Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Dottorato di Ricerca in Ingegneria dell'Informazione
Curricula Scienza e Tecnologia dell'Informazione
Ciclo XXXI

Solutions for large scale, efficient, and secure Internet of Things

Supervisore:

Prof. Andrea Zanella

Coordinatore del corso:

Prof. Andrea Neviani

Dottorando:

Daniel Zucchetto

Abstract

The design of a general architecture for the Internet of Things (IoT) is a complex task, due to the heterogeneity of devices, communication technologies, and applications that are part of such systems. Therefore, there are significant opportunities to improve the state of the art, whether to better the performance of the system, or to solve actual issues in current systems. This thesis focuses, in particular, on three aspects of the IoT. First, issues of cyber-physical systems are analysed. In these systems, IoT technologies are widely used to monitor, control, and act on physical entities. One of the most important issue in these scenarios are related to the communication layer, which must be characterized by high reliability, low latency, and high energy efficiency. Some solutions for the channel access scheme of such systems are proposed, each tailored to different specific scenarios. These solutions, which exploit the capabilities of state of the art radio transceivers, prove effective in improving the performance of the considered systems. Positioning services for cyber-physical systems are also investigated, in order to improve the accuracy of such services. Next, the focus moves to network and service optimization for traffic intensive applications, such as video streaming. This type of traffic is common amongst non-constrained devices, like smartphones and augmented/virtual reality headsets, which form an integral part of the IoT ecosystem. The proposed solutions are able to increase the video Quality of Experience while wasting less bandwidth than state of the art strategies. Finally, the security of IoT systems is investigated. While often overlooked, this aspect is fundamental to enable the ubiquitous deployment of IoT. Therefore, security issues of commonly used IoT protocols are presented, together with a proposal for an authentication mechanism based on physical channel features. This authentication strategy proved to be effective as a standalone mechanism or as an additional security layer to improve the security level of legacy systems.

Sommario

La progettazione di un'architettura generale per l'*Internet of Things* (IoT) è un compito complesso, data l'eterogeneità di dispositivi, tecnologie di comunicazione e applicazioni che sono parte di tali sistemi. Ci sono, dunque, opportunità significative per migliorare lo stato dell'arte, sia per incrementare le prestazioni del sistema che per risolvere problemi specifici. Questa tesi si concentra, in particolare, su tre aspetti dell'IoT. Per primo, si analizzano i problemi dei cosiddetti *sistemi cyberfisici*. In tali sistemi, le tecnologie IoT sono ampiamente usate per monitorare, controllare e agire su entità fisiche. Uno dei punti più critici di tali sistemi è relativo alla comunicazione, che deve essere caratterizzata da un'elevata affidabilità, da bassa latenza e da alta efficienza energetica. Sono dunque proposte alcune soluzioni per lo schema di accesso al mezzo di tali sistemi, ognuna specifica per un diverso scenario. Queste soluzioni, che sfruttano le caratteristiche dei moduli di rice-trasmissione radio di ultima generazione, si dimostrano efficaci nel migliorare le prestazioni dei sistemi in esame. Inoltre, vengono investigate tecniche di localizzazione per sistemi cyberfisici, per incrementare l'accuratezza di tali tecniche. In seguito, l'attenzione si sposta sull'ottimizzazione della rete e dei servizi per applicazioni che generano un traffico elevato, come lo streaming video. Questo tipo di traffico è comune tra i dispositivi, come gli smartphone o i visori di realtà virtuale, che non hanno limiti stringenti in termini di potenza di calcolo o consumo energetico, ma che fanno comunque parte integrale dell'ecosistema IoT. Le soluzioni proposte riescono ad incrementare la *Quality of Experience*, seppur usando meno larghezza di banda rispetto alle soluzioni attuali. Infine, viene analizzata la sicurezza dei sistemi IoT. Seppur sia spesso sottovalutato, questo aspetto è fondamentale per permettere l'ampia diffusione di tali tecnologie. Sono dunque presentati i problemi di sicurezza dei più diffusi protocolli IoT, congiuntamente ad una proposta per un meccanismo di autenticazione basato sulle caratteristiche del canale. Questa strategia di autenticazione si è dimostrata efficace sia come meccanismo unico di autenticazione che come tecnologia da abbinare ai meccanismi attuali in modo da migliorarne la sicurezza.

Acknowledgements

I would like to thank the following people, who collaborated in some of the works presented in this thesis: Federico Chiariotti, Rita Coutinho, Michele De Filippo De Grazia, Ole Grøndalen, Kashif Mahmood, Olav N. Østerbø, Chiara Pielli, Giuseppe Ravagnani, Marco Sansoni, Alberto Testolin, Lorenzo Vangelista, Marco Zorzi, and Michele Zorzi.

Special thanks go to my supervisor, Andrea Zanella. Your advice on both research as well as on my career have been invaluable.

Also, thank you to all my fellow labmates for the stimulating discussions and for all the fun we have had in the last years.

Last but not the least, I would like to thank my family for their unconditional support and love.

Table of Contents

List of Acronyms

k-NN k-Nearest Neighbors

AI asymptotic interference-limited

ANN Artificial Neural Network

AR autoregressive

AWGN additive white Gaussian noise

BLE Bluetooth Low Energy

BS Base Station

C-TDMA Contention-TDMA

CDF Cumulative Distribution Function

CF characteristic function

CPS Cyber-Physical System

CS Compressive Sensing

CSMA carrier-sense multiple access

DAG Direct Acyclic Graph

DASH Dynamic Adaptive Streaming over HTTP

FC Fusion Center

i.i.d. independent and identically distributed

IoT Internet of Things

KL Kullback-Leibler

LBT Listen Before Talk

LOS line-of-sight

LPWA Low Power Wide Area

LR Logistic Regression

LRU Least Recently Used

LS-SVM Least Squares Support Vector Machine

M2M Machine-to-Machine

MAC Medium Access Control

MARP Multirate ALOHA Reservation Protocol

MDP Markov Decision Process

MIMO multiple-input-multiple-output

ML maximum likelihood

MOS Mean Opinion Score

MPD Media Presentation Description

MPR Multi-Packet Reception

MRD Multi-Rate Decoding

MS-DS mean squared delay spread

MSE Mean Squared Error

MSSA Multirate-Split Slotted ALOHA

MTC Machine-Type Communication

MTD Machine-Type Device

NB Naive Bayes

NFC Near Field Communication

NL noise-limited

NLOS non-line-of-sight

OFDM orthogonal frequency division multiplexing

PC Popular Content

pdf probability density function

PLA physical-layer authentication

PPP Poisson Point Process

PRMA Packet Reservation Multiple Access

QoE Quality of Experience

QoS Quality of Service

R-ALOHA Reservation ALOHA

r.v. random variable

RBM Restricted Boltzmann Machine

RF Random Forest

RM Reservation Message

RMSE Root Mean Square Error

RSS received signal strength

RTT Round Trip Time

SA Slotted ALOHA

SINR signal-to-interference-and-noise ratio

SNR Signal-to-Noise-Ratio

SPR Single-Packet Reception

SSIM Structural Similarity

SVM Support Vector Machine

SVR Support Vector Regression

TDMA Time Division Multiple Access

ToA time of arrival

UWB ultra-wideband

WSN Wireless Sensor Network

Chapter 1

The Internet of Things: An overview

Building a general architecture for the Internet of Things (IoT) is a very complex task, exacerbated by the extremely large variety of devices, link layer technologies, and services that may be involved in such a system. In this introductory chapter, we analyse the main blocks of a generic IoT architecture, describing their features and requirements, and investigating the most common approaches proposed in the literature for each block. The analysis will prove the importance of adopting an integrated approach that jointly addresses several issues and is able to flexibly accommodate the requirements of the various elements of the system.

1.1 Introduction

The IoT is a paradigm in which sensors and microcontrollers are extended into the world of everyday objects and actively exchange information to achieve common goals. This technology shift is deemed to be the next stage of the information revolution after the massive spreading of the Internet in every field, and its impact is expected to be much heavier than that caused by the integration of the Internet in our lives through smartphones and other mobile devices. In fact, the IoT shall be able to seamlessly incorporate a large number of heterogeneous end systems, while providing open access to selected subsets of data for the development of digital services [11]. The integration of potentially any object into the Internet allows for new forms of interactions between humans and devices, or directly among devices, according to what is commonly referred to as the Machine-to-Machine (M2M) communication paradigm [12].

Cyber-Physical Systems (CPSs), which can be considered a subset of the IoT, are engineered systems that deeply integrate with the physical environment surrounding them. In particular, a CPS is composed by a network of elements that interact with the physical world through computation, communication, and control capabilities [13, 14]. Examples of CPSs are autonomous vehicles [15], medical devices [16], robotics [17, 18], and smart power grids [19]. Furthermore, Wireless Sensor Networks (WSNs) and many common low-power IoT applications can be included in the CPS area too.

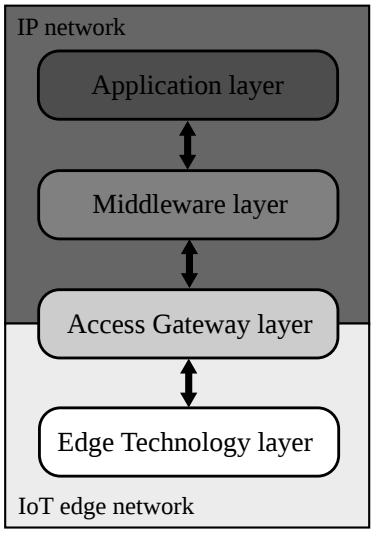


Figure 1.1: IoT architecture

Traditionally, the design of the cyber and physical parts of a system have been decoupled, while CPS emphasizes a holistic system view where the focus is on the inter-dependency and interaction of both parts of the system. In addition, CPSs link two different domains: the continuous and discrete domains of physical and cyber components, respectively. This heterogeneity makes the analysis of these systems a major challenge [14]. While many of the techniques presented in this thesis can be applied in general IoT scenarios, the focus is mainly to CPSs applications.

There is an increasing interest in both IoT and CPS areas, together with the number of their deployments. A recent investigation from Juniper Research has revealed that the number of IoT connected devices is predicted to be 50 billion in 2022, up from an estimated 21 billion by the end of 2018, a rise of 238% [20] whereas other studies [21] foresee 19.8 billions of non-phone interconnected devices against 8.6 billions of phone devices by 2023. According to the McKinsey analysis [22] the potential economic impact of IoT is going to be between $3.9 trillions and $11.1 trillions by 2025.

However, the heterogeneity of both end devices and applications complicates the already challenging development of the IoT [23], which needs to cope with massive access to the transmission channels, security issues and energy efficiency problems, which are stressed by the use of constrained end devices. To cope with these issues, the ongoing research in the scientific community addresses all layers of the protocol stack, from physical transmission up to data representation and service composition.

Although it is not straightforward to define a unified scheme for the various IoT applications, it is possible to pinpoint the basic blocks that make up every IoT architecture (Fig. 1.1) [24]:

- *Edge Technology layer*: it is the hardware layer that represents the *things* part of the IoT and consists of sensors and actuators. The main function

of this layer consists in collecting information from an environment or a system and processing this information. The end devices must be able to communicate with the Access Gateway layer in order to transmit the collected observations and to receive feedback from the upper layers. Several solutions have been proposed for efficiently managing communication among the end devices, which typically are severely constrained in terms of computational and storage capabilities, and energy capacity;

- *Access Gateway layer*: it represents the point of access to the Edge Technology layer and basically revolves around data handling, i.e., forwarding the information generated by the end devices to the middleware layer and sending data produced by the latter back to the devices. It must provide all the functions that the constrained peripheral nodes cannot bear and must support protocols for communicating with the IP world, whilst coping with the large variety of devices that may be present in the network;

- *Middleware layer*: it is the intermediate layer between the *things* and the Internet and is mainly responsible for filtering and storing the data received from the end devices. It is also responsible for enforcing the security policy in the IoT network. This layer must be able to cope with the device heterogeneity and hide it to the IoT applications in order to facilitate their access to sensor data;

- *Application layer*: it is the layer responsible for presenting the information to the final user. It provides a high level management of all involved devices in an integrated way, ensuring scalability, high availability, and the reliable and secure execution of the requested functionalities from the devices.

To cope with the intrinsic constraints of IoT scenarios, several challenges must be addressed at every layer and the development of the whole architecture also has to efficiently integrate every element without interfering with the rest of the system. The goal of this chapter is to provide an introductory overview of the protocols and technologies that can be employed in the communication between the various elements of an IoT system. The following sections illustrate the most advanced commercially available solutions for IoT systems, from physical and link layer technologies to network management and security. Drawbacks of each solution are highlighted, showing the need for improvements over the state of the art. In the remaining chapters of the thesis, then, each topic will be investigated more deeply, in order to propose techniques for solving such issues and quantify the resulting improvements.

1.2 IoT wireless technologies

The basis of an IoT system is to be found in the physical connection between the devices in the IoT network. Most connections between *things* use wireless technologies, since the number of devices to be connected is usually large and, in many scenarios, devices need to be mobile, e.g., in the case of wearable technologies or for tracking purposes, or placed in locations where wiring is not feasible or convenient.

Multi-hop short-range wireless technologies have been the first enabler of IoT networks. Things connected to these networks usually run on dedicated protocol stacks specifically designed to satisfy the device constraints, although at least one of these devices must be connected to an IP network. Devices connected to both the IoT and an IP network act as access gateways, as they allow users to communicate with *things* via traditional devices like PCs and smartphones. In this group, the most prominent technologies are the IEEE 802.15.4 family, including ZigBee and 6LoWPAN, and ZWave. These kinds of networks operate in the 2.4 GHz and 868/915 MHz unlicensed industrial, scientific and medical (ISM) frequency bands, with devices connected using a mesh topology. The distance between these devices ranges from few meters up to roughly 100 m, depending on the specific technology used and on the surrounding environment. One of the downsides of using a multi-hop technology is the need, for nodes in the network, to keep the radio circuitry on in order to forward the messages coming from other nodes, thus reducing the power efficiency of the network and reducing the battery life of nodes. These technologies have also proven to be inadequate in scenarios where the network must provide a large coverage range, as in Smart City applications.

To overcome these shortcomings, new technologies have been proposed. They can be grouped in two families: cellular IoT networks and Low Power Wide Area (LPWA) networks and, unlike multi-hop short-range wireless technologies, they enable a *place-&-play* connectivity [25], i.e., any device can be connected to the IoT network by simply placing it in the desired location and switching it on. In particular, the Third Generation Partnership Project (3GPP), which is the body that developed the specifications for the most popular cellular technologies, attempted to revamp GSM (Global System for Mobile Communications) to support IoT devices, thus implementing the Cellular IoT architecture [26]. A possible issue that arises in these types of networks is the massive number of devices that need to access the transmission channel. Since cellular technologies were not designed to provide machine-type services to a huge number of devices, the signaling and control traffic may become the bottleneck of the system [23]. To solve these issues and improve compatibility with future cellular networks, 3GPP introduced IoT-specific cellular technologies in Release 13, namely Narrow-Band IoT (NB-IoT) and Enhanced Machine-Type Communication (eMTC). These technologies are targeted at improving coverage while reducing complexity and energy consumption of cellular IoT devices [27, 28].

A possible alternative is represented by LPWA networks, which combine the use of dedicated protocol stacks tailored to constrained devices, with long coverage range. In this kind of networks, the end devices are connected to a central aggregator, generally referred to as *gateway*, which provides bridging to the IP world in a fashion similar to the access gateway in multi-hop networks. The gateway coverage range is in the order of kilometers, making it possible to serve an entire city with a limited number of gateways. A limit of these networks is the low bitrate that, however, is expected to be sufficient for many IoT services.

The first LPWA technology proposed in the IoT market is SIGFOX,[1] founded in 2009. The SIGFOX physical layer uses an Ultra Narrow Band (UNB) modulation coupled with sub-GHz bands to ensure a great coverage range. SIGFOX,

[1] `http://www.sigfox.com/`

Technology	SIGFOX	Ingenu	LoRa
Coverage range [km]	rural: 30–50 urban: 3–10	≈ 15	rural: 10–15 urban: 3–5
Frequency bands [MHz]	868 or 902	2400	various, sub-GHz
Data rate [Kbps]	0.1	0.01–8	0.3–37.5
Nodes per BS	$\approx 10^6$	$\approx 10^4$	$\approx 10^4$

Table 1.1: Comparison among LPWA radio technologies.

which acts as an operator for IoT services, already deployed its nation-wide access networks in many European countries, including France, Spain and the Netherlands, thanks to the great coverage range of their gateways, claimed to be 30–50 km in rural areas and 3–10 km in urban areas [29].

A further LPWA technology is LoRa, designed and patented by Semtech Corporation [30], which also manufactures the chipsets. Its physical layer uses a derivative of Chirp Spread Spectrum, operating in the unlicensed sub-GHz bands. LoRa systems are being deployed by telecommunication providers like Orange and Bouygues Telecom in France, Swisscom in Switzerland, and KPN in the Netherlands. While the physical layer of LoRa is proprietary, the rest of the protocol stack, known as LoRaWAN [31], is being developed by the LoRa Alliance[2], an association of industry partners dedicated to the development of LoRa solutions.

Ingenu,[3] a trademark of On-Ramp Wireless, is another example of LPWA technology. Ingenu networks, unlike most of the other LPWA technologies, operate in the 2.4 GHz band, but thanks to the use of the patented Random Phase Multiple Access technology [32], can still work over long distances. In collaboration with Meterlinq, Ingenu is deploying a nationwide network in Italy to enable smart water and smart gas monitoring, with the long-term goal to scale the network to include additional IoT applications. Also, a nationwide network is being deployed in the USA.

Tab. 1.1 shows a comparison of these LPWA radio technologies, highlighting, in particular, the differences in bitrate and declared coverage range [25].

1.3 Messaging protocols

The interaction with the specific wireless transmission technologies discussed in the previous section is typically realized by means of standard Application Program Interfaces (APIs) and communication protocols that can be logically placed on the *Access Gateway layer* of Fig. 1.1. The goal of this layer is to abstract the specificities of the lower layers and provide common ways to access the data collected by the IoT nodes. This section describes the most important protocols that are being proposed for this purpose. In particular, the focus is

[2] `https://www.lora-alliance.org/`
[3] `https://www.ingenu.com/`

on REST, MQTT and AMQP, as all of them are widely used and provide a comparable set of features. However, these communication protocols have been developed starting from different requirements and with contrasting use cases in mind, thus providing dissimilar performance in various scenarios.

REST

The Representational State Transfer (REST) is a software architecture style for building scalable web services, typically over the Hypertext Transfer Protocol (HTTP) [33], and originated from the Ph.D. thesis of Roy Fielding in the year 2000 [34]. For a service to be identified as RESTful, the following five constraints must be respected.

- *Client-server*: a RESTful service follows a client-server model, with separation of concerns.
- *Stateless*: at the server side, no information about session and client context is retained and each request is an independent transaction that is unrelated to any previous request. So, each client request needs to contain all the information necessary to serve the request and only the client holds the session state. In this way servers are simpler and scalability is enforced.
- *Layered system*: client and server may not be directly interconnected. Intermediary servers may improve system scalability by enabling load balancing and providing shared caches, and may also enforce security policies.
- *Cacheable*: clients and intermediaries can cache responses, allowing an improvement in scalability and performance.
- *Uniform interface*: the uniform interface between client and server allows each part to evolve independently. This constraint is based on two notions: first of all, individual resources must be identified in the requests and, secondly, these resources can be manipulated according to the CRUD pattern: Create, Retrieve, Update and Delete.

A further optional requirement is that servers shall be able to transfer executable code to clients.

The concept of resource is central in RESTful services: every resource is globally and uniquely identified by a Uniform Resource Identifier (URI) and is considered as an abstract entity disconnected from its representation. Since REST APIs are used almost exclusively over HTTP, in this work we will consider only REST over HTTP.

MQTT

The Message Queuing Telemetry Transport (MQTT) protocol is a lightweight event and message oriented protocol allowing devices to asynchronously communicate across constrained networks to remote systems. MQTT, version 3.1.1, has recently been standardized by the Organization for the Advancement of Structured Information Standards (OASIS) consortium [35] and has been submitted to the International Organization for Standardization (ISO) in order to

become an International Standard [36]. The MQTT protocol has been initially designed to communicate telemetry data in a M2M scenario, and therefore can work in unreliable networks with small bandwidth and high latency. The size of the message header can be as small as 2 bytes, since, in IoT and M2M scenarios, messages are typically short and control information may easily become the predominant part of the communication. The protocol has a client-server architecture: the server part is represented by a central broker that acts as intermediary among the clients, i.e., the entities that produce and consume the messages. MQTT revolves around the concept of topics, which are UTF-8 (Unicode Transformation Format, 8 bit) strings used by the broker to filter messages for each connected client. Topics are used by clients for publishing messages and for subscribing to the updates from other clients. This pub/sub mechanism avoids the need for consumer entities to continuously poll the data producers for new messages: through a topic subscription, an MQTT client receives all the messages published by other clients for that topic. MQTT libraries have been provided for all major IoT development platforms, for the two major mobile platforms, i.e., Android and iOS, and for several programming languages (Java, C, PHP, Python, Ruby, Javascript).

AMQP

The Advanced Message Queuing Protocol (AMQP) is an open Internet protocol for message exchange. AMQP version 1.0 has been standardized by the OASIS consortium [37] and successively by the ISO, as ISO/IEC 19464:2014 [38]. Its original goals were to enable communication between systems of different vendors, support messaging semantics needed in the financial service industry, be extensible to new queuing and routing strategies, and allow complete configuration of the message routing. The initial development of AMQP started from the initiative of financial institutions that needed to reliably exchange data between heterogeneous systems. Since then, this protocol has been successfully used to exchange messages in M2M scenarios. In an AMQP system, the entities that produce and consume messages over the network are linked to central messaging servers, called brokers. At the broker, inbound messages are put in different queues, waiting to be collected by message consumers. The message routing is very flexible, as it allows, e.g., to send messages in broadcast, to direct them to a single entity, or to use a topic-based pub/sub mechanism, as in MQTT.

In the rest of this section, we compare the protocols performance with a focus on the IoT scenario. Hence, we consider the following aspects: support of different IoT traffic patterns, data encoding and manipulation, reliability, and security.

1.3.1 Support of different IoT traffic patterns

In an IoT network many devices are linked to a central aggregator that monitors and operates them. An IoT network can be used to serve several purposes, which may differ in the way the messages are exchanged between network nodes. It is possible to categorize the message exchanges of an IoT network in four different communications patterns (Fig. 1.2): *telemetry*, *notifications*, *inquiries*, and *commands* [39]. Summarized results from the analysis of these patterns, together with the protocols support for them, can be found in Tab. 1.2.

Telemetry: in the telemetry pattern, the device autonomously sends data to the central aggregator with a fixed time period or at the occurrence of some events. The aggregator, upon data reception, simply stores them for further analysis. Data messages are usually small, with lengths of some tens of bytes, but frequent. Depending on the use case, the average interval between the data messages could range from hours (e.g., for environmental monitoring) to fractions of a second (e.g., for car telemetry). The HTTP protocol, being ASCII-oriented, is too verbose for short messages, since each optional header greatly increases the size of the message. For this reason, the REST approach is not very efficient in this case. MQTT, instead, was designed to have a low overhead and to be very efficient in case of short messages. AMQP, due to its many features, has a larger header than MQTT, however it provides a flow control mechanism, not present in HTTP and MQTT, to slow down the source in case the destination is unable to keep up to the rate of messages.

Notifications: refer to the central aggregator sending messages to the devices to notify them about an event. It has many characteristics in common with the telemetry pattern, but it also requires all the end devices, or their associated access gateway, to be reachable from the central aggregator. Hence, if using the REST architecture over HTTP, special attention is required to connect nodes through Network Address Translation (NAT) and firewall gates. Furthermore, each device must host an HTTP server to be able to receive the notifications, adding complexity to the device, which is usually constrained in terms of memory and computing power. None of these issues arise for MQTT and AMQP because, in those protocols, connections are always initiated by the client, so that only the message broker needs to be publicly reachable. Furthermore, to receive and send messages, nodes only need the MQTT or AMQP client, which can run in a constrained environment.

Inquiries: in this pattern the end device (or the element in charge of connecting the end device to the IP world) sends requests to the central aggregator, which successively answers with the required information. This is exactly the use case addressed by the REST architecture, since it can be seen as a traditional request-response pattern. Here the HTTP server must be placed in the central aggregator, while the end devices (or the access gateways) are equipped with an HTTP client, which is less demanding in terms of computing resources than the HTTP server. At the beginning, the MQTT protocol did not describe a way to exchange messages in a request-response pattern, so that the parties must agree beforehand on pair topics for this pattern: a topic for publishing requests and another for publishing responses. However, the OASIS MQTT Technical Committee included a mechanism to formally enable the request-response messaging pattern in MQTT 5. AMQP, instead, already supports a mechanism to enable the request-response message exchange, thus providing the flexibility needed to operate in this scenario.

Commands: in this case, the central aggregator sends a message to the end device to trigger an action and then waits for the reply by the end device containing the outcome of the action. Besides the reachability issue described in the notifications pattern, the REST architecture also fails to manage the case in which the end device is temporarily offline: messages sent while the device is unavailable are simply lost. MQTT addresses this problem with the introduction of the optional retain flag in the published message, forcing the message to be sent to all clients that, in the future, will subscribe to the corresponding

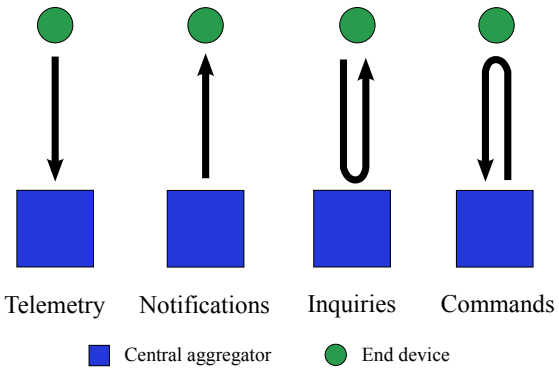


Figure 1.2: Communication patterns

Features	Telemetry			Notifications			Inquiries			Commands		
	REST	MQTT	AMQP	REST	MQTT	AMQP	REST	MQTT	AMQP	REST	MQTT	AMQP
Small overhead	✗	✓	∼	✗	✓	∼	✗	✓	∼	✗	✓	∼
Flow control	✗	✗	✓	✗	✗	✓	Not required			Not required		
Reachability behind NAT or firewall	Not required			✗	✓	✓	Not required			✗	✓	✓
Supports constrained devices	✗	✓	✓	✗	✓	✓	✗	✓	✓	✗	✓	✓
Request-response	Not required			Not required			✓	✓	✓	✓	✓	✓
Publish-subscribe	✗	✓	✓	✗	✓	✓	Not required			Not required		
Supports temporarily offline devices	Not required			✗	∼	✓	Not required			✗	∼	✓
General fit	✗	✓	✓	✗	✓	✓	✗	∼	✓	✗	∼	✓

Table 1.2: Features required for the various communication patterns and their support by the considered protocols.

topic. As mentioned before, it is also important to remark that MQTT lacks a formalized request-response mechanism to correlate the request to the end device with its response. Regarding AMQP, its message broker always saves incoming messages to the message queues, allowing them to be retrieved at a later time even if the recipients are temporarily unavailable. Furthermore, AMQP introduces a refinement that consists in a Time-to-Live indication, to remove stale messages from the queues.

1.3.2 Data encoding and manipulation

Usually, message recipients elaborate the received data depending on the type of the message content. AMQP features a rich set of metadata to describe the transmitted data, including a complete type system. Actually, AMQP type system defines some primitive types, together with constructs to extend them in order to allow the association of an AMQP value with an external type that is not present as an AMQP primitive. This feature is not present in REST nor

in MQTT. Actually, the REST architecture only allows to specify the type of message content through the HTTP *Content-Type* header, but lacks a more comprehensive metadata set. The message content is, instead, completely opaque to MQTT, which does not even allow the indication of generic information such as the *Content-Type*. In order to elaborate the message content, the parties must therefore agree beforehand to the exact format of the message, and a change in message format requires the communicating parties to be manually updated.

1.3.3 Reliability

In the IoT field, reliability refers to the absence of communication errors in the transmitted data and the guarantee that transmitted messages have been delivered to the recipients. REST over HTTP relies only on the underlying Transmission Control Protocol (TCP) to provide reliability of message exchanges, while MQTT and AMQP offer more flexible mechanisms to provide additional levels of reliability assurance [40]. In MQTT, Quality of Service (QoS) is an attribute of the individual message being published. However, due to device or link constraints, a subscribing client can set the maximum quality of service a broker can use to send messages to it, hence each message will be delivered with a QoS value that is the minimum between the value of the QoS attribute in the message and the maximum QoS accepted by the client. The QoS attribute can take three possible values: *at-most-once* (QoS level 0), in which no acknowledgement is needed, *at-least-once* (QoS level 1), which requires the transmission to be acknowledged, and *exactly-once* (QoS level 2), that requires a more sophisticated acknowledgement mechanism which involves the exchange of three acknowledgement messages. QoS level 2 is the only level that can be used for non-idempotent messages that must be delivered reliably, since it guarantees that the message is not delivered multiple times, unlike QoS level 1. However, it is to be noted that, with the highest QoS level, the overhead is large and sending messages at a high rate may degrade the system performance. AMQP has QoS properties similar to those of MQTT, supporting message queuing and delivery semantics that cover at-most-once, at-least-once and exactly-once deliveries. Furthermore, the AMQP specification also describes an optional transaction mechanism with a multiphase commit sequence, to ensure that each message is delivered as intended, regardless of failures or reboots.

1.3.4 Security

REST over HTTP, MQTT and AMQP can all be placed on top of Transport Layer Security (TLS) [41], which provides confidentiality of the data exchanged. TLS also supports authentication of the server, which is the message broker in case of MQTT and AMQP, or the HTTP server in case of REST. While TLS could also be used to authenticate clients, this is not commonly employed because it involves the generation and management of a certificate for each client that must be able to connect to the server. Instead, client authentication is typically implemented in the protocol running on top of TLS. With the REST architecture it is possible to use HTTP Basic or Digest authentication mechanisms. MQTT, instead, allows the clients to specify the username and, optionally, a password while connecting to the broker. AMQP does not provide an

authentication mechanism itself, but allows the use of the Simple Authentication and Security Layer (SASL) framework.

To conclude, the choice of the protocol to use depends on the specific use case. The most important factors to consider are: the rate at which new messages are generated, the underlying network performance, the reliability of links, the necessity of extensibility and message semantic, and the quality of service required. According to this analysis, the choices made for a specific use case are explained in Sec. 1.6.

1.4 Between the Things and the Internet: the middleware

IoT systems often deal with different types of devices, each with its own communication protocol and different requirements, that need to somehow interact with the final user. In order to meet this demand, IoT architectures require a software platform, called middleware (see Fig. 1.1), which represents an intermediate layer between the Internet and the *things* and acts as a bond joining mixed applications communicating over heterogeneous interfaces. The middleware is also in charge of masking the system complexity that is faced when interacting with the end devices, so that even the average technology-inexperienced user is able to enjoy IoT services effortlessly.

The development of a middleware in the IoT context requires the support of various functionalities. The following list summarizes the crucial issues that the middleware must address [42] [43]:

- *Interoperability*: the conceived middleware must cope with the great heterogeneity of the smart objects. Interoperability aims at device abstraction and is threefold: technical, syntactic and semantic. According to the European Telecommunications Standards Institute (ETSI) [44], technical interoperability is defined as the association of hardware or software components, systems and platforms that enable M2M communication to take place. Syntactic interoperability deals instead with data formats and asks for an agreed upon and well-defined common syntax for messages. Finally, semantic interoperability is associated with the ability of computer systems to exchange data with unambiguous and shared meaning, understandable to humans.

- *Device discovery and management*: bootstrapping is a crucial phase in the IoT as it prepares the smart objects to join the network and to interact with the other end devices, detecting all their neighbours and making their presence known. Moreover, the middleware must be aware of the context in order to work in smart environments, as the smart objects may move in a random fashion causing rapid changes in the network topology. An IoT middleware must be able to update routing information in an efficient way without affecting the overall network performance and independently of the routing protocol used. Another issue related to device management involves actuators, which may be accessed simultaneously by different applications in a contradictory way: the middleware is in charge of solving such conflicts.

- *Security and privacy aspects*: security is a key point in IoT architectures, which often deal with sensitive data. Thus the middleware must ensure authentication, confidentiality, data integrity and non repudiation and must be able to manage different roles and privileges.

- *Application abstraction*: the middleware should provide an interface for both high-level applications and end users to interact with the end devices without prior knowledge about the physical network and the implementation details.

- *Data management*: the IoT is leading to an explosion of data exchanges, thus the middleware needs to cope with enormous volumes of data. It is also necessary to have historical data stored, which allow the end user to retrieve old observations and to display time-series graphs.

Other useful features concern modularity, i.e., the possibility to add functionalities without altering the existing core, which is essential to customize the platform in a plug-and-play fashion for accommodating missing features, and the capability of supporting downlink traffic towards the end devices to enable actions from the final users. Many challenges need to be addressed in order to build an efficient, robust, scalable and real-time platform. For these reasons, it may be preferable not to develop a custom middleware from scratch, but rather to use an existing and well tested platform and adapt it to fulfill the specific system requirements, if needed. There exist many implemented middlewares; in this work three different frameworks are analysed, namely openHAB, Sentilo, and Parse, which are intended to be used along with an access gateway to provide the described functions. Their characteristics are detailed in the rest of this section.

1.4.1 openHAB

The software platform openHAB[4] targets home automation and was born in 2008 from the need of its creator, Kai Kreuzer, to integrate sensors and actuators in his own house in Darmstädt (Germany).

OpenHAB is extensible through a plug-and-play principle and interoperable thanks to the use of modules to support different communication protocols and mechanisms. Many modules have already been implemented, such as the MQTT binding, a component that allows openHAB to act as an MQTT client and hence to support a pub/sub mechanism for seamlessly interacting with the nodes. End devices are registered in openHAB as *items*. An item is a data-centric functional atomic building block: all openHAB resources are represented using this abstraction, which is independent of the technology used. In this way the final user does not need to be aware of the physical network technology employed, since he/she only needs to dialogue with openHAB via HTTP. Historical data can be stored in relational, NoSQL or round-robin databases, in IoT cloud services or in simple log files, according to the user needs. For what concerns security, TLS can be enabled for all the protocols that support it and user authentication is also possible. However, openHAB targets home automation and therefore it is designed to be used by a limited number of users, all

[4] `http://www.openhab.org/`

with complete access to the available information. The implementation of user differentiation according to given roles is on the future work list and will make it possible to assign different read and write permissions to users.

The strengths of the openHAB platform are its high modularity and the presence of bindings that support different protocols. Its main shortcomings, instead, concern the lack of a user conditional access and the internal items implementation, since items cannot be bound to a specific time and geospatial context nor customised according to the users needs. This abstraction, for example, makes it impossible to store location information to track mobile end devices; also, past measures cannot be inserted at a later time, since data cannot contain a custom timestamp.

The openHAB project gave rise to Eclipse SmartHome, a flexible framework for the smart home. Eclipse SmartHome will be the basis for the next iteration of the openHAB project, namely openHAB 2, which is still in its early stages of development.

1.4.2 Sentilo

Sentilo[5] is the product of a project started in November 2012 by the Barcelona City Council and conceived to make Barcelona a reference point in the field of Smart Cities. The name Sentilo was chosen because it means *sensor* in Esperanto, underlying the intention of openness and universality in the use of a platform.

Sentilo is an extensible open source platform that offers a REST API over HTTP, supporting all the communication patterns described in Sec. 1.3. However, Sentilo does not support other communication protocols: it is necessary to implement a bridging module for every other protocol to be used in the architecture, in order to properly translate its messages into their REST equivalent. Sensors and actuators are registered in Sentilo as uniquely identifiable items and are organized according to a hierarchical structure. It is also possible to track the location of mobile sensors. For what concerns access control, Sentilo features a token-based authentication system to identify the petitioner of the request, coupled with a privilege policy based on roles. Also, to provide confidentiality, the REST API can be used over the secure HTTPS channel.

To recap, the selling points of Sentilo are the possibility of extending its functionalities in a plug-and-play fashion, the presence of a hierarchical and slightly customizable item representation and the implementation of an authorization and role-based permission mechanism that facilitates the interaction in the same context of multiple users with different roles. Sentilo's most important drawback, instead, is its weak interoperability, as it natively supports only the REST API and cannot communicate via other protocols.

1.4.3 Parse

Parse[6] is a cloud-based data management system that allows people to quickly develop web and mobile apps. More specifically, it is a Backend as a Service (BaaS) solution, a turnkey service that adds user authentication, push notifications, social media integration, location data, and data analytics into any

[5] `http://www.sentilo.io`
[6] `https://parse.com`

app. It was acquired by Facebook in 2013 with the aim of adding Mobile BaaS capabilities to the existing platform and, as IoT backends are the logical extension of mobile backends, at the end of March 2015 Facebook announced Parse for IoT. Parse for IoT is a collection of Software Development Kits (SDKs) for connected devices, such as Arduino Yún, a microcontroller board with built-in WiFi capabilities. Parse SDKs are directly deployed on hardware platforms and provide a simple REST API. Such SDKs make devices able to receive push notifications, save data, and take advantage of the Parse Cloud. All Parse resources are represented as Parse Objects, uniquely identified and customizable. Particular objects are the *roles*, which group users with common access privileges in order to support role-based access control. Even data storage on Parse is built around a Parse Object: there is no need to explicitly create databases or tables to use Parse, since data will be automatically stored in the cloud. Finally, it is possible to extend the functionalities of Parse for the IoT by creating the so called Cloud Modules.

To sum up, the strengths of Parse are the great customization available for Objects and the presence of a solid permissions and roles structure to control user access. The major weaknesses are instead the need of installing the SDK on each device and the inability to communicate in a way different from REST. Moreover, being Parse for IoT so recent (it was officially announced just a few months ago), this tool is not widely deployed, so a proof of its real-world performance is still missing.

1.4.4 Platforms comparison

The above descriptions highlight the great effort required to develop a complete middleware that simultaneously accomplishes all the listed requirements.

All three platforms represent valuable middleware solutions for the IoT, but at the same time all of them lack some useful features. They all provide modularity, data management and application abstraction, which is typically achieved by implementing a REST API designed to be used by the user interface. Security is achieved by means of authentication and message encryption, but, unlike the others, openHAB does not offer user differentiation according to roles. However, openHAB is the only middleware, among the three described in this study, capable of supporting different communication protocols simultaneously. Therefore, openHAB is suitable for small deployments, where there is no need to distinguish among end users, whereas Sentilo, which supports user roles, may be employed in wider deployments, although it may require the implementations of protocol bridging bindings. Parse for IoT is not the best suited platform for large existing IoT networks, since it requires the SDK installation on all the hardware devices, and therefore depends on the particular sensors and network topology used.

1.5 Security in IoT systems

A central aspect of every IoT application is security, which must be guaranteed at every level of the system. IoT security, in particular, revolves around the concepts of identification, confidentiality, integrity and availability, and needs to meet the new requirements implied by the pervasive presence of the Internet

PLATFORM	OPENHAB	SENTILO	PARSE
modularity	✓	✓	✓
application abstraction	✓	✓	✓
multiple protocols support	✓	×	×
semantic and syntactic interoperability	✓	✓	✓
data management	✓	✓	✓
data storage	✓	✓	✓
items custom representation	×	✓	✓
user conditional access	×	✓	✓
add timestamp to transmitted data	×	✓	×
security	✓	✓	✓
growing community	✓	✓	✓

Table 1.3: Comparison among the platforms

in any aspect of daily life. Internet-facing services are in fact under continual attack and this does not bode well for the IoT, which relies on it and also incorporates many constrained devices for which it is hard to apply security mechanisms such as frequency hopping communication and public key encryption [45]. But as the IoT also touches many sensitive areas, security represents a challenge that cannot be neglected: attacks and malfunctionings would just outweigh any of the IoT benefits. Security experts are currently investigating whether current security mechanisms can be integrated in the IoT or new designs are required to accomplish security goals. What mainly introduces new threats is the distributed nature of IoT architectures and the use of fragile technologies, such as limited-function embedded devices in public areas where they are accessible by anyone and may be physically harmed [46]. As sensors are typically simple low power devices, they cannot even support ordinary security measures: network firewalls and protocols can manage the high-level traffic flowing through the Internet, but the protection of the endpoint devices with limited resources available to accomplish it raises new challenges and demands for revolutionary solutions.

The most important security features for IoT systems are the following.

- *Identification*: the *things* must be uniquely identified, independently of their underlying mechanisms, e.g., the IP address they are associated to. Assigning a unique identifier to devices is the basis for the authentication step and the consequent authorization phase.

- *Confidentiality*: it refers to the guarantee that information is not made available or disclosed to unauthorized individuals, entities, or processes.

Confidentiality is fundamental in an IoT scenario, in which a plethora of devices transmit messages, leading to an explosion of data. Access to these data must be controlled mainly by means of cryptographic mechanisms and users access lists.

- *Integrity*: to maintain the consistency, accuracy, and trustworthiness of data over its entire life cycle, data must not be changed in transit or altered by unauthorized people.

- *Availability*: for any information system to serve its purpose, the information must be available when it is needed. Availability may be hindered by legitimate users too, if they flood the network with requests that exhaust network resources, interrupting services available to other legitimate users.

Clearly, the IoT is prone to be more susceptible to attacks than the rest of the Internet, since billions of devices will be producing and consuming a large number of different services. From a network perspective, the sensors should open a secure communication channel with more powerful devices exploiting cryptographic algorithms and using an adequate system for exchanging the keys. A safe transmission over TCP/IP connections can be achieved by enabling Transport Layer Security (TLS), which asks the parties to authenticate themselves and provides message encryption. At the application level, security needs for different application environments are different, although data privacy, access control and disclosure of information are likely common requirements. In [24] the authors stress the crucial role of security and privacy and highlight how the public acceptance of the IoT will happen only when strong security and privacy solutions will be in place. In fact, when the Internet first appeared, no security infrastructure had been built for it. But, when the first security problems came out, the only viable solution to solve them was to treat security and privacy as add-on features. In the IoT, instead, security has to be intrinsic, hence we must find new fundamental solutions, shared among all interested parties, for addressing this challenge.

1.6 A use case: an IoT system for Smart Cities

In this section we analyse the development of a complete IoT system that targets the Smart Cities context, a project carried out by Patavina Technologies s.r.l.[7] in the city of Padova, Italy.

LoRa has been chosen as the wireless technology. From the analysis carried out in [25] and summarized in Sec. 1.2, it results that, although the declared coverage range of LoRa is slightly lower than that of the other two technologies, the transmission data rate achievable with LoRa is higher. The LoRa network is typically laid out in a *star-of-stars* topology, where the end devices are connected via a single-hop LoRa link to one or many gateways which, in turn, are connected to a common Network Server (NetServer) via standard IP protocols [31]. The NetServer represents the point of access for the LoRa network as it can support other communication protocols.

[7] `http://www.patavinatech.com/en`

The NetServer communicates with the rest of the world using MQTT. As outlined in Sec. 1.3, MQTT is a lightweight protocol that meets IoT requirements thanks to its very small message header and the pub/sub mechanism, features not provided in HTTP. AMQP was considered unsuitable because it is very complex and all of the features missing in MQTT but provided in AMQP were not necessary for the implemented architecture. Therefore, setting up AMQP clients and broker would have required much more work with very little benefit. An advantage of AMQP over MQTT is the message queueing system that allows the clients not to miss messages arrived whilst they were unavailable. Anyway, when unmissable data is sent, it is still possible to use the simple retain mechanism available in MQTT, as explained in Sec. 1.3. Patavina Technologies also developed mechanisms for detecting the unavailability of any MQTT element and activate it again within a short time, assuring that no MQTT component is affected by an extended downtime. Request and response topics and the format of the message payload have been agreed beforehand.

The selected middleware is Sentilo. In fact, Parse for IoT demands the SDK installation on hardware devices, restricting the freedom of choice on the development environment, and mandating the use of REST interfaces on the devices, a choice not well suited for the constrained devices and network technology used in our solution. On the other hand, openHAB did not provide a mechanism for user differentiation according to roles. Hence, the overall architecture sees LoRa devices that communicate with the IP world through the NetServer, which supports MQTT. The messages sent by the NetServer are processed by an Application Server (AppServer) and the new human-readable MQTT messages are forwarded to Sentilo, which is in charge of registering the nodes and their data so that authorized users can access them. Along with the network management infrastructure, a friendly website for interacting with the nodes and displaying their readings has been developed.

The system deployment corroborated the choices made by Patavina Technologies. LoRa network connectivity has been tested by installing a private network in a large and tall building (19 floors), with nodes placed also in heavily shielded positions, e.g., inside elevators in order to put the connectivity condition under strain. Other experimental tests have been carried out in Padova with the purpose of assessing the worst case coverage in an urban environment. It turned out that, in harsh propagation conditions, the LoRa technology allows to cover a cell of about 2 km of radius and, even when assuming a radius of 1.2 km to take into account a reasonable margin for interference and link budget variations, the number of gateways needed for assuring coverage in the municipality of Padova is much lower than the number of sites required by one of the major cellular operators in Italy to provide mobile cellular access over the same area. MQTT has proved to be an excellent communication protocol: the pub/sub mechanism makes it possible to automatically receive updates from nodes avoiding the polling procedure of HTTP, while the extremely small header of MQTT messages (which is even smaller than in AMQP) affects the traffic intensity in a minimal way. However, the use of MQTT required the implementation of an additional bridging module in Sentilo for the conversion of MQTT messages into HTTP messages (and viceversa for downlink traffic). Using this setting, the average delay experienced by a packet in uplink from the NetServer to the final application and the average uplink traffic intensity for a particular network setup have been evaluated. Both metrics have been

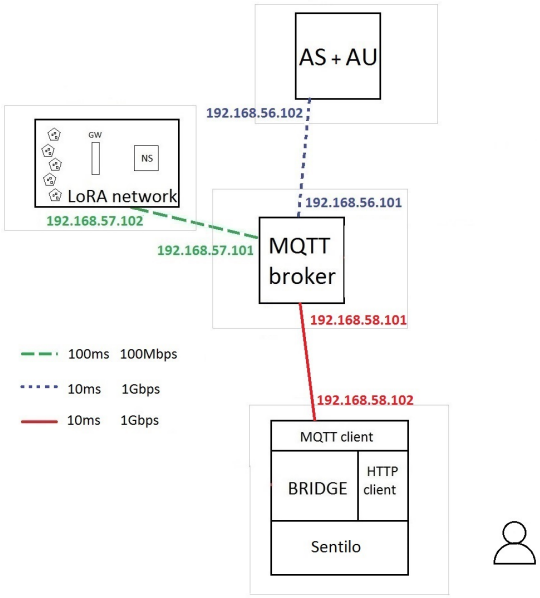


Figure 1.3: Simulation setup

calculated as the average values over more than 2.5 million packets. Downlink traffic, mainly constituted of sporadic commands targeting the sensors, has not been considered, as messages coming from the nodes are expected to be the predominant traffic in the considered scenario.

The adopted configuration is represented in Fig. 1.3 and sees the AppServer, the MQTT broker and Sentilo deployed in a cloud service, hence with very fast connections between the elements, while the NetServer is placed close to the LoRa gateways in order to minimize the latency in the LoRa network, and is connected to the MQTT broker by means of an Asymmetric Digital Subscriber Line (ADSL) link. The propagation times are around 10 ms for the connections in the cloud and 100 ms for the ADSL link. It is worth noting that the use of TLS for security reasons increases the traffic intensity because of the preliminary authentication phase and the additional message headers. Moreover, an MQTT QoS level of 2 (see Sec. 1.3) has been used in all exchanges, thus affecting both the traffic intensity - because of the acknowledgement messages - and the delay, since the broker needs to wait for the acknowledgement from the producer before sending the message to the subscribers of the involved topic. It turned out that, in addition to the LoRa traffic, each message generated by a peripheral node produces about 22 kByte of traffic resulting from the MQTT and HTTP messages exchanged in order to process the node observation and deliver it to the final user. About a quarter of such flow is comprised of HTTP traffic for dialoguing with Sentilo. The traffic associated to each link is reported in Tab. 1.4, where *brk* refers to the broker, *NetS* to the NetServer and *AppS* to the AppServer. The global traffic highly depends on the number of nodes employed and on the rate at which they transmit their messages.

TRAFFIC [KBYTE]	$NetS \rightarrow brk$	$AppS \rightarrow brk$	$brk \rightarrow Sentilo$
NO TLS	0.6	3.9	11.1
TLS	0.8	6.7	13.9

Table 1.4: Traffic intensity. Bytes exchanged at each link for one node observation

DELAY [ms]	$t_{propagation}$ (all)	t_{broker}	t_{AS}	$t_{Sentilo}$
NO TLS	130	204	29	120
TLS	130	203	28	120

Table 1.5: Transmission delays

The average delay experienced by a packet can be expressed as the sum of the various propagation times and the processing times, namely the transmission times from the NetServer to the broker, from the broker to the AppServer and vice versa, from the broker to Sentilo and from Sentilo to the final user, the time needed by the broker for handling the QoS level and identifying the interested subscribers, the processing times of the AppServer and Sentilo. Tab. 1.5 shows the average propagation and processing times from the simulations, resulting in an average transmission time of about 480 ms. Notice that this value strongly depends on the propagation times of the connection links, especially that of the ADSL, that represents the bottleneck of the transmission, whereas the processing times of the AppServer and Sentilo are in the order of just a few milliseconds.

Security in the Edge Technology layer is granted by the LoRa technology: data frames are encrypted with the scheme described in IEEE 802.15.4/2006 Annex B [47] using the AES algorithm with a key length of 128 bits. For each end device there is a specific application session key which is used by both the NetServer and the end device to encrypt and decrypt the payload field of application-specific data messages. Security on the MQTT connections is granted by enabling both authentication and TLS encapsulation: the MQTT broker provides username and password authentication and limits access to topics by using access control lists, whereas TLS ensures confidential transmission. Finally, Sentilo REST API is used over the secure HTTPS channel and Sentilo validates all HTTP requests according to the AAA architecture: Authentication, Authorization and Accounting. This means that the platform first identifies the petitioner of the request, then checks whether it is authorized to perform the requested action over the requested resource, and it finally traces the request by auditing the action and who performed it. Authentication is enabled by the mandatory use of an identification field in the HTTP headers, resulting in the so-called token-based authentication, which also allows for the authorization of a request by simply looking up the privileges associated to the involved token.

However, a weakness in Sentilo's security framework is that tokens, which are necessary to guarantee a secure and controlled access to resources, were stored

in the database in clear and, although the access to the databases requires authorization, it is always a good habit not to store passwords as they are, as a malicious attacker may find a way to access the database. There exist many hashing techniques commonly used for storing passwords, such as the MD5 algorithm and the family of the Secure Hash Algorithm (SHA). The chosen one is *bcrypt*, a cryptographic hash function (i.e., a one-way hash function, practically impossible to invert) which aims at being slow, or, more precisely, as slow as possible for the attacker while not being intolerably slow for the honest systems. It is derived from the Blowfish block cipher which uses look up tables to generate the hash, thus requiring a significant amount of memory space. This discourages attacks based on Graphics Processor Unit (GPU), which excels at doing simple manipulations on a large set of data, as it will become cumbersome to generate the hashes due to memory restrictions.

Another leak of Sentilo concerns the generation mechanism of tokens, based on the hashing of some knowable values, namely, a prefix retrievable in the source code, the name of the entity for which the token is being generated, and the creation time of this entity with a millisecond accuracy. Tokens are generated in two steps: firstly, the three mentioned elements are concatenated in a single string, and then the hash function of such string is computed according to the SHA-256 algorithm. The resulting token is a string made of 26 hexadecimal numbers. Basically, knowing the timestamp of the creation of a specific entity in Sentilo, it is possible to calculate the token associated to that entity. To prove this issue, a brute-force attack has been conducted against Sentilo to retrieve the token associated to a particular role, and this attack succeeded in a reasonable amount of time. The brute-force algorithm was single-threaded and did not rely on GPU acceleration, which is instead commonly used nowadays. Even with such non-optimized routine, just 4 ms are needed for a single attempt on an Intel® Core i7-2600 quad core processor. Thus, knowing the creation day of the entity for which the token has been generated, the token is retrievable in 4 days in the worst case.

The original token generation procedure is clearly unsafe and represents a big issues in the security of Sentilo. It is possible to considerably increase the security level by changing the token creation routine with a completely random token generator. If a brute-force attack is perpetrated by trying all possible strings of 26 hexadecimal characters, the average time for determining the correct token increases considerably. There are $16^{26} \simeq 2.03 \cdot 10^{31}$ possible combinations, but the number of tries before a success cannot be represented as a geometric random variable as the attempts, despite being independent of each other, are not identically distributed: after k wrong attempts, $16^{26} - k$ combinations remain. If we model the probability of needing at least k attempts before succeeding with a random variable, it is possible to estimate the lower and upper bounds of its cumulative distribution function by assuming identically distributed tries with the minimum and maximum probabilities, respectively. Considering that $p_{j+1} \leq p_j \forall j \geq 0$, the minimum probability of a try is that of the first attempt, i.e., $p_0 = p = 16^{-26}$, whereas the maximum probability of k tries is $p_k = 1/(1/p - k)$. For $k = 2 \cdot 10^{30}$ the probability is still low, about 0.1. With an average computing time of 4 ms per attempt, about 10^{20} years would be needed for trying $k = 2 \cdot 10^{30}$ combinations. It is evident that using a random token certainly improves security against brute-force attacks.

The described use case shows the extent of elements that must be considered

in an IoT system and the effort needed for integrating them in a solid, robust and secure system. Currently, there exist many solutions in the literature and since there is no well-established and widely-acknowledged best practice, developers should analyse the available strategies and protocols to identify those that best meet their requirements.

1.7 Structure of the thesis

As we have seen, there are significant opportunities to improve the state of the art, whether to enhance the performance of the system, or to solve actual issues in current systems, as in the security layer.

The communication system plays a major role in enabling CPSs. Message delivery must be reliable and low-latency for a wide variety of scenarios, from industrial systems with a presence of high electromagnetic interference to smart mobility in urban scenarios, characterized by a large amount of interference from neighbouring transmitting devices [15]. IoT communication technologies are widely used in CPSs to monitor, control, and act on physical entities. In Part I of this thesis we analyse the issues of such technologies. We focus, in particular, on the channel access of such systems, which becomes a critical part whenever the device density becomes very large. In particular, in Chap. 2, we study the communication patterns of some typical devices in a CPS, in order to create an accurate model. In Chap. 3 we investigate some traditional channel access technique, focusing on their behaviour in massive access scenarios. A theoretical model of one such techniques, namely ALOHA, is defined in Chap. 4 to find the optimal value of its parameters, then a generalization of the model is given to support networks using rate adaptation mechanisms. Chap. 5 proposes further improvements to ALOHA, specially tailored for M2M communications and with a focus on energy efficiency. These improvements, however, are not dependent on the type of data carried in the transmitted packets. In Chap. 6, some new channel access strategies are introduced, which leverage correlations found in data from sensor networks to allow for further reductions in energy consumption for this type of devices.

Part II is focused on how machine learning techniques can be used to optimize CPS services. Such service optimization can assume different forms. For example, as explained in Chap. 7, gathering information from the network and being able to accurately predict future traffic load helps in reducing the interference in the network and schedule network resources. The latter, in particular, can assume the form of admission control and resource management mechanisms, which can be more impactful if they are aware of the type of content transiting on the network. In this case, in fact, they can balance the network use with the QoS provided in order to guarantee a minimum service level, as shown in Chap. 8 through Chap. 10 for video streaming applications. These applications are common amongst non-constrained devices, like smartphones and augmented/virtual reality headsets, which form an integral part of the IoT ecosystem. The communication subsystem, however, does not provide only the delivery of messages between devices of a network. Positioning services, investigated in Chap. 11, also use communication techniques, often specifically designed for this purpose and enhanced by a machine learning approach. The ability to precisely know the location of its components is a fundamental require-

ment for a CPS, since its devices are often mobile and have to move in a precisely defined path to be effective and not cause harm to the environment [15].

