices and may be used as a guideline for the development of lightweight defense and attack strategies.

The studies presented in this thesis validate the importance of including the energy consumption in the MAC design when there is an interest in optimizing the devices lifetime. To achieve such goal, it is necessary to have an accurate characterization of the power consumption; a mathematical analysis gives insight on the impact that the various operations related to the channel access have on the energy efficiency and how they are interrelated. The obtained results can be helpful to evaluate the performance obtained in real deployments and to realize flexible channel access schemes able to adapt to the changing network conditions and application requirements.

The second part of the thesis concerns the channel access and data transmission mechanisms of existing standards. Ch. 7 shows how to improve the LTE standard by providing additional knowledge on the current network load so as to adapt the collision resolution procedure of the RACH phase based on the estimated load. A mechanism based on standard machine-learning techniques in fact allows one to infer the collision multiplicity during the RACH phase. This information can be used to improve the current medium access mechanism, which is sensitive to massive access overload, and thereby yield reduced transmission latency, lower power consumption and

increased supported system load. Moreover, the proposed techniques outperform the state-of-the-art threshold-based schemes for preamble detection during the RACH.

Finally, Ch. 8 describes a mathematical model for the CBAP allocations in IEEE802.11ad, which can be used to gain insight on the system performance that can be obtained with a chosen transmission schedule; this may be very helpful to understand how to match diverse traffic flows with specific requirements to the most appropriate scheduling. The proposed model is an adaptation of Bianchi's seminal model for legacy WiFi networks, which takes into account the hidden node and deafness issues related to directional communication and the intrinsic features of the standard 802.11ad.

List of publications

Journals

[J1] C. Pielli, D. Zucchetto, A. Zanella, L. Vangelista and Michele Zorzi, "Platforms and Protocols for the Internet of Things," *EAI Endorsed Transactions on Internet of Things*, vol. 15, no. 1, Oct. 2015

[J2] A. Biason, C. Pielli, M. Rossi, A. Zanella, D. Zordan, M. Kelly and Michele Zorzi, "EC-CENTRIC: An Energy-and Context-Centric Perspective on IoT Systems and Protocol Design," *IEEE Access*, vol. 5, pp. 6894–6908, Feb. 2017,

[J3] C. Pielli, P. Popovski, Č. Stefanović, and Michele Zorzi, "Joint Compression, Channel Coding, and Retransmission for Data Fidelity With Energy Harvesting," *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1425–1439, Apr. 2018

[J4] F. Chiariotti, C. Pielli, A. Zanella and M. Zorzi, "A Dynamic Approach to Rebalancing Bike-Sharing Systems," *Sensors*, vol. 18, no.2, Jan. 2018

[J5] A. Biason, C. Pielli, A. Zanella and Michele Zorzi, "Access Control for IoT Nodes with Energy and Fidelity Constraints," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3242–3257, Feb. 2018

[J6] F. Chiariotti, C. Pielli, N. Laurenti, A. Zanella and M. Zorzi,"A game-theoretic analysis of energy-depleting jamming attacks," *submitted* to Hindawi Wireless Communications and Mobile Computing

[J7] F. Chiariotti, C. Pielli, A. Zanella and M. Zorzi,"Combining dynamic rebalancing strategies and user incentives: towards a unified model of bike sharing optimization," *submitted* to Transactions on Intelligent Transportation Systems

[J8] C. Pielli, D. Zucchetto, A. Zanella and M. Zorzi,"An Interference-Aware Channel Access Strategy for WSNs Exploiting Temporal Correlation," *submitted* to IEEE Transactions on Communications

Conferences

[C1] C. Pielli, A. Biason, A. Zanella and M. Zorzi, "Joint optimization of energy efficiency and data compression in TDMA-based medium access control for the IoT," in *IEEE Globecom Workshops (GC Wkshps)*, Dec. 2016

[C2] C. Pielli, F. Chiariotti, N. Laurenti, A. Zanella and M. Zorzi,"A game-theoretic analysis of energy-depleting jamming attacks," in IEEE International Conference on Computing, Networking and Communications (ICNC 2017), pp. 100–104, Jan. 2017

[C3] C. Pielli, P. Popovski, Č. Stefanović, and M. Zorzi, "Minimizing Data Distortion of Periodically Reporting IoT Devices with Energy Harvesting," in *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2017

[C4] D. Zucchetto, C. Pielli, A. Zanella and M. Zorzi, "A Random Access Scheme to Balance Energy Efficiency and Accuracy in Monitoring Applications," in *Information Theory and Applications Workshops (ITA)*, Feb. 2018

[C5] D. Zucchetto, C. Pielli, A. Zanella and M. Zorzi, "Random Access in the IoT: An Adaptive Sampling and Transmission Strategy" in *IEEE International Conference on Communications (ICC 2018)*, May 2018