The focus of Part III is on the security of IoT systems, which is critical due to their pervasivity, scale, and their use in critical infrastructures. CPSs are particularly sensitive to these issues, since they deal with the physical world and can harm people or animals, or damage things, if they are used for a malicious intent. Therefore, security issues may transform in safety issues and, consequently, hamper the adoption of such systems. Such issues are investigated in Chap. 12, while a proposal for an enhanced authentication mechanism is presented in Chap. 13.

Finally, the work is concluded by Chap. 14, where the main findings are summarized, and a discussion on the way forward is given.

Part I

Networks in cyber-physical systems

Chapter 2

MTC source models

Machine-Type Communication (MTC) is one of the biggest factor dictating the design of 5G networks. The challenge, however, is that the traffic generated in MTC is very disparate both in volume and shape [48]. For example, there can be water level measuring sensors generating few bytes of data every hour, while, on the other side, there are surveillance cameras flooding the network with massive amount of data. Likewise, there could be sensors *uploading* data, while, on the other hand, there could be applications which require *downloading* data, such as weather maps for farmers. Secondly, groups of machine type sources, unlike most human generated traffic, may initiate transactions that are correlated both in space and time [49]. So, modeling this diverse canvas of machine type devices is essential to understand the performance of both current and future networks and for accurate network dimensioning.

2.1 Related Work

Traffic models can be broadly divided into *aggregated* and *source* models [49,50]. Aggregated models capture the traffic properties of a group of users over a cell, area, or entire network. Source models, instead, capture traffic behavior of an individual user, referred to as a source. Aggregated models provide a simple and efficient way to analyse the network behavior as a whole but, since all devices have the same parameters, it can not accurately reflect the small-scale behavior of the network, in particular if devices have very different traffic patterns. This is not an issue with source models which have, however, a higher computational cost due to the larger number of parameters.

A number of M2M traffic models have been proposed in the literature. The Third Generation Partnership Project (3GPP) designed two aggregated models, one for coordinated traffic and one for uncoordinated arrivals [51]. They are based on, respectively, a uniform and Beta distribution of arrivals. Parameters for such distributions have been extracted in [51, 52].

Most, if not all, of the source model approaches are based on state-based modeling [48, 49] as quite often traffic sources have their inherit natural states, depending on different processes in the communication, e.g., waiting states, thinking times and states where sources are active. A well known two-state source model is the ON/OFF model [48], where the source changes between

ON state (where packets are sent) and OFF state (where no packets are sent). Traffic modeling for M2M last mile wireless access is proposed in [53] where the analysis is limited to event driven and fixed scheduling traffic sources. A coupled Markovian arrival process is proposed in [54] for MTC in an automotive industry and the weakness of traditional simple models with exponential inter-arrival time distribution is highlighted. In [55], each device can be either in the *regular* or *alarm* state, according to a two-state Markov chain. Traffic is generated according to a Bernoulli process with a rate specific to the device state. Further state-of-the-art source models are presented in the following section, where we analyse their ability to accurately represent different real world data sets.

2.2 Comparison of M2M traffic models against real world data sets

Considering the large variety of Machine-Type Devices (MTDs), the M2M traffic patterns can then be very disparate and finding a comprehensive traffic model that can be used for protocol design and performance assessment is not an easy task. As a result, a number of different M2M traffic models have been proposed in the literature in the effort of balancing the model complexity and the accuracy/realism of the generated synthetic traffic traces. However, it is not yet clear what are the actual characteristics of real M2M traffic sources and, consequently, which model(s) might better represent the most common M2M traffic patterns.

This section provides two main contributions:

(i) traffic patterns generated by real MTDs used in three relevant and representative M2M services, namely logistic, parking, and metering, are studied in order to identify the main components and the most relevant features;

(ii) by using such real-world data traces, a comparison of the capabilities of three well-known M2M traffic source models to produce realistic M2M traffic traces is performed, identifying the strengths and weaknesses of the different models, and proposing some possible improvements.

2.2.1 Selected M2M traffic models

In the following, three M2M source models that offer a balance between complexity and accuracy are described: the Source Semi-Markov Model (SSMM), the Coupled Markov Modulated Poisson Process (CMMPP), and the Coupled Markovian Arrival Process (CMAP) models. All of them make use of Markov chains to model the state flow of the MTDs in a stochastic manner. However, CMAP considers only two states, corresponding to periodic and event-based traffic patterns, respectively. Instead, the SSMM model entails four states, associated to different traffic generation events, included the case of transmission bursts corresponding to the exchange of long messages. The CMMPP model, finally, can include an arbitrary number of states, which gives additional flexibility but requires a larger computational cost and makes it more prone to overfitting. In addition, CMMPP entails time-varying transition probabilities, thus enabling to model non stationary sources.

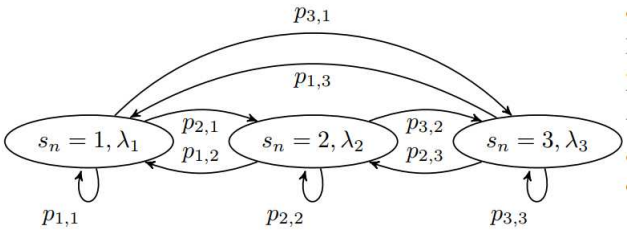


Figure 2.1: CMMPP model with three states [56].

These models are described in greater detail in the following, while in the next sections they will be tested against a real dataset with different types of M2M sources in order to identify their strengths and weaknesses, and possible improvements.

Source Semi-Markov Model (SSMM)

The SSMM model [49] targets specifically event-driven communication, along with the traditional periodic reporting messages. Each device is modeled using a Markov chain consisting of three states:

- *Periodic Update* (PU), when the device is periodically transmitting status reports to a central unit. A typical example of PU message is a smart meter reading.
- *Event Driven* (ED), when data transmission is triggered by a certain event.
- *Payload Exchange* (PE), when a larger amount of data needs to be transmitted between the sensing devices and the server after an event, which may correspond to any of the previous states (PU or ED).

Additionally, an OFF state is introduced, corresponding to the period when no data needs to be transmitted and corresponds, usually, to a deep sleep state of the device to save energy.

The transition probability to the same state is set to zero. Sojourn times and message lengths are generated according to probability distributions that are independent between states and potentially different for each of them.

Coupled Markov Modulated Poisson Process (CMMPP)

The CMMPP model is able to capture the time and space correlation of M2M traffic sources [56]. The packet generation events are modelled by means of a Markov modulated Poisson process, that is to say, a Poisson process whose arrival rate depends on the state of a Markov chain that, in turn, is associated to a certain working state of the MTD. For example, a MTD that can work in three states (as sleep, normal, and alert), may be represented by the CMMPP in Fig. 2.1, where $\{s_i; i = 0, 1, 2\}$ represent the three working states, λ_i is the packet generation rate when the MTD is in state s_i, and $p_{i,j}$ is the transition probability from state s_i to state s_j in one step.

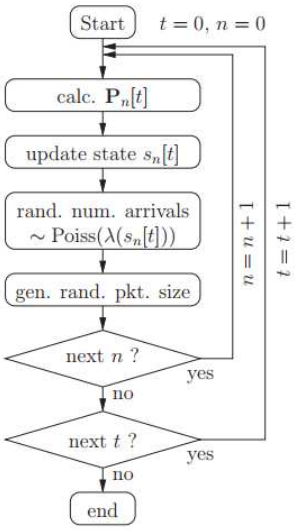


Figure 2.2: Message generation process according to the CMMPP model [56].

To introduce time and space correlation among the packet generation processes of different MTDs, the Markov transition matrix of the generic n device can be expressed as

$$P_n(t) = \delta_n \cdot \theta[t] \cdot P_C + (1 - \delta_n \cdot \theta[t]) \cdot P_U \tag{2.1}$$

where P_C and P_U are globally-defined transition probability matrices for perfectly coordinated and uncoordinated devices, respectively, the constant $\delta_n \in [0, 1]$ captures the degree of coordination of the nth MTD with respect to the other MTDs, while $\theta(t) \in [0, 1]$ is a common background process that models the time correlation among the sources. The generation of arrivals according to the CMMPP model is outlined in Fig. 2.2. This model does not explicitly include the generation of message lengths, which, however, can be added to the model following the same principle.

Coupled Markovian Arrival Process (CMAP)

CMAP has been inspired by the previous model, but some aspects have been simplified and others have been generalized, in order to increase the model flexibility [54]. More specifically, CMAP assumes that the MTDs can only be in two states, *normal* or *event-driven*. Each state is associated to different probability distribution functions for the generation of the lengths and the interarrival times of the messages. MTDs remain in the normal state for a random time interval, after which they enter the event-driven state, where they sojourn for a fixed amount of time and then return back to the normal state.

2.2.2 Analysis of real world M2M traffic sources

In this section we analyse the dataset obtained from one the biggest M2M operators in Europe (see [4]). The dataset contains traffic traces generated by three

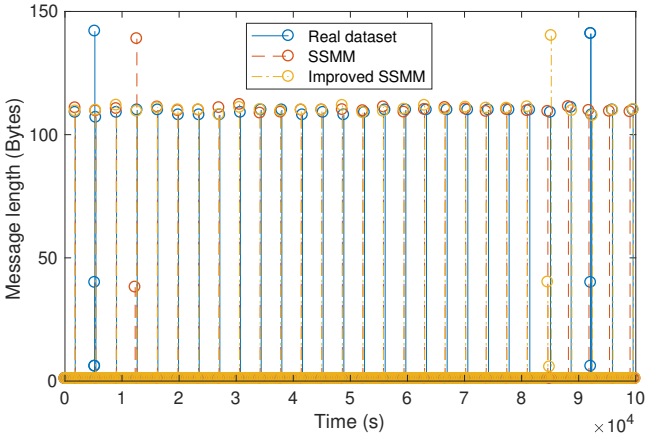


Figure 2.3: Tracking devices: time series from the dataset and realizations from the SSMM (original and modified) models.

different M2M services, namely logistic (assets tracking), smart parking, and remote electricity metering. For the first two services, we consider the packet generated by each single MTD, that is to say, by each tracking device for the logistic service, and each parking sensor for the smart parking. Instead, for the electricity metering service, we only have the traffic traces generated by a few concentrators, which are MTDs that collect and forward the readings provided by a certain number of peripheral meters. The considered services and traffic sources generate rather diverse traffic patterns. However, in all the considered cases, the downlink traffic was mostly negligible, consisting of only a few configuration messages sent during the service setup, and short acknowledgement packets with basically no impact on the network performance. Therefore, we focus on the uplink traffic only. In the following, a more in-depth analysis of the traffic generated by the three services is provided.

Tracking devices

Fig. 2.3 reports an example of the uplink traffic pattern generated by one tracking device (blue pin with solid stem). We can identify five main message types composing the time series:

1. a very large number of very short packets (1 Byte), with approximately constant inter-packet time, which are likely keep-alive messages sent during inactivity periods, when the tracked object remains still;

2. a few bursts of three messages of 5, 40, and 140 Bytes, respectively, which are always transmitted together, probably triggered by the occurrence of a certain event;

3. and a large number of long messages of about 110 Bytes, with the characteristic pattern of periodic (location) updates.

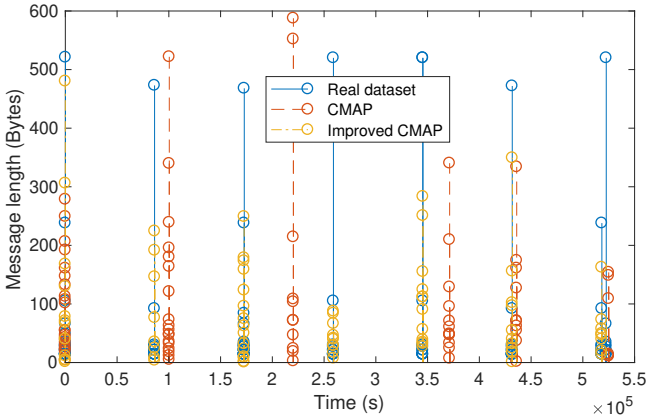


Figure 2.4: Electricity meters concentrator: time series from the dataset and realizations from the CMAP (original and modified) models.

Electricity meters concentrator

In Fig. 2.4 part of the time series generated by the electricity meter concentrators (blue pin with solid stem) is reported. We can observe that transmissions occur in bursts spaced apart by almost constant time intervals of about 24 hours. This pattern is coherent with a concentrator that gathers data from neighboring slave devices and periodically sends them in bursts to a common gateway. The messages in a burst, however, have variable lengths.

Parking sensor

The third MTD considered is a parking sensor. As we can see in Fig. 2.5, the traffic from this device can be divided in three categories:

1. A high-frequency periodic transmission of short messages (around 800 Bytes) in bursts, which are likely keep-alive or state-update messages;
2. A less regular traffic of large messages (from 1000 to 6000 Bytes), which may be generated by some event (e.g., expiration of the parking time, or occupation of the parking slot);
3. An almost periodic patter of very large messages (around 16000 Bytes), with very long periodicity (approximately 10.5 hours), which may be periodic status reports.

In the following section, we will try to replicate these empirical traces using the three models described in Sec. 2.2.1, in order to see how accurately such models can represent the different M2M traffic sources.

2.2.3 Comparison with M2M traffic models

In this section, we first attempt to set the parameters of the models described in Sec. 2.2.1 in order to reproduce the real traffic patterns presented in the previous section. Then, some possible adjustments of the models to improve

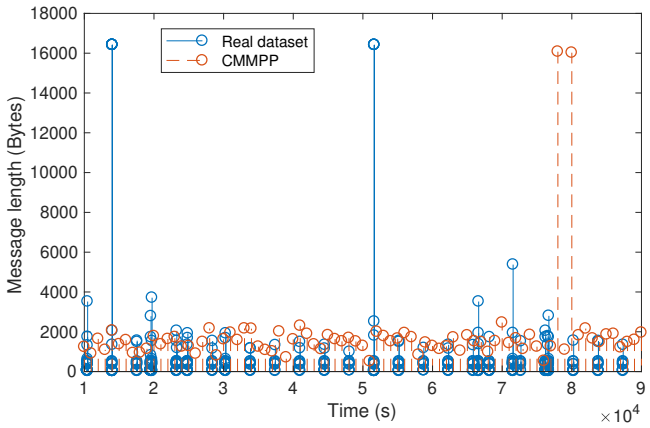


Figure 2.5: Parking sensors: time series from the dataset and realization from the CMMPP model.

the matching between synthetic and real traffic patterns are proposed. The different types of traffic sources separately are considered separately.

Modeling tracking devices

The traffic pattern shown in Fig. 2.3 can be well modeled by using an SSMM, with the following transition probability matrix:

$$P = \begin{array}{c} \\ OFF \\ PU \\ ED \\ PE \end{array} \begin{array}{c} \begin{array}{cccc} OFF & PU & ED & PE \end{array} \\ \begin{pmatrix} 0 & 0.98 & 0.02 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The other parameters are reported in Tab. 2.1. The resulting trace is reported in Fig. 2.3 (red pin with dashed stem). Strictly speaking, this model can not fully replicate the event-based transmission, due to the very deterministic behaviour of the event-based pattern for this device. In fact, the sequence of the payload lengths for such messages are always {40, 5, 140} Bytes, which can not be replicated by any combination of parameters for the ED and PE states. This issue can be easily solved by splitting PE in two states, PE1 and PE2, where the device sends the 5 and 140 Bytes messages, respectively. Therefore, the transition probability matrix becomes as follows.

$$P = \begin{array}{c} \\ OFF \\ PU \\ ED \\ PE1 \\ PE2 \end{array} \begin{array}{c} \begin{array}{ccccc} OFF & PU & ED & PE1 & PE2 \end{array} \\ \begin{pmatrix} 0 & 0.98 & 0.02 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

State	Distribution	Mean	Standard Deviation
PU	Rounded Gaussian	110 Bytes	1 Byte
ED	Rounded Gaussian	40 Bytes	1 Byte
PE	Rounded Gaussian	140 Bytes	1 Byte

Table 2.1: SSMM parameters for tracking devices

Parameter	Distribution	Mean
PU interarrival time	Exponential	400 s
PU message length	Exponential	110 Bytes
ED interarrival time	Exponential	7 s
ED message length	Exponential	140 Bytes
ED sojourn time	Deterministic	5 s

Table 2.2: CMAP parameters for tracking devices

State PE2 has the same characteristics as state PE in Tab. 2.1, while PE1 has a rounded Gaussian message length distribution with mean 6 Bytes and standard deviation 1 Byte.

With this change, the model produces a pattern more similar to the real one (yellow pin with dash-dotted stem in Fig. 2.3). To quantify this improvement we resort to a well established dissimilarity measure between time-series, that is the Kullback-Leibler (KL) divergence over the empirical distribution of message lengths [57]. The original SSMM has a KL divergence of 0.016, while the modified model reaches a value of 0.0032, showing a notable improvement over the original one. The Mean Squared Error (MSE) for the two models, calculated by linearly interpolating the modeled time-series, is 1529 Bytes^2 for the original SSMM, which was reduced to 1229 Bytes^2 when using the improved model, confirming the accuracy improvement.

We can also try to fit the same time series with the CMAP source model. In this case, the optimal fit of the model to the dataset is obtained with the parameters in Tab. 2.2.

The comparison of the actual and synthetic traces is reported in Fig. 2.6. We can see that the model performs poorly, particularly for the event-based traffic, giving a KL divergence for message length distribution of 5.92, three orders of magnitude larger than that obtained with the modified SSMM. This is because CMAP entails only one event-related state, causing the same problem observed with the original SSMM. By adding two more states, we can make this model perform as good as the modified SSMM, but with a higher complexity. Furthermore, the modeling of the interarrival times is also poor, as shown in Fig. 2.7. Very similar considerations can be drawn for CMMPP. Therefore, we can conclude that SSMM is the preferable model for this type of very deterministic and periodic traffic sources.

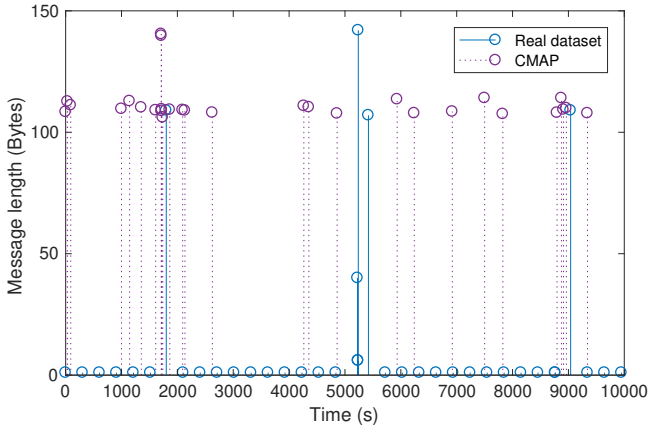


Figure 2.6: Tracking devices: time series from the dataset and realizations from the CMAP model.

Parameter	Distribution	Mean
Sojourn time in ED state	Deterministic	20 s
ED message interarrival time	Exponential	1 s
ED message length	Exponential	100 Bytes

Table 2.3: CMAP model parameters

Modeling electricity meters concentrators

The pattern reported in Fig. 2.4 exhibits constant time intervals between transmission events. Furthermore, the distribution of the message lengths is well captured by a two-state model. Therefore, the CMAP model seems to be a good candidate for this type of source. Considering that the device never sends messages between two consecutive burst sessions, the generate rate in the normal state of the model is set to zero. Therefore, the transition probability matrix P is as follows, with the first state being the normal one.

$$P = \begin{pmatrix} 0.92 & 0.08 \\ 1 & 0 \end{pmatrix}$$

The other model parameters are reported in Tab. 2.3.

A time series realization obtained with the model just described is reported in Fig. 2.4 (red pin, dashed stem). By comparing it with the real traffic pattern, we can see that it generates messages with realistic values, giving a KL divergence for message length of 0.2574, but there are issues with the time between two burst sessions. In particular, the burst from the real device is completely periodic, being the inter-burst time fixed. The model is not able to capture this aspect since the state transitions in the model are stochastic and depend on the transition probability matrix of the Markov chain. We can therefore investigate the use of a simple timer, as the one used in the ED state, to com-

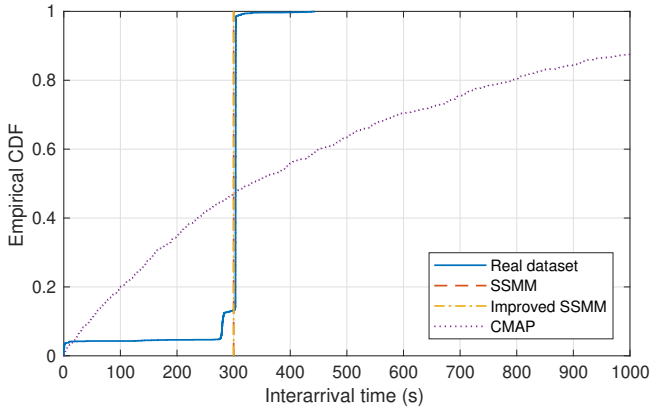


Figure 2.7: Empirical cumulative distribution function (ECDF) of interarrival times for tracking devices.

mand the transition from the regular to the ED state. The model defined in this way degenerates in a semi-Markov chain where the probability of staying in the same state is zero and the sojourn times are constants. The only stochastic aspect, then, lies in the message length of the generated traffic in the ED state. By setting the sojourn time to 86360 s, we get the model realization depicted in Fig. 2.4 (yellow pin, dash-dotted stem). Results show that this model is able to represent the time series with a high degree of accuracy. Also the arrival instants inside a burst are well represented, as depicted in the empirical Cumulative Distribution Function (CDF) for the message interarrival time in Fig. 2.8.

Modeling parking sensors

The parking sensors exhibit the most complex traffic pattern, which calls for the higher flexibility of the CMMPP model. We can define two states, *regular* and *alarm*. The short and medium sized messages are generated in the regular state, while the larger messages are generated when the system is in the alarm state. As already mentioned, the CMMPP model offers a large degree of freedom, with particular reference to:

- The transition probability matrices P_C and P_U of coordinated and uncoordinated scenarios, respectively;
- The background process Θ and the factor δ_n;
- The message length in the regular and event states.

Here the performance of only one device is analysed, therefore we only define one δ_n value. The best fit of the CMMPP model to the real data is obtained using a Beta(3,4) distribution to generate $\theta(t)$ values, a Gaussian distribution for the message lengths (with, respectively, mean of 1500 and 1600 Bytes, and standard deviation of 400 and 100 Bytes), and setting $\delta_n = 0.05$ and the following

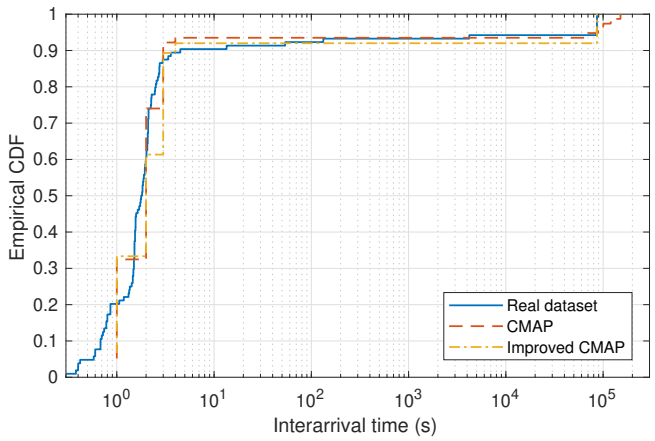


Figure 2.8: Electricity meters: time series from the dataset and realizations from the CMAP (original and modified) models.

transition probability matrices:

$$P_U = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad P_C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

We can note, from Fig. 2.5, that the approximately periodic nature of large message traffic is not captured in the model, due to the probabilistic transitions between the regular and event states. Furthermore, the high-frequency short message transmission is also not well represented. Better results could be obtained by combining this model with a semi-Markov approach, similar to that used for the electricity meter, to model the more deterministic features of the pattern, but this is left as future work.

2.3 State Modulated Traffic Models for Machine Type Communications

From the previous analysis it emerged that, in general, M2M traffic patterns have strong deterministic components that, in some cases, are overlapped with asynchronous event-driven components. On the other hand, the three source models considered in this study are all based on a stochastic framework that exhibits some limits in capturing deterministic patterns.

The model proposed in this section makes it possible to represent a wider range of M2M sources, each with very different traffic characteristics, ranging from highly regular to bursty types of traffic sources. To this end, a modulated Renewal Processes for packet level traffic modeling of single M2M type of sources is applied. A modulated Renewal Process is an extension of ordinary Renewal Processes (RPs) [58] where the distribution between arrivals may change according to the state of a modulating Markov Chain. To make the description manageable it is assumed that the number of arrivals in a state, with a particular inter-arrival time distribution, can be modeled by an integer random

variable with a particular distribution that is specific for that particular state. Hence, the total time spent in a particular state (for the modulated RP) is the sum of the inter-arrival times between the arrivals that occur in that particular state. After the total time in a state has expired, the source will move to another state according to the modulating Markov Chain, and start a new RP arrival sequence with its specific distributions (both for the inter-arrival time and the number of arrivals). The modulated RP is therefore a generalization of legacy RPs where the statistical distribution of the arrival process changes depending on a state variable. In addition, we may also specify the packet size distribution, which again may depend on the state variable. This type of parameterized model makes it possible to represent a wide array of M2M sources with varying traffic characteristics.

Modeling based on modulated RP has pros and cons. Sometimes, when the source type and its traffic generation pattern are well known, e.g., alternating between some known deterministic pattern, the modulated RP will not necessary give accurate description. However, when the patterns are more random, the traffic generation will fit well with a general stochastic modeling approach. Some of the appealing properties of the modulated RP traffic model are that they are easy to understand and simulate, and a very broad range of M2M source types can be described with such models, both for regular and bursty traffic. However, there are some drawbacks attached to this modeling approach such as the fact that the modulated RP models involve a large number of parameters, and it is therefore not easy to choose the *best* model and estimate the parameters based on recorded traces. Secondly, it is difficult to model aggregated traffic streams and analytical models based on aggregates are difficult to analyze.

2.3.1 Packet level model

The general packet level model is briefly described here. The idea behind this type of arrival process is to generalize the legacy renewal model, where we allow the distribution between arrivals to change according to the state of a modulating Markov Chain. Further, the sojourn time in a state is determined by the number of arrivals in that particular state, described by an integer random variable, and by the inter-arrival times.[1]

Description of the Modulated RP

Let us start by giving a formal description of the modulated RP. The modulated RP is described by the following stochastic variables (see Fig. 2.9):

- The (modulation) state variable I_k at k'th jump is the state of a Markov Chain, with state space $\Omega = \{1, 2, \ldots, N\}$ and transition probability matrix $Q = (q_{ij})$, with $i, j \in \Omega$, and $q_{ii} = 0$ for all $i \in \Omega$.
- If the modulating Markov Chain has performed k transitions up to time t, then the state of the system at a generic time t is $J_t = I_k$;
- When the modulation state variable I_k is in state $i \in \Omega$, i.e. $I_k = i$, then

[1] Another possibility is to model the time spent in a state as a separate random variable. This approach, however, would make the model more complicate and it is not considered in this study.

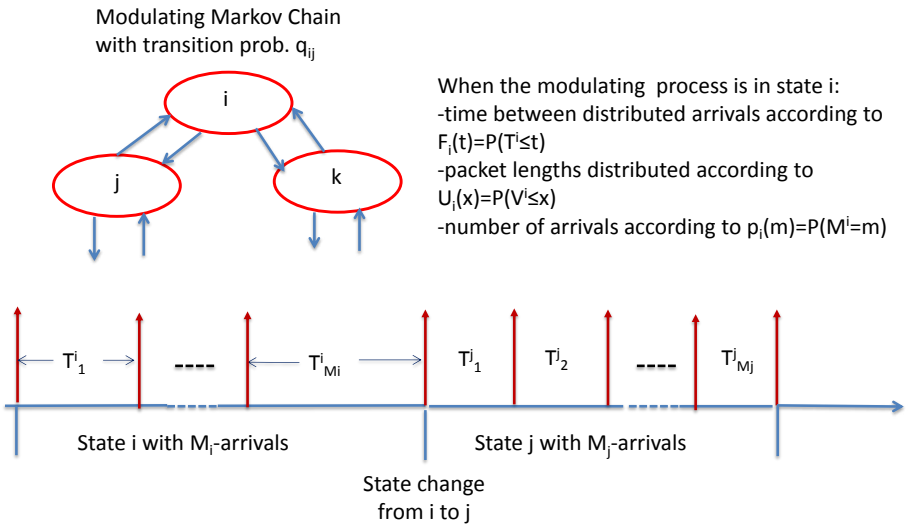


Figure 2.9: Traffic generation by the N state Modulated Renewal Process.

- packet inter-arrival times $\{T_k^i\}$ are independent and identically distributed (iid), with CDF equal to $G_i(t) = P(T^i \leq t)$, where T^i is the canonical inter-arrival time random variable;
- the corresponding packet lengths $\{V_k^i\}$ are also iid, with CDF $U_i(x) = P(V^i \leq x)$, where V^i is the canonical packet-size random variable;
- the total number M^i of packet arrivals in state $i \in \Omega$ is discrete random variable with probability mass distribution $p_i(m) = P(M^i = m)$, for $m = 1, 2, \ldots$.

For the analysis, we also have to define the probability density functions (pdfs) for the inter-arrival times, $g_i(t) = G_i'(t)$, and for the packet size in each state, $u_i(x) = U_i'(x)$. The associated Laplace Stieltjes Transforms (LSTs) are given by $f_i(s) = \int_{t=0}^{\infty} e^{-st} dG_i(t)$ and $v_i(y) = \int_{x=0}^{\infty} e^{-yx} dU_i(x)$, respectively, while the moment Generating Function (GF) of M^i is given by $P_i(z) = \sum_{m=1}^{\infty} z^m p_i(m)$. Furthermore, we denote $t_i^{(k)} = E[(T^i)^k]$, $v_i^{(k)} = E[(V^i)^k]$ and $m_i^{(k)} = E[(M^i)^k]$ as the k'th moment of inter-arrival times, packet lengths and total number of packet arrivals in state $i \in \Omega$, respectively. For ease of writing, the mean values are represented as $t_i = t_i^{(1)}$, $v_i = v_i^{(1)}$ and $m_i = m_i^{(1)}$, respectively.

Now, the sojourn time in state i where $i \in \Omega$ is readily found as the (stochastic) sum

$$S^i = T_1^i + T_2^i + \ldots + T_{M^i}^i \tag{2.2}$$

where all the T_k^i are iid and distributed as T^i.

Given the mutual independence of the involved variables, the transform of the joint distribution of the sojourn time and the number of arrivals in state i is given by $w_i(s,z) = E[e^{-sS^i} z^{M^i}] = E[E[e^{-sS^i} z^{M^i} \mid M^i]] = E[(zf_i(s))^{M^i}]$, which then yields the following functional equation

$$w_i(s,z) = P_i(zf_i(s))\,. \tag{2.3}$$

The mean and variance of S^i are hence

$$E[S^i] = m_i t_i \text{ and } \sigma^2_{S^i} = \sigma^2_{M^i} t_i^2 + m_i \sigma^2_{T^i} \tag{2.4}$$

where σ^2_X denotes the variance of a random variable X.

To be able to follow the process over several state changes we define the matrix $R(s,z) = (r_{ij}(s,z))$ where we also include the probability of the next state in the expression (2.3)

$$r_{ij}(s,z) = E[e^{-sS^i} z^{M^i} \mathbf{1}_{\{I_{k+1}=j\}} \mid I_k = i] = P_i(z f_i(s)) q_{ij} \tag{2.5}$$

or

$$R(s,z) = diag(P_i(z f_i(s)))Q\,, \tag{2.6}$$

where $\mathbf{1}_{\{I_{k+1}=j\}}$ is 1 if the next state of the modulating Markov Chain is $j \in \Omega$, and zero otherwise, while $diag(\lambda_i)$ is the diagonal matrix with (diagonal) elements λ_i, $i = 1, \ldots, N$. Hence, the process defined by the pair $\{S, I\}$ constitutes a ordinary Markov RP with LST of the generator matrix $R(s,1) = (r_{ij}(s,1)) = diag(P(f_i(s))) \cdot Q$.

Let $X^i = V_1^i + V_2^i + \ldots + V_{M^i}^i$ denote the total volume of bits arrived while being in state i. Then, the envelope process $R(s,1)$ defines the corresponding fluid model where the rate r_i in state i is taken to be the ratio between the mean volume of state i, and the mean sojourn time in that state, i.e.,

$$r_i = \frac{E[X^i]}{E[S^i]} = \frac{v_i}{t_i}\,. \tag{2.7}$$

If the number of arrivals in state i, M^i, is geometrically distributed, i.e., $p_i(m) = (1-p_i)p_i^{m-1}$ for $m = 1, 2, \ldots$, then the modulated RP will be an ordinary Markov RP with generator Matrix

$$F(t) = diag(G_i(t))\hat{Q}\,, \tag{2.8}$$

where $\hat{Q} = (\hat{q}_{ij})$ is the transition matrix for the associated Markov Chain given by

$$\hat{q}_{ij} = \begin{cases} p_i & \text{for } j = i\,; \\ (1-p_i)q_{ij} & \text{for } j \neq i\,. \end{cases} \tag{2.9}$$

Due to the memoryless property of the geometrical distribution, it is sufficient for this particular case to consider the process at packet arrival instants, where p_i represents the probability that there is no state change between two succeeding arrivals, while $1 - p_i$ gives the probability that the state changes, and q_{ij} is the conditional probability that the next state is j, given that state i is left.

If we also assume that the inter-arrival times in state i have negative exponential distribution, i.e., $G_i(t) = 1 - e^{-\lambda_i t}$, then the arrival process is an ordinary Markov Process (MP). In this case, the envelope process defined by the LST of the generator matrix (2.8) will be a MP, since the time spent in a particular state will have a negative exponential distribution. For Markov RP, several results are available in the literature, in particular refer to [59] and [60] for an update of some interesting results.

Steady state distributions

In the analysis the steady state distribution of the state variable I_k will play an important role when considering a modulated RP. At jump instances the state variable is governed by the transition matrix Q and if we take $\pi_i = \lim_{k\to\infty} P(I_k = i)$ to be the steady state probabilities for $i = 1, \ldots, N$, and let $\Pi = (\pi_i)$ be the corresponding row vector, the steady state distribution is determined by the following equations

$$\Pi = \Pi \cdot Q \quad \text{and} \quad \Pi \cdot e = 1\,, \tag{2.10}$$

where e is a column vector with ones.

If we consider the state variable at an arbitrary time, say J_t at time t, then the corresponding steady state distribution $p_i = \lim_{t\to\infty} P(J_t = i)$ is found by scaling the probabilities π_i (at state jump instances) by the mean length of a period between jumps, $E[S^i] = m_i t_i$ giving the following steady state distribution

$$p_i = \frac{m_i t_i}{C}\pi_i \quad \text{with} \quad C = \sum_{i=1}^{N} \pi_i m_i t_i = \Pi \cdot diag(m_i t_i) \cdot e\,. \tag{2.11}$$

We also have to define the marginal distributions of time between packets (independent of the states). Over a long period the number of arrivals that see the process in state i with CDF $G_i(t)$ is proportional to $\pi_i m_i$ while the total number of arrivals in the same period is proportional to $\sum_{i=1}^{N} \pi_i m_i$. Hence, the probability that we have an arrival in state i is given by $r_i = \frac{\pi_i m_i}{\sum_{i=1}^{N} \pi_i m_i}$. If we take T_{avg} as the average random variable then the CDF $G_{avg}(t) = P(T_{avg} \le t)$ will be given by

$$G_{avg}(t) = \frac{\sum_{i=1}^{N} \pi_i m_i G_i(t)}{\sum_{i=1}^{N} \pi_i m_i}\,. \tag{2.12}$$

From (2.12) we find the moments of T_{avg} to be

$$E[(T_{avg})^k] = \frac{\sum_{i=1}^{N} \pi_i m_i t_i^{(k)}}{\sum_{i=1}^{N} \pi_i m_i}\,. \tag{2.13}$$

Similar for the packet length, if we take V_{avg} as the *average* packet length distribution, the CDF $U_{avg}(t) = P(V_{avg} \le t)$ is given by

$$U_{avg}(t) = \frac{\sum_{i=1}^{N} \pi_i m_i U_i(t)}{\sum_{i=1}^{N} \pi_i m_i}\,. \tag{2.14}$$

As above the moments of U_{avg} are found from (2.14) as

$$E[(V_{avg})^k] = \frac{\sum_{i=1}^{N} \pi_i m_i v_i^{(k)}}{\sum_{i=1}^{N} \pi_i m_i}\,. \tag{2.15}$$

In fact it is also possible to derive (2.12) with somehow different arguments where $G_{avg}(t)$ should be a weighted sum of the $G_i(t)$'s, $(G_{avg}(t) = \sum_{i=1}^{N} a_i G_i(t))$ and we need to determine the a_i's. The main idea is that two succeeding arrivals should see the same average distribution. By considering the process over a long

time the probability that an arrival in state i is the last before state change is $\frac{1}{m_i}$ and that it is not is $1 - \frac{1}{m_i}$. If an arrival is the last before state change the next state is j by probability q_{ij} (with arrival distribution $G_j(t)$). For two succeeding distributions to be equal we therefore must have $G_{avg}(t) = \sum_{i=1}^{N} a_i G_i(t) = \sum_{i=1}^{N}(1 - \frac{1}{m_i})a_i G_i(t) + \sum_{i=1}^{N} \frac{1}{m_i} a_i \sum_{j=1}^{N} q_{ij} G_j(t)$. Rearranging this requires $\sum_{j=1}^{N}(-\frac{a_j}{m_j} + \sum_{i=1}^{N} \frac{a_i}{m_i} q_{ij}) G_j(t) = 0$ or $-\frac{a_j}{m_j} + \sum_{i=1}^{N} \frac{a_i}{m_i} q_{ij} = 0$ for $j = 1, \cdots, N$. Hence, $a_j = \alpha \pi_j m_j$ for some constant α, and therefore $a_j = r_j$ for $j = 1, \cdots, N$.

The equilibrium Modulated RP

If we consider a modulated RP which has reached equilibrium and start to observe the process at a certain time, say $t = 0$ and observe the process from there, then the state variable J_t will be in steady state. The time to the next packet arrival, and the number of arrivals to the next state change, denoted by $\tilde{T^i}$ and $\tilde{M^i}$, will be distributed as the residuals of the random variables of T^i and M^i, i.e.,

$$\tilde{G_i}(t) = P(\tilde{T^i} \leq t) = \frac{1}{t_i} \int_0^t (1 - G_i(\tau)) d\tau \,; \tag{2.16}$$

$$\tilde{p_i}(m) = P(\tilde{M^i} = m) = \frac{1}{m_i} \sum_{j=m+1}^{\infty} p_i(j) \,. \tag{2.17}$$

Furthermore, the LST and GF of the residual distributions may be found from (2.16) and (2.17), leading to $\tilde{f_i}(s) = \frac{1 - f_i(s)}{s t_i}$ and $\tilde{P_i}(z) = \frac{1 - P_i(z)}{(1-z) m_i}$, respectively. Observe that the residual distribution of the number of arrivals in a particular state is the probability of having exactly m arrivals before a state changes, when picking a certain arrival interval at random. For example, the residual sojourn time in state i, denoted as $\tilde{S^i}$, can be expressed as the sum of the residual inter-arrival time for the first packet, which is distributed as $\tilde{T^i}$, and then inter-arrival times of the remaining $\tilde{M^i}$ packets, which are distributed according to T^i, i.e.:

$$\tilde{S^i} = \tilde{T^i} + T_1^i + T_2^i + \ldots + T_{\tilde{M^i}}^i \tag{2.18}$$

where the all the T_k^i, all are independent and distributed according to T^i.

The joint transform of the time duration $\tilde{S_i}$ and the number of arrivals $1 + \tilde{M^i}$ until the next state change is given as

$$w_i(s, z) = z \tilde{f_i}(s) \tilde{P_i}[(z f_i(s))]$$

which by applying the relations for $\tilde{f_i}(s)$ and $\tilde{P_i}(z)$ gives

$$\tilde{w_i}(s, z) = \frac{z}{t_i m_i} \hat{w_i}(s, z) \tag{2.19}$$

where

$$\hat{w_i}(s, z) = \frac{(1 - f_i(s))(1 - P_i(z f_i(s)))}{s(1 - z f_i(s))} \tag{2.20}$$

The LST of the residual sojourn time in a given state i is now easily found from (2.19) by taking $z = 1$, giving $\tilde{w_i}(s, 1) = \frac{1 - P_i(f_i(s))}{s m_i t_i}$ as expected. To be able to follow the process over the first state changes we also define the matrix

$\tilde{R}(s,z) = (\tilde{r}_{ij}(s,z))$ where we include the probability of the next state in the expression (2.19)

$$\tilde{r}_{ij}(s,z) = E[e^{-s\tilde{S}} z^{1+\tilde{M}} \mathbf{1}_{\{I_1=j\}} \mid J_0 = i] = \tilde{w}_i(s,z) q_{ij} \tag{2.21}$$

or

$$\tilde{R}(s,z) = diag(\tilde{w}_i(s,z))Q = z\,diag(\frac{1}{t_i m_i})\hat{R}(s,z) \tag{2.22}$$

where the matrix $\hat{R}(s,z)$ can also be written as

$$\hat{R}(s,z) = diag(\hat{w}_i(s,z))Q \tag{2.23}$$

and where $\tilde{S} = \tilde{S}^i$ (given by (2.18)) and $\tilde{M} = \tilde{M}^i$ are the duration and number of arrivals in this particular state where the state variable at time $t = 0$ is i; that is $J_0 = i$.

The data volume arrived during the initial state may be written as $\tilde{X}^i = V_0^i + V_1^i + V_2^i + \ldots + V_{\tilde{M}^i}^i$ where all the V^i's are distributed according to $U_i(x) = P(V^i \leq x)$ as defined in subsec. 2.3.1.

We may now combine some of the results discussed above and consider a modulated RP in equilibrium over several state changes. By combining the results above for the initial state 2.3.1 and for the normal state 2.3.1 we obtain the following theorem.

Theorem 1. *Consider a modulated RP in equilibrium and observe the process from a random point taken to be $t = 0$ and let Y_k and L_k be the time and numbers of arrivals up to the k'th state change. Defining the matrix*

$$\begin{aligned} &R^k(s,z) = (r_{ij}^k(s,z)) \quad \textit{where} \\ &r_{ij}^k(s,z) = E[e^{-sY^k} z^{L^k} \boldsymbol{1}_{\{I_{k+1}=j\}} \mid J_0 = i] \end{aligned} \tag{2.24}$$

we then have

$$R^k(s,z) = \tilde{R}(s,z) \cdot R(s,z)^{k-1} \tag{2.25}$$

where $\tilde{R}(s,z)$ is given by (2.22) *and $R(s,z)$ is given by* (2.5).

Proof. We have $Y^k = \tilde{S}_1 + S_2 \ldots + S_k$ and $L^k = 1 + \tilde{M}_1 + M_2 \ldots + M_k$ where $\tilde{S}_1$ and $1 + \tilde{M}_1$ are the time and number of arrivals to the first state change for the initial period and S_l and M_l are the time and number of arrivals for period l; $l = 2, \cdots, k$. By inserting in (2.24) we then obtain (2.25). □

Packet counts and index of dispersion

To study the behavior of modulated RP over longer time periods, we want to find the distribution of the number of arrivals up to a certain point in time. To do this we first find the distribution of the number of arrivals in the last period which includes that point in time. We therefore first consider the process in a given time interval. Suppose that the state is $I_k = i$ and define the time up to the l'th arrival within this state as

$$S_l^i = T_1^i + T_2^i + \ldots + T_l^i \,. \tag{2.26}$$

Similarly, let $\hat{N}_t^i$ denote the number of arrivals that occur up to time t, assuming that at time $t = 0$ the state is $I_k = i$ and there is no state change in the interval

$(0,t)$. Therefore, the event $\{\hat{N}_t^i = l\}$ equals that of $\{S_l^i > t, S_{l+1}^i \leq t, M^i > l\}$. We hence have

$$\begin{aligned} &P(\hat{N}_t^i = l) = P(S_l^i > t, S_{l+1}^i \leq t, M^i > l) = \\ &(P(S_l^i \leq t) - P(S_{l+1}^i \leq t))P(M^i > l)\,. \end{aligned} \tag{2.27}$$

By defining the z-transform $\hat{H}^i(t,z) = E[z^{\hat{N}_t^i}]$ and by taking the Laplace transform $\hat{G}^i(s,z) = \int_{t=0}^{\infty} e^{-st}\hat{H}^i(t,z)dt$ and using the fact that the Laplace transform of the convolution $P(S_l^i \leq t)$ equals $\frac{f_i(s)^l}{s}$ then this give $\hat{G}^i(s,z) = \frac{1-f_i(s)}{s}\sum_{l=0}^{\infty}(zf_i(s))^l P(M^i > l) = \frac{(1-f_i(s))(1-P_i(zf_i(s)))}{s(1-zf_i(s))}$. Hence by (2.20) we have

$$\hat{G}^i(s,z) = \hat{w}_i(s,z) \tag{2.28}$$

Similarly, the initial period, $k = 0$ have to be treated somewhat differently due to the fact that the time to the first arrival is given by the residual time and the residual numbers of arrivals. By assuming $J_0 = i$ we define the time up to the l-th arrival in that period

$$\tilde{S}_l^i = \tilde{T}^i + T_1^i + \ldots + T_l^i \tag{2.29}$$

where $\tilde{T}^i$ is distributed according to residual arrival time. As above we let $\tilde{N}_t^i$ be the number of arrivals in the initial period ($k = 0$) up to a time t without any state changes in the interval $(0,t)$. The event $\{\tilde{N}_t^i = l\}$ equals that of $\{\tilde{S}_{l-1}^i > t, \tilde{S}_l^i \leq t, \tilde{M}^i > l-1\}$ leading to

$$\begin{aligned} &P(\tilde{N}_t^i = l) = P(\tilde{S}_{l-1}^i > t, \tilde{S}_l^i \leq t, \tilde{M}^i > l-1) = \\ &(P(\tilde{S}_{l-1}^i \leq t) - P(\tilde{S}_l^i \leq t))P(\tilde{M}^i > l-1)\,. \end{aligned} \tag{2.30}$$

As above, we define the z-transform $\tilde{H}^i(t,z) = E[z^{\tilde{N}_t^i}]$ and the Laplace transform $\tilde{G}^i(s,z) = \int_{t=0}^{\infty} e^{-st}\tilde{H}^i(t,z)dt$. By using the fact that the Laplace transform of the convolution $\tilde{P}(S_l^i \leq t)$ equals $\tilde{f}_i(s)\frac{f_i(s)^{l-1}}{s}$ we find $\tilde{G}^i(s,z) = z\tilde{f}_i(s)\frac{1-f_i(s)}{s}\sum_{l=0}^{\infty}(zf_i(s))^l P(\tilde{M}^i > l)$. Since $\sum_{l=0}^{\infty} z^l P(\tilde{M}^i > l) = \frac{1}{1-z} - \frac{1-P_i(z)}{m_i(1-z)^2}$ we obtain

$$\tilde{G}^i(s,z) = z\frac{(1-f_i(s))^2}{t_i s^2}\Big(\frac{1}{1-zf_i(s)} - \frac{1-P_i(zf_i(s))}{m_i(1-zf_i(s))^2}\Big) \tag{2.31}$$

We may now state the following theorem on the distribution of the number of arrivals up to a certain time t.

Theorem 2. *Consider a modulated RP in equilibrium and let N_t be the number of arrivals up to time t and let $P(N_t = n)$ be the corresponding distribution and let $H(t,z) = E[z^{N_t}]$ be the z-transform. Then the Laplace transform $G(s,z) = \int_{t=0}^{\infty} e^{-st}H(t,z)dt$ is given by the following matrix expressions*

$$\begin{aligned} G(s,z) = &\tfrac{1}{s} + \tfrac{z-1}{Cs^2}\Pi \cdot diag(m_i\tfrac{1-f_i(s)}{1-zf_i(s)}) \cdot e - \\ &\tfrac{z}{Cs}\Pi \cdot diag(\tfrac{1-f_i(s)}{1-zf_i(s)}\hat{w}_i(s,z)) \cdot e + \\ &\tfrac{z}{C}\Pi \cdot \hat{R}(s,z) \cdot [I - R(s,z)]^{-1} \cdot \hat{R}(s,z)) \cdot e \end{aligned} \tag{2.32}$$

where Π is the steady state distribution at state jumps given by (2.10)*, the constant C is given in* (2.11)*, $\hat{w}_i(s,z)$ is given by* (2.20) *and the matrices $R(s,z)$ and $\hat{R}(s,z)$ are given by* (2.5) *and* (2.23) *respectively.*

Proof. We first observe to have $N_t = 0$ we must have the residual time $\tilde{T}^i > t$ and hence,

$$P(N_t = 0) = \sum_{i=1}^{N} p_i P(\tilde{T}^i > t) \tag{2.33}$$

The condition $N_t = n$ when $n > 0$ may be attained by either having no state changes up to t or on ore more state changes, e.g., $k \geq 1$. For the latter case, we condition on the elapsed time $Y^k = y$ and arrivals $L^k = l$ up to the k'th state change. Then the numbers of arrivals in the remaining interval of length $t - y$ has to be $n - l$ (to have $N_t = n$). By integrating and summing over all possible combinations of arrivals in the two intervals and summing over all $k \geq 1$, and then multiplying by the initial state probabilities and summing over all states both at time $t = 0$ and t, one gets the following expression

$$\begin{array}{l} P(N_t = n) = \sum_{i=1}^{N} p_i P(\tilde{N}_t^i = n) + \\ \sum_{k=1}^{\infty} \sum_{i=1}^{N} \sum_{i_0=1}^{N} p_{i_0} \sum_{l=0}^{n} \int_{y=0}^{t} P(\hat{N}_{t-y}^i = n - l) \\ d_y P(Y^k \leq y, L^k = l, I^k = i \mid J_0 = i_0) \, . \end{array} \tag{2.34}$$

Now we take the transforms of the expressions (2.33) and (2.34). Laplace transform of $P(N_t = 0)$ yields $P \cdot diag(\frac{1}{s} - \frac{1-f_i(s)}{t_i s^2}) \cdot e$. Similarly, the transforms of the second term $\sum_{i=1}^{N} p_i P(\tilde{N}_t^i = n)$ is $P \cdot diag(\tilde{G}^i(s,z)) \cdot e$. Finally the third convolution part yields the sum $\sum_{k=1}^{\infty} P \cdot R^k(s,z) \cdot diag(\hat{w}_i(z,s) \cdot e$. Using $R^k(s,z)$ given by (2.25) yields $\frac{z}{C} \Pi \cdot \hat{R}(s,z) \cdot [I - R(s,z)]^{-1} \cdot \hat{R}(s,z)) \cdot e$ where we also have used that $diag(\hat{w}_i(z,s) \cdot e = \hat{R}(s,z)) \cdot e$. Collecting and substituting for P in terms of the steady state jump probabilities Π and inserting for $\tilde{G}^i(s,z)$ by (2.31) we then obtain (2.32). □

To find the Laplace transform of the first and second moments of N_t turn out to beneficial to rewrite the expression for $G(s,z)$ above by taking $\hat{w}_i(s,z) = \frac{1}{sz}(1 + \frac{z-1}{1-zf_i(s)})(1 - P_i(zf_i(s))$. This leads to the following simplification

$$G(s,z) = \frac{1}{s} + \frac{z-1}{Czs^2} \Pi \cdot diag(m_i) \cdot e + \tag{2.35}$$

$$\frac{(z-1)^2}{Czs^2} \left\{ \Pi \cdot diag\left(\frac{m_i}{1 - zf_i(s)} - \frac{1 - P_i(zf_i(s))}{1 - zf_i(s)} \right) \cdot e + \right. \tag{2.36}$$

$$\Pi \cdot diag\left(\frac{1 - P_i(zf_i(s))}{1 - zf_i(s)} \right) \cdot Q \cdot [I - diag(P_i(zf_i(s))) \cdot Q]^{-1} \cdot \tag{2.37}$$

$$\left. diag\left(\frac{1 - P_i(zf_i(s))}{1 - zf_i(s)} \right) \cdot e \right\} . \tag{2.38}$$

By (2.38) we find that the first moment is proportional to the length of the interval, while for the variance the general result is found in terms of Laplace transforms. The result is stated in the following theorem.

Theorem 3. *For the mean and variance of N_t we have the following expressions*

$$E[N_t] = t \frac{\Pi \cdot diag(m_i) \cdot e}{C} \tag{2.39}$$

$$\int_{t=0}^{\infty} e^{-st} Var[N_t]dt = -\frac{\Pi \cdot diag(m_i) \cdot e}{Cs^2} - \frac{2(\Pi \cdot diag(m_i) \cdot e)^2}{C^2 s^3} + \tag{2.40}$$

$$\frac{2}{Cs^2}\left\{\Pi \cdot diag\left(\frac{m_i}{1-f_i(s)} - \frac{1-P_i(f_i(s))}{1-f_i(s)}\right) \cdot e + \right. \tag{2.41}$$

$$\Pi \cdot diag\left(\frac{1-P_i(f_i(s))}{1-f_i(s)}\right) \cdot Q \cdot [I - diag(P_i(f_i(s))) \cdot Q]^{-1} \cdot \tag{2.42}$$

$$\left. diag\left(\frac{1-P_i(f_i(s))}{1-f_i(s)}\right) \cdot e \right\}. \tag{2.43}$$

Proof. These results follow directly from (2.38) by differentiating with respect to z to first and second order, and then finding the Laplace transform of $E[N_t]^2$. □

For large t both the mean and variance will grow with rate proportional to t, it is therefore natural to introduce the *index of dispersion of counts* (IDC) as the ratio between variance and mean

$$I_t = \frac{Var[N_t]}{E[N_t]}. \tag{2.44}$$

By Tauberian arguments, it is possible to obtain the asymptotic expansion of $Var[N_t]$ for large t. The method used is to expand the inverse matrix $[I - diag(P_i(f_i(s)))]^{-1}$ in terms of its adjoint matrix and the determinant, and then also expand both $\frac{m_i}{1-f_i(s)} - \frac{1-P_i(f_i(s))}{1-f_i(s)}$ and $\frac{1-P_i(f_i(s))}{1-f_i(s)}$ for small s.

Theorem 4. *The variance and IDC have the following asymptotic expressions for large t*

$$Var[N_t] = At + B + O(t^{-1}) \tag{2.45}$$

$$I_t = \frac{A}{D} + \frac{B}{D}t^{-1} + O(t^{-2}) \tag{2.46}$$

where A and B are constants given in terms of model parameters and $D = E[N_1] = \frac{\Pi \cdot diag(m_i) \cdot e}{C}$.

Proof of the asymptotic expansion of $Var[N_t]$ *for large t.* We first expand the different parts of the Laplace transform of $Var[N_t]$ in (2.43) for small s. We denote σ^2 and γ^3 as the variance and 3'rd central moment respectively. We find

$$P_i(f_i(s)) = 1 - m_i t_i s + \frac{1}{2}u_i^{(2)}s^2 - \frac{1}{6}u_i^{(3)}s^3 + o(s^3) \tag{2.47}$$

where

$$\begin{aligned} u_i^{(2)} &= t_i^2(\sigma_{M_i}^2 + m_i^2) + \sigma_{T_i}^2 m_i \\ u_i^{(3)} &= t_i^3(\gamma_{M_i}^3 + m_i^3) + t_i\sigma_{T_i}^2(3\sigma_{M_i}^2 + 3m_i^2 - 3m_i) + \gamma_{T_i}^3 m_i \end{aligned} \tag{2.48}$$

and

$$\frac{1-P_i(f_i(s))}{1-f_i(s)} = m_i - \frac{1}{2}w_i^{(1)}s + \frac{1}{12}w_i^{(2)}s^2 + o(s^2) \tag{2.49}$$

where

$$\begin{aligned} w_i^{(1)} &= t_i(\sigma_{M_i}^2 + m_i^2 - m_i) \\ w_i^{(2)} &= t_i^2(2\gamma_{M_i}^3 - 3\sigma_{M_i}^2 + 2m_i^3 - 3m_i^2 + m_i) + \\ &\quad \sigma_{T_i}^2(3\sigma_{M_i}^2 + 3m_i^2 - 3m_i) \end{aligned} \tag{2.50}$$

and

$$\frac{m_i}{1-f_i(s)} - \frac{1-P_i(f_i(s))}{(1-f_i(s))^2} = \frac{1}{2}\nu_i^{(1)} - \frac{1}{6}\nu_i^{(2)}s + o(s) \tag{2.51}$$

where

$$\begin{aligned}\nu_i^{(1)} &= \sigma_{M_i}^2 + m_i(m_i-1)\\ \nu_i^{(2)} &= t_i(\gamma_{M_i}^3 - 3\sigma_{M_i}^2 + m_i(m_i-1)(m_i-2))\end{aligned} \tag{2.52}$$

From the expression of the Laplace transform of the variance given by (2.43) we see that the hard part to is to expand $[I - diag(P_i(f_i(s)))Q]^{-1}$ for small s. We use the notion of adjoint matrices and use the result $adjA \cdot A = A \cdot adjA = I \cdot detA$ for non-singular NxN matrix, or $A^{-1} = \frac{adjA}{detA}$. Expanding, we have $[I - diag(P_i(f_i(s)))Q] = I - Q + s\,diag(m_i t_i)\cdot Q - \frac{1}{2}s^2 diag(u_i^2)\cdot Q + o(s^2)$. Similarly, we expand both the adjoint and the determinant $adj[I - diag(P_i(f_i(s)))Q] = H_0 + sH_1 + s^2H_2 + o(s^2)$ and $det[I - diag(P_i(f_i(s)))Q] = b_0 + sb_1 + s^2b_2 + s^3b_3 + o(s^3)$. It follows that $b_0 = 0$ since $det[I - Q] = 0$. By expanding the of identity for adjoint matrices we obtain the following equations to determine H_0

$$\begin{aligned}&H_0[I-Q] = [I-Q]H_0 = 0\\ &H_1[I-Q] + H_0 diag(m_i t_i)Q =\\ &[I-Q]H_1 + diag(m_i t_i)QH_0 = b_1\end{aligned} \tag{2.53}$$

The first equation gives $H_0 = a_0 L$ where a_0 is a constant and $L = e \cdot \Pi$. By pre- or post-multiplying the second equation by the matrix L gives $a_0 L \cdot diag(m_i t_i) \cdot Q \cdot L = b_1 L$ giving $a_0 C = b_1$ with $C = \Pi \cdot diag(m_i t_i) \cdot e$, and hence $H_0 = \frac{b_i}{C}L$. Expanding to second order of the inverse by using the expression $\frac{adj[I-diag(P_i(f_i(s)))Q]}{det[I-diag(P_i(f_i(s)))Q]}$, we finally find the following expansion for the inverse

$$\begin{aligned}&Q[I - diag(P_i(f_i(s)))Q]^{-1} = \tfrac{1}{s}\tfrac{L}{C} + \tfrac{1}{b_1}B_1 - \tfrac{b_2}{b_1}\tfrac{L}{C} +\\ &s(\tfrac{1}{b_1}B_2 - \tfrac{b_2}{b_1^2}B_1 + [\tfrac{b_2^2}{b_1^2} - \tfrac{b_3}{b_1}]\tfrac{L}{C}) + o(s)\end{aligned} \tag{2.54}$$

where we have defined $B_i = QH_i$ and further H_i and b_i is the i'th coefficients in the expansion of $adj[I - diag(P_i(f_i(s)))Q]$ and $det[I - diag(P_i(f_i(s)))Q]$ respectively. The sought constants A and B are now the coefficients of s^{-2} and s^{-1} in the expansion of the Laplace transform (2.43) above. We first observe that the coefficient of s^{-3} vanishes as expected. We find

$$\begin{aligned}A = &\tfrac{\Pi\cdot diag(\nu_i^{(1)})\cdot e}{C} - \tfrac{\Pi\cdot diag(m_i)\cdot e}{C}\\ &+2\tfrac{\Pi\cdot diag(m_i)\cdot B_1\cdot diag(m_i)\cdot e}{b_1 C} - 2\tfrac{b_2}{b_1}(\tfrac{\Pi\cdot diag(m_i)\cdot e}{C})^2\\ &-2\tfrac{(\Pi\cdot diag(m_i)\cdot e)(\Pi\cdot diag(w_i^{(1)})\cdot e)}{C^2}\end{aligned} \tag{2.55}$$

and

$$\begin{aligned}B = &2\tfrac{\Pi\cdot diag(m_i)\cdot B_2\cdot diag(m_i)\cdot e}{b_1 C} - 2\tfrac{b_2}{b_1}\tfrac{\Pi\cdot diag(m_i)\cdot B_1\cdot diag(m_i)\cdot e}{b_1 C}\\ &+\tfrac{\Pi\cdot diag(m_i)\cdot B_1\cdot diag(w_i^{(1)})\cdot e}{b_1 C} + \tfrac{\Pi\cdot diag(w_i^{(1)}))\cdot B_1\cdot diag(m_i)\cdot e}{b_1 C}\\ &-\tfrac{1}{3}\tfrac{\Pi\cdot diag(\nu_i^{(2)})\cdot e}{C} + 2[(\tfrac{b_2}{b1})^2 - \tfrac{b_3}{b_1}](\tfrac{\Pi\cdot diag(m_i)\cdot e}{C})^2\\ &+2\tfrac{b_2}{b_1}\tfrac{(\Pi\cdot diag(m_i)\cdot e)(\Pi\cdot diag(w_i^{(1)}))\cdot e}{C^2}\\ &+\tfrac{1}{3}\tfrac{(\Pi\cdot diag(m_i)\cdot e)(\Pi\cdot diag(w_i^{(2)}))\cdot e}{C^2} + \tfrac{1}{2}(\tfrac{\Pi\cdot diag(w_i^{(1)})\cdot e}{C})^2\end{aligned} \tag{2.56}$$

For the general case, the expansion of the determinant and the adjoint of $[I - diag(P_i(f_i(s)))Q]$ will be the hard part to find.

□

Remark 1 (Explicit expressions for A two and three state models). The results given in (2.55) and (2.56) above require the matrices B_1 and B_2 as the first and second order expansion of the adjoint matrix $[I - diag(P_i(f_i(s)))Q]$ as well as b_1,b_2 and b_3; the three first coefficient for the determinant. For general number of states analytical expression is hard to find unless for small value of number of states. Below, the explicit expressions for the two and three state cases are given. For $N = 2$ we find the following expression for the constant A

$$\begin{array}{l} A = \frac{1}{(m_1t_1+m_2t_2)^3}((m_1+m_2)^2(m_1\sigma_{T_1}^2 + m_2\sigma_{T_2}^2)+ \\ \quad (t_1-t_2)^2(m_2^2\sigma_{M_1}^2 + m_1^2\sigma_{M_2}^2)) \end{array} \tag{2.57}$$

For $N = 3$ the expressions are far more technical with several more parameters. We take the Q-matrix as follows

$$Q = \begin{pmatrix} 0 & q_{12} & 1-q_{12} \\ 1-q_{23} & 0 & q_{23} \\ q_{31} & 1-q_{31} & 0 \end{pmatrix} \tag{2.58}$$

and define the following auxiliary parameters

$$\begin{array}{l} r_1 = 1 - q_{23}(1-q_{31}) \\ r_2 = 1 - q_{31}(1-q_{12}) \\ r_3 = 1 - q_{12}(1-q_{23}) \end{array} \tag{2.59}$$

and the steady state probabilities at state jumps is then

$$\pi_1 = \tfrac{r_1}{r_1+r_2+r_3}, \pi_2 = \tfrac{r_2}{r_1+r_2+r_3}, \pi_3 = \tfrac{r_3}{r_1+r_2+r_3} \tag{2.60}$$

We find the following expression for the constant A

$$\begin{array}{l} A = \frac{1}{(r_1m_1t_1+r_2m_2t_2+r_3m_3t_3)^3} \\ \Big((r_1m_1+r_2m_2+r_3m_3r_3)^2(r_1m_1\sigma_{T_1}^2 + r_2m_2\sigma_{T_2}^2 + r_3m_3\sigma_{T_3}^2)+ \\ r_1\sigma_{M_1}^2\big(r_2m_2(t_1-t_2) + r_3m_3(t_1-t_3)\big)^2+ \\ r_2\sigma_{M_2}^2\big(r_1m_1(t_2-t_1) + r_3m_3(t_2-t_3)\big)^2+ \\ r_3\sigma_{M_3}^2\big(r_1m_1(t_3-t_1) + r_2m_2(t_3-t_2)\big)^2+ \\ \gamma_{12}m_1^2m_2^2(t_1-t_2)^2 + \gamma_{13}m_1^2m_3^2(t_1-t_3)^2 + \gamma_{23}m_2^2m_3^2(t_2-t_3)^2+ \\ 2m_1m_2m_3\big(r_1m_1\delta_1(t_1-t_2)(t_1-t_3)+ \\ r_2m_2\delta_2(t_2-t_3)(t_2-t_1) + r_3m_3\delta_3(t_3-t_1)(t_3-t_2)\big)\Big) \end{array} \tag{2.61}$$

where we have defined the parameters

$$\gamma_{ij} = r_ir_j(2-r_i-r_j) \quad for \quad i,j = 1,2,3 \tag{2.62}$$

and

$$\delta_i = 1 - r_i - \textstyle\prod_{j=1,j\neq i}^{3}(1-r_j) \quad for \quad i = 1,2,3 \tag{2.63}$$

Observe that we get the two state solution if two of the states have equal mean and variance of their arrival distribution. E.g. if we have $t_2 = t_3 (= t_2^*)$ and $\sigma_{T_2}^2 = \sigma_{T_3}^2 (= \sigma_{T_2^*}^2)$ we obtain the result for $N = 2$ by defining the following weighted mean and variance

$$m_2^* = \frac{r_2}{r_1} m_2 + \frac{r_3}{r_1} m_3 \tag{2.64}$$

$$\begin{aligned} \sigma_{M_2^*}^2 = \frac{r_2}{r_1}\sigma_{M_2}^2 + \frac{r_3}{r_1}\sigma_{M_2}^2 + \frac{r_2(2-r1-r2)}{r_1^2} m_2^2 + \\ \frac{r_3(2-r1-r3)}{r_1^2} m_3^2 + 2\frac{r_2+r_3-r_2 r_3-r_1}{r_1^2} m_2 m_3 \end{aligned} \tag{2.65}$$

2.3.2 Fitting two state models to recorded traces

One of the main difficulties when applying general source models is to set the model's parameters based on real measurements (traces). In our case, for each state it is necessary to find three distributions, namely for the inter-arrival times, the packet size, and the number of arrivals, in addition to the transition matrix of the modulating Markov Chain. Instead, some important parameters, like the overall statistical moments and autocorrelation of key variables or the dispersion index, are quite easy to estimate by exploiting the (supposed) ergodic nature of the involved stochastic processes. Hence, by estimating a set of parameters and requiring that these measurements match the corresponding analytical expressions (derived from the model), we obtain a set of equations that may be solved for the model parameters.

We observe that the number of statistical distributions that need to be estimated for an N-state modulated RP based on the recorded traces grows linearly with N, while the size of the transition matrix is equal to N^2.

Clearly, the larger the number of model's parameters to be estimated, the larger the required data set, and the more noisy the resulting model. It is therefore convenient to simplify the model by fixing some of the model's parameters and estimating the remaining from the available data traces. The difficulty is to choose the right balance between model complexity, and its capability to capture the more important features of the actual source process.

To set the model parameters, two classes of measurements are considered, namely:

- short-time scale: values that are meaningful over short time scales, like mean and variance of the inter-arrival times, packet size, and number of arrivals processes,
- long-time scale: values describing the packet generation process over longer time scales, like packet counts or index of dispersion.

In the following, the proposed simplified source model is detailed.

Two-state model with negative exponential arrival distribution and geometrical distribution of arrivals

The simplest (non-trivial) model consists of two modulating states, $\Omega = \{1, 2\}$, with single-parameter distributions for the inter-arrival times, the packet size and the number of arrivals in a state. More specifically, the inter-arrival times are chosen to be negative exponentially distributed random variables with mean

t_1 and t_2 for state $I_k = 1$ and $I_k = 2$, respectively, while the number of arrivals in each state is modeled as a geometrically distributed random variable with mean m_1 and m_2, respectively while the packet size is exponentially distributed. Hence, the model is fully described by the parameter set $\{t_1, t_2, m_1, m_2\}$. For this case the Laplace transform (2.43) of the variance of N_t is invertible and we find the following IDC

$$I_t = 1 + 2\frac{m_1^2 m_2^2 (t_1 - t_2)^2}{(m_1 t_1 + m_2 t_2)^2 (m_1 + m_2)} - 2\frac{m_1^3 t_1 m_2^3 t_2 (t_1 - t_2)^2}{t(m_1 t_1 + m_2 t_2)^3 (m_1 + m_2)}(1 - e^{-t\frac{m_1 t_1 + m_2 t_2}{m_1 t_1 m_2 t_2}}) \quad (2.66)$$

when $t \to \infty$, (2.66) gives

$$I_\infty = 1 + 2\frac{m_1^2 m_2^2 (t_1 - t_2)^2}{(m_1 t_1 + m_2 t_2)^2 (m_1 + m_2)}\,. \quad (2.67)$$

Moreover, if we know the IDC for a particular time t_0 and we take $F^{-1}(y)$ as the inverse function of $F(x) = \frac{1-e^{-x}}{x}$, we may solve for the exponent in (2.66), leading to

$$\frac{m_1 t_1 + m_2 t_2}{m_1 t_1 m_2 t_2} = \frac{1}{t_0} F^{-1}\left(\frac{I_\infty - I_{t_0}}{I_\infty - 1}\right)\,. \quad (2.68)$$

Suppose that from trace measurements we have estimated the four parameters, i.e., the mean, square coefficient of variation, and IDC at infinity and at a particular time t_0. We may then set up the following four equations to determine the model parameters $\{t_1, t_2, m_1, m_2\}$

$$\begin{aligned} &\tfrac{m_1 t_1 + m_2 t_2}{m_1 + m_2} = a = E[T_{avg}] \\ &2\tfrac{m_1 m_2 (t_2 - t_1)^2}{(m_1 t_1 + m_2 t_2)^2} = b = \tfrac{Var[T_{avg}]}{E[T_{avg}]^2} - 1 \\ &2\tfrac{m_1^2 m_2^2 (t_2 - t_1)^2}{(m_1 t_1 + m_2 t_2)^2 (m_1 + m_2)} = c = I_\infty - 1 \\ &\tfrac{m_1 t_1 + m_2 t_2}{m_1 t_1 m_2 t_2} = d = \tfrac{1}{t_0} F^{-1}(\tfrac{I_\infty - I_{t_0}}{I_\infty - 1}) \end{aligned} \quad (2.69)$$

where we assume that all the parameters on the right hand side, i.e., $\{E[T_{avg}], Var[T_{avg}], I_\infty, I_{t_0}\}$, are known by direct measurements. Fortunately, (2.69) yields quadratic equations by substituting $x_1 = m_1 t_1$ and $x_2 = m_2 t_2$ for which we find the following solutions

$$\begin{aligned} x_1 &= \tfrac{1}{4b^2 d\eta}\left(\Delta + (2\eta + b(\eta - 2))\sqrt{\Delta}\right) \\ x_2 &= \tfrac{1}{4b^2 d\eta}\left(\Delta - (2\eta + b(\eta - 2))\sqrt{\Delta}\right) \\ m_1 &= \tfrac{1}{4ab^2 d\eta}\left(\Delta + (2\eta - b(\eta + 2))\sqrt{\Delta}\right) \\ m_2 &= \tfrac{1}{4ab^2 d\eta}\left(\Delta - (2\eta - b(\eta + 2))\sqrt{\Delta}\right) \end{aligned} \quad (2.70)$$

where

$$t_1 = \frac{x_1}{m_1} \;\text{ and }\; t_2 = \frac{x_2}{m_2}\,. \quad (2.71)$$

and

$$\Delta = 4\eta^2 + 4\eta(\eta - 2)b + (\eta - 2)^2 b^2 \;\; and \;\; \eta = acd\,. \quad (2.72)$$

Modeling examples

In the following, the proposed method is applied for three types of M2M sources, namely

- Single electricity meter.
- Concentrator which aggregates measurements from different electricity meters.
- Parking meter.

The considered source types are taken from a real M2M network based on recorded traces for both uplink (UL) and downlink (DL). Based on the traces, different time-series are constructed and analysed. In the examples below we mainly concentrate on the packet arrival process. The measured parameters are: mean and coefficient of variation of the arrival times, and index of dispersion at two points in time, one very large and the second at 10 s. The estimated parameters are given in Tab. 2.4. We observe that the average time between packets is relatively long, e.g., in the range of one hour for the electricity meter, while for the parking meter the mean is around a couple of minutes. We have large values, i.e., in the 10-25 range, for both the square coefficient of variation and the dispersion index for large time. The index of dispersion at 10 s is less than 10 for all cases. From these parameters and by manual inspection of the recorded time-series we conclude that these traffic sources are very bursty in nature.

	$E[T_{avg}]$ [s]	$\frac{Var[T_{avg}]}{E[T_{avg}]^2}$	I_∞	$I_{10\text{sec}}$	$E[V_{avg}]$ [B]	$\frac{Var[V_{avg}]}{E[V_{avg}]^2}$
ElMeter P2P UL	2969.73	10.8499	12.4174	9.63390	425.168	1.27135
ElMeter P2P DL	3812.49	8.2299	10.0329	7.12250	39.5970	0.44533
ElMeter Conc. UL	5025.31	16.0387	15.2734	5.07427	69.5619	3.53885
ElMeter Conc. DL	5387.97	14.8912	14.7908	4.22270	32.0204	1.64640
Parking UL	162.055	20.0424	25.3383	6.89966	728.636	10.42150
Parking DL	188.133	17.1263	22.8702	4.94596	406.867	0.10329

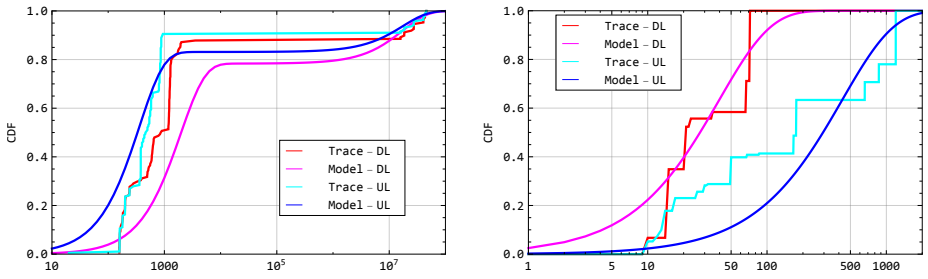
Table 2.4: Measured parameters for the sources

The estimated two-state model parameters are shown in Tab. 2.5. We observe that the resulting calculated parameters give a typical ON/OFF pattern with two very distinct states. In the first state, the inter arrival time is in the range of one second and the mean number of packets is estimated around 10 packets, while in the second mode inter-packet time is large and the mean number of packet arrivals in this state is less than two for all the sources. Hence, for all the cases we have a typical ON/OFF behaviour, with a burst of approximately ten packets and then very long time between bursts.

The corresponding distributions of the inter-arrival times and packet lengths for the different cases are shown in Fig. 2.10, Fig. 2.11 and Fig. 2.12. Compared to the estimated distributions, the CDF based on the model is smooth. However, for all the three cases, even if the match is not perfect, the similarity is

	t_1 [s]	m_1	t_2 [s]	m_2
ElMeter P2P UL	0.36137	6.86923	17597.4	1.39443
ElMeter P2P DL	0.58994	5.76723	17596.7	1.59488
ElMeter Cons. UL	1.73002	8.09070	42825.4	1.07525
ElMeter Cons. DL	2.27016	7.89402	42826.4	1.13559
Parking UL	1.26545	13.6396	1717.15	1.41026
Parking DL	1.95609	12.5222	1721.01	1.52089

Table 2.5: Calculated model parameters for the sources.



(a) Inter Arrival Time [ms]

(b) Packet Length [Bytes]

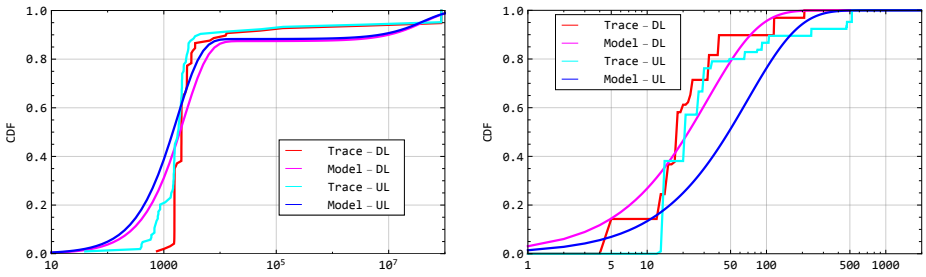
Figure 2.10: CDF of the inter arrival time and packet length obtained from the model and based on direct estimation from traces for single Electric Meter type of source.

evident. We see that the two modes manifest themselves in the form of the CDF with most of the arrivals occurring in the first mode with the relative small inter arrival times, while in mode two the time between packets will be several hours for the Electrical Metering sources. Instead, for the Parking source, the corresponding time between arrivals is typically half an hour. For the packet length distributions, the curves based on negative exponential approximations with the same mean as the empirical estimate ones have also been added.

Fig. 2.13, Fig. 2.14 and Fig. 2.15 show time-series from the traces and from simulations, using the two-state model with the calculated parameters given in Tab. 2.5. Also these figures confirm that the two-state model fits quite well with the traces, which clearly show that most of the inter-arrival times are quite small. Then there are a few observations with large time between arrivals and we clearly see that the model recreates similar behaviour.

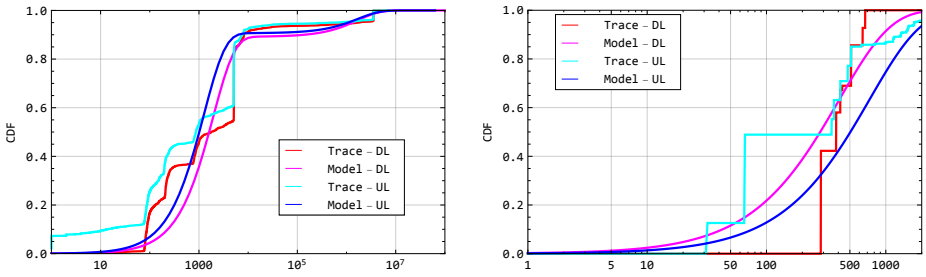
2.4 Conclusion

In this chapter we analysed some real world traffic traces, arriving to the conclusion that, in many cases, M2M traffic exhibits strong deterministic components.



(a) Inter Arrival Time [ms] (b) Packet Length [Bytes]

Figure 2.11: CDF of the inter arrival time and packet length obtained from the model and based on direct estimation from traces for Electric Meter from concentrator type of source.



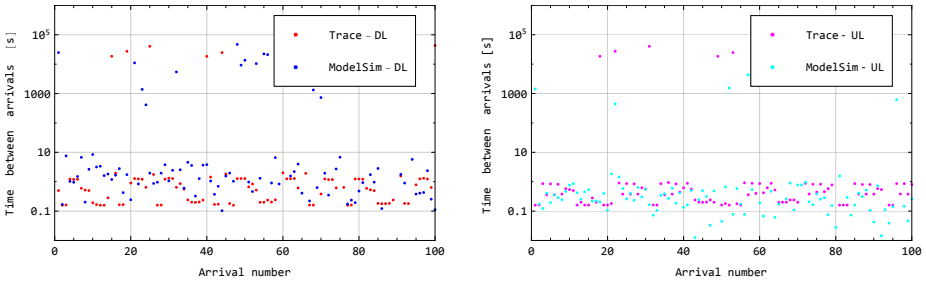
(a) Inter Arrival Time [ms] (b) Packet Length [Bytes]

Figure 2.12: CDF of the inter arrival time and packet length obtained from the model and based on direct estimation from traces for single Parking Meter type of source.

We have also discovered that none of the considered state of the art technique was able to reliably model all the traffic sources of the considered real world data set.

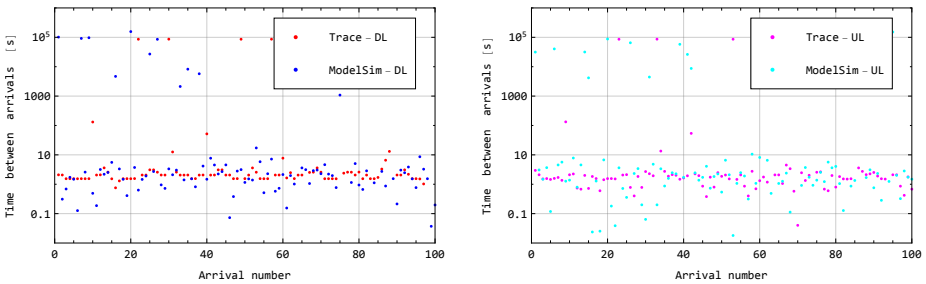
Therefore, the use of a generic packet-level model has been proposed, which is an alternating renewal process with different numbers of arrivals in each state. After all the arrivals happened for a particular state, the process changes its state, which corresponds to different distributions of the inter-arrival times and of the number of arrivals in the state. The most important performance measures for such processes have been derived, as the marginal arrival distributions and the corresponding statistical moments, and the distribution of the packet count in a given time interval has been analyzed. The variance of the packet count is found in terms of Laplace transform, and asymptotic behaviour is found for large times. For two- and three-state models, explicit results have been reported.

Finally, the work focused on a two-state model with exponentially distributed packet inter-arrival times and geometrically distributed number of ar-



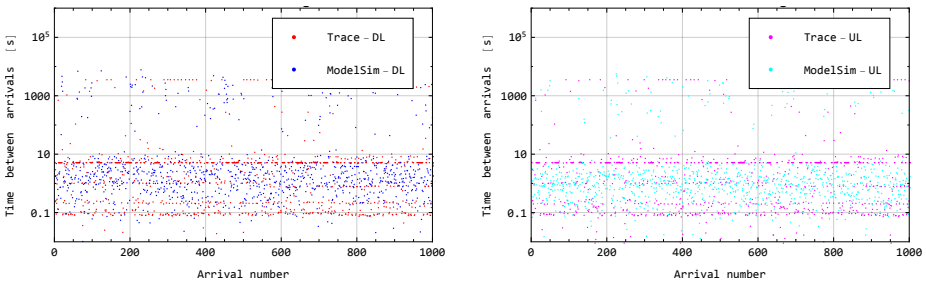
(a) Downlink (b) Uplink

Figure 2.13: Simulated and trace time series of the inter arrival times for single Electric Meter type of source.



(a) Downlink (b) Uplink

Figure 2.14: Simulated and trace time series of the inter arrival times for Electric Meter from concentrator type of source.



(a) Downlink (b) Uplink

Figure 2.15: Simulated and trace time series of the inter arrival times for single Parking Meter type of source.

rivals in a state, for a total of four model parameters. A method to determine these parameters from quantities estimated from traffic traces has been suggested. The method was tested against three types of M2M traces, showing reasonably similar stochastic characteristics to the considered traces for all three types. This proves the ability of the proposed method to adapt to such different traffic characteristics, contrary to the considered state of the art techniques.

Chapter 3

Uncoordinated access schemes for the IoT: approaches, regulations, and performance

IoT devices communicate using a variety of protocols, differing in many aspects, with the channel access method being one of the most important. Most of the transmission technologies explicitly designed for IoT and M2M communication use either an ALOHA-based channel access or some type of Listen Before Talk (LBT) strategy, based on carrier sensing. This chapter provides a comparative overview of the uncoordinated channel access methods for IoT technologies, namely ALOHA-based and LBT schemes, in relation with the ETSI and FCC regulatory frameworks. Furthermore, a performance comparison of these access schemes is performed, both in terms of successful transmissions and energy efficiency, in a typical IoT deployment.

3.1 Introduction

Forecasts say that, by 2020, IoT networks will need to handle 1.6 machine-type connections for each member of the global population [61]. A key element to enable the full realization of the IoT vision is the ubiquitous connectivity of end devices, with minimal configuration, as for the so-called *place-&-play* paradigm [23]. As anticipated in Sec. 1.2, the three main approaches to provide connectivity to IoT devices are the following.

Cellular systems. The existing cellular networks are a natural and appealing solution to provide connectivity to IoT end-devices, thanks to their world-wide established footprint and the capillary market penetration. Unfortunately, current cellular network technologies have been designed targeting wideband services, characterized by few connections that generate a large amount of data, while most IoT services are expected to generate a relatively small amount of traffic, but from a very large number of different devices. This shift of paradigm

challenges the control plan of current cellular standards, which can become the system bottleneck. For these reasons, the IoT and M2M scenarios are considered as major challenges for next generation wireless cellular systems, commonly referred to as 5G.

Short-range multi-hop technologies. This family collects a number of popular technologies specifically designed for M2M communications or Wireless Personal Area Networks (WPANs). These systems usually operate in the frequency bands centered around 2.4 GHz, 915 MHz and 868 MHz, though the 2.4 GHz is the most common choice. They are characterized by high energy efficiency and medium/high bitrates (order of hundreds of kbit/s or higher), but limited single-hop coverage area. To cover larger areas, most WPAN technologies provide the possibility to relay data in a multihop fashion, realizing a so-called *mesh network*. Examples of standards in this category are IEEE 802.15.4 [47], Bluetooth Low Energy [62], and Z-Wave, the latter having its physical and data link layers specified in ITU-T G.9959 [63].

LPWA networks. A third relevant class in the arena of IoT-enabling wireless technologies consists in the LPWA solutions. According to [64], LPWA technologies will account for 28% of M2M connections by 2020. These technologies, specifically designed to support M2M connectivity, provide low bitrates, low energy consumption, and wide geographical coverage. Almost all LPWA technologies operate at frequencies around 800 or 900 MHz, though there are also solutions working in the classic 2.4 GHz ISM band or exploiting white spaces in TV frequencies. Some relevant LPWA technologies are LoRaWAN™, Sigfox, Ingenu [25].

While cellular systems entail centralized access schemes over dedicated frequency bands, which provide high efficiency, robustness, security, and performance predictability, most of WPAN and LPWA technologies operate on unlicensed radio bands, adopting uncoordinated access schemes. The use of unlicensed bands yields the obvious advantage of lowering the operational costs of the network, while the adoption of uncoordinated channel access schemes makes it possible to simplify the hardware of the nodes, thus reducing the manufacturing costs and the energy consumption. The downside is that the lack of coordination in channel access may yield performance losses in terms of throughput and energy efficiency when the number of contending nodes increases.

To alleviate the problem of channel congestion in the unlicensed bands, radio spectrum regulators have imposed limits on the channel occupation of each device, in terms of bandwidth, time, and on the maximum transmission power. However, the Federal Communications Commission (FCC) in the USA and the Conference of Postal and Telecommunications Administrations (CEPT) in Europe have taken different approaches to limit channel congestion: the first imposes very strict limits on the emission power and favors the use of spread spectrum techniques but do not restrict the number of access attempts that can be performed by the nodes [65], while the second limits the fraction of on-air time of a device to be lower than a given *duty cycle*, or imposes the use of LBT techniques, which are also referred to as carrier-sense multiple access (CSMA) protocols [44].[1]

[1] The two terms will be used interchangeably in this study.

These precautions are actually effective when the coverage range of the wireless transmitters is relatively small (few meters), as was indeed the case for the first commercial products operating in the ISM frequency bands. However, this condition does no longer hold for LPWA solutions, which have coverage ranges in the order of 10–15 km in rural areas, and 2–5 km in urban areas, with a star-like topology that can exacerbate the mutual interference and hidden node problems. Furthermore, while short-range communication systems usually support a single, or just a few modulation schemes and transmit rates, LPWA technologies usually provide multiple transmit rates to optimize the transmission based on the distance to be covered.

Despite these quite radical changes in the transmit characteristics of the recent LPWA technologies with respect to the previous generation of the so-called Short Range Devices (SRD), the channel access methods and the regulatory constraints are still the same. In this chapter, we investigate the performance of well established uncoordinated channel access schemes in this new scenario, characterized by a huge number of devices with large coverage ranges and multi-rate capabilities. To this end, we first review the main uncoordinated access schemes used by the most common wireless communication technologies for the IoT, together with the regulatory framework. Then, the performance achieved by two popular uncoordinated access schemes in a typical LPWA network scenario is compared, considering the limits imposed by the regulations.

3.2 Uncoordinated access techniques for the IoT

Channel access schemes can be roughly divided in two main categories: coordinated and uncoordinated (or contention-based). Coordinated access schemes require time synchronization among the nodes and, hence, are more suitable for small networks (e.g., Bluetooth) or centrally controlled systems (e.g., cellular), with large traffic flows (e.g., voice or bulk data transfer). Uncoordinated access strategies, instead, are usually considered for networks with a highly variable number of devices and where a reduced manufacturing cost is required, since the more relaxed timing constraints of these strategies makes it possible to adopt low-cost oscillators and simpler components. In the following a quick overview is provided for the two main uncoordinated access schemes that are widely adopted by the transmission technologies typically associated to the IoT scenarios.

3.2.1 ALOHA-based schemes

Many protocols for M2M communication are based on pure ALOHA access schemes, according to which a transmission is attempted whenever a new message is generated by the device. Indeed, the ALOHA protocol was designed for a scenario somehow similar to that of IoT, targeting systems characterized by a large number of nodes that need to transmit short packets to a common receiver. Although the traffic per node is generally assumed to be very low, the aggregate traffic offered to the common receiver can be significant. Simplicity and efficiency at low traffic rates, thus, still make ALOHA the reference channel access choice in these scenarios.

This form of channel access may be coupled with a retransmission scheme,

according to which a packet is retransmitted until acknowledged by the receiver. However, some IoT services (e.g., environmental monitoring) can tolerate a certain amount of lost messages. In these cases, a retransmission scheme is not needed, allowing for a simplification of the device firmware and enabling a significant reduction in the energy consumption. For these reasons, ALOHA schemes are widely adopted in M2M communication as, for example, LoRaWAN and Sigfox. Furthermore, some standards that adopt LBT access techniques optionally provide an ALOHA mode of operation, as for the IEEE 802.15.4.

More sophisticated ALOHA-based protocols can be enabled when nodes are time synchronized, e.g., by means of beacons periodically broadcasted by coordinator nodes (e.g., gateways in LoRaWAN). For example, Slotted ALOHA (SA) divides the time in intervals of equal size, called slots, and allows transmissions only within slots, thus avoiding packet losses due to partially overlapping transmissions.

Hybrid ALOHA [66] extends the SA protocol by dedicating a set of timeslots to the transmission of training sequences for channel estimation, while the other slots are used for information data. If users transmit the training sequences in different timeslots, they can perform a correct channel estimation. Ideally, this allows overlapping transmissions not to collide due to advanced Multi-Packet Reception (MPR) techniques.

ALOHA has also been used to access the channel in a Near Field Communication (NFC) scenario. Pure ALOHA, however, proved to be inefficient when multiple NFC tags answer simultaneously to an identification request, generating packets collisions. This problem has been mitigated by the introduction of Framed Slotted ALOHA and its evolutions [67–70], which organize the slots in frames, and allow each device to transmit only once per frame, in a random slot.

The limit of these schemes is that packet transmission time should not exceed the slot duration. A common solution to accommodate uneven packet transmission times is to adopt a hybrid access scheme (HYB) that splits the frame in two parts: the first k slots are used by the nodes to send resource reservation messages to the controller, using a FSA access scheme, while the remaining slots in the frame are allocated by the controller to the nodes, according to the amount of resources required in the accepted reservation messages. The nodes get notified about the allocated resources by a control message that is broadcasted by the controller right after the end of the reservation phase. Variants of these basic mechanism are currently used in many different protocols as, e.g., GSM, 802.11e. However, to the best of our knowledge, the HYB approach has not yet been studied in the M2M scenario.

Another approach used to improve the performance of SA aims at reducing the contention by applying a more deterministic behavior, inspired by Time Division Multiple Access (TDMA). For example, in Reservation ALOHA (R-ALOHA) [71], slots are grouped together in frames containing the same number of slots. Whenever a node manages to transmit successfully on a slot in a frame, the same slot is automatically *reserved* to that user in the following frames. Slots that are not reserved can be accessed by all nodes as for SA.

Closely related protocols are the Packet Reservation Multiple Access (PRMA) [72, 73], where the base station explicitly acknowledges the transmission in each slot, and the Contention-TDMA (C-TDMA) [74] where, differently from PRMA, the reservation state of the slots is notified by the base station only once per

frame, thus reducing the communication overhead. These reservation-based protocols, however, are suboptimal when applied to MTC, where nodes generate packets sporadically.

The Probabilistic Time Division protocol [75] tries to dynamically balance random and scheduled access opportunities by simply adjusting the value of a single parameter a. Slots and frames are defined as in R-ALOHA and each user is associated with a *favored* slot in a frame. A node transmits on its favored slot with probability a, while with probability $1-a$ it uniformly chooses another slot in the frame. It has been shown that this protocol is immune to instability when $a > 0.7$ and, even when $a \leq 0.7$, the protocol is still less prone to instability than ALOHA. However, each node must be univocally assigned a favored slot, which makes the protocol inefficient for networks with a massive number of low traffic nodes.

The Coded Slotted ALOHA schemes employ a combination of successive interference cancellation and belief-propagation erasure decoding to iteratively remove the interference from the signal received in collided slots [76–78]. Under this scheme, nodes transmit multiple copies of each packet, which may be coded using a packet-level linear block code. When a packet is received in a non-collided slot, the receiver cancels out its contribution from all the previous collided slots where other copies of the same packet were transmitted, and the process is repeated for each new packet that gets decoded after the interference cancellation. This mechanism, however, requires multiple transmissions of each packet and inter-packet coding, which may be beyond the capabilities of very basic machine-type devices.

3.2.2 Carrier sensing schemes

When using carrier sensing techniques, each device listens to the channel before transmitting (from which the wording "Listen-Before-Talk"). The channel sensing operation is typically called *Clear Channel Assessment* (CCA) and aims at checking the occupancy of the channel by other transmitters, in which case the channel access will be delayed to avoid mutual interference that may result in the so-called *packet collisions*. The LBT schemes can differ in the way the CCA is performed and in the adopted behavior in case the channel is sensed busy.

The three most common methods to perform the CCA are the following.

- *Energy detection* (ED). The channel is detected as busy if the electromagnetic energy on the channel is above a given ED threshold.
- *Carrier sense* (CS). The channel is reported as busy if the device detects a signal with modulation and spreading characteristics compatible with those used for transmission, irrespective of the signal energy.
- *Carrier sense with energy detection* (CS+ED). In this case, a logical combination of the above methods is used, where the logical operator can be AND or OR.

The IEEE 802.15.4 standard supports all these CCA methods, along with pure ALOHA and two other modes specific for ultra-wideband communications. In an unslotted system, the backoff procedure for the IEEE 802.15.4 CCA mechanism tries to adapt to the channel congestion by limiting the rate at which

subsequent CCAs are performed for the same message. If the number of consecutive backoffs exceeds a given threshold, the message is discarded. Details about the CCA procedure in IEEE 802.15.4 networks can be found in [47], together with recommendations about the ED threshold and CCA detection time.

3.3 The regulatory framework

The use of unlicensed frequency bands by radio emitters is subject to regulations that are intended to favor the coexistence of a multitude of heterogeneous radio transceivers in the same frequency bands, limiting the mutual interference and avoiding any monopolization of the spectrum by single devices. The radio emitters operating in the ISM frequency bands are typically referred to as "Short Range Devices." However, the ERC Recommendation 70-03, emanated by the CEPT, specifies that *The term Short Range Device (SRD) is intended to cover the radio transmitters which provide either uni-directional or bi-directional communication which have low capability of causing interference to other radio equipment.* Despite the name, there is no explicit mention of the actual coverage range of such technologies. Therefore, long-range technologies operating in the ISM bands, such as Sigfox or LoRa, are still subject to the same regulatory constraints that apply to the actual short range technologies, as IEEE 802.15.4, Bluetooth, IEEE 802.11, and so on.

In the European Union, the European Commission designated the CEPT to define technical harmonization directives for the use of the radio spectrum. In 1988, under the patronage of the CEPT, the European Telecommunications Standards Institute (ETSI) was created to develop and maintain Harmonized Standards for telecommunications.

In the unlicensed radio spectrum at 868 MHz, the ETSI mandates a duty cycle limit between 0.1% and 1% over a 1 hour interval for devices that do not adopt LBT [44]. Only very specific applications, such as wireless audio, are allowed to ignore the duty cycle limitation. The duty cycle constraint can be relaxed by employing an LBT access scheme together with the Adaptive Frequency Agility (AFA), i.e., the ability to dynamically changing channel [44]. Devices with LBT and AFA capabilities, in fact, are only subject to a 2.8% duty cycle limitation for any 200 kHz spectrum. An example of technology that adopts the LBT approach is the IEEE 802.15.4 that, however, does not perfectly match the ETSI specifications, since its channel sensing period is shorter than that mandated by ETSI, which is between 5 ms and 10 ms, depending on the used bandwidth [44]. Instead, the recommendations on the LBT sensitivity, which shall be between -102 dBm and -82 dBm, are usually satisfied by commercial transceivers.

Due to the adoption by the European Union of a new set of rules for the radio equipments, called Radio Equipment Directive (RED) [79], ETSI is reviewing the related Harmonized Standards. However, devices that are compliant with the previous Radio and Telecommunication Terminal Equipment (R&TTE) Directive [80] can be placed on the market until June 17, 2017. Furthermore, devices that do not satisfy the constraints imposed by the Harmonized Standards can still be commercialized, but subject to a more comprehensive certification procedure attesting that the device meets the essential requirements of the European Directives [79]. The latest draft version of the ETSI Harmonized

Standards [81] includes some changes on the medium access procedures. In particular, the LBT technique is generalized as a *polite spectrum access* technique, while AFA is no more required. Furthermore, the LBT ED threshold has been relaxed, while the minimum CCA listening period has been increased.

The agency designated to regulate radio communications in the USA is the FCC, which also grants permits for the use of licensed radio spectrum and emanates regulations for wired communications. The FCC regulation does not impose any duty cycle restrictions to emitters operating in the 902–928 MHz band, but limits the maximum transmit power, for non-frequency hopping systems, to -1.25 dBm [65], which is significantly lower than the 14 dBm allowed by ETSI.

3.4 Performance analysis

ALOHA schemes and channel sensing techniques have been comprehensively modeled and their performance limits in terms of throughput and capacity are well understood (see, e.g., [82, 83], just to cite few). However, the use of different spreading techniques and/or modulation-&-coding-schemes to cope with the interference and to trade transmission speed for reliability, the large coverage range enabled by the LPWA technologies, the total reuse of the same frequency bands by different technologies, and the limitations imposed by the regulations to the channel access, raise the question on how effective are the classical uncoordinated channel access techniques to adequately support the expected growth of the IoT services.

In this section we shed some light on these aspects by presenting a simulation analysis of the performance achieved by ALOHA-based (specifically, pure ALOHA and HYB) and LBT access schemes in the simplest IoT scenario sketched in Fig. 3.1: a gateway (GW) receiving packets from a multitude of peripheral devices randomly spread over a wide area. Despite its simplicity, this scenario embodies most of the problems that can be expected in a real IoT deployment based on long-range technologies. In particular, we are interested in investigating how the distance from the gateway may impact the performance experienced by the node, with and without multirate capability and using either ALOHA or LBT techniques. ALOHA-based access schemes, in fact, allow the maximum energy saving in light traffic conditions, since they avoid the (even small) energy cost involved in carrier sensing. On the other hand, nodes farther away from the gateway are likely more prone to transmission failure due to interference, which however can potentially be mitigated by the use of LBT. Furthermore, the adoption of rate adaptation techniques is expected to increase the system capacity by reducing the transmit time of nodes closer to the gateway that not only will experience a lower interference probability, but will also have the chance to transmit more packets within the duty cycle limitations. It is hence interesting to investigate how much of such a performance gain will be transferred to the more peripheral nodes, and whether the LBT techniques can further improve performance in a significant manner.

Parameter		Value
Spatial node density	λ_s	10^{-3} nodes/m^2
Packet generation rate	λ_t	0.01 packets/s
Transmission power	$\mathrm{P_{tx}}$	14 dBm
Transmission frequency	f	868 MHz
Path loss coefficient	A	36.36 m^{-1}
Path loss exponent	β	3.5
Packet length	L	240 bit
Transmission bitrates	$\mathcal{R}$	$\{0.5, \ldots, 100\}$ kbit/s
Bandwidth	B_w	400 kHz
Noise spectral density	N_0	$2 \cdot 10^{-20}$ W/Hz
Duty cycle	δ_T	1%
Circuit power	P_c	16 dBm
Sensing time	T_s	0.4 ms
Sensing energy	E_s	3.98 μJ (LBT) 0.2 mJ (LBT+ETSI)
Smoothing parameter	α	0.1
Target outage probability for RA	p^*	0.05
HYB parameters		
Frame duration	T_W	60 s
Number of reservation slots in a frame	N_{RM}	80
Reservation message size	L_{RM}	24 bits
Reservation message transmit rate	R_{RM}	500 bit/s
Beacon duration	T_B	0.12 s
Resource notification message duration	T_{RA}	3.84 s

Table 3.1: Simulation parameters

3.4.1 Simulation scenario

In the simulations we consider a propagation model given by the product of the channel gain, $\gamma(d) = (Ad)^{-\beta}$, which accounts for the power decay with the distance d from the transmitter through the model parameters A and β, and the Rayleigh fading gain, which is modelled as an exponential random variable with unit mean.

We consider a limited set of possible transmission rates, namely $\mathcal{R} = \{0.5, 1, 5, 10, 50, 100\}$ kbit/s, and assume that a packet transmitted at rate $R \in \mathcal{R}$ is correctly decoded if the signal-to-interference-and-noise ratio (SINR), i.e., the received signal energy over the total noise energy plus interference energy collected by the receiver during the packet reception time, is above a certain threshold $\Gamma^\circ(R)$, which is determined from the Shannon channel capacity as

$$\Gamma^\circ(R) = 2^{R/B_\mathrm{w}} - 1 \tag{3.1}$$

where B_w is the signal bandwidth.

For the single rate case (SR), we suppose that all nodes transmit with the lowest bitrate of 500 bit/s. For the multirate scenario, instead, we consider a simple rate-adaptation mechanism that keeps a moving-average estimate of the

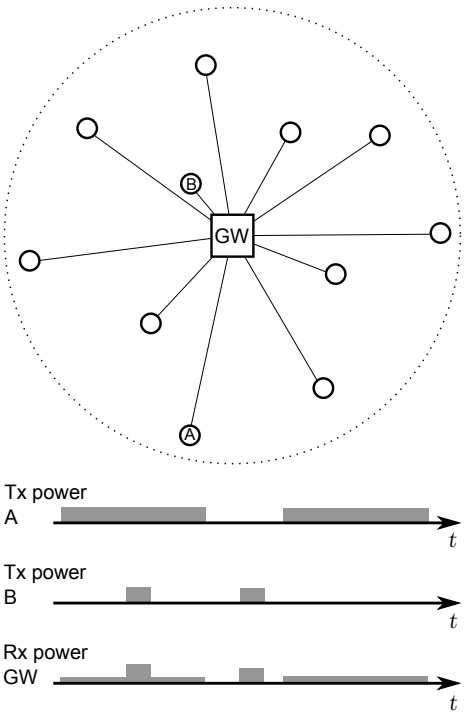


Figure 3.1: Above: simulation scenario, with multiple transmitters scattered around the common receiver (GW). Below: example of signal transmissions by nodes A and B, using different bitrates, and of received signal power at the gateway.

SINR (using a smoothing factor α) and selects the rate R^* so that the expected outage probability is not larger than $p^* = 0.05$. To improve the energy efficiency, furthermore, we assume no acknowledgement or retransmission mechanism is implemented, so that packets that are not successfully received are definitely lost.

The LBT scheme is implemented based on the IEEE 802.15.4 specifications. The ED CCA threshold is chosen to match the minimum signal power required to correctly receive a packet transmitted at the basic rate of 500 bit/s. This value is compatible with the limits on the LBT threshold imposed by ETSI [44].

As exemplified in Fig. 3.1, transmitting nodes are distributed as for a spatial Poisson process of rate λ_s [devices/m^2] over a circle with radius equal to the maximum coverage distance at the basic rate of 500 bit/s. Each device generates messages of length L according to a Poisson process of rate λ_t [packets/s]. All messages are addressed to the gateway that is placed at the center of the circle.

The setting of all the simulation parameters is reported in Tab. 3.1.

3.4.2 Transmission failure probability

We define p_{fail} as the probability that a transmitted message (including reservation messages in case of HYB) is received with SINR below threshold and,

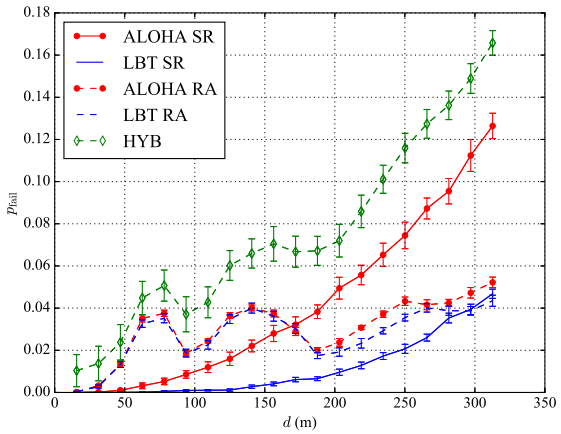


Figure 3.2: p_{fail} for ALOHA and LBT, for single rate (SR) and rate adaptive (RA) cases, with 95% confidence intervals.

hence, is not correctly decoded. For HYB, the transmission requests that are not accepted because of lack of slots in the transmission part of the frame are also included in the p_{fail}. Note that, while both the Single rate (SR) and Rate Adaptation (RA) versions of the pure-ALOHA and LBT schemes are considered here, only the RA version is considered for the HYB protocol, since this access scheme is more effective when packet transmissions have uneven duration. In Fig. 3.2 we can observe the failure probability for target nodes placed at increasing distances from the gateway. Red curves with circle markers refer to ALOHA, blue plain curves to LBT, and green dashed line with diamond markers to HYB. Solid and dashed lines have been associated to the SR and RA case, respectively.

For the SR case, we can see that the failure probability grows with the distance from the gateway, since nodes farther away have less SINR margin for successful decoding and are hence less robust to the interference produced by overlapping transmissions. In this case, carrier sense can indeed improve performance, even if the sensing range does not prevent the hidden node problem.

The downside of using LBT (not reported here for space constraints) is that up to 55% of the transmission attempts are aborted, in high traffic conditions, because the maximum number of CCAs is reached without finding an idle channel.

The adoption of RA changes significantly the performance, smoothing out the differences between the two access protocols. Indeed, higher bitrates allow the nodes near the receiver to occupy the channel for a lower period of time, thus reducing the probability of overlapping with other transmissions and improving the performance of both access schemes. Note that the change of rate with the distance is reflected by the oscillation in the failure probability that, however, remains approximately below $1 - p^*$.

Rather interestingly, HYB performs worse than the other schemes. The reason is that, in the considered scenario, the transmit time of reservation messages, always sent at the basic rate, is comparable to that of data packets sent at higher rates. Therefore, the reservation channel can become the system bot-

tleneck. The overall channel occupancy of HYB is thus significantly higher than that of the other two schemes, yielding higher failure probability.

3.4.3 Energy efficiency

Another key performance index in the IoT scenario is the *energy efficiency*, which is here defined as the ratio of the total number of bits successfully delivered to the gateway over the entire energy consumed by the node (including channel sensing and failed transmissions).

The power consumed during a transmission is modelled as the sum of a constant term, named circuit power, that represents the power used by the radio circuitry, and a term that accounts for the radiated power, which is called transmission power. When using LBT, the power required to perform the ED CCA is also added to the consumed power. Referring to the data-sheets of some off-the-shelf modules,[2] we set the circuit power to 16 dBm, the transmit power to 14 dBm, the receive power to 13 dBm, and the CCA power to 10 dBm [84,85].

In Fig. 3.3a we can see the energy efficiency for ALOHA and LBT access schemes when varying the distance of the target node from the gateway, in the SR case. We can observe that peripheral nodes exhibit lower energy efficiency because of the larger number of failure transmissions, and that the carrier sensing mechanism can alleviate this problem. The black curve marked with crosses shows the results obtained when using the parameters imposed by ETSI in the CCA procedure. As it can be seen, the energy efficiency is slightly lower than that obtained with the parameters adopted by commercial technologies, which may suggest that ETSI recommendations in this regard are possibly too conservative.

The adaptive rate case is shown in Fig. 3.3b, where the performance achieved by HYB is also shown. We can observe that both ALOHA and LBT can reach very high efficiency for nodes near the receiver, since the higher bitrates that decrease the transmit energy and the failure probability. It is worth to note that the first factor is dominant for the energy efficiency. The benefit transfers to the nodes farther away from the gateway, though the performance gain progressively reduces with the distance from the transmitter.

We also observe that, for nodes closer to the gateway, LBT shows a non-negligible energy efficiency loss with respect to ALOHA, which is even more marked when adopting the ETSI parameters. This is clearly due to the energy cost of the carrier sense mechanism, which takes a time comparable with the packet transmission time when using high bitrates. Furthermore, as revealed by the analysis of the failure probability, the carrier sense mechanism is not really worth for nodes close to the gateway when using RA, considering also that it may yield packet drops due to the impossibility of finding the channel idle within the maximum number of carrier sensing attempts. This problem would be further exacerbated in case of overlapping cells. Therefore, the use of CCA appears to be fruitless, if not detrimental, for nodes close to the gateway when RA is enabled.

Finally, we observe that the energy efficiency of HYB is the worst, being affected by both the higher failure probability observed in Fig. 3.2 and the

[2] Atmel AT86RF212B, Texas Instruments CC1125 and CC1310, and Semtech SX1272 modules.

higher energy consumption due to the transmission of resource messages and the reception of beacons. This inefficiency is more marked for nodes near the receiver, where the energy spent on control messages is actually greater than that used for the high-rate transmissions of small data packets.

3.4.4 Coexistence issues

Another important question regards the coexistence in the same area of nodes using LBT and ALOHA access schemes.

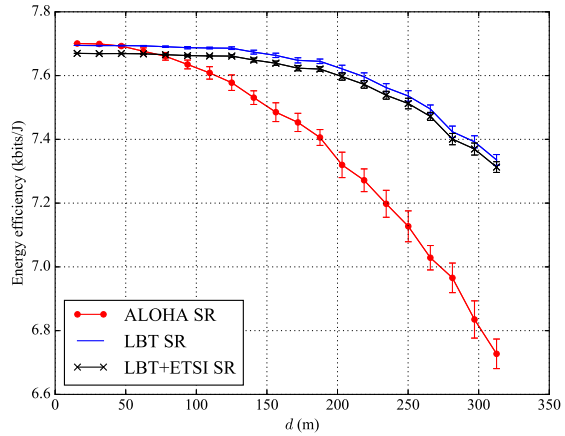
Fig. 3.4 and Fig. 3.5 report the throughput of the two access methods, defined as the overall rate of successful packet transmissions, and the energy efficiency. Curves for ALOHA (respectively LBT) have been obtained by fixing the spatial density of this type of nodes to 0.001 nodes/m^2 and increasing the spatial density of LBT (respectively ALOHA) nodes from 10^{-5} to 10^{-2} nodes/m^2.

Results in Fig. 3.4 show that the performance of ALOHA nodes is not impacted by an increase in the number of LBT nodes, while the latter suffer strong performance degradation due to the CCA mechanism that aborts a transmission attempt when the channel is sensed busy for a given number of successive attempts. We can also see that the use of multiple transmission rates can only slightly alleviate the problem, but the fragility of the LBT mechanism in presence of ALOHA traffic still remains. Similar observations can be drawn for the energy efficiency results. In both cases, the use of RA improves the energy efficiency quite significantly.