[C6] F. Chiariotti, C. Pielli, A. Zanella and M. Zorzi, "Bike sharing as a key smart city service: State of the art and future developments," in *International Conference on Modern Circuits and Systems Technologies (MOCAST)*, May 2018

[C7] M. Sansoni, G. Ravagnani, D. Zucchetto, C. Pielli, A. Zanella and K. Mahmood, "Comparison of M2M traffic models against real world data sets," in *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Sept. 2018

[C8] C. Pielli, T. Ropitault, and M. Zorzi, "The Potential of mmWaves in Smart Industry: Manufacturing at 60GHz," in *Springer International Conference on Ad-Hoc Networks and Wireless*, Sept. 2018

[C9] D. Magrin, C. Pielli, Č. Stefanović, and M. Zorzi, "Enabling LTE RACH Collision Multiplicity Detection via Machine Learning," *submitted* to *IEEE International Conference on Communications (ICC 2019)*

Bibliography

[1] N. Benvenuto and M. Zorzi, *Principles of communications Networks and Systems*. John Wiley & Sons, 2011.

[2] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, Feb 2014.

[3] A. Biral, M. Centenaro, A. Zanella, L. Vangelista, and M. Zorzi, "The challenges of M2M massive access in wireless cellular networks," *Digital Communications and Networks*, vol. 1, no. 1, pp. 1–19, 2015.

[4] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad hoc networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[5] "Cisco visual networking index: Forecast and methodology, 2015–2020," White Paper, Cisco, Apr. 2016.

[6] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of things," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1101–1102, 2012.

[7] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[8] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's INTRAnet of things to a future INTERnet of things: a wireless-and mobility-related view," *IEEE Wireless communications*, vol. 17, no. 6, Dec. 2010.

[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, Sept. 2013.

[10] A. Biason, C. Pielli, M. Rossi, A. Zanella, D. Zordan, M. Kelly, and M. Zorzi, "EC-CENTRIC: An Energy- and Context-Centric perspective on IoT systems and protocol design," *IEEE Access*, vol. 5, 2017.

[11] Y. Liang, "Efficient temporal compression in wireless sensor networks," in *IEEE 36th Conference on Local Computer Networks (LCN)*. IEEE, Oct. 2011, pp. 466–474.

[12] Nokia. (2015) LTE evolution for IoT connectivity. [Online]. Available: http://resources.alcatel-lucent.com/asset/200178

[13] Ericsson. (2015) Ericsson, AT&T and Altair demonstrate over 10 years of battery life on LTE IoT commercial chipset. [Online]. Available: https://www.ericsson.com/news/1962068

[14] Huawei. (2015) NB-IOT - Enabling New Business Opportunities. [Online]. Available: http://www.huawei.com/minisite/4-5g/img/NB-IOT.pdf

[15] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*. ACM, 2006, pp. 307–320.

[16] R. C. Carrano, D. Passos, L. C. Magalhaes, and C. V. Albuquerque, "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 181–194, First Quarter 2014.

[17] 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, Mar. 2015.

[18] D. Gunduz, K. Stamatiou, N. Michelusi, and M. Zorzi, "Designing intelligent energy harvesting communication systems," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 210–216, Jan. 2014.

[19] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, May 2009.

[20] S. T. Kouyoumdjieva and G. Karlsson, "Impact of duty cycling on opportunistic communication," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1686–1698, July 2016.

[21] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Transactions on Communications*, vol. 60, no. 1, pp. 220–230, Jan. 2012.

[22] A. Biason and M. Zorzi, "Joint transmission and energy transfer policies for energy harvesting devices with finite batteries," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2626–2640, Dec. 2015.

[23] A. Hac, *Wireless sensor network designs*. John Wiley & Sons Ltd, 2003.

[24] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, June 2002, pp. 1567–1576.

[25] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, Second Quarter 2010.

[26] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "LoRaWAN Specifications," LoRa Alliance, Tech. Rep., 2015.

[27] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and C. Petrioli, "Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems," in *Proc. IEEE Conf. on Computer Communications (INFOCOM)*, Apr. 2015, pp. 522–530.

[28] L. Gu and J. A. Stankovic, "Radio-triggered wake-up capability for sensor networks," in *Proc. IEEE 10th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, May 2004, pp. 27–36.

[29] W. Shen, T. Zhang, M. Gidlund, and F. Dobslaw, "SAS-TDMA: a source aware scheduling algorithm for real-time communication in industrial wireless sensor networks," *Wireless Networks*, vol. 19, no. 6, pp. 1155–1170, Aug. 2013.

[30] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6TiSCH: deterministic IP-enabled industrial Internet (of Things)," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, Dec. 2014.