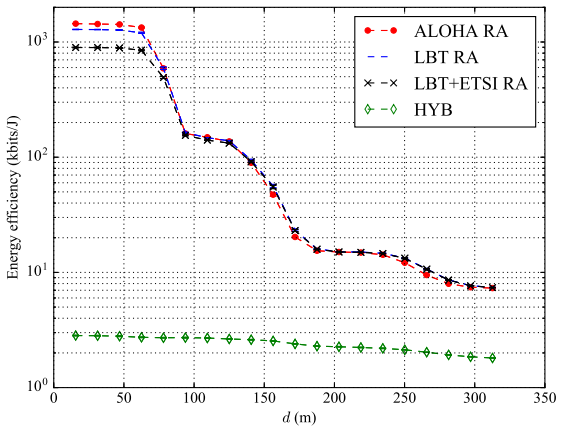
3.5 Conclusions

This chapter presented an overview of the three main uncoordinated channel access schemes, namely pure ALOHA, HYB, and LBT, in an IoT scenario. The performance of these schemes has been compared in terms of probability of successful transmission and energy efficiency, by considering the duty-cycle limitation for ALOHA, the control packets for HYB, and the CCA procedure for LBT as mandated by the international regulation frameworks.

From this analysis, it appears clear that adding rate adaptation capabilities is pivotal to maintain reasonable level of performance when the coverage range and the cell load increase. Moreover, we observed that LBT generally yields lower transmission failure probability, though packet dropping events may occur because the channel is sensed busy for a certain number of consecutive CCA attempts. This impacts on the actual energy efficiency of the LBT access scheme, which may turn out to be even smaller than that achieved by ALOHA schemes. Furthermore, we also observed that LBT performance undergoes severe degradation when increasing the number of ALOHA devices in the same cell, again because of the channel-blockage effect caused by the other transmitters. Finally, the HYB scheme proves ineffective in the considered scenario, since the reservation channel becomes the system bottleneck with short data packets. Nonetheless, hybrid solutions that adopt LBT for peripheral nodes and ALOHA for nodes closer to the receiver, or apply rate adaptation also to the reservation phase, can potentially lead to a general performance improvement of the system. In particular, the latter approach is going to be analysed in Chap. 5.



(a) Single rate (SR) case.



(b) Rate adaptive (RA) case.

Figure 3.3: Successfully received bits per unit of consumed energy, with 95% confidence intervals.

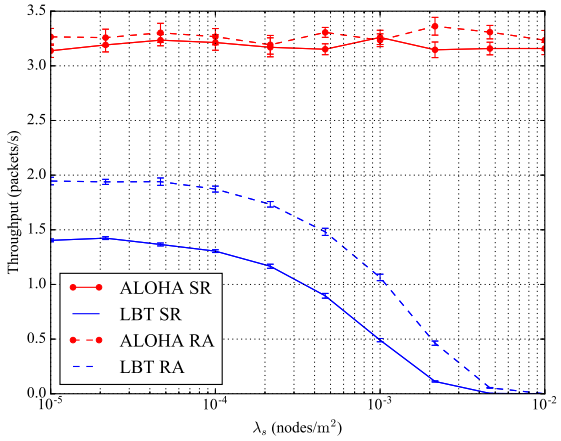


Figure 3.4: Aggregated throughput for each channel access method in the single and adaptive rate scenarios, with 95% confidence intervals.

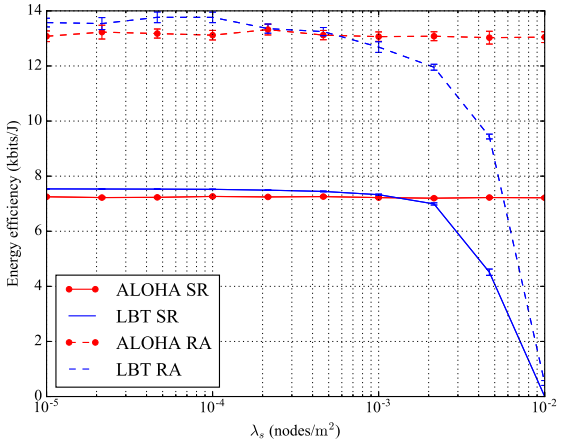


Figure 3.5: Successfully received bits per unit of consumed energy for each channel access method, in the single and adaptive rate scenarios, with 95% confidence intervals.

Chapter 4

Optimal parameter selection for ALOHA networks

This chapter focuses on ad hoc sensor networks where nodes communicate in pairs generating mutual interference, and tackles the fundamental yet still open question of how to set the transmission rate so as to maximise throughput. Elaborating on a previous stochastic geometry model, a SINR decoding threshold that depends on the transmission rate is considered, and the interplay between interference and noise power is analysed to find the optimal transmission rate. The model is then extended to include techniques, like rate adaptation or data compression, that may cause the time on air of each transmission to be different. A random component is therefore included in the time on air of the transmissions and the success probability formula in such cases is provided.

4.1 Introduction

The performance of ALOHA networks is well understood under the collision channel model, assuming that the concurrent transmission of two or more nodes results in the loss of all packets. In this scenario, taking the lead from classical throughput and delay results for fixed-length messages [86, 87], ensuing studies concentrated on systems with heterogeneous packet lengths, obtaining the delay distribution [88, 89] and bounds on the throughput [90, 91]. In particular, it has been shown that the best throughput is obtained when messages have a constant transmission time [90, 92].

Departing from the collision channel, the role of power unbalance and capture effect was clarified in [93–98]. Under the assumption that an incoming signal can be successfully decoded even in the presence of interferers provided its power is sufficiently larger than that of its contenders, remarkable improvements in terms of throughput and stability region of the protocol were shown for fixed message length. Similar trends were later derived also in unslotted systems with variable packet size [99].

A more realistic channel model accounting for path-loss and fading was fi-

nally considered in [100]. Here, the authors develop a stochastic geometry framework to analyse the performance of ALOHA *ad hoc* networks, i.e. systems in which nodes communicate in pairs rather than sending data towards a common receiver. Assuming a correct reception to take place when the SINR of the considered packet is above a fixed threshold, closed-form expressions for the success probability were derived.

While offering a powerful and flexible framework, [100] does not explore optimisation of network parameters, triggering some questions of interest. In particular, a non-trivial tradeoff arises when considering how to set the transmission rate. Indeed, for a constant payload size, higher bitrates lead to shorter transmission times and thus to a lower collision probability. On the other hand, though, increasing the rate also reduces the coding resiliency to interference, raising the SINR level required for successful decoding.

Some insights on the issue were presented in [101], considering allocation of orthogonal sub-bands to users to obtain the largest spectral efficiency. In [102], a similar approach was followed in a *slotted* ALOHA setup to optimise the network parameters for the no-fading case in the absence of noise. Derived results reveal how the amount of interference largely influences the optimal rate or bandwidth to be used. From a practical design standpoint, however, simple and robust rate-adaptation policies shall be devised, as network conditions in terms of congestion may not be known accurately. Characterising how well the configuration striking the optimal balance performs in a wide range of scenarios can thus offer relevant design hints that, to the best of our knowledge, have not been derived for asynchronous random access protocols.

This work tackles the question extending the analysis in [100]. Following a stochastic geometry approach, the SINR experienced at a receiver is related to node/traffic density, path loss, fading, and packet transmission time. The impact of the transmission bitrate on the decoding threshold is captured through the Shannon capacity law, though the framework can embed any other model. This study shows that the optimal rate depends on system parameters in a complex manner. However, a handy closed-form expression can be obtained focusing on the interference-limited region. The simplified formulation only requires knowledge of the path-loss exponent and of the transmission bandwidth, and is thus easily implementable in distributed and heterogeneous sensor networks. Moreover, numerical results show how its application does not lead to significant performance losses even in noise-limited scenarios, broadening the applicability of the derived design hints. Furthermore, an extension of the framework is provided for networks where the time on air of messages are different. This extended model is applied to different scenarios, to obtain hints on how to enhance the performance of heterogeneous networks.

4.2 System model

Following the *Poisson rain* model proposed in [100], the behaviour of an unslotted ALOHA network is captured by means of a homogeneous space-time Poisson Point Process (PPP) $\Psi = \{(X_i, t_i)\}$ of rate λ_s [packets/s/m^2]. Accordingly, a transmitter is created at time t_i and position $X_i \in \mathbb{R}^2$, occupies the channel for the duration of a single packet transmission, and then disappears. The generated message is sent to a target receiver, uniformly placed over a cir-

cle of radius r centered at the transmitter.[1] All messages have payload size L, and are sent at a bitrate $R = L/B$, where B is the on-air packet transmission time. For the sake of simplicity, we can neglect the transmission of feedback from the receivers. Furthermore, we assume that packets are not retransmitted or, equivalently, that retransmissions occur after random backoff times such that the aggregate packet arrival rate, inclusive of both new transmissions and retransmissions, is the PPP considered in the analysis.

Nodes employ a constant transmission power $\mathrm{P_{tx}}$, and wireless propagation is affected by a path-loss component $\ell(d) = (Ad)^{\beta}$, where d is the distance from the transmitter, $\beta > 2$ is the path loss coefficient, and $A > 0$ is a constant that depends on antennas gain and transmission frequency.[2] Moreover, the effect of block Rayleigh fading is captured as an exponential power factor F of unit mean. Without loss of generality, we focus on the typical receiver, i.e., a node located at the origin of the plane, whose reception starts at $t = 0$ and ends at $t = B$. Defining $I = \frac{1}{B}\int_0^B i(t)dt$ as the average interference experienced during the packet reception, we define the SINR as

$$\gamma = \frac{\mathrm{P_{tx}} F}{\ell(r)(N_s + I)}\,, \tag{4.1}$$

where N_s is the noise power. We assume the use of a capacity-achieving channel code, so that decoding at the typical receiver is successful if the SINR is above a threshold Γ°, which depends on the transmission rate as

$$\Gamma^{\circ} = 2^{R/B_{\mathrm{w}}} - 1\,, \tag{4.2}$$

where B_{w} is the transmission bandwidth. We observe that the framework can accommodate any other rate-SINR threshold model. Models that preserve the geometric dependence between Γ° and R will yield qualitatively similar results to those obtained with (4.2).

4.3 Stochastic geometry framework

In this section, a general expression for the success probability of a transmission is obtained by leveraging a stochastic geometry framework. This result will also be used in the next chapter, where a more general scenario is considered.

4.3.1 Interference characterization

As we said earlier, the success probability is defined as the probability that the SINR is higher than a threshold Γ°. Therefore, from the CDF of the exponential distribution, we have

$$p_s = \mathrm{E}\left[e^{-\Gamma^{\circ}\ell(r)(I+N_s)/\mathrm{P_{tx}}}\right] = \mathcal{L}_N(\Gamma^{\circ}\ell(r)/\mathrm{P_{tx}})\mathcal{L}_I(\Gamma^{\circ}\ell(r)/\mathrm{P_{tx}}) \tag{4.3}$$

[1] Despite its simplicity, the model is apt to describe networks with sporadic traffic or with high mobility, and has been shown to offer a good approximation of more involved descriptions accounting for static node positions and retransmissions [100].

[2] For physical reasons, one shall set $r > 1/A$. Alternatively, the channel gain can be set equal to $\max(1, \ell(r))$. Nonetheless, for mathematical tractability, we ignore this correction to the path loss component, as customary in stochastic geometry approaches. Additional investigation showed that this approximation has negligible impact on the results.

where the expectation is taken over I and N_s. $\mathcal{L}_I(s)$ and $\mathcal{L}_N(s)$ are, respectively, the Laplace transforms of the interference and the noise.

We now focus on the Laplace transform of the interference, using the *average interference constraint* approach presented in [100, Section III.B]. We can expand the term related to the instantaneous interference $i(t)$ experienced during the packet reception, obtaining

$$I = \frac{1}{B} \int_0^B \sum_{X_j \in \Psi^1(t),\, j \neq 0} \mathrm{P_{tx}} F_j / \ell(|X_j|) \mathrm{d}t. \tag{4.4}$$

Changing the order of integration and summation, and recalling the Slivnyak's theorem, which states that the law of $\Phi \setminus X$ conditional on the fact that Φ has a point at X is the same as the law of Φ [103, 104], we can write [100]

$$I = \sum_{(X_j, T_j) \in \Psi} \mathrm{P_{tx}} F_j h(T_j) / \ell(|X_j|) \tag{4.5}$$

where

$$h(s) = \int_0^B \frac{\mathbf{1}(s \leq t < s + B)}{B} \mathrm{d}t = \frac{\max(B - |s|, 0)}{B}. \tag{4.6}$$

4.3.2 Campbell's theorem for marked processes

We now review some results from the literature, which we will build on in the next sections.

Theorem 5 (Campbell's theorem [105, 106])**.** *Let Φ be a Poisson process on S with intensity $\lambda(x)$, and let $f : S \to \mathbb{R}$ be measurable. Then the sum*

$$\Sigma = \sum_{X \in \Phi} f(X) \tag{4.7}$$

is absolutely convergent with probability 1 if and only if

$$\int_S \min(|f(x)|, 1) \lambda(x) \mathrm{d}x < \infty \tag{4.8}$$

If this condition holds, then

$$\mathrm{E}\left[e^{s\Sigma}\right] = \exp\left\{ \int_S \left(e^{sf(x)} - 1 \right) \lambda(x) \mathrm{d}x \right\} \tag{4.9}$$

for any complex s for which the integral on the right converges, and, in particular, when s is pure imaginary.

If Φ is uniform of intensity λ, condition (4.8) reduces to

$$\int_S \min(|f(x)|, 1) \mathrm{d}x < \infty \tag{4.10}$$

and equation (4.9) becomes

$$\mathrm{E}\left[e^{s\Sigma}\right] = \exp\left\{ \lambda \int_S \left(e^{sf(x)} - 1 \right) \mathrm{d}x \right\} \tag{4.11}$$

We can associate to each point X of the random set Φ a random variable m_X (the *mark* of X) taking values in some space M. The distribution of m_X may depend on X but not on the other points of Φ, and the m_X for different X have to be independent. Because of the *Marking theorem* [105], the random countable subset $\Phi^* = \{(X, m_X); X \in \Phi\}$ of $S \times M$ is a Poisson process [105, Section 5.2].

This allows us to generalize the Campbell's theorem to the case of marked processes. In particular, we can consider the marked Poisson process Φ^* as described before and the sum $\Sigma^* = \sum_{X \in \Phi} f(X, m_X)$. Under a convergence condition analogous to (4.8), we can rewrite (4.9) considering the marked process as a non-marked Poisson process on $S \times M$:

$$\mathrm{E}\left[e^{s\Sigma^*}\right] = \exp\left\{\int_S \int_M (e^{sf(x,m)} - 1) p(m|x)\lambda(x)\mathrm{d}m\mathrm{d}x\right\} \tag{4.12}$$

$$= \exp\left\{\int_S \left(\mathrm{E}_M\left[e^{sf(x,m)}\right] - 1\right)\lambda(x)\mathrm{d}x\right\}, \tag{4.13}$$

where $p(m|x)$ is the conditional distribution of the mark m_X. Again, if Φ^* is uniform of intensity λ^*, (4.12) becomes

$$\mathrm{E}\left[e^{s\Sigma^*}\right] = \exp\left\{\lambda^* \int_S \int_M \left(e^{sf(x,m)} - 1\right) \mathrm{d}m\,\mathrm{d}x\right\}. \tag{4.14}$$

4.3.3 Solving the Laplace transform of the interference

We now want to find an explicit formula for the Laplace transform of the interference $\mathcal{L}_I(s)$. Note that the PPP of the interferers can be seen as a marked process, with the transmission start time and the fading as the marks.

From equations (4.14) (for the time marks) and (4.13) (for the fading marks), the Laplace transform of the interference I is

$$\mathcal{L}_I(s) = \exp\left\{-\lambda_s \int_{\mathbb{R}^2} \int_{-\infty}^{+\infty} \left(1 - \mathrm{E}_F\left[e^{-s\mathrm{P}_{\mathrm{tx}}Fh(t)/\ell(|x|)}\right]\right) \mathrm{d}t\,\mathrm{d}x\right\} \tag{4.15}$$

By applying the definition of expectation in Eq. (4.15) and transforming the integral in x using polar coordinates we have

$$\mathcal{L}_I(s) = \exp\left\{-2\pi\lambda_s \int_0^{\infty} \int_{-\infty}^{+\infty} \left(1 - \frac{1}{1 + s\mathrm{P}_{\mathrm{tx}}h(t)/\ell(u)}\right) \mathrm{d}t\,u\,\mathrm{d}u\right\}. \tag{4.16}$$

Let's solve the integral on t (remembering that $h(t)$ is an even function and that $h(t) = 0$ for $|t| > B$):

$$\begin{aligned}\int_{-\infty}^{+\infty} \left(1 - \frac{1}{1 + s\mathrm{P}_{\mathrm{tx}}h(t)/\ell(u)}\right) \mathrm{d}t &= 2B - 2\int_0^B \frac{1}{1 + s\mathrm{P}_{\mathrm{tx}}h(t)/\ell(u)}\mathrm{d}t \\ &= 2B - 2\frac{\ell(u)B}{s\mathrm{P}_{\mathrm{tx}}} \log\left(1 + \frac{s\mathrm{P}_{\mathrm{tx}}}{\ell(u)}\right).\end{aligned}$$

Putting that back in the formula for $\mathcal{L}_I(s)$, we get

$$\mathcal{L}_I(s) = \exp\left\{-4\pi\lambda_s B \int_0^{\infty} \left[1 - \frac{\ell(u)}{s\mathrm{P}_{\mathrm{tx}}} \log\left(1 + \frac{s\mathrm{P}_{\mathrm{tx}}}{\ell(u)}\right)\right] u\,\mathrm{d}u\right\}, \tag{4.17}$$

as reported in [100, Fact A.4].

4.4 Analysis for homogeneous time-on-air

As previously derived, for the considered system model, the probability of successful packet reception can be expressed as

$$p_s = \mathcal{L}_N\left(\Gamma^\circ \ell(r)/\mathrm{P_{tx}}\right) \mathcal{L}_I\left(\Gamma^\circ \ell(r)/\mathrm{P_{tx}}\right) , \tag{4.18}$$

where $\mathcal{L}_N(s)$ and $\mathcal{L}_I(s)$ are the Laplace transforms of the noise and interference, respectively. The former readily follows from the definition, while the latter is computed from (4.17) with a change of variable, obtaining

$$\begin{aligned} \mathcal{L}_N\left(\Gamma^\circ \ell(r)/\mathrm{P_{tx}}\right) &= \exp\left(-\frac{\Gamma^\circ \ell(r) N_s}{\mathrm{P_{tx}}}\right) ; \\ \mathcal{L}_I\left(\Gamma^\circ \ell(r)/\mathrm{P_{tx}}\right) &= \exp\left(-\lambda_s \frac{L}{R} r^2 \left(\Gamma^\circ\right)^{2/\beta} K'(\beta)\right) , \end{aligned} \tag{4.19}$$

where $K'(\beta) = \frac{4\pi}{\beta} \int_0^\infty u^{2/\beta - 1}(1 - u\log(1+u^{-1}))\,\mathrm{d}u$.

In the following, starting from the expression of p_s, we first derive the general expression of the optimal bitrate considering both the interference and noise contributions. Secondly, we focus on a setting of practical interest in which interference becomes dominant (asymptotic interference-limited region - AI) and get a handy closed-form result.

4.4.1 General case

When combining the noise and interference contributions, p_s follows from (4.18)-(4.19), leading to a network throughput density $S(R) = \lambda_s L\, p_s[\mathrm{bit/s/m^2}]$, where the transmission bitrate R affects p_s through the threshold Γ°, as for (4.2). The optimal bitrate can thus be derived by fixing the other network parameters and solving the maximisation problem

$$R^* = \underset{R>0}{\operatorname{argmax}}\, S(R) . \tag{4.20}$$

Since λ_s and L are constant and the exponential function is monotonically increasing, we can reformulate (4.20) as

$$R^* = \underset{R>0}{\operatorname{argmax}} \left[-\lambda_s \frac{L}{R} r^2 (2^{R/B_\mathrm{w}} - 1)^{2/\beta} K'(\beta) - \frac{2^{R/B_\mathrm{w}} - 1}{\mathrm{P_{tx}}} (Ar)^\beta N_s\right] . \tag{4.21}$$

For ease of writing, let $\alpha_1 = -r^2 K'(\beta) L$ and $\alpha_2 = -(Ar)^\beta \mathrm{P_{tx}}^{-1} N_s$ so that the objective is to maximise the function

$$f(R) = \alpha_1 \lambda_s R^{-1} \left(2^{R/B_\mathrm{w}} - 1\right)^{2/\beta} + \alpha_2 \left(2^{R/B_\mathrm{w}} - 1\right) . \tag{4.22}$$

The optimal rate R^* can be found by setting to zero the derivative of f with respect to R, obtaining

$$\begin{aligned} -\alpha_1 \lambda_s + \frac{\alpha_1 \lambda_s \log(4)}{\beta B_\mathrm{w}} R^* \left(2^{R^*/B_\mathrm{w}} - 1\right)^{-1} 2^{R^*/B_\mathrm{w}} + \\ \frac{\alpha_2}{B_\mathrm{w}} \log(2) R^{*2} \left(2^{R^*/B_\mathrm{w}} - 1\right)^{-2/\beta} 2^{R^*/B_\mathrm{w}} = 0 . \end{aligned} \tag{4.23}$$

Parameter		Value
Transmission power	$\mathrm{P_{tx}}$	14 dBm
Bandwidth	B_w	125 kHz
Path loss coefficient	A	36.38 m^{-1}
Transmitter-receiver distance	r	20 m
Packet length	L	200 bit
Noise spectral density	N_0	$2 \cdot 10^{-20}$ W/Hz
Noise power	N_s	$N_0 B_\mathrm{w}$

Table 4.1: Scenario parameters

Unfortunately, there is no closed-form solution to this transcendental equation. However, we can prove in the following that there exists a unique point $R^* > 0$ that satisfies (4.23), which is the absolute maximum of (4.22) and can be found through bisectional search.

Existence of a unique point satisfying (4.23). Setting $x^* = R^*/B_\mathrm{w}$, we can rewrite (4.23) in the form $F(x^*) = 1$, where $F(x) = \log(4)g(x)/\beta + Bh(x)$, $B > 0$, with $g(x) = \frac{x2^x}{2^x-1}$; $h(x) = \frac{x^2 2^x}{(2^x-1)^\gamma}$, and $\gamma = 2/\beta \in (0,1)$. Since the derivatives of $g(x)$ and $h(x)$ are both positive for $x \geq 0$, then $F(x)$ is monotonic increasing in x, and given that $F(0) < 1$, then there exists a unique $x^* > 0$ such that $F(x^*) = 1$.

□

4.4.2 Asymptotic interference-limited region

When the packet generation density is high, the success probability is mainly determined by the interference, which dominates the noise power. In this region, the noise term can then be neglected and the corresponding coefficient α_2 in (4.23) vanishes. Thus, a simplified expression for the optimal transmission rate in asymptotic interference-limited (AI) conditions follows:

$$R^*_\mathrm{AI} = B_\mathrm{w} \frac{2\mathcal{W}\left(-2^{-1-\beta/\log(4)}\beta\right) + \beta}{\log(4)}, \tag{4.24}$$

where $\mathcal{W}(\cdot)$ is the Lambert W function, such that $\mathcal{W}(x)e^{\mathcal{W}(x)} = x$.

It is worth noting that R^*_AI depends only on the bandwidth B_w and the path-loss exponent β, while it is independent of the packet generation rate (provided that λ_s is sufficiently high to be in AI conditions), the packet size, and even the transmitter-receiver distance. As we will see in the next section, while the throughput depends on these parameters, the optimal transmission rate does not need to be adapted to the receiver distance or to the level of interference.

4.5 Results for homogeneous time-on-air

The results in this section have been obtained by setting the parameters as in Tab. 4.1, representative for typical LPWA network deployments.

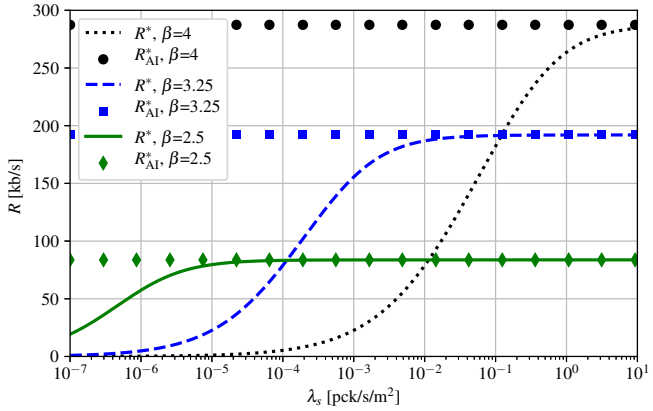


Figure 4.1: Optimal rate in the noise-limited and asymptotic interference-limited regions, for different values of β.

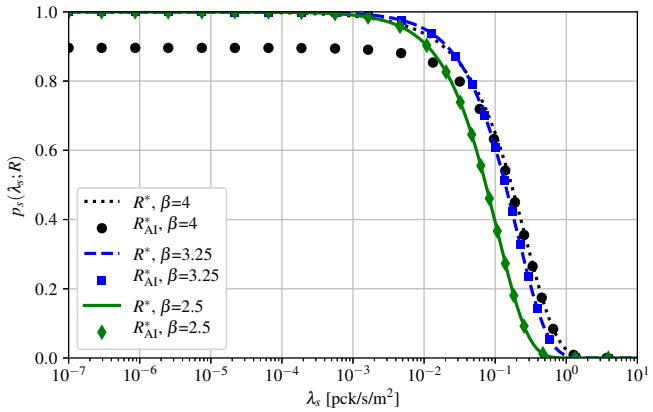


Figure 4.2: Success probability using the optimal rate (R^*) and the asymptotic optimal rate (R^*_{AI}), for different values of β.

Fig. 4.1 shows the optimal transmission rate when varying λ_s, both for the general case (lines) and the AI regime (markers). Different lines have been obtained by changing the value of the path loss coefficient β. We can observe that, before entering the AI region, the optimal transmission rate grows with the offered traffic rate λ_s, until it hits the optimal values for the AI conditions. In fact, when channel impairments play a key role for packet decoding, low bitrates pay off by offering greater resilience to fading and noise. Conversely, when interference is the main cause for losses, high bitrates tend to perform better thanks to the lower amount of interference on the channel brought by shorter transmission times. From a practical standpoint, then, an ideal rate adaptation mechanism should be able to determine the overall level of interference with respect to thermal noise to decide whether to increase or decrease its transmission bitrate.

Even under this assumption, though, the optimal rate in the general case

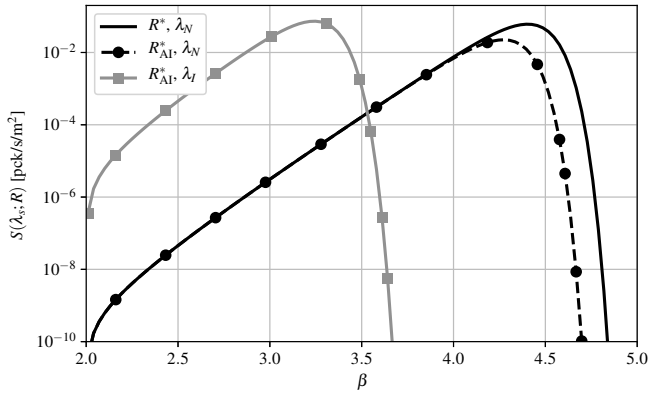


Figure 4.3: Throughput using R^* and R^*_{AI} in the noise-limited (NL) and AI regions.

depends on many parameters – e.g. the traffic density – which may be difficult to estimate in practical and distributed network deployments. On the other hand, the simple and closed-form rate expression derived for the AI region depends only on the bandwidth and on the path-loss exponent β, which can be easily estimated or fixed to the reference value for the considered scenarios (e.g. 3.5 in urban settings). The relevant question of whether setting the transmission rate to R^*_{AI} offers good performance also in the noise-limited regime thus naturally arises. To better appreciate the impact of such an approximation, the success probability that can be obtained by using the exact optimal rate R^* and the AI optimal rate R^*_{AI} are reported in Fig. 4.2. We can see that, for most values of β, no appreciable difference emerges when using R^*_{AI} in place of R^*, while a wider gap is noticed for stronger path-loss exponents, e.g. $\beta = 4$.

To better study this effect, consider a parameter $\lambda^{\mathrm{th}}(\beta)$, which is defined as the packet generation rate for which the optimal rate computed in the noise-limited (NL) region is equal to 90% of the optimal rate in the AI region, i.e., $R^* = 0.9\, R^*_{\mathrm{AI}}$. Therefore, $\lambda^{\mathrm{th}}(\beta)$ can be seen as a threshold that separates the NL- from the AI-region. An in-depth inspection of the results, not reported due to space constraints, reveals that $\lambda^{\mathrm{th}}(\beta)$ has an exponentially increasing trend in β. Thus, for a given offered traffic, a small increase in β may be sufficient to move from AI into NL conditions. We hence select two packet generation rates, namely $\lambda_I(\beta) = \lambda^{\mathrm{th}}(\beta) \cdot 100$ and $\lambda_N(\beta) = \lambda^{\mathrm{th}}(\beta)/100$, which are firmly in the AI and NL region, respectively, and evaluate the system throughput, varying β, both for R^* and R^*_{AI}. We can see in Fig. 4.3 that the use of R^*_{AI} in place of R^* in the NL region causes negligible performance loss if β is less than 4 (as typical in most practical cases), while for larger values of β the penalty is more consistent. To maximise the network throughput in a scenario with a not very large path-loss exponent, then, the devices can avoid to estimate many channel parameters (like transmitter-receiver distance, packet generation density, etc.), since they can simply use the AI optimal rate, which depends only on the path-loss exponent β and on the modulation bandwidth.

As a side note, observe that the packet length L and the process intensity λ_s appear always together in the form $\lambda_s L$ in the throughput formula, which

has been derived in Sec. 4.4. Therefore, neglecting the overhead due to the replication of the header field, the throughput of the nodes is not affected by packet fragmentation, which would decrease the packet size but increase the transmitted number of packets, keeping the product $\lambda_S L$ constant. Considering the overhead due to the fragmentation, and the fragility of the reassembly procedure (which fails if either a single segment is lost), fragmentation does not appear as beneficial in the considered scenario. However, it is important to remark that, in the considered setting, all nodes are at the same distance from their receiver, and use the same bitrate: in a more heterogeneous setting the optimal packet size might vary for the different nodes.

t_1	t_2	$h(t,b)$ for $t_1 < t < t_2$
$-\infty$	$-b$	0
$-b$	$\tau_1 = \min\{0, B-b\}$	$(b+t)/B$
$\tau_1 = \min\{0, B-b\}$	$\tau_2 = \max\{0, B-b\}$	$\min\{b, B\}/B$
$\tau_2 = \max\{0, B-b\}$	B	$(B-t)/B$
B	$+\infty$	0

Table 4.2: Value of $h(t,b)$ for different intervals in t.

4.6 Using different packet transmission times for different nodes

In the following, we consider each transmitter-receiver couple to use an independent and identically distributed (i.i.d.) transmission time b, independent of everything else, with pdf $f_b(b)$. The transmission time for the desired transmitter is denoted by B and is considered known and fixed. Therefore, b can be considered as a mark of the process Ψ presented in Sec. 4.3, which means that, to calculate the probability of transmission success p_s, we have to take the expectation with respect to b inside equation (4.16), as per equation (4.13):

$$\mathcal{L}_I(s) = \exp\left\{-2\pi\lambda_s \int_0^{\infty}\int_{-\infty}^{+\infty}\left(1 - \mathrm{E}_b\left[\frac{1}{1+s\mathrm{P}_{\mathrm{tx}}h(t,b)/\ell(u)}\right]\right) u\,\mathrm{d}t\,\mathrm{d}u\right\} \tag{4.25}$$

where

$$h(t,b) = \frac{1}{B}\max\left\{\min\left\{b + \frac{t-|t|}{2}, B - \frac{t+|t|}{2}\right\}, 0\right\}. \tag{4.26}$$

Let's rewrite $\mathcal{L}_I(s)$ changing the order of the expectation and the integral with respect to t:

$$\mathcal{L}_I(s) = \exp\left\{-2\pi\lambda_s \int_0^{\infty}\int_0^{\infty}\int_{-\infty}^{+\infty}\left(1 - \frac{1}{1+s\mathrm{P}_{\mathrm{tx}}h(t,b)/\ell(u)}\right)\mathrm{d}t\, f_b(b)\,\mathrm{d}b\, u\,\mathrm{d}u\right\}. \tag{4.27}$$

We now solve the integral with respect to t. First, we divide the interval $(-\infty, +\infty)$ in five intervals, each having a continuous behavior of $h(t,b)$, as described in Tab. 4.2. The integral in t thus becomes

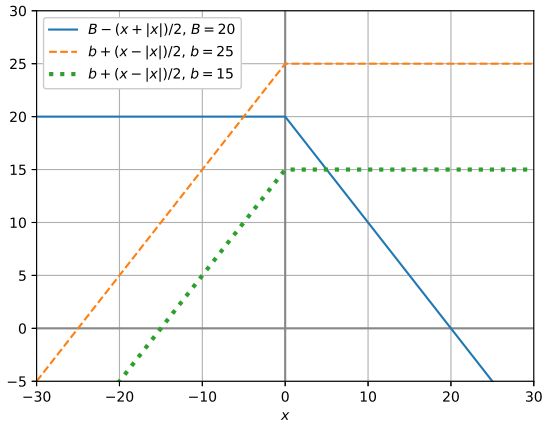


Figure 4.4: Components of $h(x,b)$: $B-(x+|x|)/2$ and $b+(x-|x|)/2$.

$$
\begin{aligned}
&\int_{-\infty}^{+\infty}\left(1-\frac{1}{1+s\mathrm{P}_{\mathrm{tx}}h(t,b)/\ell(u)}\right)\mathrm{d}t\\
&=\int_{-b}^{\tau_1}\left(1-\frac{1}{1+s\mathrm{P}_{\mathrm{tx}}(b+t)/(\ell(u)B)}\right)\mathrm{d}t\\
&+\int_{\tau_1}^{\tau_2}\left(1-\frac{1}{1+s\mathrm{P}_{\mathrm{tx}}\min\{b,B\}/(\ell(u)B)}\right)\mathrm{d}t\\
&+\int_{\tau_2}^{B}\left(1-\frac{1}{1+s\mathrm{P}_{\mathrm{tx}}(B-t)/(\ell(u)B)}\right)\mathrm{d}t\\
&=B+b-\frac{\ell(u)B}{s\mathrm{P}_{\mathrm{tx}}}\left[\log\left(1+\frac{s\mathrm{P}_{\mathrm{tx}}(b+\tau_1)}{\ell(u)B}\right)+\log\left(1+\frac{s\mathrm{P}_{\mathrm{tx}}(B-\tau_2)}{\ell(u)B}\right)\right]\\
&-\frac{\tau_2-\tau_1}{1+s\mathrm{P}_{\mathrm{tx}}\min\{b,B\}/(\ell(u)B)}\\
&=B+b-\frac{\chi(u)}{s}\log\left(\frac{(\chi(u)+s(b+\tau_1))(\chi(u)+s(B-\tau_2))}{\chi(u)^2}\right)\\
&-\frac{\tau_2-\tau_1}{1+s\min\{b,B\}/\chi(u)},
\end{aligned}
$$

where we defined $\chi(u)=\ell(u)B/\mathrm{P}_{\mathrm{tx}}$. Putting this result back into equation (4.27) we have

$$
\begin{aligned}
\mathcal{L}_I(s)=\exp\Bigg\{-2\pi\lambda_s\int_0^\infty\int_0^\infty\Bigg[&B+b\\
&-\frac{\chi(u)}{s}\log\left(\frac{(\chi(u)+s(b+\tau_1))(\chi(u)+s(B-\tau_2))}{\chi(u)^2}\right)\\
&-\frac{\tau_2-\tau_1}{1+s\min\{b,B\}/\chi(u)}\Bigg]f_b(b)\,\mathrm{d}b\,u\,\mathrm{d}u\Bigg\}
\end{aligned}
\tag{4.28}
$$

Splitting the integral on b in two parts, we can simplify τ_1 and τ_2 as indicated

b_1	b_2	τ_1 for $b_1 \leq b \leq b_2$	τ_2 for $b_1 \leq b \leq b_2$
0	B	0	$B-b$
B	$+\infty$	$B-b$	0

Table 4.3: Value of τ_1 and τ_2 for different intervals of b.

in Tab. 4.3. The Laplace transform of I is then

$$\begin{aligned}
\mathcal{L}_I(s) = \exp\Bigg\{ & -2\pi\lambda_s \int_0^\infty \Bigg\{ B \\
& + \int_0^B \left[b - \frac{\chi(u)}{s} \log\left(\frac{(\chi(u)+sb)(\chi(u)+sb)}{\chi(u)^2} \right) - \frac{B-b}{1+sb/\chi(u)} \right] f_b(b)\,\mathrm{d}b \\
& + \int_B^\infty \left[b - \frac{\chi(u)}{s} \log\left(\frac{(\chi(u)+sB)(\chi(u)+sB)}{\chi(u)^2} \right) \right. \\
& \left. - \frac{b-B}{1+sB/\chi(u)} \right] f_b(b)\,\mathrm{d}b \Bigg\} u\,\mathrm{d}u \Bigg\} \\
= \exp\Bigg\{ & -\lambda_s 2\pi \int_0^\infty \Bigg\{ B \\
& + \int_0^B \left[-2\frac{\chi(u)}{s} \log\left(\frac{\chi(u)+sb}{\chi(u)} \right) + \frac{sb^2/\chi(u)+2b-B}{1+sb/\chi(u)} \right] f_b(b)\,\mathrm{d}b \\
& - 2\frac{\chi(u)}{s} \log\left(\frac{\chi(u)+sB}{\chi(u)} \right) \int_B^\infty f_b(b)\,\mathrm{d}b \\
& + B\frac{\int_B^\infty (1+sb/\chi(u)) f_b(b)\,\mathrm{d}b}{1+sB/\chi(u)} \Bigg\} u\,\mathrm{d}u \Bigg\}
\end{aligned} \tag{4.29}$$

Remembering that, by ignoring the noise component, $p_s = \mathcal{L}_I(\Gamma^\circ \ell(r)/\mathrm{P_{tx}})$ (Eq. (4.3)), we can find the success probability of the transmission, given r (the distance between the useful transmitter and receiver) and the useful transmission length B.

Remark 2. In case the area where the interferers are distributed is finite, we have to change the integration intervals for the integral in v in Eq. (4.32). Defining d_{MIN}, d_{MAX} the minimum and maximum distance, respectively, at which interferers are located, the integration interval becomes:

$$\left[\left(\frac{d_{\mathrm{MIN}}}{r} \right)^\beta \frac{B}{T}, \left(\frac{d_{\mathrm{MAX}}}{r} \right)^\beta \frac{B}{T} \right] .$$

This result is trivially derived by changing the integration interval in Eq. (4.16) and following the same steps of the infinite area case.

4.6.1 Restricting to a specific loss function

If we consider the path-loss function to be

$$\ell(r) = (Ar)^\beta , \tag{4.30}$$

with the parameters $A > 0$ and $\beta > 2$ related to the transmission environment,[3] the success probability p_s assumes a simpler form. In particular, in Eq. (4.29), we can operate the following substitution:

$$\frac{\chi(u)}{s} = \frac{\ell(u)B/\mathrm{P_{tx}}}{\ell(r)\,\Gamma^\circ/\mathrm{P_{tx}}} = \frac{u^\beta B}{r^\beta \Gamma^\circ} = v\,. \tag{4.31}$$

The success probability is then

$$p_s = \exp\left\{ -2\pi\lambda_s \left(\frac{\Gamma^\circ}{B}\right)^{\frac{2}{\beta}} \frac{r^2}{\beta} \int_0^\infty \left\{ B + \int_0^B \left[-2v \log\left(1+\frac{b}{v}\right) + \frac{(2b-B)v+b^2}{v+b} \right] f_b(b)\,\mathrm{d}b - 2v\log\left(1+\frac{B}{v}\right) \int_B^\infty f_b(b)\,\mathrm{d}b + \frac{B}{v+B}\int_B^\infty (v+b) f_b(b)\,\mathrm{d}b \right\} v^{\frac{2}{\beta}-1}\,\mathrm{d}v \right\}. \tag{4.32}$$

4.7 Defining the message transmission time distribution

We need to define the pdf $f_b(\cdot)$ of the message transmission time b. We analyse three scenarios, described in the following sections.

4.7.1 Variable payload

In this scenario, each transmitting node is at a constant distance from its receiver and the modulation used is the same for all nodes. However, nodes can transmit packets having different payload lengths, which causes possible different transmission durations for different packets.

To calculate the success probability in this case, we start from Eq. (4.32), with the SINR threshold Γ° and the transmitter-receiver distance r fixed and given. $f_b(b)$ can then be easily extracted from the packet length pdf $f_L(l)$ leveraging the definition of rate $R = L/b$. Since L is discrete, $f_b(b)$ becomes a discrete pdf and the integral in $\mathrm{d}b$ becomes a summation.

$$p_s = \exp\left\{ -2\pi\lambda_s \left(\frac{\Gamma^\circ}{B}\right)^{\frac{2}{\beta}} \frac{r^2}{\beta} \int_0^\infty \left\{ B + \sum_{b\le B} \left[-2v \log\left(1+\frac{b}{v}\right) + \frac{(2b-B)v+b^2}{v+b} \right] f_b(b) - 2v\log\left(1+\frac{B}{v}\right) \sum_{b>B} f_b(b) + B\frac{\sum_{b>B}(v+b) f_b(b)}{v+B} \right\} v^{\frac{2}{\beta}-1}\,\mathrm{d}v \right\}. \tag{4.33}$$

[3] As noted in [100], there are other possible choices of path-loss functions that avoid the pole at $r = 0$.

4.7.2 Variable distance and adaptive modulation

In this scenario, nodes transmit packets having the same payload lengths, however the distance between a transmitter and its receiver may be different for each transmitter-receiver pair. The transmission rate is adapted based on the distance the transmission has to travel, with long distance transmission using lower rates to improve their resilience to interference, while short range transmissions can use a higher rates to lower their energy consumption.

To solve Eq. (4.32), we need to know the dependency between b and the distance r between the transmitter and its receiver. We can use two approaches:

- Consider b a function of r and the average channel noise. In this way, the transmission time does not depend on the interference and the calculation is simple. Actually, this is the only information that a device that do not cooperate with the other devices in the network can have, because it can not have real-time information about channel occupation by other devices.

- Consider b a function of the previous parameters and the interference in the network. To use this approach, we must first define a base transmission time distribution, then calculate the network interference using that f_b and calculate a new transmission time distribution using the acquired information on interference. Then, we iterate this fixed-point procedure until we converge to a final distribution, that will be the actual transmission time distribution to be used.

In the following, we will use the first approach, leaving the second approach as future work.

We define N_s the average noise experienced by a receiver, B_w the channel bandwidth, and L the message length in bits. Shannon's theorem states that the channel capacity C is

$$C = B_\mathrm{w} \log_2(1 + \overline{\Gamma}) \,, \tag{4.34}$$

where $\overline{\Gamma} = \frac{\mathrm{P_{tx}}}{(Ar)^\beta N_s}$ is the average Signal-to-Noise-Ratio (SNR). Since we consider a capacity-achieving channel code and modulation, we set the transmission rate equal to C, so we obtain

$$b = \frac{L}{B_\mathrm{w}} \left[\log_2 \left(1 + \frac{\mathrm{P_{tx}}}{N_s (Ar)^\beta} \right) \right]^{-1} \,. \tag{4.35}$$

We choose to consider the receiver position to be uniformly distributed in an annulus with inner radius r_I and outer radius r_O centered on its transmitter. Observing that $N_s(Ar)^\beta > 0$, we can find the CDF of b:

$$\begin{aligned} F_b(\alpha) = P(b \le \alpha) &= P\left(\frac{L}{B_\mathrm{w}} \left[\log_2 \left(1 + \frac{\mathrm{P_{tx}}}{N_s (Ar)^\beta} \right) \right]^{-1} \le \alpha \right) \\ &= P\left(r \le \frac{\mathrm{P_{tx}}^{1/\beta}}{A \left[N_s \left(2^{\frac{L}{B_\mathrm{w} \alpha}} - 1 \right) \right]^{1/\beta}} \right) = F_r \left(\frac{\mathrm{P_{tx}}^{1/\beta}}{A \left[N_s \left(2^{\frac{L}{B_\mathrm{w} \alpha}} - 1 \right) \right]^{1/\beta}} \right) \,, \end{aligned}$$

where $F_r(\cdot)$ is the CDF of the transmitter-receiver distance. Since

$$F_r(\gamma) = \frac{\gamma^2 - r_I^2}{r_O^2 - r_I^2} \,, \tag{4.36}$$

we have that

$$f_b(\alpha) = \frac{1}{R_O^2 - R_I^2} \frac{\mathrm{d}(\gamma^2)}{\mathrm{d}\alpha} \,. \tag{4.37}$$

Expanding $\frac{\mathrm{d}(\gamma^2)}{\mathrm{d}\alpha}$, we obtain

$$\frac{\mathrm{d}(\gamma^2)}{\mathrm{d}\alpha} = \frac{2L\log(2)}{A^2\beta B_\mathrm{w}\alpha^2} \left(\frac{\mathrm{P_{tx}}}{N_s}\right)^{2/\beta} \left(2^{L/(\alpha B_\mathrm{w})} - 1\right)^{-2/\beta-1} 2^{L/(\alpha B_w)} \,. \tag{4.38}$$

Since $0 < r_I \leq r \leq r_O$, we have that

$$0 < r_I \leq \frac{\mathrm{P_{tx}}^{1/\beta}}{A\left[N_s\left(2^{\frac{L}{B_\mathrm{w}\alpha}} - 1\right)\right]^{1/\beta}} \leq r_O$$

$$(Ar_I)^{-\beta} \geq \frac{N_s}{\mathrm{P_{tx}}}\left(2^{\frac{L}{B_\mathrm{w}\alpha}} - 1\right) \geq (Ar_O)^{-\beta}$$

$$0 < b_{\min} = \frac{L}{B_\mathrm{w}} \log_2^{-1}\left(\frac{\mathrm{P_{tx}}(Ar_I)^{-\beta}}{N_s} + 1\right) \leq \alpha$$
$$\leq \frac{L}{B_\mathrm{w}} \log_2^{-1}\left(\frac{\mathrm{P_{tx}}(Ar_O)^{-\beta}}{N_s} + 1\right) = b_{\max} \,.$$

To conclude, we have that

$$F_b(\alpha) = \begin{cases} 0 & \alpha < b_{\min} \\ \dfrac{A^{-2}\left(N_s\left(2^{\frac{L}{B_\mathrm{w}\alpha}} - 1\right)/\mathrm{P_{tx}}\right)^{-2/\beta} - r_I^2}{r_O^2 - r_I^2} & b_{\min} \leq \alpha \leq b_{\max} \\ 1 & \alpha > b_{\max} \end{cases} \tag{4.39}$$

$$f_b(\alpha) = \begin{cases} \dfrac{\frac{2L\log(2)}{A^2\beta B_\mathrm{w}\alpha^2}\left(\frac{\mathrm{P_{tx}}}{N_s}\right)^{2/\beta}\left(2^{L/(\alpha B_\mathrm{w})} - 1\right)^{-2/\beta-1} 2^{L/(\alpha B_w)}}{r_O^2 - r_I^2} & b_{\min} \leq \alpha \leq b_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{4.40}$$

4.8 Example applications of the heterogeneous time on air model

To show the flexibility of this model, we analyse the performance of two ALOHA systems, which are both a specialization of the general scenario described in Sec. 4.7.2. In the first, all transmitter-receiver pairs are at the same distance but can choose between three different transmission rates. In the second, the bitrate is adapted based on the transmitter-receiver distance. Unless otherwise stated, the value of the parameters are as given in Tab. 4.4.

Parameter		Value
Spatio-temporal transmission density	λ_s	10^{-6} pcks/s/m^2
Transmission power	$\mathrm{P_{tx}}$	14 dBm
Transmission frequency	f	868 MHz
Path loss coefficient	A	$4\pi f/c \simeq$36.38 m^{-1}
Path loss exponent	β	2.5
Transmitter-receiver distance	r	200 m
Packet length	L	200 bit
Bandwidth	W	125 kHz
Noise spectral density	N_0	$2 \cdot 10^{-20}$ W/Hz
Noise power	N_s	$N_0 W$

Table 4.4: Scenario parameters

4.8.1 Fixed distance, different bitrates

We assume that the transmission bitrate can take three possible values: R_1, R_2, and R_3, with probability p_1, p_2, and p_3, respectively. Messages have a fixed length L. The rate distribution is such that $\overline{B} = \mathrm{E}[B] = \sum_i p_i L/R_i = L/R_2$. We set $\overline{B} = 2.389 \cdot 10^{-3}$ s, $B_1 = 0.2$ ms, $B_3 - B_2 = B_2 - B_1$, and $p_1 = p_3$. Results are shown in Fig. 4.5.

We observe that using a single bitrate provides a higher peak throughput, however the additional bitrates increase the stability region, which is important, e.g., in massive access scenarios.

4.8.2 Different distances, adaptive bitrate

This scenario considers the devices able to adapt their transmission bitrate according to their distance to the receiver according to the Shannon's capacity formula. To increase resilience against fading, in the bitrate selection strategy the reference received power is chosen such that the actual received power will be higher than the considered value 95% of the time.

Fig. 4.6 shows the success probability depending on the distance from the receiver for this rate adaptation strategy against the single rate case. The rate used in this latter case is the lowest rate available for the rate adaptation strategy, so as to maintain the same coverage area.

We can note that the success probability using a single rate is decreasing in the distance, with success rates higher than in the rate adaptation case for nodes at the minimum distance. That is because nodes near the receiver in the rate adaptation case use a rate which is much higher than in the single rate case. On the contrary, the success probability at the maximum distance is higher for the rate adaptation case than for the single rate case, even if both strategies use the same bitrate. This is because, when all devices use only the lowest rate, the average level of interference in the channel is higher than in the rate adaptation case, therefore, when comparing the performance of a device using the lowest rate in both scenarios, the lowest average interference level in the rate adaptation case enables higher success probabilities. We can also see that the rate adaptation strategy is able to enforce fairness between devices at

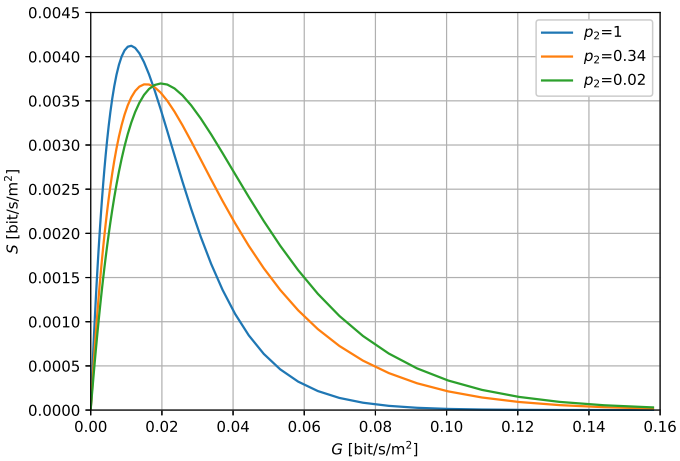


Figure 4.5: Throughput S against offered traffic G for different packet size distributions.

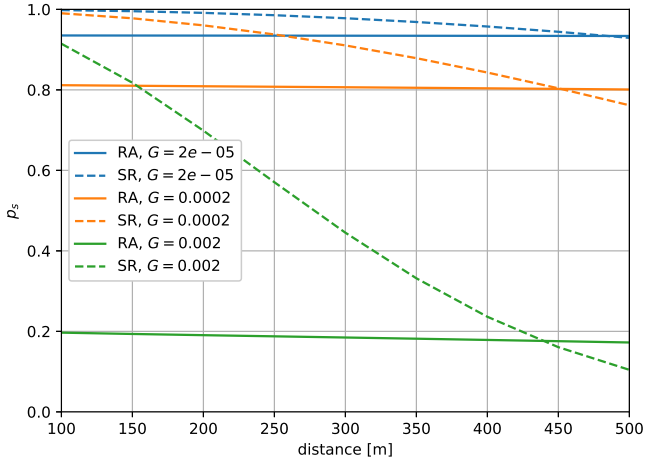


Figure 4.6: Success probability for different transmitter-receiver distances, for the single rate (SR) and rate adaptive (RA) cases.

different distances.

4.9 Conclusions

In this section, a method has been provided for the optimisation of the transmission rate with respect to the aggregate throughput in ALOHA ad hoc networks with homogeneous links. The first result was a simple expression of the optimal rate in the AI region, which depends only on the path-loss exponent β and on the bandwidth and, thus, is easy to calculate in real-world scenarios where devices may have limited knowledge on the surrounding environment. An approach to tackle the optimisation problem was also proposed for the NL region. In such conditions, however, it was shown how the choice of rate does not affect network performance significantly.

We then extended the derived stochastic geometry model to networks where the time on air of messages could be different. The extended model is general enough to be able to consider, e.g., messages with different payload lengths and bitrate adaptation techniques. To show the flexibility of this model, results have been explicitly calculated for two scenarios, where the time on air heterogeneity was caused by the use of different bitrates, while the message payload length was fixed. Results were able to provide insights on the effectiveness of the use of multiple bitrates, which is going to be exploited in the following chapters.

Chapter 5

Multi-rate ALOHA Protocols for Machine-Type Communication

To mitigate the scalability limits of ALOHA protocols, some MTC technologies feature multiple transmission rates and, more importantly, advanced receivers that can detect the modulation scheme of a packet *on the fly*, without the need to transmit the packet header at a known basic rate. In this chapter, two multi-rate Medium Access Control (MAC) protocols are proposed, named Multirate-Split Slotted ALOHA (MSSA) and Multirate ALOHA Reservation Protocol (MARP), which are designed to better exploit such multi-rate capabilities of the wireless technologies. By means of extensive simulations, the performance over the legacy ALOHA protocol in the specific scenario of MTC, characterized by a massive number of nodes which sporadically transmit short packets, are studied.

5.1 Introduction

The classic ALOHA protocol is known to suffer scalability problems when the channel access rate increases, so that it does not appear suitable to sustain massive MTCs. To mitigate this scalability issue, many MTC technologies support multiple modulation and coding rates, in order to better adapt the transmission to the channel conditions, thus reducing the transmission time and the energy consumption of the nodes. However, a major problem in contention-based multi-rate systems is that the collision probability increases with the duration of the packet transmission, so that nodes using lower transmission rates get generally penalized by a higher failure probability due to the interference produced by other nodes. Because of that, some technologies feature advanced receivers that are capable of Multi-Rate Decoding (MRD) and MPR.

MRD refers to the capability of the receiver to detect the modulation scheme of the incoming signal *on the fly*, without the need for any signalling, as in conventional wireless transmissions. Without MRD, the indication of the coding scheme used for the payload of the packet is typically embedded in the packet

header, which is always sent at a basic rate to make it possible its decoding at the receiver. In this way, however, the transmission efficiency (i.e., the ratio between the transmission time of the packet payload and that of the entire packet, included the header) decreases with the bitrates of the payload, and the efficiency loss becomes even more important for short packets, as for MTC. Instead, MRD makes it possible to use the same coding scheme for the entire packet (header and payload), thus achieving the same transmission efficiency for all the bitrates and enabling the use of higher bitrates also in MTC.

MPR, instead, refers to the capability of the receiver to decode multiple signals in parallel. This feature is obtained by using multiple receive chains at the receiver operating on different frequency channels and/or using advanced signal processing algorithms to exploit the coding gain of the various modulation schemes, or iteratively decoding and canceling the overlapping components of the compound received signal by using Successive Interference Cancellation techniques [107]. Currently, commercial receivers typically support MPR only for packets that are transmitted on different frequency channels or with different modulation schemes. MPR, thus, increases the system capacity by enabling transmissions on multiple orthogonal (or quasi orthogonal) channels.

While these advanced features clearly increase the capacity of the systems, further gain can be obtained by changing the MAC protocol in order to better exploit such features. Following this principle, here two multi-rate ALOHA-based MAC protocols, named MSSA and MARP, are proposed. MSSA and MARP are both based on a time frame structure that splits the resources based on the transmission rates of the nodes, so that nodes contend only with other nodes that use the same transmission rate. In this way, the channel access should gain in fairness.

The main difference between the two proposals is that MSSA makes use of the (rate-split) slotted ALOHA protocol to transmit data packets, while in MARP the slotted ALOHA protocol is used to transmit short Reservation Messages (RMs) that, if successfully delivered, will grant the node exclusive channel access to transmit longer data packets. The main novelties of these protocols, hence, are: i) the splitting of the access resources based on the node's data rates; ii) the transmission of the control information (RMs, beacons, acknowledgments) at the same rate of the associated data packets; iii) the dynamic adaptation of the frame duration and organization to the amount and rate-distribution of the channel access requests.

The performance of the proposed protocols in terms of throughput and energy efficiency are studied by means of extensive simulations and compared with that of the baseline ALOHA access procedure adopted by common LPWA technologies.

5.2 Multi-Rate ALOHA protocols

In this study, the focus is on a simple (but common) MTC scenario, where a finite but large number of machine-type devices are connected to a common base station/gateway using a single frequency channel, shared by all users. Nodes transmit short packets of fixed length, according to a certain packet generation process, which is assumed independent for each node. Furthermore, nodes are supposed to be able to adapt their transmission rate to the average channel

Figure 5.1: Allocation of transmission windows for each rate in a MSSA frame.

gain towards the gateway. The set of available transmission rates is denoted as $\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$, with $R_1 < \ldots < R_k$.

The receiver is capable of recognizing and decoding each such rate without the need of any control information (MPR feature). For what concerns the capability of simultaneous decoding of multiple packets, two opposite cases are distinguished: Single-Packet Reception (SPR), where the receiver can handle only one packet at a time; and full MPR, where, instead, the receiver can simultaneously decode packets sent at different rates, which are supposed to not interfere one another. Note that, in the latter case, the performance analysis boils down to that of systems using a single transmission rate.

In the following, the two proposed multi-rate ALOHA-based protocols are described in detail.

5.2.1 The MSSA protocol

The MSSA protocol is based on the Frame Slotted ALOHA structure where, however, the slot duration depends on the transmission rate. More specifically, as depicted in Fig. 5.1, the time is divided in frames, which are split in multiple transmission windows, one for each bitrate. The windows, in turn, are organized in slots, whose length depends on the bitrate associated to the window, and is sufficient to contain a packet transmitted at that rate. Each node, then, transmits its packet on a random slot of the window reserved to its transmission rate.

It is assumed that the duration T_F of the time frame cannot be changed, depending on a number of uncontrollable factors, such as the maximum acceptable delay, clock drift, beacon size, and so on. Therefore, we can only act on the way the time frame is split among the different windows, i.e., on the number n_i of slots assigned to the ith transmission window, reserved to the transmission rate R_i, with $i = 1, \ldots, k$.

To find the optimal values of such $\{n_i\}$, we can resort to an approximate Poisson model, which yields a simple optimization problem that can be solved using standard methods. Denoting by L_{pck} the size of data packets, the slot duration in the ith window will be equal to $\frac{L_{\text{pck}}}{R_i}$. Therefore, any feasible slot allocation must satisfy the condition:

$$\sum_{i=1}^{k} n_i \frac{L_{\text{pck}}}{R_i} \leq T_F \,. \tag{5.1}$$

Denoting by W_i the average number of packets successfully sent in the ith window, our aim is to find the values of $\{n_i; i = 1, \ldots, k\}$ that maximize the aggregate average system throughput, given by $W = \sum_{i=1}^{k} W_i$, while satisfying the feasibility condition (5.1).

Figure 5.2: Example of resource allocation for reservation and data transmission subframes inside a MARP frame.

Now, let G_i denote the aggregate packet generation rate of the nodes that transmit at rate R_i, $i = 1, \ldots, k$. Then, the average number of transmission attempts in the ith window of a frame will be equal to $A_i = G_i T_F$. Assuming Poisson arrivals,[1] the average number of non-overlapping packet transmissions in the ith window of a frame is given by

$$W_i = A_i e^{-\frac{A_i}{n_i}} = G_i T_F e^{-\frac{G_i T_F}{n_i}} \ . \tag{5.2}$$

Putting all the pieces together, the optimization problem can then be expressed as

$$\begin{aligned} \max_{n_1,\ldots,n_k} \ & \sum_{i=1}^{k} G_i T_F e^{-\frac{G_i T_F}{n_i}} \\ \text{s.t.} \quad & \text{condition} \\ & \{n_i \in \mathbb{N};\ i = 1, \ldots, k\}. \end{aligned} \tag{5.3}$$

This multi-rate optimization problem can be solved using the *Differential Evolution* technique, a heuristic approach to nonlinear optimization that was initially proposed for continuous and unconstrained problems [108], but has been later extended to mixed-integer constrained problems [109].

5.2.2 The MARP protocol

The MARP protocol is largely based on MSSA, with the difference that the contention-based access method is used to transmit RMs that, if accepted, will grant exclusive access to the channel for the data transmission. Therefore, as illustrated in Fig. 5.2, each time frame is divided into a reservation subframe of constant duration T_R and a data transmission subframe of variable duration T_D. The channel access in the reservation subframe is managed according to the MSSA protocol, but in place of data packets, the nodes will transmit RMs, each specifying the required resources (i.e., the expected transmission time of the associated data packet) and a packet identifier. A different RM must be transmitted for each data packet that needs to be sent.

After the reservation subframe, the base station broadcasts a beacon that contains the identifiers of the RMs that have been successfully received and the assigned transmission window in the following transmission subframe. The beacon is transmitted in a multi-rate mode, reflecting the rate of the accepted RMs, in ascending bitrate order.[2] The beacon starts with some control fields

[1] We observe that, in the MTC scenario, the Poisson-arrival model is reasonable, since we have a large population of nodes, each with low packet transmission probability.

[2] This assumption implies that the downlink channel is (at least) as good as the uplink channel, which is reasonable when the gateways are more powerful than the peripheral nodes

(such as the indication of the beginning of the next frame) transmitted at the basic rate. Each rate switching during the transmission of the beacon is preceded by a rate-switching flag, so as to let the receivers change the demodulation scheme accordingly. After the reception of the feedback for its RM, a device can stop listening to the beacon in order to save energy. Hence, the devices can keep their radio on just for the transmission of their RMs, the reception of the parts of the beacon of interest, and the transmission of the data packet (if allowed).

The data transmission window starts immediately after the end of the beacon and lasts as long as needed to transmit all the packets of accepted RMs. Therefore, the duration T_D of the transmission subframe and, consequently, that of the whole frame may change from frame to frame. Nonetheless, assume that the mean frame duration, $\overline{T}_F$, is given. Therefore, as for MSSA, the protocol parameters that can be optimized are the number n_i of reservation slots to be assigned to the ith rate in the reservation subframe. The feasibility condition, however, has to keep into account the duration of the reservation subframe, the transmission time of the multi-rate beacon, and the transmission time of the data packets for the accepted RMs. In the following, the expressions of these three terms are found.

Denoting by L_{RM} the length of a reservation message, the duration of the reservation subframe can be expressed as

$$T_R = \sum_{i=1}^{k} n_i \frac{L_{\mathrm{RM}}}{R_i} \,. \tag{5.4}$$

Indicating by w_i the number of accepted RMs sent at rate R_i, the transmission time of the beacon can be expressed as

$$T_B = \frac{L_H}{R_1} + \sum_{i=1}^{k} \frac{L_F + w_i L_{FB}}{R_i} \,; \tag{5.5}$$

where L_H denotes the size of the beacon header, always transmitted at the basic rate, while L_F and L_{FB} indicate the size of the rate-switching flag and of the beacon segment, respectively, which are transmitted at the different rates. To simplify the analysis, we consider $L_{FB} = L_{\mathrm{RM}}$.[3]

Finally, the duration of the transmission subframe is given by

$$T_D = \sum_{i=1}^{k} \frac{w_i L_{\mathrm{pck}}}{R_i} \,. \tag{5.6}$$

Summing all the terms together we get the frame duration:

$$T_F = \frac{L_H}{R_1} + \sum_{i=1}^{k} \frac{n_i L_{\mathrm{RM}} + w_i (L_{\mathrm{pck}} + L_{\mathrm{RM}}) + L_F}{R_i} \,. \tag{5.7}$$

and can transmit with higher power. If the symmetry assumption does not hold, however, the protocol can still work by suitably scaling the transmission rate of the beacon message, with a small performance degradation.

[3]This can be considered as a worst-case scenario, since a compression function can be used to shorten the beacon size.

Note that, since $\{w_i\}$ are random variables, so are T_B, T_D, and T_F. Now, under the Poisson-arrival assumption, the mean number of packets transmitted at rate R_i in a frame is given by

$$W_i = \mathrm{E}\left[w_i\right] = G_i \overline{T}_F e^{-\frac{G_i \overline{T}_F}{n_i}} \tag{5.8}$$

where $\mathrm{E}\left[\cdot\right]$ denotes the statistical expectation operator, and $\overline{T}_F$ is the mean frame duration. The feasibility condition for MARP can then be obtained by using (5.8) in the expectation of the right-hand side of (5.7), which gives

$$\overline{T}_F \geq \frac{L_H}{R_1} + \sum_{i=1}^{k} \frac{n_i L_{\mathrm{RM}} + G_i \overline{T}_F e^{-\frac{G_i \overline{T}_F}{n_i}} \left(L_{\mathrm{pck}} + L_{\mathrm{RM}}\right) + L_F}{R_i} \,. \tag{5.9}$$

Finally, the optimization problem can be formulated as

$$\begin{aligned} \max_{n_1,\ldots,n_k} \quad & \sum_{i=1}^{k} G_i \overline{T}_F e^{-\frac{G_i \overline{T}_F}{n_i}} \\ \text{s.t.} \quad & \text{condition (5.9)} \\ & \{n_i \in \mathbb{N}\,;\; i = 1, \ldots, k\}. \end{aligned} \tag{5.10}$$

This multi-rate optimization problem can again be solved using the Differential Evolution technique. Note that, when considering the special case of single-rate transmissions (i.e., when $k = 1$), the optimization problem can be solved by a simple exhaustive search on the number of reservation slots.

5.3 Performance analysis

This section presents the results of an extensive simulation study, where the performance of MSSA, MARP, and the legacy SA protocol have been compared both in terms of throughput and energy efficiency. The simulator has been implemented in Python 3 using the SciPy scientific libraries.

5.3.1 Simulation scenario

In the simulation, transmitters are uniformly distributed over a circle centered at the gateway, with density λ_s nodes per squared meter. The transmissions are affected by path loss, Rayleigh fading, and white noise, in addition to interference. Therefore, the received signal power is given by $\mathrm{P}_{\mathrm{tx}}(Ad)^{-\beta}F$, where P_{tx} is the transmission power, A and β are path-loss parameters, d is the transmitter-receiver distance and F is an exponential random variable with unit mean, modeling the Rayleigh fading. A packet transmitted at rate R is successfully received if the SINR exceeds a threshold $\Gamma_r = 2^{R/B_{\mathrm{w}}} - 1$, where B_{w} is the channel bandwidth, according to the Shannon formula. The cell radius is such that nodes at the edge have 95% of success probability in absence of interference.

Each node generates new data packets of length L_{pck} according to a Poisson process of rate λ_t. Packets generated during a frame are all transmitted in the next one. In the multi-rate case, the available transmission rates are $\mathcal{R} = \{0.5, 2, 10, 100\}$ kbit/s, which well represent the bitrates usually supported by

Parameter		Value
Spatial node density	λ_s	0.1 nodes/m^2
Packet generation rate	λ_t	$\{3 \cdot 10^{-7}, \ldots, 3 \cdot 10^{-4}\}$ packets/s
Transmission power	$\mathrm{P_{tx}}$	14 dBm
Transmission frequency	f	868 MHz
Path loss coefficient	A	36.36 m^{-1}
Path loss exponent	β	3.5
Packet length	L_{pck}	240 bit
Transmission bitrate (single-rate scenario)	R	0.5 kbit/s
Transmission bitrates (multi-rate scenario)	$\mathcal{R}$	$\{0.5, \ldots, 100\}$ kbit/s
Bandwidth	B_{w}	400 kHz
Noise spectral density	N_0	$2 \cdot 10^{-20}$ W/Hz
Frame duration	$\overline{T}_F$	100 s
RM size	L_{RM}	40 bits
Beacon flag size	L_F	8 bits
Beacon header size	L_H	32 bits

Table 5.1: Simulation parameters

commercial IoT transmission technologies. For the single-rate scenario, the bitrate is set to $R_1 = 0.5$ kbit/s.

The values of all the simulation parameters are listed in Tab. 5.1.

5.3.2 Performance metrics

The throughput is defined as the average number of successfully delivered data packets per unit time, and indicated as S.

The energy cost of a protocol is obtained by adding up the energy spent to transmit control and data packets, and to receive the feedback (if any).

Since devices transmit infrequently, we can suppose that they do not listen to every beacon, rather they wake up when a new packet is ready for transmission and wait until the next beacon reception to get time synchronized. After that, devices can perform channel access following the adopted MAC protocol. Note that, after wake up, a node waits on average half a time frame for the next available beacon.

Therefore, let $\mathrm{P_{rx}}$ denote the power consumed during reception. Denoting by $P(R)$ the probability that a node transmits with rate R, the average energy consumption to transmit a packet in MSSA and SA is simply given by

$$\mathcal{E}_{\mathrm{MSSA}} = \mathrm{P_{rx}} \left(\frac{\overline{T}_F}{2} + \frac{L_H}{R_1} \right) + \mathrm{P_{tx}} \sum_{i=1}^{k} \frac{L_{\mathrm{pck}}}{R_i} P(R_i) \; ; \tag{5.11}$$

where the first term of the sum accounts for the energy spent to wait for and receive a short synchronization beacon, which is supposed to be L_H bits long and always transmitted at the basic rate, while the second term accounts for the packet transmission energy.

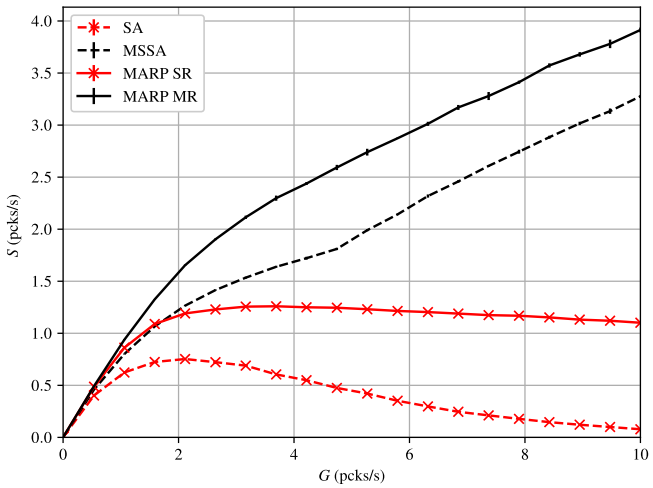


Figure 5.3: Throughput of SA and MARP, with 95% confidence intervals.

For the MARP protocol, instead, we have

$$\mathcal{E}_{\text{MARP}} = \text{P}_{\text{rx}} \left(\frac{\overline{T}_F}{2} + \frac{L_H}{R_1} + \sum_{i=1}^{k} \frac{W_i L_{\text{RM}} + L_F}{R_i} \right) + \text{P}_{\text{tx}} \sum_{i=1}^{k} \frac{L_{\text{RM}} + L_{\text{pck}} W_i / n_i}{R_i} P(R_i) \,, \quad (5.12)$$

where, again, the first term accounts for the energy to receive the beacon (that, however, also carries feedback information), while the second term gives the packet transmission energy. Finally, W_i is given in (5.8). Note that $\mathcal{E}_{\text{MARP}}$ is an upper bound to the actual energy consumption because it accounts for the reception of the entire beacon, even though nodes can stop the reception earlier, as previously explained.

The values used for the power consumption in transmission (P_{tx}) and reception (P_{rx}) mode have been extracted from the datasheets of some off-the-shelf modules.[4]

5.3.3 Throughput analysis

In Fig. 5.3 we can see that the MARP significantly outperforms the other schemes in terms of throughput. In particular, the maximum throughput for the single-rate and multi-rate MARP is higher than that of SA and MSSA, respectively.[5] Also, for the single-rate MARP, the decreasing trend of the throughput after the peak is much less marked than in SA, allowing for a wider stability region. For example, when considering an offered traffic equal to the 80% of

[4] Atmel AT86RF212B, Texas Instruments CC1125 and CC1310, and Semtech SX1272 modules.

[5] Note that, in the single-rate case, the MSSA achieves the same performance as the standard SA.

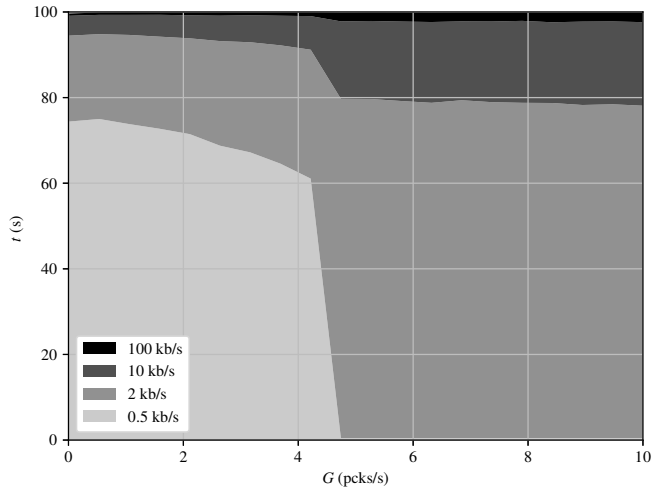


Figure 5.4: Average frame time used for transmission of packets in MSSA, grouped for bitrate.

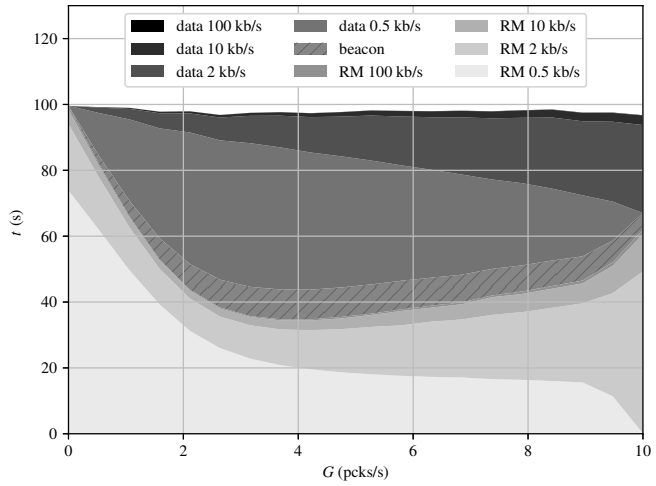


Figure 5.5: Average frame time used for transmission of RMs, beacon, and data packets in MARP, grouped for bitrate.

their maximum throughput, single-rate MARP is capable to sustain a temporary overload of the offered traffic 9 times larger than that in the stable working point, while SA enters the instability region if the offered traffic exceeds 2.75 times the value at the stable working point.

To understand the origin of this gain, we can analyse Fig. 5.4 and Fig. 5.5, which report the average fraction of frame time occupied by the transmission of data and control packets at the different rates, for MSSA and MARP, respectively. We can see that, when the offered traffic grows, the optimization routine tends to allocate less and less slots to the lower rates, in order to maximize the

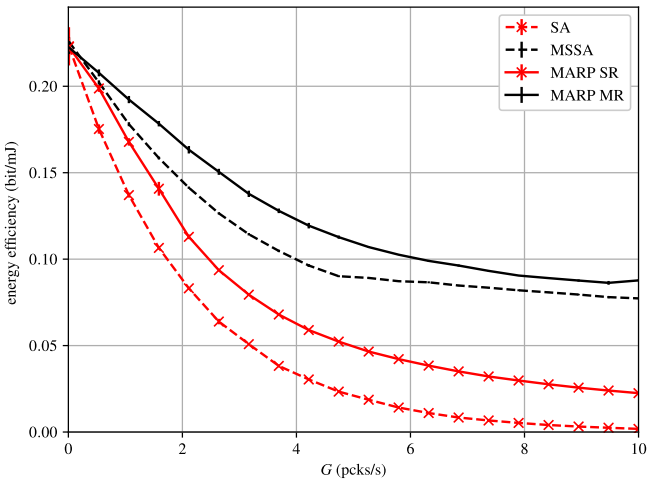


Figure 5.6: Number of bits successfully transmitted per unit energy, with 95% confidence intervals.

overall number of successful transmissions per unit time. This is equivalent to artificially reduce channel contention by forcing some nodes to silence, which is a way to preserve system stability when the channel is overloaded.

In Fig. 5.5 we can also see that the average frame duration is actually less than the imposed value of $\overline{T}_F = 100$ s because of the random fading that causes some of the RMs to be lost even in absence of interference.

5.3.4 Energy efficiency analysis

In Fig. 5.6 we can see the energy efficiency (i.e., the amount of successfully transmitted information bits over the energy consumed in all transmission attempts) of SA, MSSA, and MARP protocols. The curves have been obtained by using (5.11) and (5.12), where $\overline{T}_F$ and $P(R_i)$ have been extracted from the simulations. In the single-rate scenario, the cost for the beacon listening in MARP is significant, so, in a very low traffic environment, we can save energy using the traditional SA. However, in massive access scenarios, the higher energy consumption of MARP is compensated by a higher success rate, allowing for a better efficiency than SA. The same is true for the multi-rate case, where the energy efficiency of MARP is lower than that of MSSA only for extremely low traffic.

5.4 Conclusions

In this chapter, two variations of the ALOHA protocols have been proposed to better support the multi-rate transmission and reception capabilities of modern wireless technologies for MTC. The first extends SA by reserving time windows to different bitrates, while the second introduces RMs in order to reduce the fraction of time occupied by collided packets on the channel.

A simulative performance analysis has been carried out to assess the improvements over legacy ALOHA-like protocols. Results show that the proposed protocols provide higher throughput and make it possible to sustain higher traffic than SA. In particular, MARP achieves the best performance when the traffic offered to the channel becomes critical. MARP also offers better energy efficiency than MSSA, both in the single-rate and multi-rate cases, thanks to the short duration of the RMs, guaranteeing a collision-free channel access to the comparatively large data packets. The overhead of MARP RMs is significant only when collisions are very uncommon, making it the overall best performing protocol for any networks with non-negligible size.

Chapter 6

Random Access Schemes to Balance Energy Efficiency and Accuracy in Monitoring Applications

Performance and efficiency of channel access protocols can be improved by considering the content of the messages they are transmitting. Compared to the approaches in the previous chapter, that were agnostic to the message content, this chapter proposes some random access schemes specifically tailored for monitoring applications, where messages contain data points of a time series. By exploiting correlations in time series, these schemes have a very high energy efficiency while guaranteeing a predefined accuracy in the time series reconstruction.

6.1 Introduction

The continuous decrease in cost and size of sensing devices enables their use in various applications, in particular related to environmental and industrial sensing [110–114]. Monitoring activities can be roughly divided in two classes, based on their objective. In the first category, the aim is to collect data in order to enable its statistical analysis and thereby perform trend analysis or predict future values. In the second case, 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. The main difference in the two use cases is that the former analyses the time series as a whole, including the past history of the monitored signal, while the latter only considers its current realization. Accordingly, the definition of the target performance metric, which is related to the error of the signal reconstructed from the samples at the control station, is different for the two cases. For signal prediction, indeed, it makes sense to consider the cumulative signal estimate error up to the current time, possibly using a weight function to smooth out the impact of the past errors [8]. In contrast, when the target is related to event triggering, the error metric is

calculated independently for every sample, since past values are not of interest when new data is available [5].

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. The first constraint to consider is the energy consumption of both sensing and data transmission/reception procedures for battery powered devices. Another critical point is the use of a massive number of sensing devices to increase the spatial accuracy of the data. This in fact yields a large traffic on the wireless channel at the risk of reducing the amount of successful transmissions to the Fusion Center (FC) due to collisions, and may eventually result in a reduction of the monitoring accuracy. Most of the work that deals with signal compression and data monitoring does 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 further 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 behaviour often changes if observed for a sufficiently long time. Also, intervals when data has a large variance, and therefore is difficult to predict, are 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 [115] 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 [115]:

Sampling compression leverages data correlation in space and time to reduce the sampling activity of devices. The FC is then in charge of recovering the missing samples using only the received information.

Data compression processes the sampled measures in order to limit the length of messages directed to the FC.

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.

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 study investigates new compression and communication protocols that target a given QoS, measured in terms of reconstruction error at the FC, while minimizing the energy consumption of the devices. The focus is on a static scenario with a massive number of sensors.

6.2 Related work

Here, the significant amount of work on compression techniques for WSNs is reviewed, following the classification introduced by [115].

Sampling Compression

Many techniques that operate at this level exploit only the temporal correlation between samples to reduce the sampling activity of devices. For example, [111] estimates the maximum frequency of the time series via the Fast Fourier Transform and sets the sampling rate according to the Nyquist theorem. Other similar approaches use Kalman filters [116], Bollinger bands [117], or linear programming techniques [118]. Ref. [119] proposes an Exponential Double Sampling-type predictor to dynamically change the sampling interval in order to maintain the error below a given threshold. In addition, [110] considers issues related to multi-hop networks, where the routing of messages containing sensor readings can have a significant impact on the total energy cost. Ref. [120] applies temporal sampling compression to WSNs with energy harvesting sensors, with the objective to dynamically vary the sampling rate based on the amount of available energy on each device.

Differently from techniques exploiting temporal correlation, spatial correlation can be used in order to select only a subset of devices which will acquire and transmit sensor data [121–124]. The other devices will, instead, stay in *sleep mode* to preserve energy. The chosen device subset changes at every sampling interval to consume energy equally from all devices. To improve accuracy and lower the energy consumption, [122, 124] also exploit correlation between sensors attached to the same communication device. The reading precision and the energy cost of reading a sample, in fact, may be different for different sensors [115, 125, 126].

The joint use of both spatial and temporal correlation has the potential to further decrease the energy consumption of the sensing system. A possible way to exploit this is to group sensors in clusters to capture spatial correlation and, for each of them, acquire readings from only one sensor, which is periodically rotated. The sampling interval is then adjusted based on the acquired data [127,128]. A different approach uses Compressive Sensing (CS),[1] which is based on the observation that even a small number of random projections of a sparse signal may contain enough information to recover the whole original signal with excellent accuracy [129,130]. This enables the representation of sensor readings using fewer samples than those required by the sampling theorem. Therefore, in a WSN that uses CS, a number of measurements smaller than the required spatio-temporal granularity can be transmitted to the FC, where the missing data will be recovered [131–133]. Furthermore, in multi-hop sensor networks, CS can be used to add and compress data of messages in transit at intermediate network nodes [134].

[1] A number of studies differentiate between CS, which exploits only temporal correlation of a single sensor, and Distributed CS, which exploits also correlation across multiple sensors. To keep the description short and focused, such a distinction is not considered here.

Data compression

When using data compression, we can compress samples to limit the length of messages directed to the FC [135]. Following this approach, [136] quantizes the difference between consecutive samples using a differential pulse code modulation scheme, while the study in [137] derives the optimal quantization and transmit power levels when using a quadrature amplitude modulation.

Communication compression

Communication compression reduces the number of transmissions and their time-on-air. An implementation of this technique consists in exploiting a model of the sensed signal, which is shared by the sensors and the FC. The measurement at each sample interval is compared with the model prediction. If the prediction error is too large, the sensing device sends to the FC the correct measurement, otherwise no data transmission occurs. The model parameters may be updated in real time when the prediction error starts diverging. This approach has been used in [138] and [139], but while the former applies this technique separately for each node and, thus, uses only the temporal correlation of the data, the latter exploits the space-time correlation of the data, taking inspiration from the MPEG encoding [140].

Note that communication compression affects only the transmission strategy, while it does not affect sensor readings. Therefore, devices have to acquire sensor readings, possibly following a sampling compression strategy, according to the desired temporal accuracy, even if some of those data will not be transmitted. By performing sampling compression alone, however, it is still possible to obtain large energy savings if the energy needed to make the measurements is large. In fact, it has been shown that the sensing energy of some sensors can be larger than the energy used by the radio transmitter [115, 125, 126].

6.3 System model

We now focus on a scenario where several sensor nodes monitor some processes of interest and report their measurements to a common FC. The FC is a powerful device connected to the energy grid, while the sensors are battery-powered and with limited computational capabilities. The goal is to minimize the energy consumption of the sensor nodes, while guaranteeing a minimum level of accuracy and reliability of the monitoring service at the FC.

In the following, the channel and network model used in the rest of this chapter are described.

Network model

The first assumption is that the network is organized in a star topology, with the end devices at one-hop distance from the FC and operating based on the same strategy.

Furthermore, suppose that the network dynamics is slowly varying, so that the transmission scheme needs to be only seldom updated. Note that, even though the channel status may vary because of the fading component, the transmission strategy is based on the expected channel conditions, which are static.

Therefore, the proposed policy will not depend on the absolute slot index, which is hence neglected in the following.

The FC has no restrictions in terms of energy availability and of computational and storage capabilities. The sensor nodes, instead, are simpler devices that are off the grid and need to efficiently manage their finite energy resources.

Depending on the channel conditions and the interference caused by the other nodes, a transmitted packet may be lost. A successful transmission is immediately acknowledged by a downlink packet, which is sent using high power in a dedicated channel and, hence, is always correctly received. If no data is transmitted, or it is lost, then the FC produces an estimate $\hat{x}_k$ based on the last received data and the time-correlation characteristics of the signal model.

Channel model

The sensor nodes are at known distances from the FC and transmit wirelessly over Rayleigh fading channels. Also, assume that a device at distance r sets its transmit power $\mathrm{P}_{\mathrm{tx}}(r)$ in order to fully compensate its path-loss $\ell(r)$, so that the average received power $\overline{\mathrm{P}}_{\mathrm{rx}} = \mathrm{P}_{\mathrm{tx}}(r)/\ell(r)$ is the same for every transmitter, irrespective of r.[2]

This work considers a SA random channel access scheme, which avoids the need to centrally coordinate the channel access, 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. We may consider a transmission to be successful if the average SINR at the receiver is larger than a predefined threshold [141].

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 SINR at the FC experienced by the generic node 0, when other m nodes transmit, can be expressed as

$$\gamma(m) = \frac{\overline{\mathrm{P}}_{\mathrm{rx}} F_0}{N_s + \sum_{i=1}^{m} \overline{\mathrm{P}}_{\mathrm{rx}} F_i} \,. \tag{6.1}$$

where N_s is the noise power and the F_i terms are independent realizations of an exponentially distributed random variable (r.v.) with unit mean, accounting for Rayleigh fading for each device.

In order to better exploit the SA channel access method, we can adapt the modulation and coding scheme to the packet size, in order for the time-on-air to be always equal to the slot time. Therefore, the energy consumption is constant for each packet transmission, but longer packets will experience a higher error probability, as they are sent at higher bitrates. Using Shannon's bound as an approximation, the SINR threshold for a packet of size L is set to

$$\Gamma^{\circ}(L) = 2^{L/(TB_{\mathrm{w}})} - 1 \,, \tag{6.2}$$

where B_{w} is the transmission bandwidth and T is the time slot duration.

[2]We can suppose the maximum transmission power to be large enough (or, equivalently, the cell radius to be small enough) so that the path loss inversion can be applied for every node in the network.

The transmission success probability can then be expressed as

$$\begin{aligned} p_s(L) &= \Pr\left[\gamma(m) > \Gamma^\circ(L)\right] = \\ &= \Pr\left[F_0 > \left(\frac{N_s}{\overline{\mathrm{P}}_{\mathrm{rx}}} + \sum_{i=1}^{m} F_i\right)\Gamma^\circ(L)\right] = \\ &= e^{-N_s\Gamma^\circ(L)/\overline{\mathrm{P}}_{\mathrm{rx}}}\,\mathrm{E}\left[e^{-\Gamma^\circ(L)\sum_{i=1}^{m}F_i}\right], \end{aligned} \tag{6.3}$$

where the expectation is computed with respect to the interference distribution, conditional to the presence of the target transmitter. We can adopt a stochastic geometry reasoning and model the sensing devices that transmit in a given slot as a 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 space coordinate 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 [106]. The fading coefficients $\{F_i\}$ can then be seen as marks of this PPP, making it possible to apply Campbell's theorem for marked processes [106]. We then have

$$\begin{aligned} &\mathrm{E}_\Psi\left[e^{-\Gamma^\circ(L)\sum_{i=1}^{m}F_i}\right] \\ &\qquad = \exp\left(-\int_{\mathbb{R}^2}\int_0^\infty\left(1-e^{-\Gamma^\circ(L)\varphi}\right)\lambda_s(x,\mathtt{P})e^{-\varphi}\,\mathrm{d}\varphi,\mathrm{d}x\right), \end{aligned} \tag{6.4}$$

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 outside the cell, i.e., assuming $\lambda_s(x,\mathtt{P}) \equiv \lambda(\mathtt{P})$ for all $|x| \le r_{\max}$ and $\lambda_s(x,\mathtt{P}) \equiv 0$ otherwise, we get

$$\begin{aligned} \mathrm{E}_\Psi\left[e^{-\Gamma^\circ(L)\sum_{i=1}^{m}F_i}\right] &= \exp\left(-\lambda_s(\mathtt{P})\pi r_{\max}^2\int_0^\infty\left(1-e^{-\Gamma^\circ(L)\varphi}\right)e^{-\varphi}\,\mathrm{d}\varphi\right) \\ &= \exp\left(-\lambda_s(\mathtt{P})\pi r_{\max}^2\frac{\Gamma^\circ(L)}{\Gamma^\circ(L)+1}\right). \end{aligned} \tag{6.5}$$

Replacing this result into (6.3) we finally get

$$p_s(L) = \exp\left(-\Gamma^\circ(L)\left(\frac{N_s}{\overline{\mathrm{P}}_{\mathrm{rx}}} + \frac{\lambda_s(\mathtt{P})\pi r_{\max}^2}{\Gamma^\circ(L)+1}\right)\right). \tag{6.6}$$

Notice that the success probability depends on the adaptive transmission strategy through two parameters, namely the packet size L, and the persistency constant $\mathtt{P}$ that, in turn, is equal to the reciprocal of the mean period between consecutive transmissions of a node. In the following, this last parameter is referred to as *mean sleeping period*, and denoted by $S(L)$, being a function of the packet size L.

6.4 A semi-deterministic strategy for single value reporting

This section proposes a compression and communication protocol able to guarantee a desired accuracy in data gathering applications that are interested in

instantaneous values of some sensed parameter. Differently from previous works in the literature, the proposed strategy operates at multiple levels. At the sampling compression level, it optimizes the sensing rate by exploiting temporal correlation of the sensed measurements; at the data compression level, it sets the quantization accuracy based on the future expected squared reconstruction error; at the communication compression level, it avoids transmission of data if the prediction of the FC is still valid.

6.4.1 Signal model

Each sensor i monitors a time-correlated process $\{x_{n,i}\}_{n\geq 0}$, which is modeled as a first-order autoregressive (AR) process

$$x_{n,i} = \alpha x_{n-1,i} + u_n \quad i \in \mathcal{N}, \tag{6.7}$$

where n denotes the time step, α is the correlation parameter and the process noise is zero-mean normal, $u_n \sim \mathcal{N}(0, \sigma^2)$, and independent of any other u_m. We also assume that $|\alpha| < 1$, so that $x_{n,i}$ is a stable process.

Each sensor can decide when to sense its process and whether to transmit the measured data, since it is assumed that sensing and transmissions both consume energy. Moreover, if in slot n a device chooses to transmit $x_{n,i}$, it can also decide how much information to send, so that it may transmit a distorted version $\tilde{x}_{n,i}$ of $x_{n,i}$ to the receiver. $L_{\max}$ is the size of the original data, while $L_n \in \{0, \ldots, L_{\max}\}$ is the size of the packet sent by the device in a slot n. Reducing the amount of information introduces an error $v(L_n)$, whose statistical distribution depends on the type of data processing performed by the node. Hence, device i sends $\tilde{x}_{n,i} = x_{n,i} + v(L_n)$. If the transmission is successful, the receiver is able to perfectly reconstruct $\tilde{x}_{n,i}$. Otherwise, the FC maintains an estimate $\hat{x}_{n,i}$ of $x_{n,i}$ based on the previously received data. 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. The reconstruction accuracy at the FC is in terms of the squared error $|x_{n,i} - \hat{x}_{n,i}|^2$. Note that this error is zero only if node i transmitted in slot n (no estimation error at the FC, i.e., $\hat{x}_{n,i} = \tilde{x}_{n,i}$), and $L_n = L_{\max}$ (no distortion introduced at the source, i.e., $\tilde{x}_{n,i} = x_{n,i}$).

6.4.2 Optimization problem

Our objective is to determine the optimal duration $M^\star$ 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 ε is bounded, and (ii) the sensor's lifetime is maximized.

As already said, each transmission consumes the same amount of energy, including the circuit energy for operational mode switching. 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 M of the sleeping phase (while not violating the QoS constraint). Basically, the goal is to find

$$M^\star \triangleq \max_{L,\, M} M(L)\,, \tag{6.8}$$

Algorithm 1 Alternate optimization

1: $\mathbf{M}_L \leftarrow$ vector of size $L_{\max}$ ▷ Contains $M_L(L)\ \forall\, L$
2: **for** $L = 1, \ldots, L_{\max}$ **do**
3: Determine $M_0(L)$ ▷ Sleeping duration if $p_s = 1$
4: Initialize $p_s = 1$, $M(L) = 1$
5: **while** $M(L)$ has not converged **do**
6: $M(L) \leftarrow \lfloor M_0(L) + 1 - 1/p_s \rfloor$
7: $p_s \leftarrow$ Eq. (6.3) with I depending on $M(L)$
8: $\mathbf{M}_L(L) \leftarrow M(L)$
9: $M^\star = \max \mathbf{M}_L$, $L^\star = \operatorname{argmax} \mathbf{M}_L$

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

$$\Pr\left[|x_j - \hat{x}_j|^2 > \varepsilon\right] < v_{\text{th}}\,, \tag{6.9}$$

i.e., the probability that the squared reconstruction error exceeds ε is no larger than a predefined threshold v_{th} at any time. The optimal packet size $L^\star$ is the one that maximizes (6.8) under the constraint (6.9).

6.4.3 Analysis

Before delving into the analysis of the transmission strategy, we investigate some tradeoffs in the choice of M 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 M, the less frequent the transmissions. However, M cannot be too large, in order to respect the QoS constraint (6.9). Similarly, decreasing L increases the signal reconstruction error, but also the success probability, because transmissions will use more robust modulations (see (6.2)).

Therefore, determining the optimal transmission strategy is not trivial. For a given value of L, the proposed iterative approach alternately optimize the sleeping phase duration and determine the corresponding probability of successful transmission, until convergence, so that it derives $M_L(L)$, i.e., the optimal value of M for the chosen L. An outer optimization process is then performed to determine the value $L^\star = \operatorname{argmax} M_L(L)$ that yields $M^\star = M_L(L^\star)$.

This procedure is summarized in Algorithm 1. For a fixed L (Line 2), we first compute the duration $M_0(L)$ of the sleeping phase as if there were no interference, i.e., $M_0(L)$ only depends on the QoS requirement (6.9) (Line 3). The interference caused by the other nodes makes transmissions prone to losses. Let $M(L)$ be the number of time slots a node waits from the last successful transmission. Then, assuming that in case of packet loss a device transmits again in the next slot, the expected time elapsed between two consecutive successful transmissions is

$$\sum_{j=0}^{+\infty} (M(L) + j)(1 - p_s(L, I))^j p_s(L, I) = M(L) + \frac{1}{p_s(L, I)} - 1, \tag{6.10}$$

which should not be larger than $M_0(L)$ to satisfy the QoS constraint (6.9). Therefore we get

$$M(L) = \lfloor M_0(L) + 1 - 1/p_s \rfloor\,. \tag{6.11}$$

At the first iteration, assume no packet losses ($p_s = 1$), and thus $M(L) \equiv M_0(L)$. Then (Line 7), we update the success probability p_s using Eq. (6.3), where the number of interferers I depends on the frequency of transmissions which is influenced by $M(L)$. This alternate optimization is repeated until $M(L)$ converges. Then, $L^\star$ and $M^\star$ are those that maximize the value of $M(L)$ as in Line 9.

The following explains how to derive $M_0(L)$, while in Sec. 6.3 it is described how to update p_s based on the persistency constant `P` which, in this case, is equal to $1/M$.

Sleeping phase duration

Now we derive the optimal duration of the sleeping phase $M_0(L)$ for a given L when there is no interference. Exploiting Eq. (6.7), we can relate the signal at slot $n+j$ to that at slot n as follows

$$\begin{aligned} x_{n+j} &= \alpha^j x_n + \sum_{k=1}^{j} \alpha^{j-k} u_{n+k} = \alpha^j x_n + w_j \\ &= \alpha^j \tilde{x}_n - \alpha^j v(L) + w_j = \alpha^j \tilde{x}_n + e_j(L) \; ; \end{aligned} \tag{6.12}$$

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

The estimation error w_j is the linear combination of j i.i.d. zero-mean gaussian r.v.s and, therefore, it is itself a zero-mean normal r.v., $w_j \sim \mathcal{N}(0, \sigma_j^2)$, with variance

$$\sigma_j^2 = \sum_{k=1}^{j} \alpha^{2(j-k)} \sigma^2 = \sum_{k=0}^{j-1} \alpha^{2(j-1-k)} \sigma^2 = \frac{1-\alpha^{2j}}{1-\alpha^2} \sigma^2 \, . \tag{6.13}$$

We observe that σ_j^2 is a concave and non-decreasing function of j with a horizontal asymptote at $\sigma^2/(1-\alpha^2)$ as $j \to +\infty$. 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 j 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 $\tilde{x}_{n+j} = \alpha^j \tilde{x}_n$, so that the squared reconstruction error after j slots from the last received data is $|e_j(L)|^2$. Based on its definition (see Eq. (6.12)), $e_j(L)$ is non-decreasing in j, therefore, to guarantee the QoS constraint (6.9), it is sufficient to evaluate the CDF of the squared error at ε only for $j \equiv M$. For analytical tractability, in the following it is assumed $v(L) \sim \mathcal{N}(0, \omega^2(L))$, although the 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, $e_M(L)$ can be modeled as the weighed sum of two independent normal r.v.s and, thus, is also normally distributed:

$$e_M(L) \sim \mathcal{N}\left(0, \; \alpha^{2M} \omega^2(L) + \sigma_M^2\right) . \tag{6.14}$$

Accordingly, the largest value M_0 that satisfies the QoS constraint for the given L is

$$M_0(L) = \max\left\{M : 2 - 2\phi(\sqrt{\varepsilon}, M, L) < v_{\text{th}}\right\}, \tag{6.15}$$

where $\phi(y, M, L)$ is the CDF of $e_M(L)$ calculated in y. Eq (6.15) gives the maximum duration of the sleeping phase of a sensor when transmission is always successful ($p_s = 1$) such that (6.9) holds true. If the chosen v_{th} is too small, since M is discrete, the set in (6.15) 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 $M_0 = 1$.

Transmission strategy

With the procedure explained in Algorithm 1, after a successful transmission a device remains silent for $M^\star - 1$ slots before attempting to transmit again, where $M^\star$ yields an expected time between two consecutive successful transmissions such that the QoS requirement (6.9) is satisfied. However, based on how signal $\{x_n\}$ actually evolves, the squared error after $M^\star$ slots may be larger or smaller than threshold ε. In the latter case, the device could extend the sleeping phase by $M'(x_{n+M^\star})$ additional slots to further save energy. Here, we study how to determine $M'(x_{n+M^\star})$.

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

$$\begin{aligned}
\epsilon_j(L^\star) &= |x_{n+M^\star+j} - \hat{x}_{n+M^\star+j}| \\
&= \left|\alpha^j x_{n+M^\star} + w_j - \alpha^{M^\star+j}\tilde{x}_n\right| \\
&= \left|\alpha^j x_{n+M^\star} + w_j - \alpha^{M^\star+j}(x_n + v(L^\star))\right|
\end{aligned} \tag{6.16}$$

where $w_j \sim \mathcal{N}(0, \sigma_j^2)$ was defined in (6.12). The other terms are known by the transmitter, because $x_{n+M^\star}$ and x_n are the new and old sensed data, and the processing error $v(L^\star)$ depends on $L^\star$, already obtained with Algorithm 1. Then, the error is normally distributed: $\epsilon_j(L^\star) \sim \mathcal{N}\left(\mu_j(L^\star), \sigma_j^2\right)$, with mean

$$\mu_j(L^\star) = \alpha^j\left(x_{n+M^\star} - \alpha^{M^\star}x_n - \alpha^M v(L^\star)\right). \tag{6.17}$$

Since $\epsilon_j(L)$ 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 $\epsilon_j(L)$, i.e.,

$$\frac{1}{\sigma_j^2}\epsilon_j^2(L^\star) \sim \chi_1^2\left(\frac{\left(\mu_j(L^\star)\right)^2}{\sigma_j^2}\right). \tag{6.18}$$

Interference and communication parameters		
Time slot duration	T	100 ms
Density of sensor devices	λ_s	0.001 nodes/m^2
Cell radius	$r_{\max}$	500 m
Received power[3]	$\overline{\mathrm{P}}_{\mathrm{rx}}$	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 $P[\lvert x_n - \hat{x}_n\rvert^2 > \varepsilon]$	v_{th}	0.2
Kulau et al. strategy [117]		
Maximum sleep time	$t_{\max}$	50 slots
Weighting exponent	ϕ_{bb}	2
Sliding window size	n_s	30

Table 6.1: Simulation parameters.

It is then straightforward to compute the complementary CDF of $\epsilon_j^2(L^\star)$ for a given value of ε and derive M' as the largest j for which such value is lower than v_{th}, as done to find $M_0(L)$.

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 M slots, ignoring the dynamic sampling rate adaptation carried out by each device.

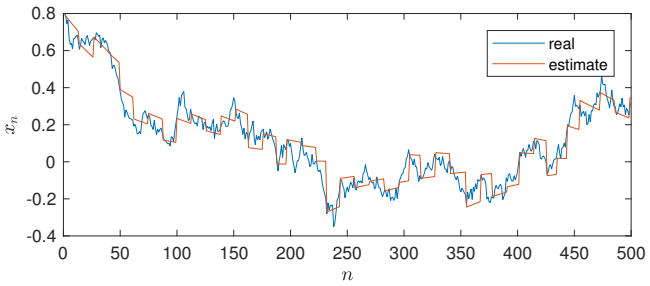
6.4.4 Numerical evaluation

To analyse the performance gain of the proposed system, a simulation is performed with parameters set as in Tab. 6.1.

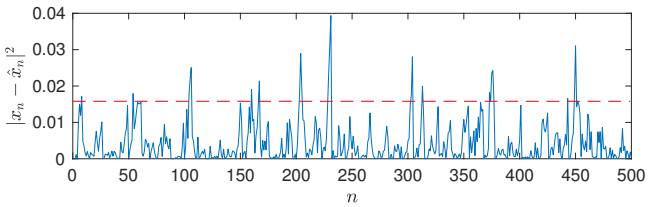
Fig. 6.1 shows an example of the original time series and the corresponding estimate with the technique described in Sec. 6.4.3. The match is very good, and the squared error is almost always below the given threshold ε. Supported by this first result, the proposed strategy (named *dynamic*) is compared 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. The other one is the sample rate adaptation technique described by Kulau et al. [117], 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

$$t_{\mathrm{wait}}(n) = \frac{t_{\max}}{1 + (b_{\mathrm{bb}}\, \sigma_{\mathrm{bb}}(n))^{\phi_{\mathrm{bb}}}} \,, \tag{6.19}$$

[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 ε.

Figure 6.1: Example of a time series and its estimate with the proposed dynamic technique (with $\varepsilon = 0.0158$).

where $\sigma_{\text{bb}}(n)$ is the standard deviation of the last n acquired samples, $t_{\max}$ is the maximum sleeping duration and b_{bb} is an hyperparameter defining the width of the Bollinger bands.

Fig. 6.2 shows the probability that the squared error at the FC stays within threshold ε, as ε increases. To guarantee a fair comparison, b_{bb} is set so that the probability that the squared error is lower than ε is the same as for the proposed strategy.[4] We can see that the proposed technique respects the QoS constraint with a large margin. Fig. 6.2 actually shows that the proposed dynamic policy is overly conservative, since the interference level considered is obtained by the static optimization of M, 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, in the optimization routine of Algorithm 1 we consider every message to be repeated the expected number of transmissions needed to get a successful reception (see Eq. (6.11)). Instead, a more 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 is set to 40.5 mW, and the sensing energy is set to 495 µJ.[5] As we can see in Fig. 6.3, the proposed strategy is between 15% and 50% more efficient than the technique described in [117], 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

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

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

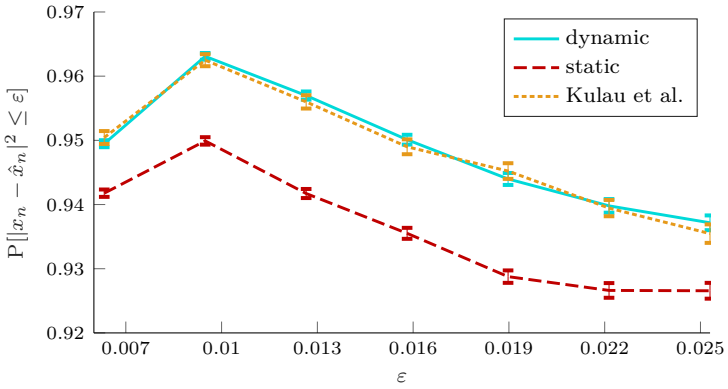


Figure 6.2: Probability that the squared error stays within threshold ε, with 95% confidence intervals.

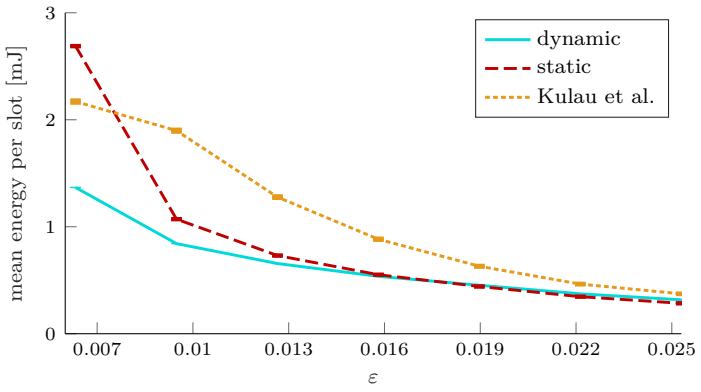


Figure 6.3: Mean energy consumed per slot, with 95% confidence intervals.

accurate. However, if the value of ε 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. 6.3, the energy efficiency loss can be quite significant and the static strategy may outperform the dynamic one.

6.5 A semi-deterministic strategy for the reporting of integral values

The model presented here is similar to the previous one, but, rather than focusing on the time evolution of the actual processes measured by the sensor nodes, here we are concerned with applications that focus on the *integral* of such measurements. Although these two scenarios are seemingly similar, the resulting optimization strategies are significantly different, as are their possible future developments.

6.5.1 Scenario

As mentioned, we focus on a scenario where several sensor nodes monitor some processes of interest and report their measurements to a common FC, which is interested in tracing the time integral of each single process x_n. 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.

In the remainder of this section, first the model of the monitored process is introduced and then the measurement procedure is described.

Monitored process model

Assume that time is slotted and that each sensor monitors a signal $\{x_n\}_{n\geq 0}$, where n is the time index.[6] Also, consider the signal x_n to be time-correlated and that it can be modeled as a first-order AR process

$$x_n = \alpha x_{n-1} + u_n\,, \quad n \geq 1\,, \tag{6.20}$$

where α is the correlation parameter, and u_n represents the process noise, which is i.i.d. over time and modeled as a zero-mean gaussian r.v., i.e., $u_n \sim \mathcal{N}(0, \sigma^2)$. The integral process is then defined as $y_n = \sum_{k=1}^{n} x_n$. To guarantee that both x_n and y_n are asymptotically stationary, we set $|\alpha| < 1$.

Process measurement model

Assume that the sensor nodes can sample the monitored process x_n with a certain, maximum accuracy, and then perform a lossy compression of their measurements to reduce the size of the transmitted messages. Moreover, suppose that the node also transmits the measurement of the integral process y_n with maximum accuracy. 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. The correct reception of a data packet, therefore, makes it possible to completely nullify the estimate error of the integral measure y_n 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.

Therefore, the current data sent by a node can be modeled as $\tilde{x}_n(L) = x_n + v(L)$, where $v(L)$ represents the error due to the lossy compression of the original signal x_n. For the sake of simplicity, assume $v(L)$ 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. In this work, the distortion function considered, also used in [142], is

$$\omega^2(L) = a\left(\frac{L - L_0}{L_{\max} - L_0}\right)^{-b} - 1 \tag{6.21}$$

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.

[6] For ease of notation, the sensor index is omitted. However, in general, different sensors may monitor different phenomena.

The temporal correlation of the signals can also be exploited to reduce the sampling and transmission rates and, by that, save energy by switching to a sleep mode. If data x_n is not received (because the measurement was either not taken by the sensor or not successfully delivered to the FC), the FC estimates it based on the last received data and the data correlation profile, obtaining $\hat{x}_n$. Therefore, 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 particular, consider the absolute value of the cumulative error $\mathcal{E}_n$ after n slots since the last successful transmission, i.e.,

$$\mathcal{E}_n = y_n - \hat{y}_n = \sum_{j=1}^{n} (x_j - \hat{x}_j) \ , \tag{6.22}$$

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

We observe that, under the considered assumptions, the error (6.22) 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. Note that, even if the focus is on y_n, the current measure x_n 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 we can use index 0 to indicate the slot when the last message from a given sensor was received and consider n as the number of slots passed since the last successful reception. Then, by leveraging on the temporal correlation profile, we can write:

$$x_n = \alpha^n x_0 + \sum_{\ell=1}^{n} \alpha^{n-\ell} u_\ell = \alpha^n \tilde{x}_0 - \alpha^n v(L) + \sum_{\ell=1}^{n} \alpha^{n-\ell} u_\ell, \tag{6.23}$$

where $\tilde{x}_0$ is the latest compressed data sample available at the FC. Considering that u_j has zero mean, the estimate with minimum 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 $\sum_{\ell=1}^{n} \alpha^{n-\ell} u_\ell$. It follows that

$$\begin{aligned} \mathcal{E}_n = \sum_{j=1}^{n} (x_j - \hat{x}_j) = \sum_{j=1}^{n} \left[-\alpha^j v(L) + \sum_{\ell=1}^{j} \alpha^{j-\ell} u_j \right] = \\ = \sum_{j=1}^{n} -\alpha^j v(L) + \sum_{j=1}^{n} \sum_{\ell=1}^{j} \alpha^{j-\ell} u_\ell \ . \end{aligned} \tag{6.24}$$

We can express the first error term in (6.24), which is associated to the distortion due to data compression, as

$$\mathcal{E}'_n = \sum_{j=1}^{n} -\alpha^j v(L) = \frac{\alpha^{n+1} - \alpha}{\alpha - 1} v(L), \tag{6.25}$$

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 (6.24) becomes

$$\mathcal{E}''_n = \sum_{j=1}^{n}\sum_{\ell=1}^{j} \alpha^{j-\ell} u_j = \sum_{\ell=1}^{n} u_\ell \sum_{j=\ell}^{n} \alpha^{j-\ell} = \sum_{\ell=1}^{n} u_\ell \frac{1-\alpha^{n-\ell+1}}{1-\alpha} \,. \tag{6.26}$$

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

$$\begin{aligned}\sigma_e^2(n) &= \sum_{\ell=1}^{n} \sigma^2 \left(\frac{1-\alpha^{n-\ell+1}}{1-\alpha}\right)^2 \\ &= \frac{\sigma^2}{(1-\alpha)^2}\left(n - 2\frac{\alpha(\alpha^n-1)}{\alpha-1} + \frac{\alpha^2(\alpha^{2n}-1)}{\alpha^2-1}\right). \end{aligned} \tag{6.27}$$

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

$$\sigma_t^2(n) = \sigma_e^2(n) + \left(\frac{\alpha^{n+1}-\alpha}{\alpha-1}\right)^2 \omega^2(L), \tag{6.28}$$

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.