[31] Y. Wu, X. Y. Li, Y. Li, and W. Lou, "Energy-efficient wake-up scheduling for data collection and aggregation," *IEEE Trans. on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 275–287, Feb. 2010.

[32] G. C. Madueño, Č. Stefanović, and P. Popovski, "Reliable and efficient access for alarm-initiated and regular M2M traffic in IEEE 802.11 ah systems," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 673–682, Oct. 2016.

[33] E. Paolini, Č. Stefanović, G. Liva, and P. Popovski, "Coded random access: applying codes on graphs to design random access protocols," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 144–150, June 2015.

[34] A. Zanella and M. Zorzi, "Theoretical analysis of the capture probability in wireless systems with multiple packet reception capabilities," *IEEE Trans. on Communications*, vol. 60, no. 4, pp. 1058–1071, Apr. 2012.

[35] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver, "Low energy operation in WSNs: A survey of preamble sampling MAC protocols," *Computer Networks*, vol. 55, no. 15, pp. 3351–3363, Oct. 2011.

[36] M. D. Jovanovic and G. L. Djordjevic, "Reduced-frame TDMA protocols for wireless sensor networks," *Int. Journal of Communication Systems*, vol. 27, no. 10, pp. 1857–1873, Oct. 2014.

[37] M. R. Lenka, A. R. Swain, and M. N. Sahoo, "Distributed slot scheduling algorithm for hybrid CSMA/TDMA MAC in wireless sensor networks," in *Proc. IEEE Conf. on Netw., Architecture and Storage (NAS)*, Aug. 2016.

[38] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. ACM 1st Workshop on Mobile Cloud Computing (MCC)*, Aug. 2012, pp. 13–16.

[39] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.

[40] A. Bogliolo, V. Freschi, E. Lattanzi, A. L. Murphy, and U. Raza, "Towards a true energetically sustainable WSN: a case study with prediction-based data collection and a wake-up receiver," in *Proc. IEEE 9th Symp. on Industrial Embedded Systems (SIES)*, June 2014, pp. 21–28.

[41] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic, "Edge mining the Internet of Things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3816–3825, Oct. 2013.

[42] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco, "Practical data prediction for real-world wireless sensor networks," *IEEE Trans. on Knowledge and Data Engineering*, vol. 27, no. 8, pp. 2231–2244, Aug. 2015.

[43] D. Zordan, B. Martinez, I. Vilajosana, and M. Rossi, "On the performance of lossy compression schemes for energy constrained sensor networking," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 1, pp. 15:1–15:34, Nov. 2014.

[44] T. Schoellhammer, E. Osterweil, B. Greenstein, M. Wimbrow, and D. Estrin, "Lightweight temporal compression of microclimate datasets," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*. IEEE Computer Society, 2004, pp. 516–524.

[45] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, "Sensing, compression, and recovery for WSNs: Sparse signal modeling and monitoring framework," *IEEE Trans. on Wireless Communications*, vol. 11, no. 10, pp. 3447–3461, Oct. 2012.

[46] Y. Li and Y. Liang, "Temporal lossless and lossy compression in wireless sensor networks," *ACM Trans. on Sensor Networks*, vol. 12, no. 4, pp. 37:1–37:35, Oct. 2016.

[47] C. W. Tsai, C. F. Lai, M. C. Chiang, and L. T. Yang, "Data mining for Internet of Things: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 77–97, First Quarter 2014.

[48] B. D. Fulcher and N. S. Jones, "Highly comparative feature-based time-series classification," *IEEE Trans. on Knowledge and Data Engineerin*, vol. 26, no. 12, pp. 3026–3037, Dec. 2014.

[49] F. Ganz, D. Puschmann, P. Barnaghi, and F. Carrez, "A practical evaluation of information processing and abstraction techniques for the Internet of Things," *IEEE Internet of Things Journal*, vol. 2, no. 4, pp. 340–354, Aug. 2015.

[50] S. L. Howard, C. Schlegel, and K. Iniewski, "Error control coding in low-power wireless sensor networks: When is ECC energy-efficient?" *EURASIP Journal on Wireless Communication Networks*, vol. 2006, no. 2, pp. 1–14, Apr. 2006.

[51] M. Miozzo, D. Zordan, P. Dini, and M. Rossi, "Solarstat: Modeling photovoltaic sources through stochastic Markov processes," in *2014 IEEE International Energy Conference*, May 2014, pp. 688–695.

[52] M.-L. Ku, Y. Chen, and K. R. Liu, "Data-driven stochastic models and policies for energy harvesting sensor communications," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 8, pp. 1505–1520, Aug. 2015.

[53] H. A. Nguyen, A. Förster, D. Puccinelli, and S. Giordano, "Sensor node lifetime: An experimental study," in *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, March 2011, pp. 202–207.