Remark 3. Although the analysis described in this section is tailored to the AR model, the procedure can be adapted to different signal models. The core step consists in characterizing the expected error at lag n (as in Eq. (6.22)). To this purpose, it is necessary to identify the temporal correlation of the signal and the best estimate that can be made (based on the correlation model) so that the estimation error is minimized. Also the distortion $v(L)$ introduced by data compression can be modeled differently, depending on the actual algorithm used for compression.

6.5.2 Channel access scheme

As discussed at the beginning of Sec. 6.5.1, the goal is to design a transmission strategy that, given a desired level of QoS, prolongs the lifetime of the devices. The proposed scheme assumes a duty-cycled operation mode, where transmissions are performed after each sampling.

As mentioned, 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 (under the QoS constraint) the mean period between its consecutive transmissions. In the following, this last parameter, which is the reciprocal of the persistency constant `P`, is referred to as *mean sleeping period*, and denoted by $S(L)$, being a function of the packet size L.

In particular, capitalizing on both the sampling and data compression approaches described in the introduction, we need to determine i) the mean du-

ration $S^\star$ of the sleeping window,[7] 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. Viceversa, 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 reported in Sec. 6.5.1, the reconstruction error $\mathcal{E}_n$ is a normal r.v., therefore its magnitude, $|\mathcal{E}_n|$, is half-normally distributed with scale parameter $\sigma_t(n)$. Also, 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, we can consider the error at the end of a window, $\mathcal{E}_W$, and define the QoS as an upper threshold v_{th} on the mean probability that $|\mathcal{E}_W|$ exceeds a given value ε.

The optimization problem can then be formulated as follows

$$S^\star \triangleq \max_{L \in \mathcal{L}} S(L)\,, \tag{6.29a}$$

$$\text{subject to:} \qquad \mathrm{E}\left[\Pr\left(|\mathcal{E}_{W-1}| > \varepsilon\right)\right] < v_{\text{th}}\,, \tag{6.29b}$$

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 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 $\mathrm{p_{tx}} = p_s(L)/S(L)$. Therefore, the condition (6.29b) can be expressed as

$$\sum_{w=1}^{\infty} \mathrm{p_{tx}} \left(1 - \mathrm{p_{tx}}\right)^{w-1} Q_{\text{hf}}\left(\varepsilon; \sigma_t(w-1)\right) < v_{\text{th}}\,; \tag{6.30}$$

where $Q_{\text{hf}}(\cdot)$ is the complementary CDF of the half-normal distribution and $\sigma_t(\cdot)$ is the square root of the variance given by (6.28), with $\sigma_t(0) = 0$.

To determine $S^\star$ (and the associated $L^\star$), an iterative approach is proposed, which is detailed in Algorithm 2. For each possible $L \in \mathcal{L}$, the corresponding optimal mean sleeping duration $S(L)$ is computed. This is obtained through an alternated optimization of the duty cycle and determining the corresponding success transmission probability, until convergence. Then, $L^\star = \operatorname{argmax} S(L)$ is chosen, which yields $S^\star = S(L^\star)$ (Line 8). The iterative procedure to de-

[7] 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 2 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: $S \leftarrow \max\{S(L) : \text{cond. (6.30) holds true}\}$
6: $p_s(L) \leftarrow$ eq. (6.6) with $\mathtt{P} = 1/S$
7: $\mathbf{S}(L) \leftarrow S$
8: $S^\star = \max \mathbf{S}(L)$, $L^\star = \operatorname{argmax} \mathbf{S}(L)$

Interference and communication parameters		
Slot duration	T	0.1 s
Density of sensor devices	λ	{500, 1000} devices/km^2
Cell radius	$r_{\max}$	500 m
Received power[8]	$\overline{\mathrm{P}}_{\mathrm{rx}}$	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 $\mathrm{E}\left[\Pr\left(\mathcal{E}_n > \varepsilon \,\middle|\, S(L)\right)\right]$	v_{th}	0.3

Table 6.2: Simulation parameters

rive $S(L)$ for a given L corresponds to the instructions in the `while` cycle in Algorithm 2. Initially, the success probability is set to 1, as if there were no interference, and the algorithm determines the corresponding mean sleeping duration S, i.e., the one that satisfies the QoS requirement (6.30) when $p_s(L) = 1$ (Line 3). 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 $p_s(L)$ is hence updated for the current value of the mean sleeping period S, evaluating (6.6) with $\mathtt{P} = 1/S$ (Line 6). The procedure is repeated iteratively until convergence (Lines 4-6).

6.5.3 Numerical evaluation

Fig. 6.4 shows the optimal value of the mean sleeping period, $S^\star$ when varying the value of ε, for two values of the node density λ_s. We can observe that the sleep period grows with ε, as expected since the QoS constraint becomes

[8] 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.

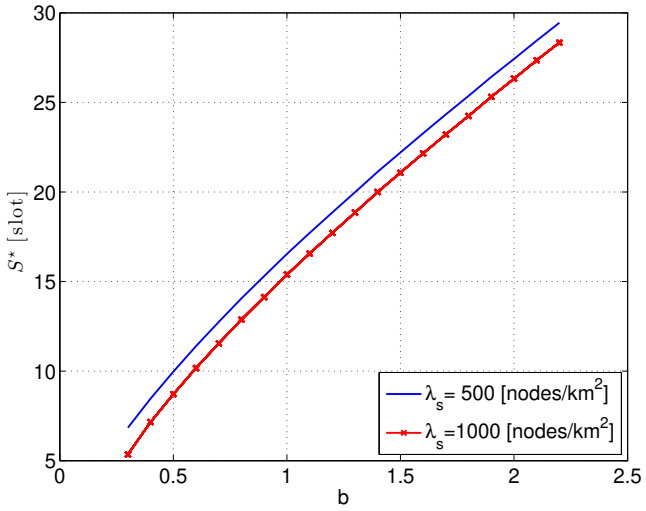


Figure 6.4: Optimal mean sleeping period $S^\star$ for different thresholds b, for different node densities λ_s.

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 ε.

To better assess the performance of the proposed strategy, it is compared 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 $\rho(\varepsilon)$. In order to get similar results between the proposed and the naive strategies in terms of QoS (i.e., equal $\Pr(|\mathcal{E}_n| > \varepsilon)$), $\rho(\varepsilon)$ grows from 1.7 to 1.95 as ε is varied from 0.3 to 2.2. The simulation parameters are reported in Tab. 6.2.

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. The energy efficiency ε is defined 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

$$\varepsilon = \frac{\mathrm{P_{tx}}T + E_s}{S^\star} \ . \tag{6.31}$$

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. 6.5, where we can see 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.

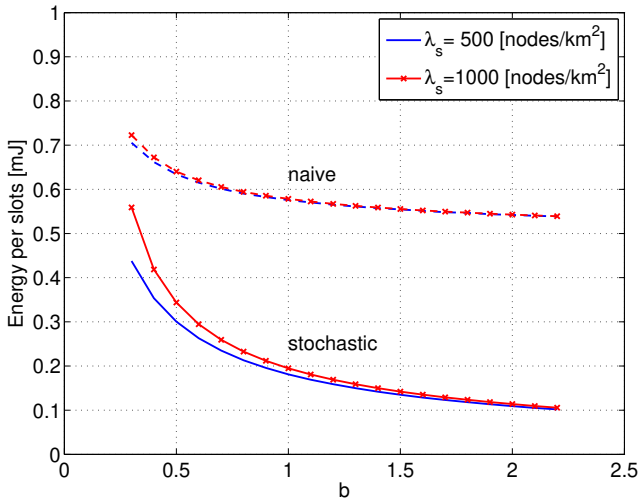


Figure 6.5: Average energy used per slot for sensing and transmission, for different node densities λ_s. Solid lines: proposed protocol (stochastic). Dashed lines: naive protocol.

6.6 A random strategy for the reporting of instantaneous values

In the following, we aim at designing a compression and communication protocol that targets a given QoS, measured in terms of reconstruction error at the FC, while minimizing the devices energy consumption. The goal is to detect when the monitored signal exceeds a predefined threshold. The use case is very similar to that in Sec. 6.4, but with a major difference. While in Sec. 6.4 a deterministic channel access strategy has been considered, here, at each time slot, a device transmits with a certain probability that depends on the time elapsed since its last correct transmission and the expected error at the FC. Differently from Sec. 6.4, here data compression is not used; the proposed strategy is a mix between sampling compression and communication compression. In fact, the devices solely sense the environment when they want to transmit, thus in a probabilistic way. Moreover, this framework is general and can accommodate different scenarios: it is possible to use different functions for the transmission probability and different models for the monitored signal, where the only requirement is to have a characterization of the expected prediction error over time.

6.6.1 System model

As in the previous sections, each sensor node tracks a temporal signal of interest $\{x_n\}$, where $n \in \mathbb{N}$ is the slot index. Different devices may measure different signals, but the node index is omitted for the sake of a simpler 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. Also, it is assumed that each sensing operation requires a fixed amount of energy.

6.6.2 Channel access scheme

We consider a probabilistic slotted ALOHA channel access scheme. Each device has a probability $\mathrm{p_{tx}}(j;\boldsymbol{\varphi})$ of waking up to transmit a packet, which depends on the number of slots j 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 $\{\mathrm{p_{tx}}(\,\cdot\,;\boldsymbol{\varphi})\}$ that guarantee a desired level of accuracy in the tracking process and, at the same time, optimize the energy usage.

In Sec. 6.6.1 it has been assumed that each sensing and transmission operation requires the same amount of energy. Consequently, maximizing the lifetime of a device is equivalent 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 v_{th} on the average outage probability. The outage probability after j slots since the last received data is defined as the probability that the squared signal prediction error exceeds a threshold ε, i.e.,

$$p_{\mathrm{out}}(j) = \Pr\left[|x_{n+j} - \hat{x}_{n+j}|^2 > \varepsilon\right], \tag{6.32}$$

where n is the last slot where a sample was correctly received by the FC, while x_{n+j} and $\hat{x}_{n+j}$ are the actual and the estimated signal in slot $n+j$, respectively. Note that we suppose that the outage probability depends only on the lag j and not on the absolute time n 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 $p_{\mathrm{out}}(0) = 0$. So, formally, our optimization problem is

$$\boldsymbol{\varphi}^{\star} = \underset{\boldsymbol{\varphi}}{\operatorname{argmax}}\ \mathrm{E}\left[\tau_{\mathrm{tx}} | \mathrm{p_{tx}}(\,\cdot\,;\boldsymbol{\varphi})\right] \tag{6.33a}$$

$$\text{s.t.}\ \ \mathrm{E}_j\left[p_{\mathrm{out}}(j)\right] < v_{\mathrm{th}} \tag{6.33b}$$

where $\mathrm{E}[\cdot]$ denotes the expectation operator, that, with the subscript j, is intended to be applied to the distribution of the random variable j. The optimal transmission probability function is then given by $\mathrm{p_{tx}}(\,\cdot\,,\boldsymbol{\varphi}^{*})$. In Sec. 6.6.3, a model for the time evolution of the signal $\{x_k\}$ is proposed, and the corresponding outage probability is derived. The QoS constraint (6.33b) can be expressed as

$$\overline{p}_{\mathrm{out}} = \mathrm{E}_j\left[p_{\mathrm{out}}(j)\right] = \sum_j p_{\mathrm{out}}(j)\pi_j < v_{\mathrm{th}} \tag{6.34}$$

where π_j is the probability that, at any given time, the last successful transmission happened j slots before.

Solving Problem (6.33) is not trivial. First, we determine the expression of the QoS constraint as a function of the transmission probability function $\mathrm{p_{tx}}(\,\cdot\,;\boldsymbol{\varphi})$. To this end, we have to derive the expression of the number of slots τ needed to *successfully* deliver a message (hence, $\tau \geq \tau_{\mathrm{tx}}$) in terms of the transmission probabilities and the probability p_s of successful transmission. Then,

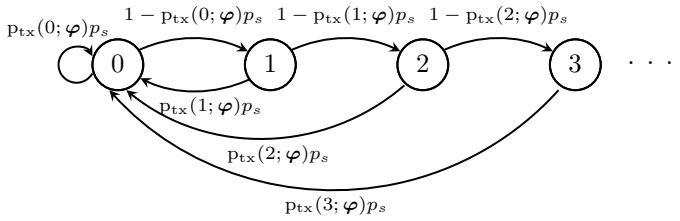


Figure 6.6: Markov chain for the node state.

using the results of Sec. 6.3, we express p_s as a function of the mean transmission probability, which, in turn, depends on τ. An alternate optimization allows us to derive both p_s and τ, which are interrelated. Finally, we leverage the knowledge about the success probability to obtain the objective function, i.e., the mean time between two successful transmissions, given the transmission probability function $\mathrm{p_{tx}}(\,\cdot\,;\boldsymbol{\varphi})$.

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 Markov Chain (MC) in Fig. 6.6. 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 $\mathrm{p_{tx}}(i;\boldsymbol{\varphi})p_s$ and $1-\mathrm{p_{tx}}(i;\boldsymbol{\varphi})p_s$, respectively, where we recall that $\mathrm{p_{tx}}(i;\boldsymbol{\varphi})$ 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.

Assuming $\mathrm{p_{tx}}(j;\boldsymbol{\varphi})$ and p_s are given, the probability mass distribution of τ equals the steady-state probability vector of the MC in Fig. 6.6. Writing the equilibrium equations, we get

$$\begin{aligned}
&\pi_1 = \pi_0(1-\mathrm{p_{tx}}(0;\boldsymbol{\varphi})p_s)\\
&\pi_2 = \pi_1(1-\mathrm{p_{tx}}(1;\boldsymbol{\varphi})p_s) = \pi_0\prod_{k=0}^{1}(1-\mathrm{p_{tx}}(k;\boldsymbol{\varphi})p_s)\\
&\vdots\\
&\pi_i = \pi_{i-1}(1-\mathrm{p_{tx}}(i-1;\boldsymbol{\varphi})p_s) = \pi_0\prod_{k=0}^{i-1}(1-\mathrm{p_{tx}}(k;\boldsymbol{\varphi})p_s) \qquad (6.35)
\end{aligned}$$

with the additional normalization constraint

$$\sum_{i=0}^{\infty}\pi_i = 1\,. \qquad (6.36)$$

By combining equations (6.35) and (6.36), we obtain

$$\pi_0 = \left\{1+\sum_{i=1}^{\infty}\prod_{k=0}^{i-1}(1-\mathrm{p_{tx}}(k;\boldsymbol{\varphi})p_s)\right\}^{-1}\,. \qquad (6.37)$$

while π_i for any $i > 0$ is given by (6.35).

Observation. The sum in (6.37) has an infinite number of terms, which in practice is difficult to evaluate for an arbitrary choice of the transmission probability function $\mathrm{p_{tx}}(\,\cdot\,;\boldsymbol{\varphi})$. However, when the system is stable, the sum in (6.37) converges, so that it can be approximated with the desired level of accuracy by considering a finite number $\tilde{n}$ of terms.

Success probability p_s

Consider Eq. (6.6), where $\lambda_s(\mathtt{P}) = \lambda \mathrm{E}_n\left[\mathrm{p_{tx}}(j;\boldsymbol{\varphi})\right]$ and Γ° does not depend on the packet size L, which is now fixed. We have

$$p_s = \exp\left(-\lambda\pi R_c^2\, \mathrm{E}_j\left[\mathrm{p_{tx}}(j;\boldsymbol{\varphi})\right] \frac{\Gamma^\circ}{1+\Gamma^\circ} - \Gamma^\circ \frac{N_s}{\overline{\mathrm{P}}_{\mathrm{rx}}}\right), \tag{6.38}$$

which depends on the mean transmission probability

$$\mathrm{E}_n\left[\mathrm{p_{tx}}(n;\boldsymbol{\varphi})\right] = \sum_{i=0}^{\infty} \mathrm{p_{tx}}(i;\boldsymbol{\varphi})\pi_i\,. \tag{6.39}$$

The steady-state probabilities $\boldsymbol{\pi} = [\pi_0, \pi_1, \dots]$ are computed as described earlier. Notice, however, that $\boldsymbol{\pi}$ and p_s are strictly intertwined, as one is needed in order to derive the other and viceversa. To deal with this issue, we can use a fixed-point approach: we initially set $p_s = 1$ and compute the corresponding steady-state probabilities as in (6.35), which are used to update the probability of successful transmission as in (6.38), and so forth until convergence. We prove in the following that such point of convergence exists, therefore the iterative method is always able to terminate.

Proof of the existence of the point of convergence. Eq. (6.38) can be written in the following format

$$p_s = A \exp\left(-B\mathcal{A}(p_s)\right) \tag{6.40}$$

where $A > 0$ and $B > 0$ collect the constant terms and coefficients in (6.38), while $\mathcal{A}(p_s) = \sum_{i=0}^{\infty} \mathrm{p_{tx}}(i,\boldsymbol{\varphi})\pi_i$ is the "area" covered by the discrete function given by the Hadamard (entrywise) product of the vectors $\boldsymbol{p}_{\mathrm{tx}}(\boldsymbol{\varphi}) = [\mathrm{p_{tx}}(0,\boldsymbol{\varphi}), \mathrm{p_{tx}}(1,\boldsymbol{\varphi}), \dots]$ and $\boldsymbol{\pi}$, the latter depending on p_s.

We observe from (6.35) that the steady-state distribution $\boldsymbol{\pi}$ is such that $\pi_i \geq \pi_{i+1}$ for any i. Furthermore, consider two values p_s and $p_s' < p_s$; it is

$$\frac{\pi'_{i+1}}{\pi_{i+1}} = \frac{\pi'_i(1 - p'_s\,\mathrm{p_{tx}}(i,\boldsymbol{\varphi}))}{\pi_i(1 - p_s\,\mathrm{p_{tx}}(i,\boldsymbol{\varphi}))} \geq \frac{\pi'_i}{\pi_i} \tag{6.41}$$

for any i. Because of the normalization condition, we then have a state k such that $\pi'_i/\pi_i \leq 1$ for all $i \leq k$, and $\pi'_i/\pi_i \geq 1$ for all $i > k$. As p_s progressively decreases towards zero, the state drift of the Markov Chain increases, so that the probabilities of being in the lower states decrease, while those of being in the higher states increase, i.e., the steady-state distribution tends to flat.

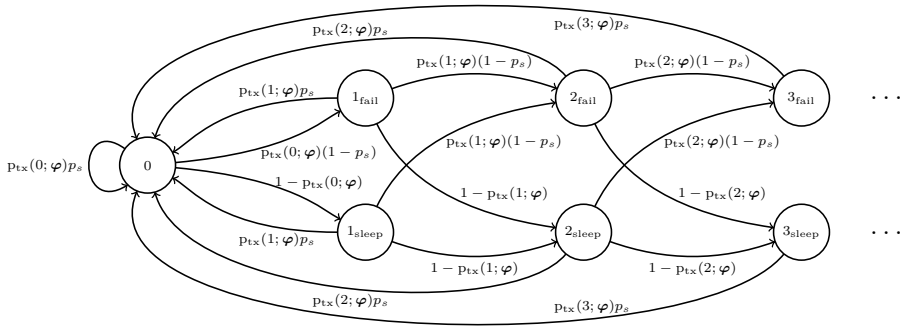


Figure 6.7: Markov chain for the node state with explicit indication of *failure* or *sleep* status.

Also, we have

$$\mathcal{A}(p_s) - \mathcal{A}(p_s') = \sum_{i=0}^{\infty} \mathrm{p_{tx}}(i,\boldsymbol{\varphi})(\pi_i - \pi_i') = \\ = \sum_{i=0}^{k} \mathrm{p_{tx}}(i,\boldsymbol{\varphi})(\pi_i - \pi_i') + \sum_{i=k+1}^{\infty} \mathrm{p_{tx}}(i,\boldsymbol{\varphi})(\pi_i - \pi_i')\,. \quad (6.42)$$

Focusing on the second term in (6.42) and considering that (i) $\mathrm{p_{tx}}(i;\boldsymbol{\varphi})$ is monotonic non-decreasing with i, and (ii) $\pi_i - \pi_i' \leq 0$ for $i > k$, it results that $\sum_{i=k+1}^{\infty} \mathrm{p_{tx}}(i,\boldsymbol{\varphi})(\pi_i - \pi_i') \leq \mathrm{p_{tx}}(k+1,\boldsymbol{\varphi}) \sum_{i=k+1}^{\infty}(\pi_i - \pi_i')$. Since the steady-state probabilities sum to 1, we can also see that $\sum_{i=k+1}^{\infty}(\pi_i - \pi_i') = (1 - \sum_{i=0}^{k} \pi_i) - (1 - \sum_{i=0}^{k} \pi_i') = -\sum_{i=0}^{k}(\pi_i - \pi_i')$. Thus, (6.42) becomes

$$\mathcal{A}(p_s) - \mathcal{A}(p_s') \leq \sum_{i=0}^{k} (\pi_i - \pi_i') \left(\mathrm{p_{tx}}(i,\boldsymbol{\varphi}) - \mathrm{p_{tx}}(k+1,\boldsymbol{\varphi})\right) \leq 0\,, \quad (6.43)$$

i.e., the area function turns out to be monotonic non-increasing with p_s. Therefore, $f_2(p_s) \triangleq A \exp\left(-B\mathcal{A}(p_s)\right)$ is monotonic non-decreasing with p_s.

Since $\mathcal{A}(p_s) \geq 0$ and $A = \exp\left(-\Gamma^{\circ} \frac{N_s}{\overline{\mathrm{P}}_{\mathrm{rx}}}\right) < 1$, it results that $f_2(p_s) < 1$. Also, $\mathcal{A}(p_s) = \sum_{i=0}^{\infty} \mathrm{p_{tx}}(i,\boldsymbol{\varphi})\pi_i \leq \sum_{i=0}^{\infty} \pi_i < \infty$, so that $f_2(p_s) > 0$.

Functions $f_1(p_s) \triangleq p_s$ and $f_2(p_s)$ are continuous in the interval $p_s \in (0,1)$. Function $f_1(p_s)$ is a straight line from 0 to 1, while $f_2(p_s)$ is a monotonically non-increasing function from $x_1 > 0$ to $x_2 < 1$, thus have at least one intersection. □

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 analyse the objective function.

Mean time $\mathrm{E}[\tau_{\mathrm{tx}}]$ between transmissions

The objective function (6.33a) is the expected time between two consecutive transmission attempts (regardless of their outcome). In order to derive its ex-

pression in terms of the transmission probabilities $\{\mathrm{p_{tx}}(j;\boldsymbol{\varphi})\}$, we introduce an MC equivalent to that of Fig. 6.6, 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 remains unchanged. As shown in Fig. 6.7, 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 $\mathrm{p_{tx}}(i;\boldsymbol{\varphi})p_s$.
- State $(i+1)_{\mathrm{fail}}$ 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 $\mathrm{p_{tx}}(i;\boldsymbol{\varphi})(1-p_s)$.
- State $(i+1)_{\mathrm{sleep}}$ if the device sleeps in the current slot. This happens with probability $1-\mathrm{p_{tx}}(i;\boldsymbol{\varphi})$.

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

$$\begin{cases} \tilde{\pi}_0 = \pi_0 \\ \tilde{\pi}_{i_{\mathrm{fail}}} = \mathrm{p_{tx}}(i-1;\boldsymbol{\varphi})(1-p_s)\,\pi_{i-1} \qquad i>0 \\ \tilde{\pi}_{i_{\mathrm{sleep}}} = (1-\mathrm{p_{tx}}(i-1;\boldsymbol{\varphi}))\,\pi_{i-1} \qquad i>0\,. \end{cases} \tag{6.44}$$

Note that $\tilde{\pi}_{i_{\mathrm{fail}}} + \tilde{\pi}_{i_{\mathrm{sleep}}} = (1-\mathrm{p_{tx}}(i-1;\boldsymbol{\varphi})p_s)\,\pi_{i-1} = \pi_i$ for $i > 0$ (see (6.35)).

To compute the expected time $\mathrm{E}[\tau_{\mathrm{tx}}]$ between two transmission attempts, we introduce $T_{\mathrm{tx}}(i)$, 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 these states correspond to a 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.,

$$\Pr[T_{\mathrm{tx}}(i) \geq h] = \prod_{k=0}^{h-1}(1-p_{\mathrm{tx}}(i+k)), \tag{6.45}$$

This yields

$$\mathrm{E}[T_{\mathrm{tx}}(i)] = \sum_{h=1}^{+\infty}\Pr[T_{\mathrm{tx}}(i)\geq h] = \sum_{h=1}^{+\infty}\prod_{k=0}^{h-1}(1-p_{\mathrm{tx}}(i+k)). \tag{6.46}$$

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

$$\mathrm{E}[\tau_{\mathrm{tx}}] = A\left(\sum_{i=1}^{+\infty}\mathrm{E}[T_{\mathrm{tx}}(i)]\,\tilde{\pi}_{i_{\mathrm{fail}}} + \mathrm{E}[T_{\mathrm{tx}}(0)]\,\tilde{\pi}_0\right) \tag{6.47}$$

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

$$A = \frac{1}{\sum_{i=1}^{+\infty}\tilde{\pi}_{i_{\mathrm{fail}}} + \pi_0} \tag{6.48}$$

In this way, the objective function (6.33a) is completely defined.

Summary of relations

The following is a guideline through the entire procedure to explicitly write the optimization problem described in (6.33) and obtain a numerical evaluation of the optimal policy.

We introduce an MC to model the number of slots since the last successful transmission, and derive its steady-state probabilities $\boldsymbol{\pi}$, reported in (6.35) and (6.37). Such probabilities depend on the expected outcome of a transmission; we can thus use a stochastic geometry reasoning to derive the success probability p_s, which is reported in (6.38) and depends on the mean transmission probability $\mathrm{E}_j\left[\mathrm{p_{tx}}(j;\boldsymbol{\varphi})\right]$. 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, we adopt a fixed-point iteration approach 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 (6.34).

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 in (6.47) and (6.48).

6.6.3 Proposed scenario

The framework described and analysed in Sec. 6.6.2 is rather general and can accommodate different scenarios. In particular, it is possible to employ arbitrary signal models and transmission probability functions. Here, a specific model for the monitored signals is considered and it is employed to derive the corresponding outage probability, which is needed to define the QoS constraint (6.33b). Also, a possible parameterized model for the transmission probability function is proposed.

Signal model

To keep the analysis simple, here we consider the AR model with degree 1 already introduced in the previous sections. Therefore, the time series evolves as

$$x_n = \alpha x_{n-1} + u_k\,, \quad n > 0\,, \tag{6.49}$$

with α a non zero constant and $u_n \sim \mathcal{N}(0,\sigma^2)$ a zero mean Gaussian innovation term, with variance σ^2. As before, we set $|\alpha| < 1$, so that $\{x_n\}$ is a stable process.

The signal in slot $n+j$ can then be expressed in terms of the signal in slot n as $x_{n+j} = \alpha^j x_n + w_j$, where $w_j = \sum_{k=1}^{j} \alpha^{j-k} u_{n+k}$. The signal estimated by the FC is $\hat{x}_{n+j} = \alpha^j x_n$, while the estimation error w_j is a zero-mean normal r.v., $w_n \sim \mathcal{N}(0,\sigma_j^2)$, with variance reported in (6.13).

In conclusion, the squared error after j steps from the last known value, $|x_{n+j} - \hat{x}_{n+j}|^2$, follows a Gamma distribution, $w_j^2 \sim \mathrm{Gamma}(K_j,\theta_j)$, where the shape and scale parameters are $K_j = 1/2$ and $\theta_j = 2\sigma_j^2$, respectively. The outage probability (6.32) becomes

$$p_{\mathrm{out}}(j) = 1 - F_{w_j^2}(\varepsilon), \tag{6.50}$$

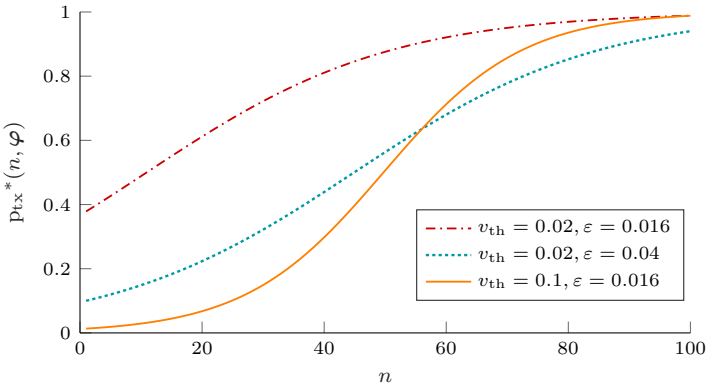


Figure 6.8: Transmission probability function resulting from the optimization procedure ($\lambda = 0.1$ devices/m^2).

where $F_{w_j^2}(\cdot)$ is the cumulative density function of the squared estimation error w_j^2 at lag j.

Although the AR model is very simple, it can provide a good representation of real-world time-correlated time series. The performance of the proposed algorithm used with a real time series, modeled as an AR signal, is shown in Sec. 6.6.4.

Transmission probability function

The model in Sec. 6.6.2 assumes that the transmission probability function can be defined by a set of parameters $\boldsymbol{\varphi}$. Intuitively, the transmission probability should increase (or, at least, not decrease) with the lag j in order to limit the estimate error that tends to grow with j. 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. 6.6, as explained previously.

We can, for example, model the transmission probability function as a generalized sigmoid

$$\mathrm{p_{tx}}(j; \boldsymbol{\varphi}) = \frac{1}{1 + e^{-\varphi_1 (j - \varphi_2)}}, \quad j \geq 0; \tag{6.51}$$

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 distributions, thus being particularly suitable for our purpose. However, we remark that this framework can be applied to any other parametric probability distribution function. Fig. 6.8 shows some examples of the curve $\mathrm{p_{tx}}(j; \boldsymbol{\varphi})$ for different QoS constraints when the device density is $\lambda = 0.1$ devices/m^2.

Solution

From (6.13) and (6.50), it is apparent that, after each successful transmission, the outage probability steadily grows in time, till the next successful trans-

Interference and communication parameters		
Time slot duration	T	100 ms
Cell radius	$r_{\max}$	100 m
Received power[9]	$\overline{\mathrm{P}}_{\mathrm{rx}}$	1 nW
Transmission bandwidth	B_{w}	125 kHz
Noise power	N_s	$1.25 \cdot 10^{-15}$ W
Signal model and QoS parameters		
AR model parameters	α	0.99
	σ^2	0.001
Initial value	x_0	0.8
Threshold on $\mathrm{E}_\tau\left[p_{\mathrm{out}}(\tau)\right]$	v_{th}	0.1
Error threshold	ε	0.04
Kulau et al. strategy [117]		
Maximum sleep time	$t_{\max}$	50 slots
Weighting exponent	ϕ_{bb}	2
Sliding window size	n_s	30
EDSAS [119]		
EWMA coefficient – long	ρ_{long}	0.2
EWMA coefficient – short	ρ_{short}	0.8
EWMA reset threshold	η	1

Table 6.3: Simulation parameters.

mission, which occurs after τ steps. However, the relation between τ and the optimization variable $\boldsymbol{\varphi}$ is quite complex. Also, the optimization problem (6.33) is in general not convex, so that analytical solutions cannot be found. Consequently, the solution to the problem has been found by using numerical methods, specifically the routines available in the MATLAB Optimization Toolbox.

6.6.4 Numerical evaluation

The proposed strategy has been validated by means of simulations to prove the scalability of this approach and to show the improvements compared to the state of the art. In particular, the performance has been studied in terms of QoS, i.e., outage probability, and energy efficiency. The values of the parameters used in the simulations are reported in Tab. 6.3. 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.

In Fig. 6.9 we can see an example of the estimate of the proposed algorithm for $\lambda = 0.001$ devices/m^2 and $\varepsilon = 0.08$. In the following, the effect of the device density on the performance is analysed and the proposed strategy is compared against two other state-of-the-art approaches from the literature.

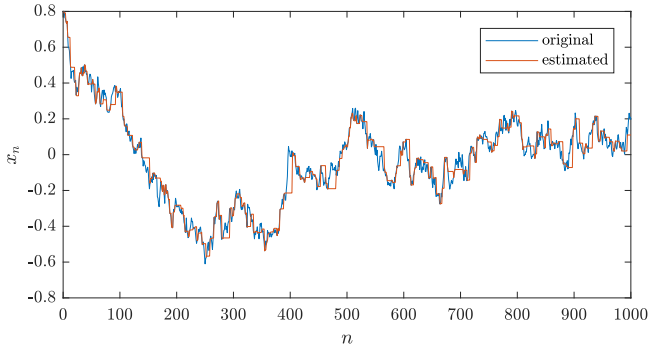


Figure 6.9: Example of original and estimated time series with $v_{\text{th}} = 0.1$.

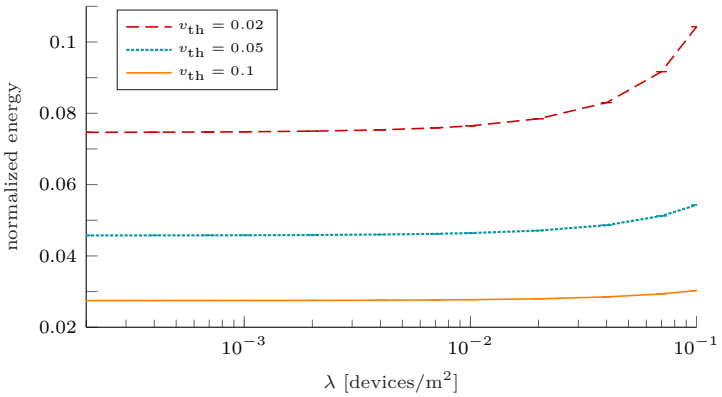


Figure 6.10: Energy consumed for increasing device density.

Scalability on device density

To test the strategy in a massive access scenario, the outage probability is evaluated for increasing values of the devices density λ. The parameters of the AR signal used for the simulation are given in Tab. 6.3. 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. 6.10 shows the corresponding normalized energy consumption. 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.

[9]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.

	ε	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 6.4: Parameters for Kulau et al. and EDSAS strategies to yield the same error as the proposed strategy for $\lambda = 10^{-3}$ devices/m^2.

Comparison with previous strategies

Here, the proposed strategy is compared with two other techniques in the literature that address directly our use case, and that represent the typical approaches to this problem. One is the strategy by Kulau et al., already introduced previously. The other strategy, named Exponential Double Smoothing-based Adaptive Sampling (EDSAS) [119], 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 δ. EDSAS starts with a 1-step prediction and, as long as the prediction error stays below δ, the sampling interval K is increased by 1 (until $S_{\max}$); when the error exceeds δ, K is decremented by 1. In more detail, the strategy uses the Wright's extension to the Exponential Double Sampling technique, where a K_n-step prediction at time n is calculated as $\hat{x}_{n+K_n} = Y_n + K_n M_n$. The coefficients Y_n and M_n are, respectively, the estimate and the trend of the signal at time n, and they are given by

$$Y_n = (1 - V_n)(Y_{n'} + K_{n'} M_{n'}) + V_n x_n \; ; \tag{6.52}$$

$$M_n = (1 - U_n) M_{n'} + U_n (Y_n - Y_{n'})/K_{n'} \; , \tag{6.53}$$

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

$$V_n = V_{n'}/(b_n + V_{n'}) \, ; \qquad b_n = (1 - \alpha_{\mathrm{E}})^{k_{n'}} \; ; \tag{6.54}$$

$$U_n = U_{n'}/(d_n + U_{n'}) \, ; \qquad d_n = (1 - \beta_{\mathrm{E}})^{k_{n'}} \; , \tag{6.55}$$

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

The algorithm includes 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 ($S_n = \rho x_{n'} + (1 - \rho) S_{n'}$) 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.

Simulation outcome Fig. 6.11 shows the outage probability as ε 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, the values of

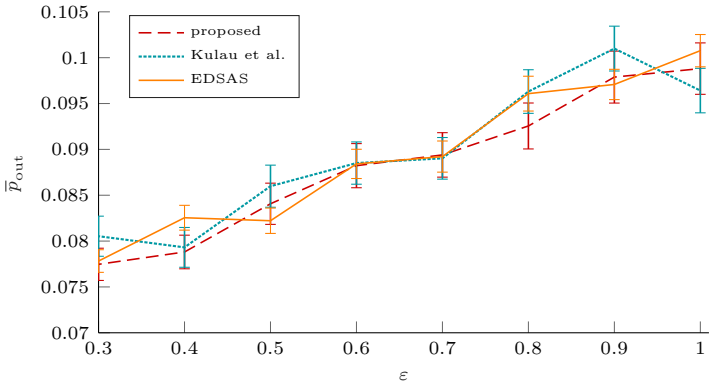


Figure 6.11: Outage probability of the considered schemes for $\lambda = 10^{-3}$ devices/m^2, with 95% confidence intervals.

b_{bb} (for the Kulau et al. strategy), α_{E}, β_{E}, and $S_{\max}$ (for EDSAS) are set so that the outage probability is almost the same as for the proposed strategy. The detailed values of these parameters are reported in Tab. 6.4. Moreover, since the two abovementioned techniques are not tailored to AR signals, real world time series are used in these simulations.[10] Note that, to use the proposed strategy with a 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. The resulting outage probability is better than the imposed one due to the impossibility to completely capture the real signal behaviour with an AR model, causing a small efficiency loss.

In Fig. 6.12 we can observe the energy efficiency of the considered protocols. 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, it evaluates 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, the proposed strategy explicitly takes into account the effect of the interferers on the ability of the FC to estimate the time series. Therefore, while the other approaches neglect the effect of collisions, so that a lower transmission rate will always result in a reduction of the estimation errors (at the cost of higher energy consumption), the proposed strategy keeps into account that a lower transmission rate 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 the [119] does not specify how to set the parameters, and thus it is necessary to manually tune the algorithm so that the outage probability (Fig. 6.11) matches that of the proposed algorithm.

[10] The used time series come 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

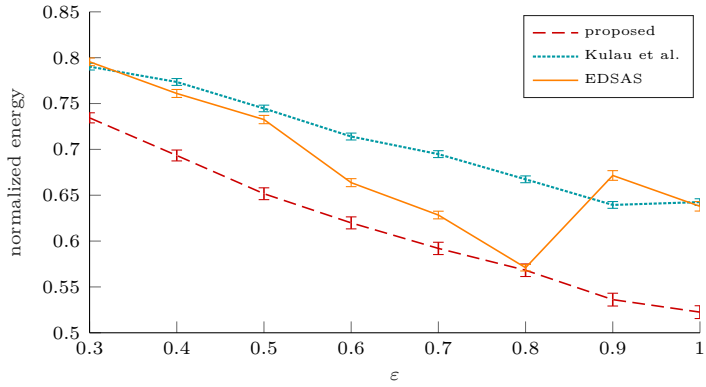


Figure 6.12: Energy consumed by the considered schemes for $\lambda = 10^{-3}$ devices/m^2, with 95% confidence intervals.

6.7 Conclusions

In this study, three novel channel access schemes for constrained devices in WSNs have been presented. The common goal was to provide an accurate estimate of the monitored signal at the FC while maximizing the energy efficiency. The three techniques operate in different scenarios: for the techniques described in Sec. 6.4 and Sec. 6.6 the interest was in instantaneous values of a time series, therefore they aim at minimizing the error for each sample, while the technique in Sec. 6.5 has been designed to minimize the cumulative error of the time series. To meet the required QoS level, these techniques operate at multiple levels, by balancing the optimal time interval between transmissions and their optimal size. Unlike most of the state of the art techniques, the role played by interference is also considered, as collisions impinge on both the QoS and the energy consumption, especially in dense scenarios. This allows the proposed strategies to obtain better performance than state of the art algorithms, as shown in the numerical evaluation. The downside is the need for a more complex algorithm to determine the transmission parameters that, on the other hand, is still sufficiently lightweight to be run on embedded microcontrollers and makes it possible to fine-tune the tradeoff between QoS requirements and energy efficiency.

Part II

Machine learning techniques for CPS service optimization

Chapter 7

Cell Traffic Prediction Using Joint Spatio-Temporal Information

The evolution of cellular networks from 4G to 5G will rely on adaptive techniques in order to manage the increasing complexity of mobile systems [143]. Up to now, cellular networks were designed using worst-case dimensioning, but the increasingly strict capacity, latency and energy efficiency requirements, together with the lower profit margins, make a smarter approach appealing to network operators.

Anticipatory networking [144] is one of the most promising approaches in smart network adaptation: the idea is to exploit knowledge of the dynamics of the system in order to predict future network states and tailor the configuration to the expected profile. There are several possible contexts for the prediction, from a single user's channel gain [145] to large-scale mobility patterns [146]. This study uses a joint exploitation of spatio-temporal data to improve the prediction accuracy of standard regression methods. Several such methods from the literature are tested on a publicly available dataset, highlighting the advantages of the proposed approach.

7.1 Related work and contribution

In the scientific literature, cell load prediction techniques are studied because of the potential gain they can provide to the performance of the network in a wide range of scenarios, such as energy efficient communications and dynamic network planning. In [147] the authors propose to use prediction techniques based on traffic matrices collected for groups of Base Stations (BSs) under the same coordinator in order to optimize the sleeping time of network elements, while in [148] a classification and prediction method is applied to temporal information given by Call Data Records in order to decide when and where it is appropriate to deploy femto-cells. The spatio-temporal relation between cells is

analysed in [149], where insights on the predictability of the traffic in a cellular network are given; however, the authors do not attempt to predict future values of the cell load, but use large-scale traffic patterns to examine the correlation. The study in [150] uses traffic variations in cell neighborhoods, using a Markov decision process model, in order to enable energy saving techniques. There are other studies that consider the spatio-temporal context in cellular networks, but their focus is on the prediction of mobility of users [151,152]. These can be then exploited in association with some knowledge of the network topology, as done in [153].

The novelty of this work with respect to previous studies is that, here, machine learning techniques that exploit temporal and spatial data jointly are employed: a cell's future load depends not only on its previous values, but also on the loads of neighboring cells. This joint approach can improve the prediction accuracy, especially in the noisiest and most challenging cases. This work focus is on medium-term prediction with a range of tens of minutes; such a range is still usable for network optimization, but is not as noisy and unpredictable as short-term cell load.

7.2 Prediction techniques

All the techniques presented in this study are based on the exploitation of spatio-temporal data, which was first proposed by Ohashi *et al.* [154]. In order to jointly consider the spatial and temporal data, we need to define the concept of *spatio-temporal neighborhood*. If a cell at a given instant is characterized by its position in space and time, given by the vector (x, y, t), we define the distance between two points as

$$d_{i,j} = \sqrt{\left(\frac{x_i - x_j}{d_0}\right)^2 + \left(\frac{y_i - y_j}{d_0}\right)^2 + \alpha \left(\frac{t_i - t_j}{T}\right)^2}, \tag{7.1}$$

where d_0 is the inter-cell distance and T is the time interval between measurements. Note that the spatio-temporal distance between different instants is non-zero even if the cell is the same, i.e., the spatial distance is 0. The parameter $\alpha \geq 0$ is a weighting factor to combine the spatial and temporal measures.

The spatio-temporal neighborhood of a point m can then be defined as the set of the discrete points in the dataset whose distance from m is smaller than some radius β:

$$N_m^\beta = \{p : d_{m,p} < \beta\}. \tag{7.2}$$

The points belonging to the spatio-temporal neighborhood are contained in an ellipsoid in space-time, and, given the same β, a smaller α includes in the neighborhood points which are further away in time. The cell load values z_p of the points within the neighborhood can be used in the prediction. In addition to the pure values, we also use as input a series of indicators that capture some of the most relevant dynamics of the cell load, as in [154].

Three indicators are implemented, which are listed below:

- The *weighted mean* is an average of the cell load values in the neighbor-

hood, weighted by their spatio-temporal distance, and is given by:

$$\omega(N_m^\beta) = \frac{1}{|N_m^\beta|} \sum_{p \in N_m^\beta} \frac{z_p}{d_{m,p}} \tag{7.3}$$

- The *spread* is the standard deviation of the cell load values in the spatio-temporal neighborhood:

$$\sigma(N_m^\beta) = \sqrt{\frac{1}{|N_m^\beta|} \sum_{p \in N_m^\beta} (z_p - \bar{z})^2}, \tag{7.4}$$

  where $\bar{z}$ is the arithmetic mean of the cell load of all the points in the neighborhood.

- The *weighted tendency* is given by the ratio between the weighted means with two radii $\beta_1 < \beta_2$ (following [154], the chosen values are $\beta_2 = \beta = 2\beta_1$):

$$\tau(N_m^{\beta_1,\beta_2}) = \frac{\omega(N_m^{\beta_1})}{\omega(N_m^{\beta_2})}. \tag{7.5}$$

  This indicator summarizes the trend of the cell load as it approaches the target location. For example, if $\tau(N_m^{\beta_1,\beta_2}) > 1$, then the load on the closest points in time and space is larger than that of farther points.

While in [154] the indicators are added to a purely temporal prediction, in this work the cell load values of all the points in the spatio-temporal neighborhood are also used as predictors.

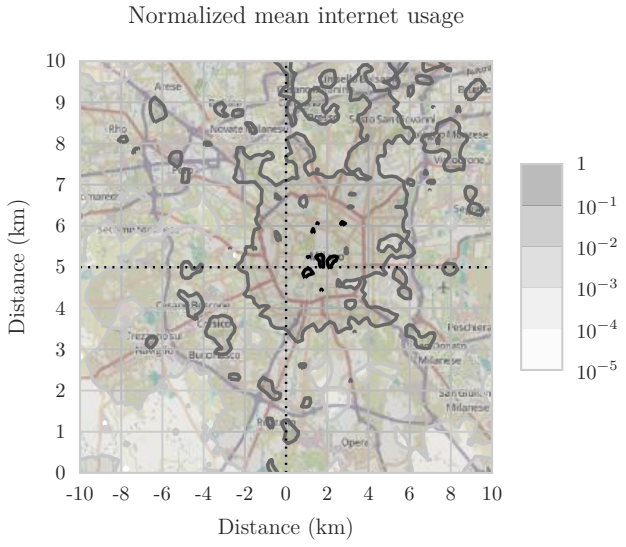


Figure 7.1: Normalized average internet usage map.

7.2.1 Prediction algorithms

The performance of several well-known prediction algorithms are tested using the input described above. The used algorithms represent the state of the art for prediction with time series [155, 156], and are briefly described below:

- The simplest tested method was the basic *multiple linear regression* [157], using least squares as a loss function.
- Given the highly variable nature of the data, some regularization techniques have been implemented in order to avoid the risk of overfitting; three methods of *regularized linear regression* have been used.
  - *Ridge regression* [158] is a shrinkage method that adds a square penalty to the least squares loss, weighted by a regularization parameter λ_R.
  - *Lasso regression* [159] is a shrinkage method very similar to ridge regression, but uses a linear penalty instead of a square penalty.
  - *Elastic net regression* [160] is a linear combination of the lasso and ridge regularization techniques, and is particularly useful when the number of predictors is larger than the number of observations and in the presence of highly correlated predictors.
- Support Vector Machines (SVMs) are mostly known as a classification tool, but they can be adapted to output real numbers, giving us the *Support Vector Regression (SVR)* technique [161]. This work uses SVR with a linear kernel, which has a regularization parameter C.
- *Random Forest (RF)* [162] is an ensemble estimator that consists of a number of regression trees, whose output is the average output of all the trees. For optimal performance, the trees' decisions should be uncorrelated, and dataset bagging and random training techniques are employed to obtain this property.
- *Artificial Neural Networks (ANNs)* [163] are well-known learning tools which use back-propagation to learn an objective function. In this work, the stochastic gradient descent method of back-propagation is used, using the tanh activation function.

7.3 Results

All the prediction methods described above were trained and tested using the *Telecom Italia Big Data Challenge 2014* dataset,[1] which contains the records of the internet usage for a grid of square cells with 200 m sides (which makes $d_0 = 200$ m in Eq. (7.1)) in the city of Milan, Italy, for the last two months of 2013. The data had a sampling period of 10 minutes (i.e., $T = 600$ s in Eq. (7.1)). The normalized mean internet usage is overlaid on a map of the city in Fig. 7.1.

[1] `https://dandelion.eu/datamine/open-big-data/`

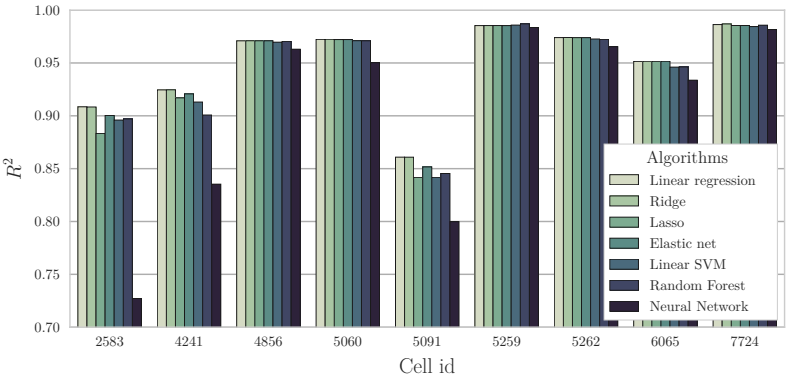


Figure 7.2: Performance of the tested regression methods.

For computational reasons, only the load of a small but representative subset of cells have been predicted, namely, the cells with id 2583, 4241, 4856, 5060, 5091, 5259, 5262, 6065 and 7724. These cells were selected because they are placed in different areas of the city and they show different traffic patterns. In particular, cells 2583 and 4241 have an average traffic that is close to the average traffic for the whole city, cells 5060, 5091 and 7724 show very high peak usage, and cells 4856, 5259, 5262 and 6065 have a very high average traffic.

The metric chosen for the results is the coefficient of determination R^2 [164], which is a commonly used metric in the regression literature, and gives an indication of how well the regression model describes the observed data.

7.3.1 Parameter optimization

All the parameters of the prediction algorithms were optimized by exhaustive search with 10-fold cross-validation, after dividing the dataset into training, validation and testing sets. The chosen values of the parameters are listed in Tab. 7.1.

Parameter	Value	Description
λ_R	[1.637e-6, 0.074]*	Ridge regularization parameter
λ_L	[1e-06, 4.665e-6]*	Lasso regularization parameter
$\lambda_{R,E}$	[0, 1.105e-5]*	Ridge regularization (elastic net)
$\lambda_{L,E}$	[0, 4.665e-6]*	Lasso regularization (elastic net)
C	[0.22, 34.081]*	SVR linear kernel penalization term
N_t	200	Number of RF trees
γ	10^{-3}	ANN learning rate
N_{iter}	10^4	Maximum ANN iterations
ε	10^{-10}	ANN convergence tolerance

*These parameters were optimized for each cell.

Table 7.1: Parameters used in the simulation.

The values of the spatio-temporal weighting factor α and of the neighborhood radius β were optimized for each cell and are listed in Tab. 7.2, for a number of

neighbors from 27 to 46.

Cell id	α	β	Number of neighbors
2583	0.25	2	27
4241, 4856	2.25	3	25
5060	0.09	2	46
5091	0.19	2	28
5259, 5262, 6065	0.12	2	37
7724	0.19	2	28

Table 7.2: Optimal neighborhood definition for each cell.

7.3.2 Prediction results

Fig. 7.2 shows the prediction accuracy on the test set for each regression method. The figure clearly shows that the ANN is not an accurate method, probably due to an insufficient training set size, whereas the other algorithms often have a similar performance. The reason is that the cell load can be easily predicted in most cells, and therefore the differences among different algorithms are minimal. On the other hand, in cells with poor prediction accuracy different methods show some performance difference. This reveals that, when the behavior of the load in a cell is less predictable, the prediction performance can be improved using different algorithms and additional context information. Indeed, the simple linear regression and ridge regression have a slightly better performance in cells 2583, 4241 and 5091, which are all located in peripheral areas of the city, close to major traffic roads or hubs (Via Gianbellino for cell 2583, the A1 highway for cell 4241, and Linate airport for cell 5091). In locations like these, with high mobility and bursty traffic, the benefit of combining spatial and temporal information is intuitive, and the performance improvement can be seen in Fig. 7.3. While only temporal or spatial data is sufficient in the highly predictable cells, the same 3 cells mentioned above show a marked improvement in the R^2 score when spatio-temporal data are considered jointly in the prediction. It is also worth noting that the use of temporal indicators does not result in a significant improvement by itself, but only when combined with the spatio-temporal neighborhood data.

The most accurate prediction methods are also the simplest: both training and parameter optimization for the linear, ridge, lasso and elastic net algorithms were significantly faster than for RF, SVR and ANN. This offsets the increased complexity due to the bigger size of the neighborhood due to the inclusion of the spatial dimension in its definition.

7.4 Conclusions and future work

This study applied several regression methods taken from the literature, combined with joint spatio-temporal information with indicators, to predict the future cell load on a 10 minutes scale. The data used to perform the training and evaluation of the different methods is from the Telecom Italia network in Milan.

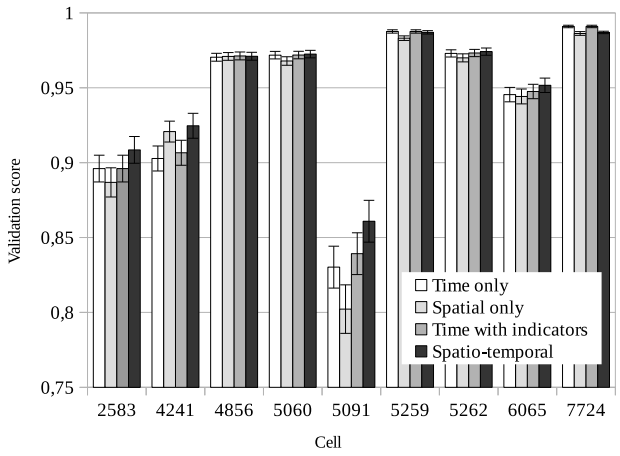


Figure 7.3: Performance of the prediction algorithms for different neighborhood definitions.

The results prove the usefulness of joint spatio-temporal information in the most difficult prediction scenarios, confirming the importance of context information for network optimization.

Future work on the prediction methods might consider the introduction of new indicators which could capture network-specific dynamics, along with a more in-depth study of the effect of the neighborhood size on the prediction accuracy.

Chapter 8

Introduction to adaptive video streaming

Nowadays, the most appealing but also the most bitrate demanding services are those providing high-quality videos to users playing real-time streaming or progressive download applications. The deployment of heterogeneous high-speed access points, such as LTE femto-cells and WiFi hotspots, dramatically increases the number of users accessing the network, which has an impact on the performance of both uplink and downlink channels. In particular, mobile video traffic is currently generating most of the mobile traffic worldwide. By the end of 2021, mobile data traffic is predicted to rise to 49 exabytes, with the share of video traffic rising from 60% (2016) to 78% (2021) [165].

To cope with this increased traffic, mobile operators need to increase the network capacity to effectively support high-quality and bitrate demanding services with the available network resources, while keeping mobile infrastructure costs at a reasonable level. A good trade-off between perceived Quality of Experience (QoE) to be offered to the mobile users and smart use of network resources is achieved by dynamically adapting the coding rate of the requested videos to the available transmission resources. As observed in [166], reducing the encoding rate of a video is indeed much less critical in terms of QoE degradation than increasing the packet loss probability or the delivery delay.

In this chapter we introduce a widely used adaptive streaming technology, namely MPEG Dynamic Adaptive Streaming over HTTP (DASH), that will be the basis for the QoE improving techniques in the following chapters.

8.1 Adaptive streaming technologies

Adaptive bitrate streaming is a technique that enables optimum multimedia streaming over telecommunication networks across a wide range of devices and connection speeds. Its main peculiarity is the ability to detect and monitor user's available bandwidth and CPU capacity to adapt in real-time the video flow bit rate accordingly.

In particular, adaptive streaming is a method of multimedia streaming where the source content is encoded at multiple bit rates, then each coded content is splitted in segments with duration of a few seconds. Retrieving a manifest file,

the client can be aware of the presence of these multiple encoded versions and the location of the various segments. Now the client is able to retrieve the segments to playback the whole multimedia content choosing, for each temporal interval, the segment relative to the desired quality level. This choice can be made in an autonomous way by the client, based on available network bandwidth and on CPU capacity of user's device.

A key difference between streaming technologies is the type of used streaming protocol. While in the past the most adopted solutions used protocols like RTP with RTSP, nowadays adaptive streaming technologies are almost exclusively based on HTTP. This allows to have various advantages with respect to other solutions, in particular:

- it allows the reuse of existing server infrastructure, without the need to have dedicated servers as in the case for RTP streaming;
- it is firewall-friendly, because with HTTP protocol the video streaming packets are generally not blocked by firewalls;
- it can exploit existing HTTP cache infrastructure to offer video segments from a nearer location to the user with respect to the original server, enabling faster video delivery.

8.2 Introduction to MPEG DASH

DASH [167] is an ISO standard developed by the *Motion Picture Experts Group* that defines an adaptive bitrate streaming technique based on HTTP.

DASH development started in 2010, evolving into a Draft International Standard in January 2011 and an International Standard in November 2011. The MPEG-DASH standard, first published in April 2012 as ISO/IEC 23009-1, has been updated on July 2013, incorporating some amendments and corrigenda.

MPEG-DASH is the first HTTP-based adaptive streaming solution that arose at the level of international standard. It was preceded by similar, but proprietary, adaptive streaming technologies, like Adobe's *HTTP Dynamic Streaming*, Apple's *HTTP Live Streaming* [168] and Microsoft's *Smooth Streaming*. The objective for MPEG-DASH was to replace those technologies by incorporating their strong points into a widely implemented and vendor-independent standard, in order to enable the use of a single technology for multimedia streaming on all platforms. To reach this objective, the standardization group worked together with the most important stakeholders, like Adobe, Apple, Microsoft, Netflix and Qualcomm, and with other standardization bodies, in particular with 3GPP, that was developing a similar technology, called *Adaptive HTTP Streaming* (AHS) [169].

Nowadays the standard is implemented in various products and gained traction as the only available technology allowing adaptive bitrate streaming on devices from different vendors.

8.3 DASH data model

MPEG-DASH defines a media content delivery model where the control is primarily client-side. In fact, clients may request data, using HTTP protocol, from

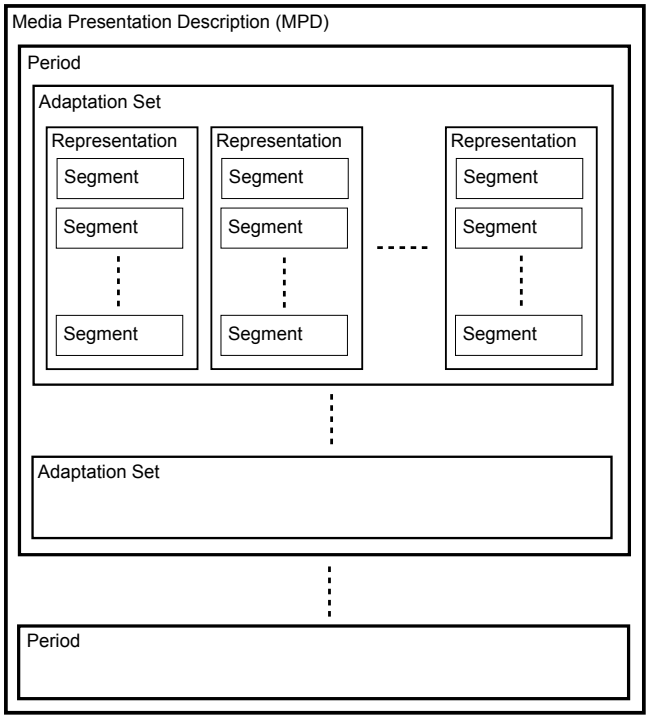


Figure 8.1: DASH data model

standard web servers that have no DASH-specific capabilities. Because of that, the DASH standard focuses on data formats used in data exchanges and not on client and server procedures.

The set of deliverable encoded versions of media content, along with their description, forms a *Media Presentation*. A DASH Media Presentation is described by an XML manifest file called Media Presentation Description (MPD) [167].

Media content is composed by one or more contiguous *periods* in time. These periods could represent parts or episodes of a main program, interleaved with inserted advertisement periods. The set of the available coded versions of media content must be consistent throughout a period, i.e., the available languages, subtitles, bitrates, etc. can not change within a period.

In a period, material is divided in *adaptation sets*. An adaptation set represents a set of coded version of a media component. For example, there could be an adaptation set for the main video component and a separate one for the main audio component. Other components, like subtitles or other audio tracks, could have a dedicate adaptation set each. Those media components could also be provided in multiplexed form. In this case, interchangeable versions of the *multiplex* may be described with a single adaptation set. An example for this case is an adaptation set containing both the main audio and main video for a period, with additional components being provided in additional adaptation sets.

An adaptation set contains a set of *representations*. A representation describes a deliverable encoded version of one or multiple media content components. Each representation in an adaptation set is sufficient to render the

```
<?xml version="1.0"?>
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500000S" type="static"
    mediaPresentationDuration="PT0H0M12.00S" profiles="urn:mpeg:dash:profile:full:2011">

    <ProgramInformation> <Title>akiyo0_dash.mpd</Title> </ProgramInformation>

    <Period duration="PT0H0M12.00S">
        <AdaptationSet segmentAlignment="true" maxWidth="352"
            maxHeight="288" maxFrameRate="25" par="352:288">

            <Representation id="1" mimeType="video/mp4" codecs="avc1.640016" width="352"
                height="288" frameRate="25" sar="1:1" startWithSAP="1" bandwidth="6772590">
                <BaseURL>akiyo0_dashinit.mp4</BaseURL>
                <SegmentList timescale="1200000" duration="5952000">
                    <Initialization range="0-865"/>
                    <SegmentURL mediaRange="866-4205261" indexRange="866-969"/>
                    <SegmentURL mediaRange="4205262-8393927" indexRange="4205262-4205365"/>
                    <SegmentURL mediaRange="8393928-10158885" indexRange="8393928-8393995"/>
                </SegmentList>
            </Representation>

            <Representation id="2" mimeType="video/mp4" codecs="avc1.640016" width="352"
                height="288" frameRate="25" sar="1:1" startWithSAP="1" bandwidth="5973738">
                <BaseURL>akiyo2_dashinit.mp4</BaseURL>
                <SegmentList timescale="1200000" duration="5952000">
                    <Initialization range="0-865"/>
                    <SegmentURL mediaRange="866-3709849" indexRange="866-969"/>
                    <SegmentURL mediaRange="3709850-7403297" indexRange="3709850-3709953"/>
                    <SegmentURL mediaRange="7403298-8960607" indexRange="7403298-7403365"/>
                </SegmentList>
            </Representation>

            <Representation id="3" mimeType="video/mp4" codecs="avc1.640016" width="352"
                height="288" frameRate="25" sar="1:1" startWithSAP="1" bandwidth="5184079">
                <BaseURL>akiyo4_dashinit.mp4</BaseURL>
                <SegmentList timescale="1200000" duration="5952000">
                    <Initialization range="0-865"/>
                    <SegmentURL mediaRange="866-3220504" indexRange="866-969"/>
                    <SegmentURL mediaRange="3220505-6425239" indexRange="3220505-3220608"/>
                    <SegmentURL mediaRange="6425240-7776118" indexRange="6425240-6425307"/>
                </SegmentList>
            </Representation>

        </AdaptationSet>
    </Period>
</MPD>
```

Figure 8.2: Example of an MPD manifest file

contained media components, but, grouping together several representations in a single adaptation set, the Media Presentation author states that those representations represent perceptually equivalent contents. This means that clients can dynamically switch between representations in an adaptation set in order to adapt to network conditions or other factors. Switching refers to the presentation of decoded data of one representation up to a certain time instant, and the presentation of decoded data of another representation from that instant onwards. If both representations are included in the same adaptation set, and the client switches properly, the media playout is perceived seamless across the switch.

Within a representation, the content may be divided in time into *segments*. In order to access a segment, an URL is provided for each segment.

Segments description in the MPD manifest file could be expressed in one of the following ways:

- *SegmentBase*: this description in used when only a single media segment is provided per representation. In this case, an URL (with an optional byte range) is reported for each representation, which references the file containing the segment for the considered representation. An example exploiting the possibility to make HTTP/1.1 byte-range requests follows:

```
<Representation id="1" mimeType="video/mp4" codecs="avc1.4d401f"
  width="1280" height="720" bandwidth="2073921">
  <BaseURL>car-20120827-88.mp4</BaseURL>
  <SegmentBase indexRange="708-1183">
    <Initialization range="0-707" />
  </SegmentBase>
</Representation>
```

- *SegmentList*: in this case the description of each representation includes a list of segment URLs, one for each segment of the considered representation. Each segment URL is composed by a file location and, optionally, a byte range, allowing to make byte-range requests according to HTTP/1.1 specification. A self-explanatory example for this case follows:

```
<Representation id="1" mimeType="video/mp4" codecs="avc1.640016"
  width="352" height="288" bandwidth="6772590">
  <BaseURL>akiyo0_dashinit.mp4</BaseURL>
  <SegmentList timescale="1200000" duration="5952000">
    <Initialization range="0-865"/>
    <SegmentURL mediaRange="866-4205261" indexRange="866-969"/>
    <SegmentURL mediaRange="4205262-8393927" indexRange="4205262-4205365"/>
    <SegmentURL mediaRange="8393928-10158885" indexRange="8393928-8393995"/>
  </SegmentList>
</Representation>
```

- *SegmentTemplate*: in this case, the list of segment URLs is expressed by a template and some replacement rules that allows to swap special identifiers with appropriate dynamic values assigned to segments. The simplest case is when the template is made by a fixed part and an index that assumes increasing values for successive segments. In this way it's possible to use DASH technology for streaming of live media content, where segments are delivered to clients while successive ones are still being generated, making impossible the creation of a segment URLs list beforehand. A simple

example of this case, where `$Number$` is the placeholder for the segment number, could be:

```
<Representation id="1" mimeType="video/mp4" codecs="avc1.640016"
  width="352" height="288" bandwidth="10059517">
  <SegmentTemplate timescale="1200000" media="seg_bowing0$Number$.m4s"
  startNumber="1" duration="2304000" initialization="seg_bowing0init.mp4"/>
</Representation>
```

8.4 Typical DASH client operation

The typical DASH client procedure to retrieve and render a media stream consists of the following steps:

1. the client retrieves the MPD manifest file from the server and parses it to be aware of all available media components and their representations;

2. the retrieval of the media starts with the download of first segments relative to the desired media components. Usually, the low bitrate version of first segments are chosen, because of the unknown network conditions. In this way, it is also possible to get a faster start of video playout. MPD manifest may also indicate the necessity to retrieve an initialization segment, containing information needed to initialize the media engines for enabling playout of the media segments. If this is not the case, segments are said to be self-initializing, because each of them contains all the necessary information for its decoding.

3. The client estimates network conditions from metrics calculated from previous segments download. These metrics will be helpful in choosing the bitrate of the next media segments to retrieve.

4. Successive segments are retrieved using the metrics calculated in the preceding step. In case of not self-initializing segments, if the new segment belongs to a different Representation with respect to the previous one, the initialization segment for that Representation must be retrieved in order to correctly decode the new segment.

5. Steps 3-4 are repeated until all desired media components are completely retrieved.

8.5 Additional DASH features

DASH technology provides additional features, such as:

- being codec independent, it works with H.264, WebM and other codecs, allowing this technology to be future-proof and adaptable to new codec that will be developed;

- the possibility to support all encryption schemes and DRM techniques specified in ISO/IEC 23001-7 standard enables its use in commercial streaming services;

- it allows for dynamic ads insertion, useful again for commercial streaming services;
- it entails special features to support live streaming, like the possibility to fragment the MPD manifest and download each fragment separately (used to update the manifest with new information that become available after the stream start).

Chapter 9

QoE Multi-Stage Machine Learning for Dynamic Video Streaming

An emerging and promising trend to address the rapid growth of video traffic in cellular networks is the development of solutions that are aware of the QoE of the end users and exploit this knowledge to optimize the network parameters. However, predicting the QoE perceived by the users in different conditions remains a major challenge. In this chapter, a machine learning approach to support QoE-based video admission control and resource management is proposed. More specifically, the approach implements a multi-stage learning system that combines the unsupervised learning of video features from the *size* of H.264-encoded video frames with a supervised classifier trained to automatically extract the quality-rate characteristics of unknown video sequences. This QoE characterization is then used to manage simultaneous video transmissions through a shared channel in order to guarantee a minimum quality level delivered to the final users.

9.1 Introduction

As already mentioned, a good trade-off between great user QoE and efficient network use is achieved by dynamically adapting the coding rate of the video flows to the available transmission resources. However, the perceived QoE at a certain encoding rate depends on the video content itself, e.g., the dynamics of the scene, the mobility of the source and frame-by-frame motion, etc., which are not easy to predict. Knowing these characteristics would make it possible to adjust the video rates according to the available transmission resources, so as to maximize the QoE of the users.

In this chapter, a cognitive approach for video delivery in communication-constrained scenarios is presented. The basic idea is to combine unsupervised and supervised machine learning techniques to infer the Quality-Rate (Q-R) characteristics[1] of the video sequences from high level information, readily avail-

[1]The Q-R characteristic is often expressed in the literature in terms of rate-distortion curve,

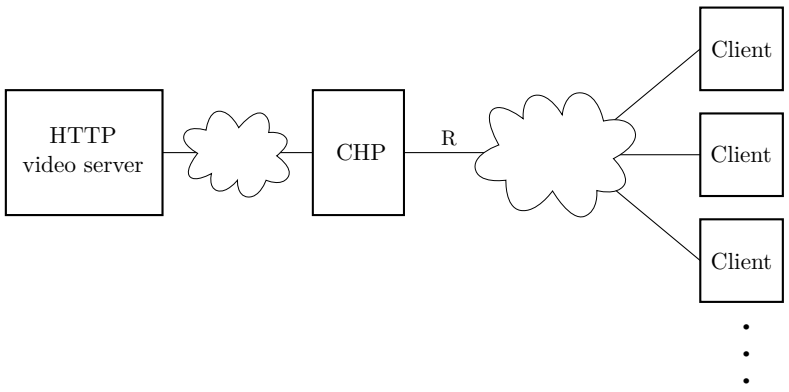


Figure 9.1: Reference scenario: the Cognitive HTTP Proxy (CHP) implements the RM and VAC mechanisms to manage the rates of the active video flows across the bottleneck link of capacity R [bit/s].

able at the network layer.

Consider a scenario where a number of mobile users request video content from some remote servers, using a shared channel. Also, assume that videos are provided by the servers in the form of short chunks of a few seconds each (called *video segments*), which are then delivered to the mobile users through HTTP sessions, similarly to DASH [167]. Therefore, there is no need to maintain long streaming sessions between server and mobile users, dramatically simplifying mobility management. Each video streaming session starts with an HTTP request sent by the mobile user to the video server for the list of the titles and formats of the available videos [170]. Each DASH file is indeed associated to a MPD that provides information characterizing the video file and the available locations of the segments, and may contain multiple representations for the same media, that is, multiple versions with different resolutions and bitrates. A DASH client is then able to dynamically select the desired representation of each chunk of the video and to get it via HTTP.

While the DASH framework is well established, the quality-adaptation policy is still open to investigation. Typically, the policies adopted by legacy DASH clients are based on local measurements, such as the number of buffered video segments at the client side, or the estimated average downlink throughput. Instead, the actual Q-R characteristics of the streamed videos, or the number and type of contending flows, are not commonly considered.

In this work, a more systematic approach is followed, proposing a solution where the rate of each competing video flow is determined in a centralized manner by a *Cognitive HTTP proxy* (CHP), as represented in Fig. 9.1. The CHP can be instantiated in the access router of a private network, to control the video traffic towards the hosts of the network. Furthermore, leveraging the upcoming Network Function Virtualization (NFV) paradigm, instances of the CHP can be activated where multiple video flows merge into the same shared link, in order to provide minimum performance guarantees to multimedia flows and/or blocking excess video traffic.

The proxy intercepts all HTTP requests, performs traffic classification, and

which conveys the same information, though presented in a different form.

applies *Video Admission Control* (VAC) and *Resource Management* (RM) algorithms to improve the QoE of the clients. In particular, the CHP will be able to intercept, interpret, and modify the DASH packets exchanged by video clients and servers, thus performing a dynamic adaptation of the video quality according to a certain utility function, which depends on the Q-R characteristic of *each single video*, which is automatically estimated by using a multi-stage machine learning approach. The Q-R characteristic is able to summarize, in a single function, the map between QoS and QoE parameters. Such map is necessary since the VAC and RM mechanisms aim to maximize the QoE while satisfying some QoE constraints of the network (such as maximum channel capacity or minimum end-to-end delay).

Crucially, the proposed method does not require to process the original content of the video frames, but only uses information readily available at the network layer after the encoding process, namely the *size of the video frames*, with some other parameters that can be easily retrieved from the MPD file associated to the video, such as the structure of the Group-of-Pictures (GOP) used during the encoding, the resolution of the video, and the frame rate. The rationale is that the Q-R function of a video is closely related to the dynamics of its content that, in turn, impacts the spatial and temporal redundancy of the video frames and, consequently, the size of the frames generated by the encoder [171, 172]. Highly dynamic videos, containing complex spatial and temporal structure, will likely result in larger frame sizes, while more static videos will be likely encoded in frames of smaller and more homogeneous size.

To test the proposed method, a training dataset has been built, containing the frame sizes for a number of HD and CIF video clips, encoded at different compression levels. The dataset was then used to perform the unsupervised training of a Restricted Boltzmann Machine (RBM) [173]. The RBM captures the latent features of the input data, thus providing a high-level representation of the video segments at different compression levels, which can be exploited by supervised learning algorithms to estimate the Q-R characteristics of unknown videos. In this study, the average Structural Similarity (SSIM) index [174] of the frames in a GOP is considered as a measure of the perceived quality of a video segment. Note that SSIM is not the only objective metric for QoE assessment of video sequences, nor is necessarily the best in all cases. The Q-R characteristics of a video can indeed be expressed with other metrics, either full reference (i.e., where the evaluation system has access to the original media) like the NTIA-Video Quality Metric General Model [175] and the MOVIE index [176], or no-reference, e.g., Video BLIINDS [177].

We observe that the SSIM focuses on the spatial dimension only, i.e., the quality of the image captured by the frames, while neglecting the time dimension that can be crucial to correctly assess the degradation of the visual experience due to gaps in the video streaming (freezing and rebuffering events) or sharp variations of the visual quality of the video frames. As it will be better discussed in Sec. 9.8, however, when the link bitrate is known (as assumed in this study), suitable VAC and RM algorithms can choose the bitrates of the video segments to fit into the available channel capacity, thus avoiding that the client runs out of frames to play out. In this scenario, where the temporal aspect of the QoE metric is less critical, SSIM represents a reasonable low-complexity choice. In addition, consider that the proposed framework can be applied to other QoE metrics with a qualitative similar Q-R characteristics (i.e., such that the quality

increases with the frame size and the data rate).

To summarize, based on a machine learning scheme, the Q-R characteristics (in terms of SSIM vs normalized bitrate) of unknown videos is estimated from the distribution of the coded frame sizes. This characterization is then fed into QoE-aware VAC and RM algorithms. Simulations show that combining unsupervised feature extraction and linear classification provides better results than a more basic approach that tries to extract the SSIM characteristics directly from the raw data. A further result is that QoE-based VAC and RM algorithms make a better use of the available transmission resources than content-agnostic schemes and provide a valuable tool for quasi-realtime adaptive video streaming applications.

9.2 Related Work

In this section we first review the state of the art on DASH adaptation logics and then consider the literature on the objective quality metrics for video sequences, which represent the two main building elements of the proposed approach.

9.2.1 Adaptation logics for DASH video streaming

As briefly mentioned in the introduction, in the DASH framework, the video clips are split in short time segments, which are encoded at different compression levels and stored at the video server as independently addressable and reproducible multimedia objects. This makes it possible to download any of the available versions of each video segment, thus enabling the dynamic adaptation of the video rate (and, in turn, quality) to the channel conditions in order to guarantee good video quality, uninterrupted play out, and smooth quality variations.

For example, the scheme proposed in [178] privileges the stability of the video bitrate over instantaneous video quality, thus adopting a conservative approach when increasing the bitrate that also yields a lower probability of freezing events. Probe and Adapt (PANDA) [179] makes use of active channel probing to estimate the path throughput and adapt the video rate accordingly. To prevent fluctuations due to cross-traffic variations, however, the scheme adopts a conservative rate-increasing strategy when the channel capacity grows and hysteresis margins to avoid frequent rate switches. A similar but simpler heuristic was presented by Petrangeli *et al.* in 2014 [180]. The mechanisms proposed in [181] uses only buffer state information to adapt the video bitrate, resorting to channel capacity estimation only during transient periods. As shown in [182], however, such simple schemes may not be able to guarantee high video quality, even when the channel capacity is constant. More complex approaches make use of predictive or Markov Decision Process techniques to model the variations of the channel capacity and find the optimal adaptation strategy [183–185]. The main limit of these approaches is the computational load: the model is too complex to be solved at runtime. To overcome this issue, some recent works apply reinforcement learning techniques to automatically learn the best adaptation strategy from the past experience [186]. However, this approach is limited by the training time of the machine-learning algorithms, which grows very quickly with the size of the state space [187]. Alternatively, the state space can be

roughly quantized to speed up the learning process to the detriment of the achieved performance [188, 189].

The work in [190] defines the bitrate adaptation strategy as an optimization problem, applied to a Markovian channel model. The function that links the video bitrate to the video quality, however, is not bound to any perceived quality metric. In [191], the authors developed a network-side mechanism to adapt video bitrate based on information from the clients that, however, are required to be all compliant with this protocol, which limits its practicality. A heuristic cross-layer algorithm for wireless networks is presented in [192], where both end-to-end bandwidth estimation and measurements from the WiFi link are used to determine the frame quality to download from the server.

For a more comprehensive overview of existing adaptation techniques for DASH, the reader is referred to [193].

The main focus of this work is not to provide another adaptation algorithms for DASH. Rather, the purpose is to propose a new methodology to infer the Q-R characteristics of each single video sequence and to show how this information can be successfully exploited in a DASH framework. For this reason, the considered scenario is rather simple, where the transmission resources reserved to video contents are constant and can be arbitrarily assigned to the different flows. Therefore, rate-adaptation is only required to reallocate channel resources when new video flows are accepted into the system or active ones are terminated. The proposed Q-R estimation technique can be combined with more sophisticated DASH algorithms and employed in more challenging scenario, whose investigation however is left to future work.

9.2.2 Objective quality metrics

Prior work on video detection over communication networks mainly focuses on extracting objective networking and quality metrics. In [194] the authors classify videos based on selected common spatial-temporal audio and visual features described by the MPEG-7 compliant content descriptors. Due to the complexity of the method, the authors make use of principal component analysis (PCA) to reduce the set of features under study. Nevertheless, this work is strictly dependent on the MPEG-7 multimedia format.

The work in [195] marks the packets using a pre-congestion notification mechanism in order to detect congestion in the network. A linear programming method is then used to assign a quality level to each video flow, in order to maximize a revenue function. The considered quality levels, however, are only described by video resolution and bitrate, not by a metric that properly evaluates the perceived quality.

The authors of [196] exploit a measurement-based admission control mechanism for video flows in order to maximize the number of admitted video requests in a network. Again, the considered metric is the video bitrate, while the perceived video quality is not considered. Also, this technique requires to know the state of the entire network in order to solve the admission control problem, which may be infeasible in large networks.

Further related work focuses on quality prediction models to capture the behavior of video scenes. The authors of [197] propose an objective model to predict the quality of 3D videos in the presence of frame losses, which is based on the header information of the video packets at different ISO/OSI layers. This

model is able to roughly capture the SSIM of some video clips based on the size of the lost frames and via deep packet inspection, which is usually avoided by operators in cellular deployments due to complexity and users' privacy concerns. Also, a model to extract the channel induced distortion in a no-reference fashion is described in [198]. The described algorithm exploits the received prediction residuals, coding modes, and the received and concealed motion vectors to compute an approximation of the SSIM index, therefore still requiring deep packet inspection. In any case, in [199], the authors claim that the frame loss probability, which is mainly a network metric, provides only limited insight into the video quality perceived by the user. Ref. [200] describes a model to map network QoE factors to a QoE value, whose complexity however makes it unsuitable for online applications. Other studies use learning techniques to predict video QoE from traffic data, including factors as the frequency of bitrate variations and the freezing events. However, the accuracy of the predicted QoE values is rather coarse [201, 202].

In this work, video test sequences are analysed and grouped based on the relation between video compression rate and SSIM. It is widely recognized that the SSIM index provides a more accurate QoE indication than more traditional metrics, like PSNR and mean square error (MSE), which have proven to be inconsistent with perceptual experience. Although the SSIM characterization of a video sequence is computationally expensive, many studies have shown that the extraction of perceived quality metrics, like SSIM, from the features of the encoded video is feasible. In [203] an artificial neural network is used to extract the SSIM of a video sequence using information on quantization parameters, frame structure, and motion vectors. The authors of [204] approximate the SSIM using, instead, an extension of the Support Vector Machine (SVM), namely the ϵ-Support Vector Regression. In this case, the considered features are derived from the frame structure, the quantization parameters, and the motion vectors. A much larger feature space is considered in [205], where 20 features describing the frame structure, motion vectors, and texture information are fed into a model, which is estimated using multipass polynomial regression. A simpler linear regression is employed in [206], which, however, is able to estimate both SSIM and Video Quality Metric (VQM) [175] for noisy channels using features related to motion vectors, the mean residual energy of the frames and error concealment information. In a related way, [207] describes the use of a multi linear regression technique to extract different video quality metrics (including PSNR, SSIM, and VSSIM) from the video bitrate and frame rate, and from information on motion vectors and on frame and group-of-picture structures. All of these methods, however, require the extraction of a large number of features from the video stream, thus requiring deep packet inspection and considerable computational cost.

Machine learning algorithms represent the state of the art in many classification tasks, especially when the structure of the domain is difficult to characterize. The problem of automatic video processing is closely related to that of image processing, with the additional complexity given by the temporal dimension of the data. In the so-called "content-based" video retrieval [208], for instance, a range of different techniques can be applied depending on the task of interest, e.g., video indexing, scene recognition and/or classification, object tracking, and motion detection. In recent years, advances in the theory and practice of probabilistic graphical models and statistical learning led to the development of

extremely powerful deep learning systems, which achieve state-of-the-art performance in several machine vision tasks [209,210]. Although the main application of these systems has been primarily focused on still frames, there have also been successful extensions to the temporal domain [211, 212].

All the above-mentioned machine learning methods, however, are usually applied at the pixel level, or to some higher-level representations obtained after additional pre-processing of the raw images. Nevertheless, for the task of classifying different videos depending on the dynamics of their content, it is assumed that the relevant information is still preserved after the video has been encoded to be sent on a transmission channel.

[171] showed that SSIM can be compactly represented by means of polynomial curves that can be associated to each video. Tagged videos can then be handled by simple traffic shaping mechanisms in case of network congestion or under-provisioned network resources. The idea of representing the Q-R curve as a polynomial function is well known in the literature. For example, considering the distortion expressed as the PSNR, the Bjøntegaard model [213] approximates a Q-R curve by a third order logarithmic polynomial fitting, based on experimental observations [214]. Another polynomial fitting based on PSNR and MPEG-2 encoding is described in [215].

Therefore, this chapter describes a technique for automatically extracting a set of features that can be used to describe the relevant characteristics of the original videos, using only information available at the network level.

9.3 Video analysis

The video dataset employed in this study is an expanded version of the one used in [171], where, in particular, a set of HD video clips have been added. For the reader's convenience, the video analysis framework described in [171] is reported here.

We need to evaluate the objective QoE of the videos with the SSIM index, which is a full reference metric that measures the image degradation in terms of perceived structural information change, thus leveraging the tight interdependence between spatially close pixels that contain the information about the objects in the visual scene [174]. SSIM is calculated via statistical metrics (mean, variance) computed within a square window of size $N \times N$ (typically 8×8), which moves pixel-by-pixel over the entire image. The measure between the corresponding windows X and Y of two images is computed as follows:

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)} \tag{9.1}$$

where μ and σ^2 denote the mean and variance of the luminance value in the corresponding window, and c_1 and c_2 are variables to stabilize the division with weak denominator (the interested reader is referred to [174] for additional details).

The range of the SSIM index goes from 0 to 1, which represent the extreme cases of totally different or perfectly identical frames, respectively. Tab. 9.1 shows the mapping of SSIM to Mean Opinion Score (MOS), which assesses the subjective perceived video quality on a scale of 5 values, from 1 (bad) to 5 (excellent), as reported in [216].

SSIM	MOS	Quality	Impairment
≥ 0.99	5	Excellent	Imperceptible
$[0.95, 0.99)$	4	Good	Perceptible but not annoying
$[0.88, 0.95)$	3	Fair	Slightly annoying
$[0.5, 0.88)$	2	Poor	Annoying
< 0.5	1	Bad	Very annoying

Table 9.1: Mapping SSIM to MOS scale

The analysis of the SSIM has been first applied to a pool of $V = 38$ CIF video clips, taken from standard reference sets.[2] Successively, the analysis has been replicated on a set of 28 HD videos. Each video has been encoded into H.264-AVC format. To test the robustness of the proposed approach to the specific encoding algorithm, the Joint Scalable Video Model (JSVM) reference software [218] has been used for CIF videos and the x264 encoder [219] for HD videos. The encoding has been done at $C = 18$ increasing compression levels (i.e., quantization points) for the CIF videos, and $C = 33$ levels for the HD videos, which correspond to as many quality levels. Note that there are no scene transitions inside each video sequence. The SSIM of a frame encoded at compression level c is obtained by comparing the decoded frame with the full quality version of the same frame. For practical reasons, the average values of the SSIM index computed for all frames of each video is considered.

Denote by $r_v(c)$ the transmit rate of video $v \in \{1, \ldots, V\}$ encoded at rate $c \in \{1, \ldots, C\}$, with $r_v(1)$ being the maximum (i.e., full quality) rate. To ease the comparison between different video clips, it is convenient to normalize the video rates to the full quality rates. Moreover, following the Weber-Fechner's law that postulates a logarithmic relation between the intensity and the subjective perception of a stimulus, we can introduce a logarithmic measure of the normalized rate, here named *Rate Scaling Factor* (RSF) and defined as

$$\rho_v(c) = \log(r_v(c)/r_v(1)) \,.$$

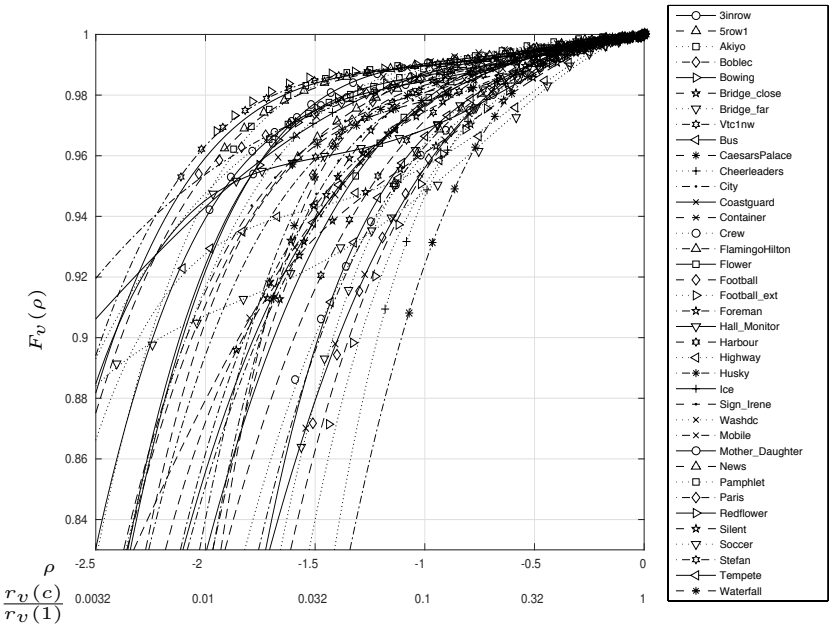
The dynamics of the video content impact the perceived QoE for a certain RSF value, as clearly shown in Fig. 9.2 (on the next page) where markers correspond to the average SSIM of each video clip when varying the RSF ρ, while lines represent a 4-degree polynomial interpolation of such points. More generally, we observe that the SSIM characteristics of a video v can be approximated by an n-degree polynomial expression, which takes the form

$$F_v^{(n)}(\rho) \simeq 1 + a_{v,1}\rho + a_{v,2}\rho^2 + a_{v,3}\rho^3 + \ldots + a_{v,n}\rho^n \,.$$

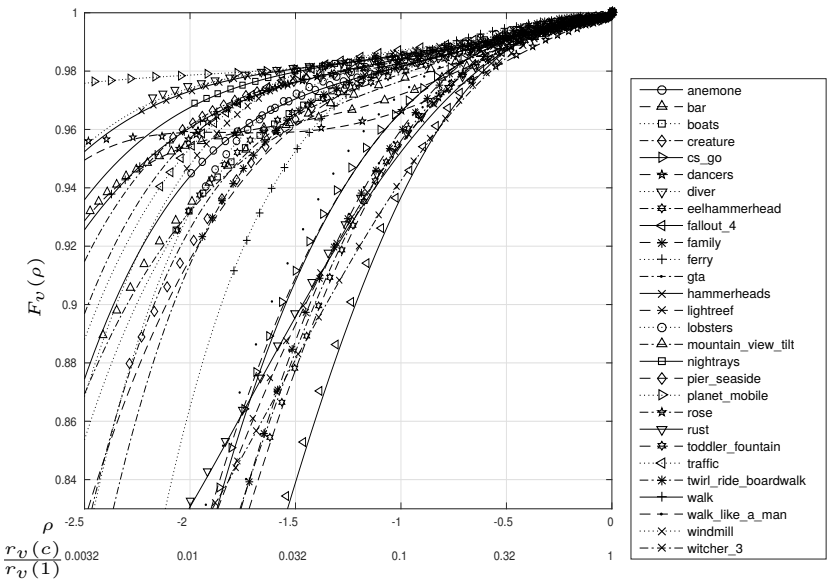
The vector of coefficients $\mathbf{a}_v = \{a_{v,i}\}$, called *SSIM coefficients* in the following, provides a compact description of the relation between the perceived QoE and the RSF of a video v.

From Fig. 9.2, we observe that, in general, the 4-degree polynomial $F_v^{(4)}(\rho)$ provides a quite accurate approximation of the SSIM values in the range of ρ of practical interest, for both the CIF and the HD videos in the test set. We observe that the relationship between QoE and QoS parameters is, in general, very complex, depending (among other factors) on metrics such as GOP size/structure,

[2] Video traces can be found in [217], ftp://132.163.67.115/MM/cif



(a) CIF video set



(b) HD video set

Figure 9.2: SSIM of the different video clips when varying the RSF: markers show empirical values, lines are obtained by the 4-degree polynomial approximation $F_v^{(4)}(\rho)$.

frame-rate, resolution, etc. (see, e.g., [220]). The curves reported in Fig. 9.2 have been obtained for a certain combination of parameters (frame rate, GOP structure, resolution), only changing the quality factor (c) of the H.264 encoder. Nonetheless, similar Q-R curves are obtained by changing the encoding parameters, i.e., considering different combinations of the GOP length and composition, frame rate, and resolution (not reported here for space constraints). In a real setting, most of these parameters will remain fixed within each video segment, so that the proposed approach is valid for each specific DASH request. It is hence conceivable to tag each video segment with the SSIM coefficients which provide a compact representation of its QoE characteristics that, in turn, can be used by RM and VAC algorithms, as discussed in the next section.

9.4 Machine Learning approach to video classification

The exact SSIM characterization of a video sequence using (9.1) is computationally demanding and infeasible in many practical cases. Following the rationale described in [172, 221], to overcome this limitation the presented approach uses a machine learning technique that provides a fairly accurate estimation of the SSIM characteristics of a video from the *size* of the frames coded in a GOP. As previously mentioned, the idea is that the SSIM characteristic of a video is closely related to the dynamics of its content, and that this information is preserved in the structure of the corresponding sequence of frame sizes after the encoding. However, extracting the SSIM characteristics of a video directly from the raw data, i.e., the frame sizes, is problematic because of the non-linear and hidden interrelations between the two quantities.

The fundamental idea behind the proposed approach is to train a generative model to capture these non-linearities, providing an alternative representation of the input data that is amenable to classification even by means of linear discrimination methods. More specifically, the proposed learning framework consists of two main phases. First, *unsupervised learning* is used to extract an abstract representation of the raw data that captures descriptive features of the video. A subsequent *supervised learning* phase is then performed to create a mapping between the abstract representations and the corresponding SSIM coefficients of the related videos. These two learning phases are detailed in the following.

9.4.1 Unsupervised phase: the Restricted Boltzmann Machine

The proposed approach relies on a powerful family of generative models which can be implemented as stochastic recurrent neural networks known as Boltzmann Machines [222]. They can be interpreted as probabilistic graphical models, where connections between units are symmetric, i.e., with equal weight in either direction. The input to the network is given through a layer of visible (i.e., observed) units, which are fully connected to another layer of hidden units that are used to model the latent features of the data. If there are no connections among units of the same layer, we obtain the so-called Restricted Boltzmann Machine (RBM) [173], which is graphically represented in Fig. 9.3.

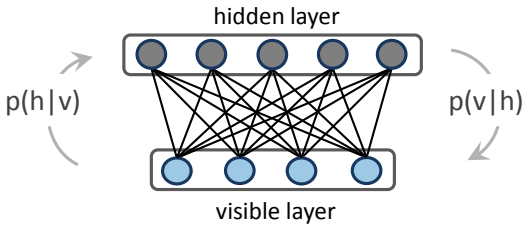


Figure 9.3: Graphical representation of a Restricted Boltzmann Machine.

The behavior of the network is driven by an energy function E, which implicitly defines the joint distribution of the units by assigning a probability value to each of their possible configurations:

$$p(v, h) = \frac{e^{-E(v,h)}}{Z} \tag{9.2}$$

where v and h are column vectors containing the values of the visible and hidden units, respectively, and Z is a normalizing factor known as partition function. The energy function is parameterized according to the weights of the connections between visible and hidden units:

$$E(v, h) = -b^\top v - c^\top h - h^\top W v\,, \tag{9.3}$$

where W is the matrix of connections weights and b and c are two additional parameters known as unit biases.

RBMs can be efficiently trained by using the contrastive divergence algorithm [223], which consists in alternating a positive and a negative phase. During the positive phase (*inference*), visible units are clamped to the values of the data observed in the training set. The network then propagates activations to hidden units, according to the weights of the connections. If we consider binary units for simplicity, during the positive phase the network observes the values of the visible units and activates each hidden unit h_j according to the conditional probability:

$$p(h_j = 1|v) = \sigma(c_j + \sum_i v_i w_{ij})\,,$$

where σ is the sigmoid logistic function, c_j is the bias term of the hidden unit h_j, and w_{ij} is the weight of its connection with the visible unit v_i. The entire vector of hidden unit activations constitutes an *internal representation* of the pattern observed in the visible units. During the negative phase, instead, hidden units are fixed and activations are propagated backward to the visible units in a similar fashion, in order to accurately *reconstruct* the original input vector. Each visible unit v_i is therefore activated according to the conditional probability:

$$p(v_i = 1|h) = \sigma(b_i + \sum_j h_j w_{ij})\,. \tag{9.4}$$

The objective of the learning process is to find a good set of weights W, so that the function E will assign low energy (i.e., high probability) to configurations of units that allow to obtain accurate reconstructions of the input patterns (i.e., maximum likelihood learning). This can be accomplished by performing

gradient descent over the likelihood function of the training data. It turns out that the derivative of the log-probability of a training vector v with respect to a particular weight w_{ij} is surprisingly simple:

$$\frac{\partial \log p(v)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \,, \tag{9.5}$$

where the first term on the right-hand side of (9.5) represents the empirical expectations computed on the training data, while the second term refers to the expectations computed according to the actual model distribution. We can use this quantity to compute how each weight should be changed at each learning step:

$$\Delta w_{ij} = \eta(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}) \tag{9.6}$$

where η is a small constant called learning rate. Due to the stochastic dynamics of RBMs, computing model expectations requires to gradually change the state of the network until it settles to *thermal equilibrium*, usually by running computationally expensive Gibbs sampling algorithms [224]. However, contrastive divergence makes it possible to efficiently train large-scale RBMs by approximating the log-probability gradient. The reader could refer to [225,226] for more details about learning in RBMs and for the explanation of important additional parameters of the algorithm (e.g., weight decay and momentum).

In our case, the training set consists of vectors of frame sizes for each GOP of the videos in the dataset. Unsupervised learning tunes the RBM model parameters (i.e., the connections weights) with the objective of reproducing the patterns presented in the visible layer, thereby minimizing the reconstruction error. At the beginning, weights are randomly initialized to small values (close to zero) and the reconstruction will be very poor. However, the learning process iteratively adapts the weights until the network is able to accurately reproduce the observed patterns. At the end of this unsupervised learning phase, the values taken by the units in the hidden layer provide an alternative and, hopefully, more expressive representation of the input vector, i.e., of a certain sequence of frame sizes in a GOP.

9.4.2 Supervised phase: the linear classifier

After a good model of the data has been learned, an additional *read-out* module can be put on top of the hidden layer of the RBM to perform a supervised classification task, which in our case consists in estimating the SSIM coefficients $\mathbf{a}_v$ for each new GOP. The idea is that some characteristics of the data are not directly visible in the raw input patterns, but can be discovered by the feature extraction process during the unsupervised learning phase. Once the RBM has learned good internal representations of the patterns by modeling their underlying causes, it should be easier to perform a supervised classification task starting from those abstract representations.

A simple linear classifier is used as a read-out module. The discrimination between the possible classes is therefore performed by exploiting a linear combination of the data features. This choice is motivated by observing that the non-linearities of the data should be captured by the generative model during the unsupervised learning phase, which creates more separable representations

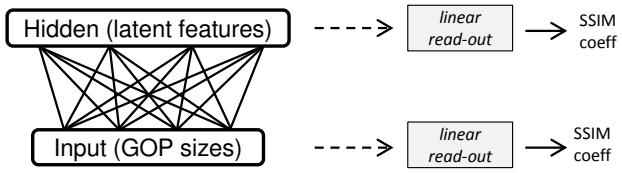


Figure 9.4: Scheme of the proposed learning framework, on which unsupervised feature extraction (left) is followed by supervised linear read-out (right).

that could be easily read out even by a linear method. In many machine learning scenarios, this strategy has shown to be very effective and is usually adopted by the so-called "kernel methods" exploited in *Support Vector Machines* [227], which first perform a non-linear projection of the data into a different (usually higher-dimensional) feature space, and then apply a linear optimization method to compute the maximum margin separating hyperplane.

Within this perspective, the accuracy of linear read-out can be considered as a coarse measure of how well the relevant features of the data are explicitly captured by the generative model [225]. Therefore, the use of a linear classifier makes it easier to understand the quality of the internal representations learned by the RBM, because we can directly compare the classification accuracy obtained using the raw input patterns with that obtained from the internal representations of the RBM. Moreover, a linear classifier is preferred in this scenario due to its greater generalization ability even in the presence of a limited training set. Indeed, a more powerful, non-linear algorithm would be more prone to overfitting. A schematic representation of this process is given in Fig. 9.4.

It is worth remarking that, once the unsupervised training phase is completed, the internal representation of the input data provided by the RBM can be used to perform supervised training of multiple read-out modules, with different purposes. For instance, it is possible to train a linear classifier that recognizes the GOPs belonging to the same video, or that classifies the GOPs according to the similarity of the video dynamics, and so on [221]. This is indeed one of the major advantages of combining unsupervised and supervised learning approaches, with the former providing an alternative representation of the input data that eases the selection of useful features by the latter.

9.5 Learning framework performance

In this section the performance of the proposed RBM-based learning framework is evaluated with respect to a linear classifier that acts directly on the raw data, i.e., the frame sizes contained in a GOP.

9.5.1 Dataset and learning parameters

The system is tested on the video dataset described in Sec. 9.3. In order to make the size of the data uniform, only the first 15 GOPs of each CIF video, and 13 GOPs for HD videos, are used, thereby discarding shorter videos. Thus, the used dataset is composed by 34 CIF videos, for a total of 510 data patterns

(GOPs), and 28 HD videos, for a total of 364 data patterns. The quality of learning in a RBM gets worse when the patterns in the training set are drawn from very different, heterogeneous distributions. In particular, in this case it can be observed that by merging the GOP patterns corresponding to both CIF videos and HD videos resulted in the emergence of a much less effective set of features. The reason is that the sole frame size is likely insufficient to capture the complex Q-R relationships for generic encoding parameters, while it is sufficient when the other parameters (namely, the GOP structure and size, and the video resolution) are fixed. A possible solution to overcome this problem is to train a different learning model for each representative combination of video encoding parameters. Another possibility may consist in expanding the input patterns by also explicitly including some information about the encoding parameters, such as the resolution of the video or the GOP structure. In this work, the first solution is considered, leaving the latter for future studies.

Therefore, two different training sets have been created, one containing samples derived from CIF videos and the other containing samples derived from HD videos, and separate RBMs were trained on each dataset. The encoding format for input patterns consisted of GOPs formed by a single inter-coded frame (I) followed by 15 predicted frames (P), which is a common format for GOPs of 16 frames. However, control simulations (not reported here) showed that the used approach still works even using other GOP formats, e.g., with a different number of frames and/or a different pattern (sequence of I and P frames within a GOP), provided that the RBM is adapted to the new input and properly trained.

Due to the limited size of the datasets, the performance of the system has been tested using a *k-fold cross-validation* technique [228]. To this aim, the dataset of CIF videos has been partitioned into 34 subsets (folds), each including all the 15 GOPs of a specific video. The RBM was then trained using 33 folds (training set), and its generalization performance was computed on the left-out fold (test set). This way, 34 different RBMs were trained, each time changing the left-out video to be used as test, and the mean estimation accuracy over all the 15 GOPs is reported. The input to the RBM consisted of 32 visible units, which represented the sizes of the 16 frames in a GOP, coded with two different compression levels $c = 1$ (full quality) and $c = 9$ (intermediate quality). Only these two levels have been included in order to limit the amount of patterns in the training set, with the goal of more clearly establishing how well the system could generalize to previously unseen compression levels. Furthermore, the intermediate qualities were used instead of the lowest ones because the aim is to estimate with greater accuracy the high SSIM region of the Q-R curves rather than the low-quality tail, considering that, in practical applications, the latter region is of scarce interest because of the very poor visual quality of the videos.

The same procedure was applied for the dataset of 28 HD videos clips, where however the intermediate quality corresponded to a parameter $c = 18$, since the number of available quality layers for HD videos was 33, against the 18 levels of the CIF videos.

The I and P frame sizes of each GOP were normalized between 0 and 1, which corresponded to the minimum and maximum frame sizes, as this is the usual format of the input patterns used for training neural networks. The size of the hidden layer determines the complexity of the generative model, since the

number of free parameters in the model is given by the number of connection weights. Different layer sizes were tested, with a number of units varying between 50 and 200, finding that the obtained results are robust with respect to this parameter. Results presented in the following have been obtained with a network having 70 hidden units.

Tests were conducted using a publicly available efficient implementation of RBMs that exploits Graphic Processing Units (GPUs) to parallelize the learning algorithm [229]. Unsupervised learning occurred using a mini-batch scheme with mini-batch size of 13, learning rate of 0.001, weight decay of 0.00001, and a momentum coefficient of 0.9. With the current settings of the machine learning parameters, the learning phase converged after about 50 epochs without exceeding one minute of running time. Regarding the supervised phase, a linear classifier can be implemented as a single layer perceptron, on which iterative learning is performed using the delta-rule. However, an equivalent but computationally more efficient method has been used, which relies on the calculation of a pseudo-inverse matrix and is readily available in some high-level programming languages such as Python or MATLAB [225].

Note that the unsupervised and supervised learning processes are performed only once. Once the RBM and the coupled linear classifier are trained, the estimation of the SSIM coefficients for unknown videos is extremely simple, and can be performed online in negligible time.

9.5.2 Coefficients estimation accuracy

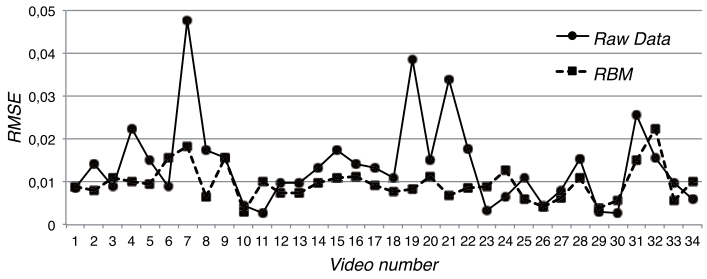
Here we assess whether the internal representation learned by the RBM allows to estimate the n SSIM coefficients for each video in the test set. The quality of the estimation is evaluated in terms of RMSE between the exact and the estimated SSIM-rate characteristics, i.e.,

$$\text{RMSE} = \sqrt{\frac{1}{\rho_{min}} \int_{\rho_{min}}^{0} \left(F_v^{(4)}(\rho) - \tilde{F}_v^{(n)}(\rho) \right)^2 d\rho}$$

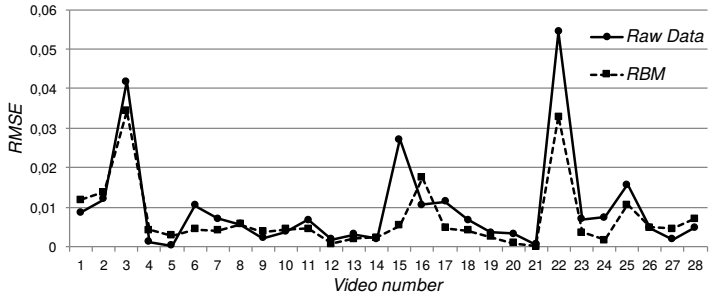
where $\rho_{min} \simeq -3$ is the minimum value of RSF of interest, while $F_v^{(4)}(\rho)$ is the reference SSIM-rate curve, and $\tilde{F}_v^{(n)}(\rho)$ is the n-degree polynomial (9.3), with coefficients estimated by the classifier.

The dashed line with square markers in Fig. 9.5 shows the mean estimation accuracy on the 15 GOPs contained in each of the 34 videos of the CIF test set (a), and on the 13 GOPs of the 28 videos in the HD test set (b). To better appreciate the performance of the RBM-based learning architecture, the graph also reports the RMSE for the SSIM curves obtained by applying the linear classifier directly on the raw data patterns (solid line with circle markers). We see that the internal representation learned by the RBM model is indeed capable of capturing critical features of the data, thereby allowing to increase the estimation accuracy for almost all test videos.

Fig. 9.6 offers a visual comparison between the exact and estimated SSIM curves for two different videos with prototypical trends (see corresponding points in Fig. 9.5 to have an idea of their average RMSE error). In particular, Fig. 9.6a shows that the curve estimated using the RBM internal representations (solid line) clearly exhibits a better alignment with the exact SSIM curve (dashed



(a) CIF video set.

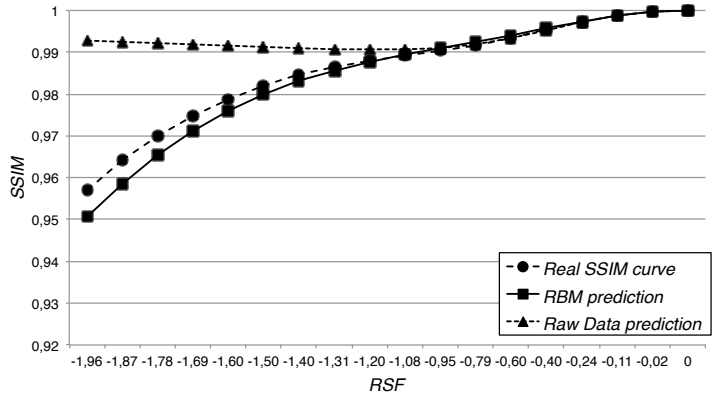


(b) HD video set.

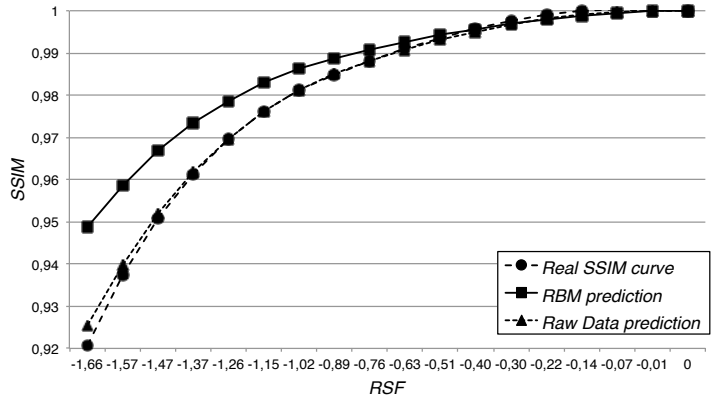
Figure 9.5: Root Mean Square Error (RMSE) of the estimated SSIM-rate curve for each video in the CIF video test set (a) and HD video test set (b), with $n = 4$. Polynomial coefficients estimation is given by applying a linear classifier on raw input data (circle markers) or on the hidden layer of the RBM (square markers).

line) than the curve obtained directly from raw data (dotted line). Even in the few cases where the RMSE is worse for RBM prediction, as that reported in Fig. 9.6b, the RBM estimation of the SSIM curve still remains good.

As explained, the complexity of the coefficients estimation increases with the degree n of the polynomial. On the other hand, high-degree polynomials offer a better approximation of the actual SSIM-rate characteristics. It is therefore interesting to investigate the accuracy of the SSIM estimation when varying the degree n of the polynomial. To this end, for each video in the CIF dataset, we report in Fig. 9.7 the RMSE of the SSIM estimation obtained by considering 2, 3 and 4-degree polynomials. Similar results were obtained for HD videos. Quite interestingly, we observe that there is no absolute winner: the optimal choice of n depends on the characteristics of each video. In the next section, we will investigate the practical impact of such estimation differences in the performance of video admission control and resource allocation algorithms.



(a) Predicted and real curves for video number 2: RBM prediction shows better precision.



(b) Predicted and real curves for video number 11: raw data prediction shows better precision, but RBM prediction is still acceptable.

Figure 9.6: Examples of predicted polynomial curves with respect to the ideal curve, for two different videos.

9.6 Performance Analysis of Cognitive RM and VAC Algorithms

In this section, we first revisit the approach presented in [171], which is used in this study in conjunction with the learning framework of Sec. 9.5. Then, we discuss the role of the play-out buffer and derive a simple analysis to determine the amount of pre-buffered content that guarantees a freezing probability lower than a given threshold.

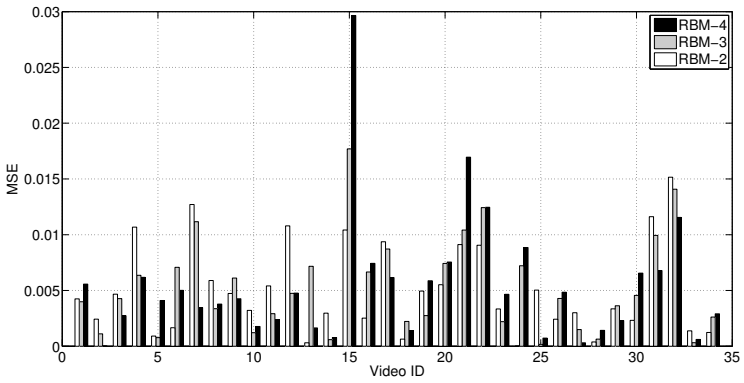


Figure 9.7: 2, 3 and 4-degree prediction error for each video of the dataset.