[54] M. I. Chidean, E. Morgado, M. Sanromán-Junquera, J. Ramiro-Bargueño, J. Ramos, and A. J. Caamaño, "Energy efficiency and quality of data reconstruction through

data-coupled clustering for self-organized large-scale wsns," *IEEE sensors journal*, vol. 16, no. 12, pp. 5010–5020, June 2016.

[55] P. Castiglione, O. Simeone, E. Erkip, and T. Zemen, "Energy management policies for energy-neutral source-channel coding," *IEEE Transactions on Communications*, vol. 60, no. 9, pp. 2668–2678, Sept. 2012.

[56] C. Tapparello, O. Simeone, and M. Rossi, "Dynamic compression-transmission for energy-harvesting multihop networks with correlated sources," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 6, pp. 1729–1741, Dec. 2014.

[57] D. Zordan, T. Melodia, and M. Rossi, "On the design of temporal compression strategies for energy harvesting sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1336–1352, Feb. 2016.

[58] S. Knorn, S. Dey, A. Ahlén, D. E. Quevedo *et al.*, "Distortion minimization in multi-sensor estimation using energy harvesting and energy sharing," *IEEE Trans. Signal Processing*, vol. 63, no. 11, pp. 2848–2863, June 2015.

[59] L. Bing, "A dynamic TDMA protocol based on correlation in wireless sensor networks," in *2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC)*. IEEE, June 2016, pp. 245–248.

[60] C. Pielli, A. Biason, A. Zanella, and M. Zorzi, "Joint optimization of energy efficiency and data compression in tdma-based medium access control for the iot," in *2016 IEEE Globecom Workshops (GC Wkshps)*. IEEE, Dec. 2016, pp. 1–6.

[61] A. Biason, C. Pielli, A. Zanella, and M. Zorzi, "Access control for iot nodes with energy and fidelity constraints," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3242–3257, May 2018.

[62] T. Berger, "Rate-distortion theory," *Wiley Encyclopedia of Telecommunications*, 2003.

[63] D. Zordan, M. Rossi, and M. Zorzi, "Rate-distortion classification for self-tuning iot networks," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2017, pp. 857–863.

[64] C. Pielli, Č. Stefanović, P. Popovski, and M. Zorzi, "Joint compression, channel coding and retransmission for data fidelity with energy harvesting," *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1425–1439, Apr. 2018.

[65] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[66] M.-L. Ku, W. Li, Y. Chen, and K. R. Liu, "Advances in energy harvesting communications: Past, present, and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1384–1412, Second Quarter 2016.

[67] K. E. Zachariadis, M. L. Honig, and A. K. Katsaggelos, "Source fidelity over fading channels: performance of erasure and scalable codes," *IEEE Transactions on Communications*, vol. 56, no. 7, July 2008.

[68] I. E. Aguerri and D. Gunduz, "Expected distortion with fading channel and side information quality," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, June 2011, pp. 1–6.

[69] J. N. Laneman, E. Martinian, G. W. Wornell, and J. G. Apostolopoulos, "Source-channel diversity for parallel channels," *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3518–3539, Oct. 2005.

[70] S. Zhao, D. Tuninetti, R. Ansari, and D. Schonfeld, "On achievable distortion exponents for a gaussian source transmitted over parallel gaussian channels with correlated fading and asymmetric SNRs," *IEEE Transactions on Information Theory*, vol. 62, no. 7, pp. 4135–4153, July 2016.

[71] F. Etemadi and H. Jafarkhani, "A unified framework for layered transmission over fading and packet erasure channels," *IEEE Transactions on Communications*, vol. 56, no. 4, pp. 565–573, Apr. 2008.

[72] D. W. K. Ng, R. Schober, and H. Alnuweiri, "Secure layered transmission in multicast systems with wireless information and power transfer," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, June 2014, pp. 5389–5395.

[73] R. V. Bhat, M. Motani, and T. J. Lim, "Distortion minimization in energy harvesting sensor nodes with compression power constraints," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[74] M. S. Motlagh, M. B. Khuzani, and P. Mitran, "On lossy joint source-channel coding in energy harvesting communication systems," *IEEE Transactions on Communications*, vol. 63, no. 11, pp. 4433–4447, Nov. 2015.

[75] M. Calvo-Fullana, J. Matamoros, and C. Antón-Haro, "Reconstruction of correlated sources with energy harvesting constraints in delay-constrained and delay-tolerant communication scenarios," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1974–1986, Mar. 2017.

[76] N. Hu, Y. D. Yao, and Z. Yang, "Analysis of cooperative TDMA in Rayleigh fading channels," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 3, pp. 1158–1168, Mar. 2013.

[77] J. K. Lee, H. J. Noh, and J. Lim, "Dynamic cooperative retransmission scheme for TDMA systems," *IEEE Communications Letters*, vol. 16, no. 12, pp. 2000–2003, Dec. 2012.

[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, Oct. 2013.

[79] D. G. Costa, L. A. Guedes, F. Vasques, and P. Portugal, "Partial energy-efficient hop-by-hop retransmission in wireless sensor networks," in *11th IEEE International Conference on Industrial Informatics (INDIN)*, July 2013, pp. 146–151.

[80] C. Pielli, C. Stefanovic, P. Popovski, and M. Zorzi, "Minimizing data distortion of periodically reporting IoT devices with energy harvesting," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2017, pp. 1–9.

[81] D. Zordan, R. Parada, M. Rossi, and M. Zorzi, "Automatic rate-distortion classification for the IoT: towards signal-adaptive network protocols," in *2017 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Dec. 2017, pp. 1–7.

[82] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[83] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-static multiple-antenna fading channels at finite blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4232–4265, July 2014.

[84] E. Altman, *Constrained Markov decision processes*. CRC Press, 1999, vol. 7.

[85] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 2005, vol. 1, no. 3.

[86] M. A. Razzaque and S. Dobson, "Energy-efficient sensing in wireless sensor networks using compressed sensing," *Sensors*, vol. 14, no. 2, pp. 2822–2859, Feb. 2014.

[87] U. Kulau, J. van Balen, S. Schildt, F. Büsching, and L. Wolf, "Dynamic sample rate adaptation for long-term IoT sensing applications," in *3rd IEEE World Forum on Internet of Things (WF-IoT)*, Dec. 2016, pp. 271–276.

[88] A. Pal and K. Kant, "On the feasibility of distributed sampling rate adaptation in heterogeneous and collaborative wireless sensor networks," in *25th International Conference on Computer Communication and Networks (ICCCN)*, Aug. 2016, pp. 1–9.

[89] M. Wu, L. Tan, and N. Xiong, "Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications," *Information Sciences*, vol. 329, supplement C, pp. 800–818, Feb. 2016.

[90] C. Karakus, A. C. Gurbuz, and B. Tavli, "Analysis of energy efficiency of compressive sensing in wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1999–2008, May 2013.

[91] V. Shah-Mansouri, S. Duan, L.-H. Chang, V. W. S. Wong, and J.-Y. Wu, "Compressive sensing based asynchronous random access for wireless networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2013, pp. 884–888.

[92] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in *International Conference on Information Technology: Coding and Computing (ITCC'05)*, vol. 2, Apr. 2005, pp. 8–13.

[93] F. Marcelloni and M. Vecchio, "Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization," *Information Sciences*, vol. 180, no. 10, pp. 1924–1941, May 2010.

[94] D. Tulone and S. Madden, "PAQ: Time series forecasting for approximate query answering in sensor networks," in *Proceedings of the Third European Workshop on Wireless Sensor Networks (EWSN 2006)*, Feb. 2006, pp. 21–37.

[95] S. Goel and T. Imielinski, "Prediction-based monitoring in sensor networks: Taking lessons from MPEG," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 5, pp. 82–98, Oct. 2001.

[96] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, "Energy management in wireless sensor networks with energy-hungry sensors," *IEEE Instrumentation Measurement Magazine*, vol. 12, no. 2, pp. 16–23, Apr. 2009.

[97] D. Zucchetto, C. Pielli, A. Zanella, and M. Zorzi, "Random access in the IoT: An adaptive sampling and transmission strategy," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[98] ——, "A random access scheme to balance energy efficiency and accuracy in monitoring applications," in *Proceedings of the Information Theory and Applications Workshop (ITA 2018)*, Feb. 2018, pp. 1–6.

[99] A. Iyer, C. Rosenberg, and A. Karnik, "What is the right model for wireless channel interference?" *IEEE Transactions on Wireless Communications*, vol. 8, no. 5, pp. 2662–2671, May 2009.

[100] F. Fazel, M. Fazel, and M. Stojanovic, "Random access compressed sensing for energy-efficient underwater sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1660–1670, Sept. 2011.

[101] N. Kumar, F. Fazel, M. Stojanovic, and S. S. Naryanan, "Online rate adjustment for adaptive random access compressed sensing of time-varying fields," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 48, Apr. 2016.

[102] L. Wu, P. Sun, M. Xiao, Y. Hu, and Z. Wang, "Sparse signal ALOHA: A compressive sensing-based method for uncoordinated multiple access," *IEEE Communications Letters*, vol. 21, no. 6, pp. 1301–1304, June 2017.

[103] M. Haenggi, *Stochastic geometry for wireless networks*. Cambridge University Press, 2013.

[104] M. Gupta, L. V. Shum, E. Bodanese, and S. Hailes, "Design and evaluation of an adaptive sampling strategy for a wireless air pollution sensor network," in *36th IEEE Conference on Local Computer Networks*, Oct. 2011, pp. 1003–1010.

[105] C. Pielli, F. Chiariotti, N. Laurenti, A. Zanella, and M. Zorzi, "A game-theoretic analysis of energy-depleting jamming attacks," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, Jan. 2017, pp. 100–104.

[106] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *IEEE Network*, vol. 20, no. 3, pp. 41–47, May-June 2006.

[107] D. R. Raymond, R. C. Marchany, M. I. Brownfield, and S. F. Midkiff, "Effects of denial-of-sleep attacks on wireless sensor network MAC protocols," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 367–380, June 2009.

[108] Y. W. Law, M. Palaniswami, L. V. Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, pp. 6:1–6:38, Feb. 2009.

[109] B. Mihajlov and M. Bogdanoski, "Analysis of the WSN MAC protocols under jamming DoS attack," *IJ Network Security*, vol. 16, no. 4, pp. 304–312, July 2014.

[110] A. D. Wood, J. A. Stankovic, and G. Zhou, "DEEJAM: Defeating energy-efficient jamming in IEEE 802.15.4–based wireless networks," in *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2007, pp. 60–69.

[111] X. Cao, D. M. Shila, Y. Cheng, Z. Yang, Y. Zhou, and J. Chen, "Ghost-in-ZigBee: Energy depletion attack on ZigBee-based wireless networks," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 816–829, Oct. 2016.

[112] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," in *Proceedings of the 3rd ACM Workshop on Wireless Security*. ACM, Oct. 2004, pp. 32–42.

[113] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, Fourth Quarter 2009.

[114] X. Liang and Y. Xiao, "Game theory for network security," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 472–486, First Quarter 2013.

[115] G. Thamilarasu and R. Sridhar, "Game theoretic modeling of jamming attacks in ad hoc networks," in *Proceedings of 18th Internatonal Conference on Computer Communications and Networks, 2009 (ICCCN 2009)*, June 2009, pp. 1–6.

[116] B. Wang, Y. Wu, K. J. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 877–889, Apr. 2011.

[117] A. Garnaev, Y. Hayel, and E. Altman, "A bayesian jamming game in an OFDM wireless network," in *2012 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*. IEEE, May 2012, pp. 41–48.

[118] L. Chen and J. Leneutre, "Fight jamming with jamming-a game theoretic analysis of jamming attack in wireless networks and defense strategy," *Computer Networks*, vol. 55, no. 9, pp. 2259–2270, Mar. 2011.

[119] M. Brownfield, Y. Gupta, and N. Davis, "Wireless sensor network denial of sleep attack," in *Proceedings of the IEEE Information Assurance Workshop (IAW)*, June 2005, pp. 356–364.

[120] G. Alnifie and R. Simon, "A multi-channel defense against jamming attacks in wireless sensor networks," in *Proceedings of the 3rd ACM workshop on QoS and security for wireless and mobile networks*. ACM, Oct. 2007, pp. 95–104.

[121] R. K. Mallik, R. A. Scholtz, and G. P. Papavassilopoulos, "Analysis of an on-off jamming situation as a dynamic game," *IEEE Transactions on Communications*, vol. 48, no. 8, pp. 1360–1373, Aug. 2000.

[122] G. Thamilarasu, S. Mishra, and R. Sridhar, "Improving reliability of jamming attack detection in ad hoc networks," *International Journal of Communication Networks and Information Security*, vol. 3, no. 1, pp. 57–66, Aug 2011.

[123] A. Gupta, A. Nayyar, C. Langbort, and T. Başar, "A dynamic transmitter-jammer game with asymmetric information," in *IEEE 51st Annual Conference on Decision and Control (CDC)*, Dec. 2012, pp. 6477–6482.

[124] B. DeBruhl, C. Kroer, A. Datta, T. Sandholm, and P. Tague, "Power napping with loud neighbors: Optimal energy-constrained jamming and anti-jamming," in *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks*. ACM, July 2014, pp. 117–128.

[125] L. Xiao, T. Chen, J. Liu, and H. Dai, "Anti-jamming transmission Stackelberg game with observation errors," *IEEE Communications Letters*, vol. 19, no. 6, pp. 949–952, June 2015.

[126] D. T. Hoang, D. Niyato, P. Wang, and D. I. Kim, "Performance analysis of wireless energy harvesting cognitive radio networks under smart jamming attacks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 1, no. 2, pp. 200–216, June 2015.

[127] A. Garnaev, W. Trappe, and A. Petropulu, "Equilibrium strategies for an OFDM network that might be under a jamming attack," in *Proceedings of the 51st Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2017, pp. 1–6.

[128] E. Altman, K. Avrachenkov, and A. Garnaev, "Jamming game with incomplete information about the jammer," in *Proceedings of the 4th International ICST Conference on Performance Evaluation Methodologies and Tools*, 2009, pp. 65:1–65:9.

[129] Y. E. Sagduyu, R. Berry, and A. Ephremides, "MAC games for distributed wireless network security with incomplete information of selfish and malicious user types," in *Proceedings of the International Conference on Game Theory for Networks*, May 2009, pp. 130–139.

[130] Y. E. Sagduyu, R. A. Berry, and A. Ephremides, "Jamming games in wireless networks with incomplete information," *IEEE Communications Magazine*, vol. 49, no. 8, Aug. 2011.

[131] D. Yang, G. Xue, J. Zhang, A. Richa, and X. Fang, "Coping with a smart jammer in wireless networks: A Stackelberg game approach," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 4038–4047, Aug. 2013.

[132] T. Basar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1999, revised from the 2nd ed. published in 1995 by Academic Press, New York.

[133] D. Abreu, "On the theory of infinitely repeated games with discounting," *Econometrica: Journal of the Econometric Society*, pp. 383–396, March 1988.

[134] M. Chiani, D. Dardari, and M. K. Simon, "New exponential bounds and approximations for the computation of error probability in fading channels," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 840–845, July 2003.

[135] J. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, Sept. 1951.

[136] C. E. Lemke and J. T. Howson, Jr, "Equilibrium points of bimatrix games," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 2, pp. 413–423, June 1964.

[137] J. C. Harsanyi, "Games with incomplete information played by "Bayesian" players part II. Bayesian equilibrium points," *Management Science*, vol. 14, no. 5, pp. 320–334, Jan. 1968.

[138] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton university press, 2007.

[139] N. K. Bose and P. Liang, "Neural network fundamentals with graphs, algorithms, and applications (mcgraw-hill series in electrical computer engineering)," 1996.

[140] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[141] T. Dozat, "Incorporating Nesterov momentum into Adam," Stanford University, Tech. Rep., 2015.[Online]. Available: http://cs229. stanford. edu/proj2015/054 report. pdf, Tech. Rep., 2015.

[142] S. Sesia, M. Baker, and I. Toufik, *LTE - The UMTS Long Term Evolution: from theory to practice*. John Wiley & Sons, 2011.

[143] G. C. Madueno, Č. Stefanović, and P. Popovski, "Reengineering GSM/GPRS towards a dedicated network for massive smart metering," in *IEEE International Conference On Smart Grid Communications (SmartGridComm)*. IEEE, Nov. 2014, pp. 338–343.

[144] M. Y. Cheng, G. Y. Lin, H. Y. Wei, and A. C. C. Hsu, "Overload control for machine-type-communications in LTE-Advanced system," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 38–45, June 2012.

[145] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the Random Access Channel of LTE and LTE-A Suitable for M2M Communications? A Survey of Alternatives," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 4–16, First Quarter 2014.

[146] D. Magrin, C. Pielli, C. Stefanovic, and M. Zorzi, "Enabling LTE RACH collision multiplicity detection via machine learning," *arXiv preprint arXiv:1805.11482*, 2018.

[147] F. J. López-Martínez, E. del Castillo-Sánchez, E. Martos-Naya, and J. T. Entrambasaguas, "Performance evaluation of preamble detectors for 3GPP-LTE physical random access channel," *Digital Signal Processing*, vol. 22, no. 3, pp. 526–534, May 2012.

[148] P. Li and B. Wu, "An effective approach to detect random access preamble in LTE systems in low SNR," *Procedia Engineering*, vol. 15, pp. 2339–2343, 2011.

[149] T. Kim, I. Bang, and D. K. Sung, "An enhanced PRACH preamble detector for cellular IoT communications," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2678–2681, Dec. 2017.

[150] X. Yang and A. O. Fapojuwo, "Enhanced preamble detection for PRACH in LTE," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, Apr. 2013, pp. 3306–3311.

[151] 3GPP, "MTC simulation results with specific solutions," 3rd Generation Partnership Project (3GPP), TR R2–104662, Aug. 2010.

[152] S. Duan, V. Shah-Mansouri, and V. W. S. Wong, "Dynamic access class barring for M2M communications in LTE networks," in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2013, pp. 4747–4752.

[153] N. K. Pratas, H. Thomsen, Č. Stefanović, and P. Popovski, "Code-expanded random access for machine-type communications," in *IEEE Globecom Workshops*, Dec. 2012, pp. 1681–1686.

[154] N. K. Pratas, Č. Stefanović, G. C. Madueno, and P. Popovski, "Random Access for Machine-Type Communication Based on Bloom Filtering," in *IEEE Global Communications Conference*, Dec. 2016, pp. 1–7.

[155] G. C. Madueno, Č. Stefanović, and P. Popovski, "Efficient LTE access with collision resolution for massive M2M communications," in *IEEE Globecom Workshops (GC Wkshps)*, Dec. 2014, pp. 1433–1438.

[156] G. C. Madueno, N. K. Pratas, Č. Stefanović, and P. Popovski, "Massive M2M access with reliability guarantees in LTE systems," in *IEEE International Conference on Communications*, June 2015, pp. 2997–3002.

[157] 3GPP, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) conformance testing," TS 36.141 V.11.3.0.

[158] ——, "Study on RAN improvements for machine-type communications," TR 37.868 V.11.0.0.

[159] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[160] G. Athanasiou, P. C. Weeraddana, C. Fischione, and P. Orten, "Communication infrastructures in industrial automation: The case of 60 GHz millimeterWave communications," in *IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, Sept. 2013, pp. 1–6.

[161] H. Shokri-Ghadikolaei and C. Fischione, "The transitional behavior of interference in millimeter wave networks and its impact on medium access control," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 723–740, 2016.

[162] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "IEEE 802.11ad: directional 60 GHz communication for multi-Gigabit-per-second Wi-Fi," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 132–141, Dec. 2014.

[163] IEEE 802.11 WG, "IEEE 802.11ad, amendment 3: Enhancements for very high throughput in the 60 GHz band," Dec. 2012.

[164] S. Kutty and D. Sen, "Beamforming for millimeter wave communications: An inclusive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 949–973, 2016.

[165] B. Satchidanandan, S. Yau, P. Kumar, A. Aziz, A. Ekbal, and N. Kundargi, "Track-MAC: An IEEE 802.11ad-compatible beam tracking-based MAC protocol for 5G

millimeter-wave local area networks," in *International Conference on Communication Systems & Networks (COMSNETS)*. IEEE, Jan. 2018, pp. 185–182.

[166] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, 2000.

[167] Q. Chen, J. Tang, D. T. C. Wong, X. Peng, and Y. Zhang, "Directional cooperative mac protocol design and performance analysis for ieee 802.11 ad wlans," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2667–2677, 2013.

[168] P. Chatzimisios, A. C. Boucouvalas, and V. Vitsas, "IEEE 802.11 packet delay-a finite retry limit analysis," in *IEEE Global Telecommunications Conference, (GLOBECOM'03)*, vol. 2. IEEE, 2003, pp. 950–954.

[169] J. W. Robinson and T. S. Randhawa, "Saturation throughput analysis of IEEE 802.11 e enhanced distributed coordination function," *IEEE Journal on selected areas in communications*, vol. 22, no. 5, pp. 917–928, 2004.

[170] F.-Y. Hung and I. Marsic, "Performance analysis of the IEEE 802.11 DCF in the presence of the hidden stations," *Computer Networks*, vol. 54, no. 15, pp. 2674–2687, 2010.

[171] K. Chandra, R. V. Prasad, and I. Niemegeers, "Performance analysis of IEEE 802.11 ad MAC protocol," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1513–1516, 2017.

[172] C. Hemanth and T. Venkatesh, "Performance analysis of contention-based access periods and service periods of 802.11 ad hybrid medium access control," *IET Networks*, vol. 3, no. 3, pp. 193–203, 2013.

[173] M. N. Upama Rajan and A. V. Babu, "Saturation throughput analysis of ieee 802.11 ad wireless lan in the contention based access period (cbap)," in *IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*. IEEE, 2016, pp. 41–46.

[174] F. Babich and M. Comisso, "Throughput and delay analysis of 802.11-based wireless networks using smart and directional antennas," *IEEE Transactions on Communications*, vol. 57, no. 5, 2009.

[175] M. N. Upama Rajan and A. V. Babu, "Theoretical maximum throughput of IEEE 802.11 ad millimeter wave wireless LAN in the contention based access period: With two level aggregation," in *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2017, pp. 2531–2536.

[176] A. Akhtar and S. C. Ergen, "Directional mac protocol for ieee 802.11 ad based wireless local area networks," *Ad Hoc Networks*, vol. 69, pp. 49–64, 2018.

[177] H. Shokri-Ghadikolaei, L. Gkatzikis, and C. Fischione, "Beam-searching and transmission scheduling in millimeter wave communications," in *IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 1292–1297.

[178] E. Khorov, A. Ivanov, A. Lyakhov, and V. Zankin, "Mathematical model for scheduling in IEEE 802.11 ad networks," in *Wireless and Mobile Networking Conference (WMNC)*. IEEE, 2016, pp. 153–160.

[179] W. L. Stutzman and G. A. Thiele, *Antenna theory and design*. John Wiley & Sons, 2012.