9.6.1 SSIM-based RM and VAC algorithms

Given a mechanism to infer the QoE characteristics of a video, we can now develop VAC and RM mechanisms that make use of such information. Consider a framework where different video clips are multiplexed into a shared link of capacity R by the Cognitive HTTP Proxy (CHP) that performs VAC and RM (see Fig. 9.1). In general, the RM module should detect changes of the link capacity (e.g., due to concurrent data flows or fading phenomena in wireless channels) and trigger an optimization procedure that adapts the video rates to maximize a certain utility function. In this work, a more favorable (but still practical) scenario is considered, in which a fixed and constant capacity is reserved to video flows, which are then isolated from best-effort traffic. In Sec. 9.8 possible extensions of the work to more challenging scenarios will be discussed.

The VAC module determines whether or not a new video request can be accepted without decreasing the QoE of any video below a threshold F^* negotiated, for instance, between the operator and video consumers. To this end, the VAC invokes the RM module to get the best resource allocation for all the videos potentially admitted into the system and, then, computes the expected SSIM of each video by using (9.3). If the estimated SSIM is below F^* the last video admission request is refused, otherwise the video is accepted and the rates of the videos in the system are adapted to the new allocation of the transmission resources determined by the RM module. To avoid sharp quality changes in the ongoing video streams, the video rates can be adapted progressively, with a step that depends on the actual gap between the current and the target SSIM of each video. Such smoothing techniques will be briefly discussed in Sec. 9.8, though a detailed analysis of these and other possible improvements is left to future work.

Formally, let R denote the average available transmission capacity of the link that can be allotted to the videos, and let $\Gamma = \{\gamma_v\}$ be an allocation vector that assigns to the vth video a fraction γ_v of R, with $\gamma_v = 0$ indicating that the video is not accepted into the system. Although the H.264 encoding can only offer a discrete set of transmit rates, in the formulation of the optimization problem it is temporarily assumed that video encoding rates can be tuned in a

continuous manner.[3] Under this assumption, the RSF of the vth video can be expressed as

$$\tilde{\rho}_v = \log\left(\frac{\gamma_v R}{r_v(1)}\right).$$

The optimization problem addressed by the RM module can then be defined as follows:

$$\Gamma_{\text{opt}} = \underset{\Gamma}{\text{argmax}}\, U(\Gamma, R, \{F_v\}) \quad \text{s.t.} \quad \sum_v \gamma_v \leq 1\,,$$

where $\{F_v\}$ denotes the set of SSIM functions of the videos, while $U(\cdot)$ denotes the *utility function* considered by the optimization algorithm. Two baseline utility functions, which reflect different optimization purposes, are considered:

Rate Fairness (RF)

Resources are distributed to all active videos proportionally to their full quality rate, without considering the impact on the perceived QoE. In this case, the optimal rate allocation for the ith video is simply given by

$$\gamma_{\text{opt},v} = \frac{r_v(1)}{\sum_j r_j(1)}\,,$$

so that the RSF of each video equals $\tilde{\rho} = \log(R/\sum_j r_j(1))$.

SSIM Fairness (SF)

Resources are allocated according to a max-min fairness criterion with respect to the SSIM of the different videos:

$$U(\Gamma, R, \{F_v\}) = \min_v F_v(\tilde{\rho}_v)\,.$$

Note that under the assumption of continuous rate adaptation, the SF criterion yields the same SSIM, say φ, to all active videos. Given this target SSIM, the RSF for each video can be easily found as $\tilde{\rho}_v = F_v^{-1}(\varphi)$, where F_v^{-1} is the inverse of the QoE function F_v (which is monotonic in the range of interest). Therefore, the optimization problem can be solved by searching for the maximum φ that satisfies the rate constraint in (9.6.1), i.e.,

$$\varphi^* = \max\left\{\varphi : \frac{1}{R}\sum_v r_v(1) 10^{F_v^{-1}(\varphi)} \leq 1\right\}.$$

and the associated rate-allocation vector is given by

$$\gamma_v = 10^{F_v^{-1}(\varphi^*)}\frac{r_v(1)}{R} \quad \text{for all } v \in V\,.$$

[3] This assumption will be removed in the simulations.

Mapping to admissible encoding rates

Once the target allocation vector $\Gamma = \{\gamma_v\}$ has been determined under the assumption of continuously encoding rates, we need to find a feasible allocation vector $\Gamma^\circ = \{\gamma_v^\circ\}$ such that, for each video v, there exists an encoding rate $r_v(c) = \gamma_v^\circ R$. The solution is obtained through the following recursive policy. For each video v, we find the minimum compression level $\hat{c}$ for which the encoding rate does not exceed the allotted capacity, i.e.,

$$\hat{c} = \min\{c \,:\, r_v(c) \leq \gamma_v R\}\,.$$

We then select the video v for which the gap between $r_v(\hat{c})$ and $\gamma_v R$ is minimum, and set $\gamma_v^\circ = r_v(\hat{c})/R$. Hence, we update the amount of available resources as $R \leftarrow R - r_v(\hat{c})$ and repeat the process iteratively over the remaining videos.

9.6.2 Play-out buffer analysis

We observe that the considered RM algorithms always guarantee that the aggregate bitrate of the downloaded video segments does not exceed the available channel capacity. Consequently, the *size*[4] of the play-out buffer at the client side will also remain approximately constant in time, except for small oscillations due to the variations of the GOP rates around their mean, which can be smoothed out by buffering a few GOPs of video before starting the playback. In this way, it is possible to avoid freezing events, while guaranteeing quick starting of the video play. In the following, we perform an approximate analysis of the necessary play-out buffer size to guarantee a smooth video playback with low probability of freezing and rebuffering events.

Let τ_v be the time duration of each GOP in the video sequence v. Furthermore, let $s_v^h(c)$ be the size of the hth GOP of the video, when encoded at compression level c. In principle, these values can be determined by the video server and passed to the client (and the CHP) through the MPD descriptor. However, for the sake of simplicity and generality, these values are modeled as i.i.d. random variables, with mean $s_v(c) = \mathrm{E}\left[s_v^h(c)\right]$ and standard deviation $\sigma_v(c)$, and it is assumed that only these two parameters are passed to the client/CHP.

Let n_0 be the number of GOPs that are buffered by the client before starting the playback. When the playback starts, a GOP is fetched from the buffer every τ_v seconds, while new GOPs arrive into the buffer from the network at uneven intervals. A freezing event occurs whenever the time to download n new GOPs exceeds the time to play $n_0 + n$ GOPs or, in other terms, when the aggregate size of n GOPs, $S_v(n;c)$, exceeds the total number of bits $D_v(n)$ that can be downloaded by the client in the period $(n+n_0)\tau_v$. Assuming that the RM determines the source rates by conservatively considering only a fraction $\alpha \in [0,1]$ of the available link rate R, we have that $s_v(c) = \alpha\tau_v\gamma_v^\circ R$, so that the aggregate size of the n GOPs is $S_v(n;c) = \sum_{h=1}^{n} s_v^h(c)$, with mean $\mu = n s_v(c) = n\alpha\tau_v\gamma_v^\circ R$, while the total amount of data that can be downloaded in the playing time of $n + n_0$ GOPs is $D_v(n) = \tau_v\gamma_v^\circ R(n_0 + n)$. The freezing probability can then be expressed as $P_f(n;c) = \Pr[S_v(n;c) \geq D_v(n)] = \Pr[S_v(n;c) \geq \mu(1+\delta)]$,

[4] As customary, the size of the play-out buffer is here intended in terms of playing time of the buffered video content, whose size in bytes depends on the compression level of the video sequence.

where $\delta = \frac{n+n_0}{n\alpha} - 1$. We wish to determine the value of n_0 such that $P_f(n;c) \leq P_f^*$ for all n, where P_f^* is the maximum acceptable freezing probability. Applying the Chernoff bound, we then get

$$P_f(n;c) \leq \exp\left(-\frac{2\delta^2\mu^2}{n\Delta_v(c)^2}\right) ,$$

where $\Delta_v(c)$ is the difference between the max and the min GOP sizes. Posing the right-hand side of (9.6.2) lower than or equal to P_f^* we get the following conservative criterion to choose the size of the play-out buffer:

$$n_0 \geq f_0(n;\alpha) = \frac{\alpha\Delta_v(c)}{s_v(c)}\sqrt{n\log\left(\frac{1}{\sqrt{P_f^*}}\right)} - n(1-\alpha)$$
$$= \beta\sqrt{n} - (1-\alpha)n , \quad (9.7)$$

where, for ease of writing, we set

$$\beta = \frac{\Delta_v(c)}{\tau_v\gamma_v R}\sqrt{\log\left(\frac{1}{\sqrt{P_f^*}}\right)} .$$

The right-hand side of (9.7) reaches it maximum for $n^* = \frac{\beta^2}{4(1-\alpha)^2}$, for which we get $f_0(n^*;\alpha) = \frac{\beta^2}{4(1-\alpha)}$. Denoting by $n_{\max}$ the maximum number of GOPs in a video stream, we can then set

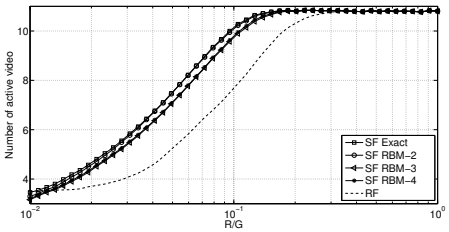
$$n_0 = \beta\sqrt{\min\{n_{\max}, n^*\}} - (1-\alpha)\min\{n_{\max}, n^*\} .$$

Using this approximation, it is possible to tune the play-out buffer size to the characteristics of the specific video stream. Note that, the smaller α (i.e., the larger the fraction of the link rate that is not allocated to the sources to leave some capacity in case of need), the smaller the play-out buffer required to avoid freezing events. However, the value of c will also be affected by α, since the RM will choose more compressed versions of the video streams to fit into the shrunk channel capacity αR. For a given P_f^*, there is then a tradeoff between the delay to start the play out, which is approximately equal to $\alpha n_0 \tau_v$, and the quality of the streamed video.

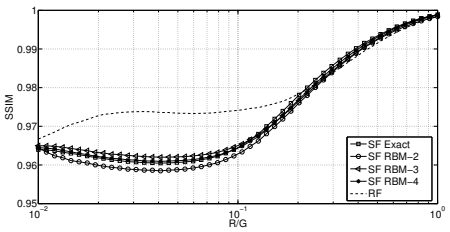
Considering the test videos used in this study, by setting $\alpha = 1$ (which allows for maximum video quality), we obtain $\Delta_v(c)/s_v(c) \leq 0.35$ for all videos and all values of c. With such values, Eq. (9.6.2) returns a buffer size of $n_0 \simeq 10$ GOPs (about 3.6 seconds with GOP of 12 frames) when considering video sequences of up to $n_{\max} = 500$ GOPs (about 3 minutes) and a freezing probability threshold $P_f^* = 5\%$, while $n_0 = 12$ GOPs for $P_f^* = 1\%$.

9.7 Simulation results

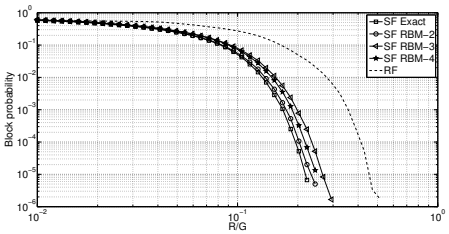
Here the results of the simulation study are presented, showing the potential benefits, in terms of QoE and blocking probability of the video connections, that can be achieved by adopting the proposed mechanisms.



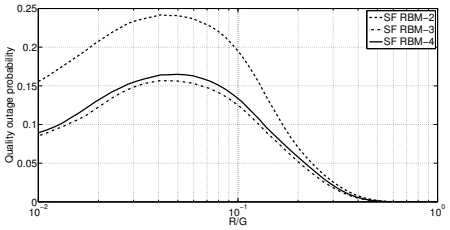
(a) Average number of admitted videos.



(b) Average SSIM of the admitted videos.



(c) Probability that a video request is blocked.



(d) Probability that the video quality is below threshold at least once during the video session.

Figure 9.8: Performance comparison of the proposed algorithms *RF* and *SF* when varying the channel capacity, where *SF-Exact* is the result based on the exact SSIM curve, while *SF-RBM-n* is based on the n-degree polynomial estimation given by the RBM model.

9.7.1 Simulation scenario

To compare the performance of the VAC and RM algorithms described above, the simulated scenario consists of a transmission link that is shared among the users, e.g., the outbound link towards the public Internet of a LAN. The VAC mechanism (running in the edge router/proxy) intercepts all requests for new video streaming sessions, and checks whether the additional traffic flow can be accommodated without dropping the QoE of the active videos below a certain SSIM threshold that is set to $F^* = 0.95$, which corresponds to good quality (MOS of 4, see Tab. 9.1).

The video generation process is simulated as a Poisson process with $\lambda = 0.66$ requests/s, where each video request refers to a video randomly picked from the dataset. The simulation provides a high-level picture of the system, neglecting the low-level details of the HTTP protocol. Each new video request triggers the VAC and RM modules, which use the Q-R curve for that video as estimated by the RBM algorithm to perform their decisions. When a new video is admitted into the system, or an active video completes its playback, the RM algorithm reassigns the resources, according to the chosen policy. Note that, while the VAC and RM operate on the *estimated* Q-R curves, the performance shown in the result section refers to the *actual* SSIM of the active videos. Denoting by T the average duration of a video sequence, we then have an offered load of $\lambda T \simeq 11$ videos, which corresponds to an aggregate rate request for full video quality of about $G \simeq 161$ Mb/s.

Video requests are processed by the VAC algorithms described in Sec. 9.6, and resources are allocated accordingly. In particular, four different flavors of the SF algorithm are considered, corresponding to different choices of the SSIM function $F_v(\rho)$, namely:

- *SF-Exact* based on the exact SSIM curve, i.e., $F_v(\rho) = F_v^{(4)}(\rho)$;
- *SF-RBM-n* based on the n-degree polynomial estimation given by the RBM model, i.e., $F_v(\rho) = \tilde{F}_v^{(n)}(\rho)$, with $n \in \{2, 3, 4\}$.

The simulation has been implemented using MATLAB, without the use of external libraries. Results are obtained in a practical, but somehow favorable scenario, where the link capacity is stable and known and the Q-R characteristic of each video is fixed in time.

9.7.2 Results

The algorithms are compared in terms of: (i) average number of admitted videos, (ii) average SSIM of admitted videos, (iii) blocking probability of a video request, and (iv) quality outage probability, i.e., probability that the quality of an accepted video drops below the minimum threshold F^* during the session. Note that with SF-Exact there is no quality outage, therefore this performance index captures the impact of the SSIM estimation errors of the RBM-based methods.

Fig. 9.8 shows the performance indices when varying the channel rate R with respect to the nominal average rate request G for full-quality videos. At first glance, we observe that the SF policies always perform better than RF, and accept more videos with above-threshold quality. This confirms that content-aware admission and resource allocation policies are much more effective than traditional content-agnostic policies in a QoE framework. It is interesting to observe in Fig. 9.8b that the average SSIM of the active videos is well above the minimum required quality threshold F^*. The reason is that actual video rates obtained with the different (discrete) compression levels are used, so that resource allocation is not able to use all the channel capacity, leaving part of it unused. This effect is minimized when $R/G \simeq 0.05$. If the video coder were able to provide any desired bitrate value, the quality for all video would have been equal to F^*, when considering a sufficiently large G. From Fig. 9.8d we also note that the smaller the margin between the mean SSIM and F^*, the larger the quality outage probability of the SF-RBM schemes. Having a smaller margin, in fact, offers less protection to SSIM estimation errors. When the average SSIM is way larger than F^*, instead, the probability that a SSIM estimation error causes the actual video quality to drop below F^* is very low.

For what concerns the SF algorithms, we observe in Fig. 9.8a that, on average, the SF-RBM polynomial approximations perform quite closely to the SF-Exact scheme. Hence, the RBM-based prediction is nearly optimal and proves the goodness of the training phase. A closer look at the results reveals that SF-RBM-2 is slightly looser than the other SF schemes in the admission process, allowing a moderately larger number of videos in the system, with a little lower average SSIM, as shown in Fig. 9.8b. From Fig. 9.8d, however, we note that the degree-2 approximation exhibits the largest quality outage probability, which negatively impacts the system performance due to the aforementioned nearly optimal number of admitted videos. Conversely, the SF-RMB-3 and SF-RMB-4 schemes perform in a comparable manner, with a very small advantage of SF-

RBM-3 over SF-RBM-4 in terms of quality outage probability. Thus, we might suggest the use of degree-3 predictions due to the slightly lower computational complexity and amount of signaling required in the system.

9.8 Improvements and open challenges

The study presented in the previous sections was mainly intended to prove the effectiveness of the machine-learning approach to gain knowledge on the Q-R characteristics of a video sequence from high-layer parameters and to show how such a knowledge can be exploited by network management algorithms to improve the service offered to the users. The analysis has been carried out by considering a practical, but somehow favorable scenario, in which homogeneous video sequences are assumed, with fixed and known Q-R characteristics and stable communication resources. Furthermore, other important QoE metrics have been neglected, such as the effect of sharp quality variations.

This section provides a preliminary discussion of some possible extensions of the proposed approach to overcome these limits, leaving a more detailed analysis to future work. Given its superior performance, only the SSIM-fairness RM criterion is considered. As a first step, some of the assumptions regarding the Q-R characteristics of the video sequences and the QoE metrics are relaxed, but still assuming that the multimedia flows are guaranteed a constant bitrate R. Then, the case where the channel capacity may vary over time is addressed.

9.8.1 Limiting video quality variations

To avoid sharp variations of the video quality due to the adaptation mechanisms, it is possible to resort to the smoothing/hysteresis techniques proposed in the DASH literature. However, the knowledge of the Q-R characteristics of each video sequence makes it possible to choose the step of the rate adaptation in a way that makes the quality variation less perceivable. Consider, for example, the reduction of the SSIM of current videos from φ to φ' to make space for a newcomer. If the quality variation $\varphi - \varphi'$ is small, so that the SSIM gap is barely perceivable, then the rate change can be performed immediately, irrespective of the actual rate gap, and the new video can be directly admitted with quality φ'. If, instead, the SSIM gap is perceivable, then the rates should be smoothly changed and the new video might be admitted with some delay and/or with a lower initial quality which is progressively and smoothly increased till φ'. We observe that the proper implementation of these mechanisms would require the definition of a function $d(\varphi, \varphi', t)$ that quantifies the quality degradation due to variations of the SSIM from φ to φ' in a time t. However, the identification of such a function is still an open and interesting research challenge.

9.8.2 Varying Q-R characteristics

The video clips considered in this analysis were homogeneous in terms of Q-R characteristics. In general, however, the Q-R curve may vary in consecutive video segments, e.g., because of scene changes. In this case, the VAC becomes more complex. If the Q-R curve is known in advance for all the video segments, the VAC can potentially predict the resource assignments for the whole

duration of the video sequences (assuming the current system conditions would not further change) and check whether the SSIM would always be satisfactory. Moreover, it is possible to design rate adaptation algorithms that temporarily increase the resource share assigned to a flow (or reduce the video quality of that flow) in order to fill the play-out buffer in prevision of future segments of the same video with higher rate requests.

To formalize these concepts, we can define $g_v^\ell(\varphi)$ as the size of the ℓ segment of video v, when encoded at a level that yields SSIM φ. Adopting a conservative approach, we may replace the feasibility condition in (9.6.1) with the following

$$\frac{1}{n_s}\sum_{\ell=1}^{n_s}\sum_{v} g_v^\ell(\varphi) \leq RT_s, \quad \text{for } n_s = 1, 2, \ldots, N_{s,\max}\,,$$

where T_s is the time duration of a video segment, and $N_{s,\max}$ is an acceptable time horizon (e.g., the least number of residual segments for the ongoing flows). Therefore, (9.8.2) is satisfied when the aggregate bitrate required to download each of the video segments at quality φ never exceeds the link capacity. A new video is accepted into the system only if the maximum φ that satisfies (9.8.2) is not lower than the threshold F^*. A more aggressive (and resource-efficient) strategy may consider a dynamic adaptation of φ, while avoiding sharp quality variations. In this case, the feasibility condition can be expressed as

$$\frac{1}{n_s}\sum_{\ell=1}^{n_s}\sum_{v} g_v^\ell(\varphi_\ell) \leq RT_s, \quad \text{for } n_s = 1, 2, \ldots, N_{s,\max}\,, \tag{9.8}$$

$$\text{s.t. } d(\varphi_\ell, \varphi_{\ell+1}, T_s) \leq d^*\,. \tag{9.9}$$

where $d(\cdot)$ is the function described in Sec. 9.8.1, and d^* is the maximum acceptable degradation due to quality variations. The analysis of these approaches, however, is left for future work.

9.8.3 Variable link capacity

The analysis carried out so far assumes that the link capacity reserved to multimedia flows is constant over time. In many practical cases, however, the multimedia contents share the channel with other flows, so that the capacity available to video flows may vary in time. In this case, the RM algorithm should be able to estimate the new available rate and adapt the quality of the on-going flows accordingly. Since the capacity estimate is generally noisy, however, it is not possible to guarantee a minimum SSIM, or to completely avoid the risk of freezing or sharp quality variations.

To gain insights on the possible effects of noisy channel estimates, we can model the link rate experienced when downloading the hth GOP of video v as $r_v^h = r_v(c) + w_v^h$, where $r_v(c)$ is the link capacity estimated by the RM algorithm and w_v^h is an estimate error term assumed to be random, with zero mean and variance $\sigma_{r,v}^2$. Building upon the analysis developed in Sec. 9.6.2, we can now express the freezing probability as $P_f(n) = \Pr\left[S_v(n) \geq D'(n)\right]$ with $D'(n) = \tau_v\left(\gamma_v^\circ R(n+n_0) + \sum_{h=1}^{n+n_0} w_v^h\right) = D_v(n) + Y(n)$ where $Y(n)$ has zero mean, so that $\mathrm{E}\left[S_v(n) - Y(n)\right] = \mu$, as in Sec. 9.6.2. Then, repeating the steps

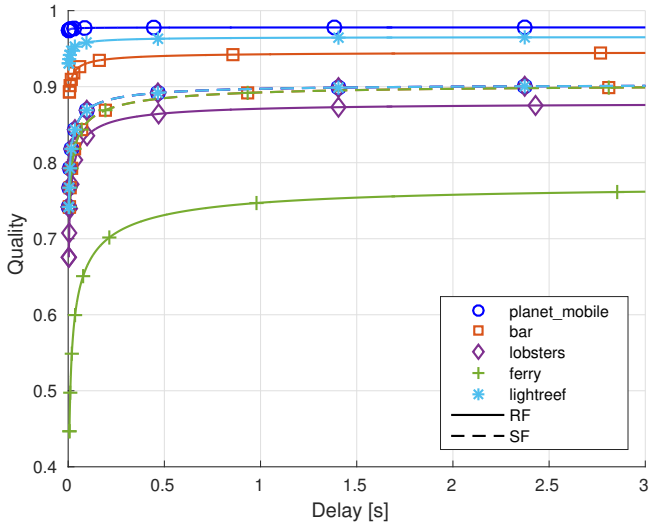


Figure 9.9: Video quality and initial playback delays for different values of α, using RF and SF.

of Sec. 9.6.2, we get

$$n_0 \geq \frac{\alpha \Delta'_v(c)}{s_v(c)} \sqrt{n \log\left(\frac{1}{\sqrt{P^*_f}}\right)} - n(1-\alpha) = \beta' \sqrt{n} - (1-\alpha)n$$

where $\Delta'_v(c) \geq \Delta_v(c)$ because of the additional variance due to rate estimation errors. Approximating $\Delta_v(c)'$ as $k\sqrt{\sigma_v^2(c) + \tau_v^2 \sigma_{r,v}^2}$ (i.e., increasing the variance of the GOP size to account for the channel capacity fluctuations), we get

$$\beta' = \frac{\alpha \Delta'_v(c)}{s_v(c)} \sqrt{\log\left(\frac{1}{\sqrt{P^*_f}}\right)} \simeq \frac{k\sqrt{\sigma_v^2(c) + \tau_v^2 \sigma_{r,v}^2}}{\gamma_v^\circ(c) R \tau_v} \sqrt{\log\left(\frac{1}{\sqrt{P^*_f}}\right)} \tag{9.10}$$

$$n_0 = \beta' \sqrt{\min\{n^{*\prime}, n_{\max}\}} - (1-\alpha)\min\{n^{*\prime}, n_{\max}\} \tag{9.11}$$

with $n^{*\prime} = \frac{\beta'^2}{4(1-\alpha)^2}$.

Clearly, the size of the play-out buffer impacts the initial delay τ_0. A rough estimate of τ_0 can be obtained by assuming that the aggregate size of the initial n_0 GOPs is equal to $n_0 s_v(c) = n_0 \alpha \tau_v \gamma_v^\circ R$ and that these GOPs are downloaded at the assigned share of the nominal link rate, i.e., $\gamma_v^\circ R$, so that we get $\tau_0 = \alpha n_0 \tau_v$. From this result and (9.11), we see that the smaller α, the lower τ_0. On the other hand, the smaller α, the lower the quality of the segments downloaded by the CHP. There exists then a tradeoff between the initial playback delay and the average quality of the video when varying α. Fig. 9.9 shows such a tradeoff for a few sample videos, when using both the SF (dashed line) and RF (solid lines) RM algorithms. The plot has been obtained by setting $k = 7$, $\sigma_v(c)/s_v(c) = 5\%$, $P^*_f = 5\%$, and $\sigma_{r,v} = 0.01$. The results show that SF makes it possible not only to offer the same quality to all video sequences, but also to

provide the same playback delay for a certain quality level. RF, instead, can give better quality (or lower playback delay) to certain videos, while others will suffer very poor quality, even when the initial delay is allowed to be large.

9.9 Conclusions and future directions

This chapter described a framework for video admission control in wireless systems that exploits machine learning algorithms to optimize resource management. By means of simulation, it has been shown that the proposed framework outperforms offline video analysis techniques in terms of the trade-off between QoE delivered and computational costs.

Further improvements of the proposed method could be obtained by extending the unsupervised learning phase by using a richer input vector, including other encoding parameters, and a deeper architecture, thereby considering a hierarchical generative model of the data distribution [209]. However, more complex models usually need larger training datasets, which must provide enough statistical information to extract a good set of descriptive features. An important step would therefore be to also increase the amount of data used to train the generative model, which can be accomplished by collecting more videos or integrating other available datasets into the framework. Finally, exploiting unsupervised learning to build an expressive set of high-level features allows great flexibility to the proposed framework, which can be used to transfer knowledge across several tasks [230].

Chapter 10

Just-In-Time Proactive Caching For DASH Video Streaming

Following the widespread adoption of adaptive video streaming algorithms on the client side, this study proposes a pre-fetching proxy cache, to be placed at the network's edge, which will predict the quality that the client will request for the following segment. The proxy predicts the future network conditions and models the system as a Markov Decision Process (MDP), in order to find the optimal decision for the proxy, given the current network conditions. This Just-in-Time caching technique pre-fetches the segment just before the client requests it, aiming to decrease the total time spent by the client downloading the segments, and indirectly increasing the user's QoE, as the DASH client will perceive better network conditions.

10.1 Introduction

Caching and pre-fetching are well known techniques to improve the user QoE in video streaming applications. Caching can be defined as the temporary storage of an object for future use, and pre-fetching can be defined as the action of requesting an object that is expected to be needed in the near future. The introduction of proxy caches in the network has been studied, along with various caching strategies. The purpose of this is to make the network less vulnerable to congestion by making the same content available in places other than the server, so there are fewer requests sent directly to the server, and users can get better performance by streaming the content from the closest available cache. There have also been some attempts to study the impact of pre-fetching on user QoE [231,232]; however, the cache often needs a considerably large size in order to yield an acceptable cache hit percentage.

The study of caching and pre-fetching solutions is widely saturated, however, a caching strategy which involves a pre-fetching technique that predicts the client's future behavior is yet to be explored. Therefore, the main objective of this work is to study the impact on the user QoE of the introduction of

a predictive proxy cache in the edge of a network, in a connection between a DASH client and a DASH server.

By placing a cache closer to the requesting client, the Round Trip Time (RTT) of the connection will decrease, reducing the time taken by the client to download a video segment. This will induce the client to perceive a better channel capacity, enabling it to request segments with higher qualities. The aim of the proxy cache is to increase the user's QoE by: i) increasing the video quality; ii) reducing the amplitude of the variation in the video quality; iii) reducing the frequency of the video quality variation; iv) reducing the frequency of stalling events; v) reducing the duration of stalling events.

Since real caches have a limited storage capacity, there must be an efficient way to determine which segments to store in the cache over time. To achieve this, a probabilistic approach is considered to predict which video segment the client will request, given the current network conditions. This prevents the unnecessary storage of unused video segments as well as the unnecessary use of bandwidth to pre-fetch these segments from the server.

10.2 State of the Art

This section describes the most relevant works published in the literature concerning pre-fetching strategies.

A network awareness study is conducted by Bronzino *et al.* [233], where the authors aim to optimally use the available end-to-end bandwidth by using intermediate nodes to cache video content closer to the client, thus distributing the traffic load over time. Ultimately, the solution aims to improve the QoE for the end user. The proposed solution involves moving the decision on the segment's quality into the network, by introducing a controller and a cache in the edge of the network. In this study, the client will only request the required video segment, and will receive that segment with the bitrate pre-fetched by the controller, which will be stored in the cache. The study takes advantage of the fact that the available bandwidth is easier to predict having a general view of the network infrastructure resources available in the network. Using the information on the client playback and buffer status, the controller exploits the available resources and chooses an appropriate bitrate for the segment it will download to store in the cache. The bitrate selection algorithm used in this study chooses a combination of bitrates for the given sequence of segments to be downloaded at the time (bitrate path). The chosen bitrates are the ones which lead to the highest QoE, given the current network conditions. This algorithm runs within a given time frame, returning the bitrate path with the highest QoE when the time frame ends.

A limitation that is addressed by this study is that in the long run, it is possible that when a new time frame begins, the network conditions will be different from the ones that were considered in the previous time frame, and this may lead to a significant QoE drop if the network conditions are less favorable at this time. Another limitation that can be seen in this study is the fact that the algorithm must be run once for every decision that must be made. Even though the proposed solution introduces some additional costs for the content provider, the authors claim that the resource costs are minimal and that the achieved gain in QoE outweighs the computational costs.

In [231], Liang *et al.* combine both caching and pre-fetching to improve the performance in terms of byte-hit ratio and video bit rates. The architecture of this solution is composed of three modules: a cache manager, a pre-fetch manager and a request pool. The cache manager handles all user requests and video segments received from the content server. If there is a cache-miss, it sends a request to the request pool, which in turn forwards the request to the content server. The cache manager also generates pre-fetch requests for every user request, and sends those to the pre-fetch manager. It only generates pre-fetch requests for successive video segments with the same bit rate as the current request. When the video segment arrives, if the cache is full, the cache manager makes a decision on whether to keep or discard the segment, based on its utility. The pre-fetch manager decides whether the received pre-fetch request should be sent to the request pool or not, based on its current usage.

Evaluation is done by comparing to three alternatives: Least Recently Used (LRU)-based caching approach, a popularity-based caching approach called Popular Content (PC) which caches the top 100 most popular in advance, and an aggressive pre-fetching approach. The study shows that the Average Per-User Throughput (APUT) is 50% higher compared to LRU and PC, and 31% higher when compared to the aggressive pre-fetching, for a cache size of 1GB. In terms of byte-hit ratio, the architecture improves the performance by nearly 84% compared to the aggressive pre-fetching, and a performance gain between 5 and 8 times larger when compared to LRU and PC. This study, however, does not take into account the volatile behaviour of the channel, as it always chooses to pre-fetch the same bit rate as was requested previously.

In [232], Krishnappa et al. investigate the advantages of having a pre-fetching and caching scheme for Hulu (a free hosting service of professionally created video for films and TV shows). The pre-fetching scheme is based on caching the most popular videos of the week provided by the Hulu website. It is compared to the conventional LRU caching. Results show that this yields a hit ratio of up to 77.69% but requires a storage of 236 GB. When evaluating the performance of pre-fetching the popular videos list, it is noted that a maximum hit ratio of 44.2% is obtained when pre-fetching 100 videos, corresponding to a cache storage of 10GB. For the same storage space, the LRU caching scheme yields a hit ratio of 45.53%; however, in this case 5767 videos are downloaded, compared to only 100 when pre-fetching.

Binging is a new trend which has also been studied. Binging is when a user watches multiple episodes of a television programme in rapid succession, typically by means of DVDs or digital streaming. For example, Claeys *et al.* [234] take advantage of the recent trend. Studies show that users stream on average 2.3 episodes per viewing, and so 57% of the streaming sessions could be announced in advance by a proxy. If these announcements were to be made, it would enable a simple prediction for future segment requests and subsequent episodes could be cached in advance, allowing for an improved QoE.

The evaluation of this study is based on the byte hit ratio. It notes an increase in performance of 54% in comparison to the LRU caching strategy. A limitation on this approach is that it does not take into account the possibility of the user ending the session before the episode ends. If this were to happen, many segments would be stored in cache with no purpose, as they would not be served to the client. Also, the bandwidth that would be used to pre-fetch these segments could have been used to serve other clients.

Figure 10.1: Schematic of the considered scenario.

Zhang *et al.* [235] present a dependency-aware caching algorithm which takes into account a dynamic network condition. The study assumes multiple users and multiple requests per user, and therefore aims to improve QoE for all users generally and not for a specific user. The algorithm is based on the profit of caching a certain segment, which is defined by the increase in utility of caching that segment. The utility of caching a segment depends on the available bandwidth, the segment size, the number of active client sessions and the number of requests per session. The algorithm decides to cache segments in descending order of profit, and depending on how full the cache storage is.

10.3 System Model

As depicted in Fig. 10.1, consider a DASH client that is streaming a video by downloading consecutive segments from a server. A proxy is placed between the client and the server, intercepting the client's segment requests and answering them directly if the chosen segment is present in its cache. Assume that the proxy proactively tries to predict the next segment that the client will request and pre-fetches it from the server; if the proxy pre-fetches and stores the correct adaptation, the latency the client experiences is far lower, improving the user QoE.

Let $a_c(n)$ be the adaptation chosen by the client for the n-th segment, and $a_p(n)$ be the one pre-fetched by the proxy. We can distinguish two different scenarios:

- *Cache hit*: if $a_p(n-1) = a_c(n)$, and the proxy has finished downloading the pre-fetched segment when the client's request arrives, the client downloads the segment from the proxy, which requests the next predicted adaptation to the server at the same time;
- *Cache miss*: if $a_p(n-1) \neq a_c(n)$ or the pre-fetching is not complete, the client's request is forwarded to the server, and either the proxy abstains from pre-fetching the next segment (resulting in another cache miss) or its pre-fetching will be in direct competition for the link between server and proxy with the client's download. In any case, the client is prioritized with respect to the proxy, so the latter can only use the bandwidth that is not used by the client.

We can model the scenario above as an MDP, a class of Markovian model defined by a state space S and an action space A, both finite and discrete, a state transition matrix M whose elements are the transition probabilities between states s_n and s_{n+1}, and a reward function $r(s_n, s_{n+1}, a_p(n))$.

The action space of the problem is represented by the proxy's pre-fetching choices $a_p(n)$: the solution to the problem is the policy $\Pi^* : S \longrightarrow A$ which maximizes the expected reward function for the next step. Note that, as explained above, refraining from pre-fetching a segment is a valid action and should be included in the problem definition. The objective of the proxy is to maximize the client's reward function, which depends on the user QoE for the video; we also assume that the proxy knows the adaptation logic the client is running and can then predict the client's actions in any given situation, in order to preserve the Markov property. In order to model the problem as an MDP, we need to define the reward function, the system state and the transition matrix.

10.3.1 Reward function

As discussed in Sec. 10.2, the QoE of a video client depends on the visual quality of the current segment, the quality variation between segments, and the playout freezing events due to rebuffering. In the following, a reward function that captures these aspects is introduced and that, in turn, can be used to derive policies that maximize the QoE of video streaming customers.

In this work, the bitrate is considered a proxy for picture quality, but the framework supports any objective QoE metric, which could be pre-computed by the server and served to the client along with the MPD or computed live using an appropriately trained deep neural network [221].

We define the reward function for the pre-fetched segment n as follows:

$$r(q_{n-1}, q_n, \phi_n) = q_n - \beta \|q_n - q_{n-1}\| - \gamma \phi_n, \tag{10.1}$$

where ϕ_n is an indicator variable that is equal to 1 in case a rebuffering event happens. The first term on the right-hand side accounts for the benefit of a higher quality q_n of the video, while the following two negative terms are penalty factors due to *quality variations* in consecutive frames and *rebuffering events*, respectively. The coefficients β and γ are weighting factors that regulate the relative importance of the three penalty terms. Note that the structure of the reward function (10.1) was first proposed and validated by De Vriendt *et al.* in [236], and is used as a comprehensive QoE metric by several algorithms in the literature [183, 186, 237].

The weights β and γ are here used to select different points in the trade-off between a high instantaneous quality, a constant quality level, and a smooth playback. The desired operational point might depend on several factors, including user preferences and video content, and tuning these parameters is outside the scope of this work.

The optimal policy $\Pi^* : S \longrightarrow A$ is defined as the policy that maximizes the expected value $E[r_n | s_n, \Pi^*]$ in any state.

10.3.2 MDP definition

Since q_n is directly involved in the reward calculation, its value should be included in the state definition in order to fit the definition of the MDP. Two other parameters indirectly affect the state transitions and the reward: the buffer level and the capacity C_n experienced by the client. Let B_n denote the buffer level at the beginning of the n-th segment download, and consider it part of the state

definition. The other parameter, the capacity, is composed of three values: the capacity of the link between client and proxy $C_{CP}(n)$, the capacity of the link between proxy and server $C_{PS}(n)$, and an indicator variable H_n which is equal to 1 in case of a cache hit and 0 otherwise. If $H_n = 1$, $C_n = C_{CP}(n)$, while in the cache miss scenario $C_n = \min(C_{CP}(n), C_{PS}(n))$.

The complete state of the MDP is then a 5-tuple: $s_n = (B_n, q_n, C_{CP}(n), C_{PS}(n), H_n)$. Here it is assumed that the policy implemented by the client is known to the proxy, which should be able to determine the probability distribution of the client's actions in any given state s_n.

10.3.3 Small-scale model

The definition of an MDP to represent the video streaming scenario has one major issue: since the download of different segments in different network conditions will take different amounts of time, the Markovian assumption can not be justified without some extra steps.

In order to overcome this problem, the capacity of the links between client and proxy (C_{CP}) and between proxy and server (C_{PS}) is modeled as two independent Markov processes, with a time step T which should be far smaller than the average download time of a segment. If we denote the number of undelivered bits in the segment at step t as b_t, we get:

$$b_{t+1} = b_t - C_t T, \tag{10.2}$$

where C_t is the capacity experienced by the client. Let T_{setup} be the number of time steps that the client needs to send the request and receive the response from the server or proxy, during which no useful bits can be downloaded; since the capacity of the client is Markovian, we can calculate the probability that an adaptation $a_c(n)$ will take N time steps, given the initial capacities and the proxy's action:

$$P(N(a_c(n)) = N | C_t^{CP}, C_t^{PS}, T_{\text{setup}}, H_n, a_p(n)) = \sum_{\mathbf{C}} p(\mathbf{C}) \delta\left(F(a_c(n)), C\right), \tag{10.3}$$

where $F(a_c(n))$ is the frame size for adaptation $a_c(n)$, $\mathbf{C}$ is the vector of channel capacities experienced by the client from time $t + T_{\text{setup}} + 1$ to $t + N$, and C is the sum of all its elements. The solution to this equation can be computed recursively for any initial combination $(N, F(a_c(n)), C_t^{CP}, C_t^{PS})$ and stored for later use. The time from one client decision to the next is simply $TN(a_c(n))$.

In order to get a cache hit for segment $n + 1$, two conditions have to be met: the proxy has to correctly predict the adaptation the client will require, and it has to download the segment before the client requests it (i.e., the proxy download needs to take $M < N$ time steps). The probability of the latter can be computed as:

$$P(M(a_p(n)) = M | C_t^{PS}, H_n) = \sum_{\mathbf{C}} p(\mathbf{C}) \delta\left(F(a_p(n)), C\right), \tag{10.4}$$

where $\mathbf{C}$ now indicates the vector of proxy-server channel capacities from time $t + T_{\text{setup}} + 1$ to time $t + M$. In this case, the capacity for the proxy is equivalent

to $C_{PS}(t)$ if $H_n = 1$, and $C_{PS}(n) - \min(C_{CP}(n), C_{PS}(n))$ otherwise, as it has to share the link with the client. In the latter scenario, M and N are not independent, and so must be computed together.

In this way, the large-scale transition from one state $s_n = (B_n, q_{n-1}, C_{CP}(n), C_{PS}(n), H_n)$ to the next $s_{n+1} = (B_{n+1}, q_n, C_{CP}(n+1), C_{PS}(n+1), H_{n+1})$ can be modeled as a Markov process; the joint computation of all variables yields the transition matrix of the MDP.

10.3.4 Solution

Since the problem has a finite horizon, it is possible to solve the MDP analytically:

$$a_p^*(s_n) = \underset{a_p \in A}{\operatorname{argmax}} \sum_{s_{n+1} \in S} E\left[r_{n+1} | s_{n+1}\right] P(s_{n+1} | s_n, a_p). \tag{10.5}$$

The two parts of the equation can be simply determined:

$$E[R_n | s_n] = \sum_{N=0}^{\infty} r(q_{n-1}, q(a_c(n)), u\left(NT - B_n\right)) P(N(a_c(n) = N | s_n), \tag{10.6}$$

where $u\left(\cdot\right)$ is the Heaviside step function. The last term can be calculated using (10.3), which was derived from the small-scale channel model. The probability $P(s_{n+1}|s_n, a_p(n))$ is computed directly from the small-scale model in Sec. 10.3.3. The expected reward from all the actions in all states can be then computed, and a simple *argmax* operation is enough to determine the optimal policy.

This brute-force policy calculation is computationally feasible for problems of relatively limited size and with a short-term temporal horizon; for larger and more long-term oriented problems, solutions such as reinforcement learning can be considered, but this is beyond the scope of this work and will be analysed in a future extension of it.

10.4 Results

This section presents a simulative comparison between the previously described proactive pre-fetching proxy, a pre-fetching proxy that pre-fetches the next segment having the same quality as the one previously downloaded by the client, and the scenario with no proxy. The client runs the algorithm described in [237].

10.4.1 Simulation scenario

The simulation scenario is simple: a proxy is placed between a DASH client and a server, with two independent channels between the client and the proxy and between the proxy and the server. The first has an RTT of 50 ms, while the second has an RTT of 200 ms. As described in Sec. 10.3.3, the two channels are modeled as independent Markov processes with transition matrices $\pi_{C,PS}$ and

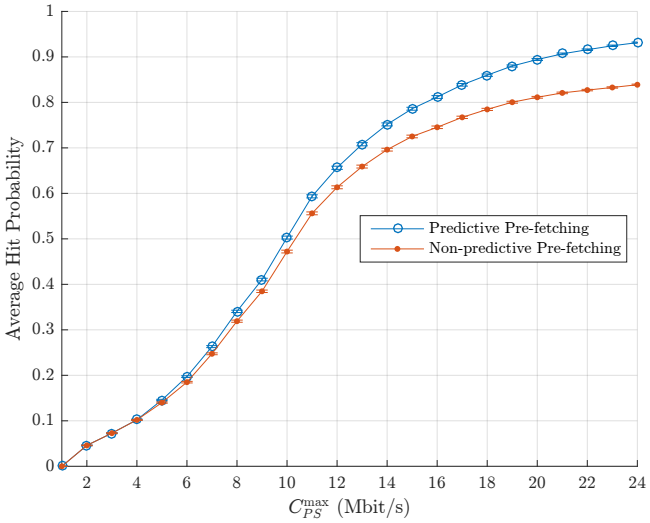


Figure 10.2: Average hit probability, with 95% confidence intervals.

$\pi_{C,CP}$, defined as

$$\pi_C = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0.05 & 0.9 & 0.05 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0.05 & 0.9 & 0.05 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0.9 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0.05 & 0.9 & 0.05 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0.1 & 0.9 \end{bmatrix}.$$

The size of this square matrix is different for the client-proxy and proxy-server channels. In particular, the states represent the link capacity measured in Mbit/s, starting from 1 up to a maximum channel capacity. The maximum channel capacity for the client-proxy link is set to $C_{CP}^{\max} = 6$ Mbit/s, so the six states for the client-proxy channel model correspond to available capacities of $\{1, 2, 3, 4, 5, 6\}$ Mbit/s. The same holds for the proxy-server link; however, its maximum capacity $C_{PS}^{\max}$ has been varied from 1 Mbit/s to 24 Mbit/s to explore different scenarios, including the ones where the proxy-server link capacity is actually lower than the client-proxy capacity, which is a challenging setting for a prefetching proxy. The time step T for the small scale model was set to 100 ms.

In the following simulations the values $\beta = 6$ and $\gamma = 10$ are used as the weighting factors in the reward function in (10.1).

10.4.2 Hit probability

First, we analyse the average cache hit probability when varying the maximum proxy-server link capacity (Fig. 10.2).

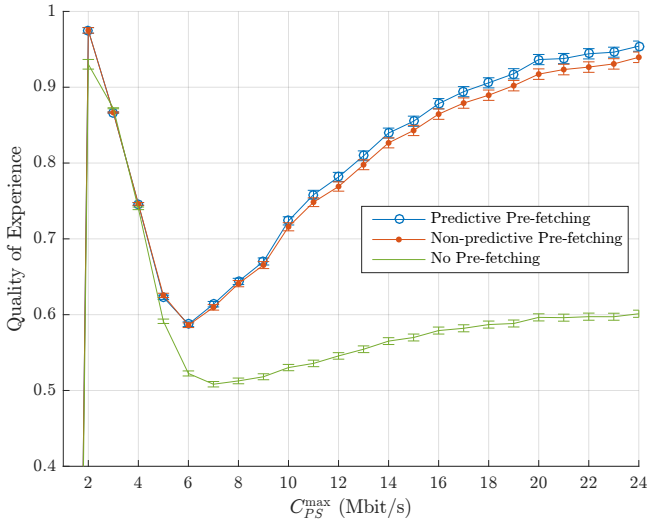


Figure 10.3: QoE, with 95% confidence intervals.

For very low values of $C_{PS}^{\max}$ it is unlikely that the proxy is able to pre-fetch segments before the client requests them. In order to better explain this statement, consider the scenario where the next segment requested by the client has the same quality as the currently downloaded one. For the proxy to be able to successfully pre-fetch the next segment, the proxy-server bandwidth available to the proxy has to be at least equal to the client-proxy bandwidth. As soon as a cache miss happens, since the proxy-server capacity is low, the client requests the segment from the server using almost all of the available bandwidth, thus preventing the proxy to download the correct quality for the next segment. This causes an avalanche of cache misses, which lower the average hit probability.

Instead, when the maximum proxy-server link capacity is high, the hit probability is large, particularly for the predictive proxy. This proves that the predictive proxy is able to accurately predict the next segment quality even when the number of states in the channel model, and, therefore, its capacity variability, is large. As expected, the same quality pre-fetching strategy offers an inferior performance. Since the proxy pre-fetches the different qualities in order of how likely they are to be requested, as the maximum proxy-server capacity approaches to infinity, the proxy is able to pre-fetch any segment quality without considering the capacity limitation, thus yielding an average hit probability close to 1.

10.4.3 Average QoE

In order to measure the overall QoE, a metric proposed in [237] is used. This metric is a linear combination of the average video quality $\bar{q}$ and its standard deviation σ_q, both normalized by the maximum available quality $q_{\max}$, and a parameter F that models the influence of stalling events.

$$\text{QoE} = 5.67\frac{\bar{q}}{q_{\max}} - 6.72\frac{\sigma_q}{q_{\max}} - 4.95 \cdot F + 0.17\,, \tag{10.7}$$

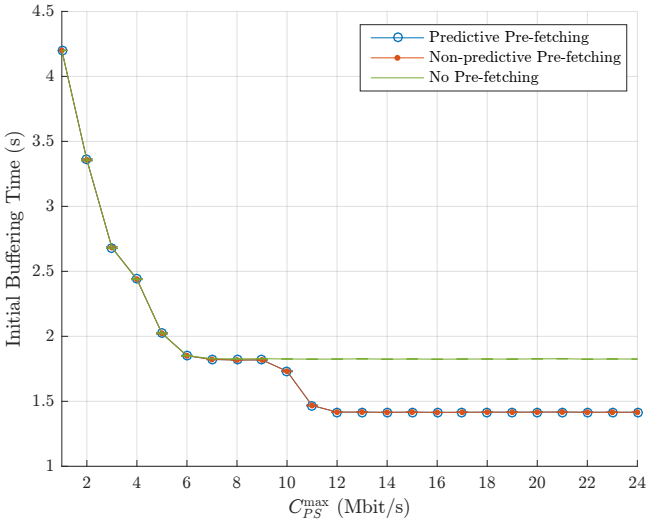


Figure 10.4: Initial buffering time, with 95% confidence intervals.

with F defined as

$$F = \frac{7}{8} \max\left(\frac{\log(\phi)}{6} + 1, 0\right) + \frac{1}{8} \cdot \frac{\min(\psi, 15)}{15}\,, \tag{10.8}$$

where ϕ is the frequency of stalling events and ψ is their average duration. Fig. 10.3 shows a large increase in QoE by using a caching proxy and, in particular, by using a predictive pre-fetching strategy.

For low proxy-server maximum capacity, the client, as explained earlier, is almost always downloading the segment from the server. With such a low proxy-server capacity, the channel model has very few states, causing a channel with low capacity but also low variability. In this scenario, the client adaptation algorithm is able to predict the channel very well, thus avoiding quality switches and rebuffering, offering large QoE even when playing low quality segments.

From there, increasing the proxy-server maximum capacity brings higher variability to the channel, while still forcing the download of low quality segments. This decreases the QoE value up to a point, which in our scenario is $C_{PS}^{\max} = 6$ Mbit/s, where the large channel variability is compensated by the possibility to play high quality segments. From there, the QoE increases with the maximum proxy-server capacity.

10.4.4 Initial buffering time

The time needed to fill the buffer before the actual start of the playout is not considered in the QoE metric plotted in Fig. 10.3. Therefore, for completeness, its behavior is shown in Fig. 10.4 for different maximum proxy-server capacities. The playout is considered started, and therefore the initial buffering period ended, when the buffer contains the first 3 seconds of video. We observe that, for $C_{PS}^{\max}$ lower than 9 Mbit/s, the performance with and without the proxy is the same, because of the low hit probability which forces the client to download the segment directly from the server. Instead, for larger maximum proxy-server

capacity values the client benefits from the presence of a proxy, reducing the initial buffering time. The initial delay stabilizes for $C_{PS}^{\max} \approx 2C_{CP}^{\max}$, since, in this case, the probability of the proxy-server link being the bottleneck is almost zero.

10.4.5 Advantages of the scheme

In conclusion, the benefit of using a pre-fetching proxy is clear: Fig. 10.3 shows a significant increase in the QoE for both pre-fetching schemes. The predictive pre-fetching scheme increases the QoE only slightly with respect to the simple non-predictive scheme, but its advantage lies in the significantly higher hit rate: since every cache miss means that a segment is downloaded by the proxy but never used by the client, increasing the hit rate greatly improves the efficiency of the system. A predictive pre-fetching system, installed before the "last mile" in a DSL or cellular network, can help improve end users' QoE without imposing a significant additional load on the core network.

10.5 Conclusions

In this work, a DASH video streaming scenario including a proactive pre-fetching proxy has been modeled as an MDP, and the optimal strategy for the proxy has been found.

Although the benefits in terms of QoE are not huge when compared to a simpler proxy which pre-fetches the next segment at the same quality as the one currently streamed by the client, the combined increase in QoE and cache hit rate means that the predictive proxy can exploit its better awareness of the scenario to provide a better quality while wasting less bandwidth and reducing the load on the server.

Further improvements to the proxy could be achieved by using reinforcement learning in order to maximize the long-term QoE instead of just the instantaneous one.

Chapter 11

Features selections and machine learning techniques for Non-LOS detection in UWB transmissions

In CPSs, positioning is an important service, not only because it provides the control loop with information about device operation, but also as an aid to the communication subsystem, enabling improved performance on bitrate adaptation or beamforming techniques. Indoor systems often use triangulation techniques based on ultra-wideband (UWB) pulses to perform accurate ranging estimates, since satellite positioning services are not available inside buildings. However, the ranging accuracy can be greatly degraded in case of non-line-of-sight (NLOS) conditions of the propagation channel, which are therefore crucial to correctly identify. This chapter provides a systematic study that includes multiple machine learning algorithms and signal features, in order to identify which features are more informative for each machine learning technique, and which combination performs the best. Furthermore, a technique exploiting multiple signals received from different directions is proposed.

11.1 Introduction

UWB location systems are seen as a promising solution to enable robust and precise positioning services in indoor environments [238]. The most common technique to position of a device in UWB systems is to estimate its distance to some reference points, called beacons, by considering the time of arrival (ToA) or the time difference of arrival (TDoA) of the signals transmitted by such beacons, and then applying triangulation or multilateration techniques [239]. Instead, the received signal strength (RSS) or angle of arrival (AoA) techniques are less popular, since they do not exploit the fine space resolution of impulsive signals and thus offer lower accuracy [240, 241].

The main source of ranging errors in ToA UWB ranging is the presence of NLOS components in the received signals, which introduce a positive time bias

that, in turn, results in an overestimate of the transmitter-receiver distance [241–243]. A common and simple strategy to mitigate this type of ranging errors is to detect the received signals with a strong NLOS component, which can then be discarded or weighted less than the line-of-sight (LOS) signals when performing the ranging estimate [241, 244].

Therefore, much effort has been dedicated to the study of effective and efficient ways to discriminate between NLOS and LOS propagation conditions in UWB transmissions. The basic idea consists in exploiting some specific features in the received UWB impulse that are likely affected by NLOS propagation, such as kurtosis, delay spread, energy, and others. However, the extent to which these features are influenced by the propagation conditions strongly depends on the characteristics of the environment, thus making difficult the definition of threshold-based classification criteria.

One possible approach to circumvent these difficulties is to train machine-learning algorithms to automatically classify the received UWB impulse as LOS or NLOS. However, which signal features are more informative and what is the most promising machine learning algorithm to recognize NLOS propagation conditions in UWB transmissions is still unclear.

This chapter sheds some light on these problems by performing a comparative study among different machine learning algorithms. More specifically, we consider SVM, which has been widely used in the literature for such a purpose, with other four machine learning techniques that are considered state-of-the-art in many classification tasks, namely k-Nearest Neighbors (k-NN), ANN, Naive Bayes (NB), and Logistic Regression (LR). Altogether, the five techniques compared in this study cover a broad variety of approaches: generative, discriminative, distance-based, and regression models. The analysis is performed on real measurements collected in heterogeneous environments, thus making it possible to i) identify the signal features that are more useful in this classification task, and ii) assess the robustness of the NLOS identification techniques to variations of the building materials in the structure. A distinctive trait of this analysis is the use of an antenna array at the UWB receiver, which makes it possible to perform the classification task by considering the impulses simultaneously received from multiple directions. It will be shown how the accuracy of the NLOS detection task monotonically increases with the number of available antenna elements at the receiver, though with diminishing gain after a certain value. In connection with this analysis, we are also going to discuss a possible methodological pitfall in the validation of the classifiers due to the angular correlation of the received UWB signals.

In summary, the original contribution of this work consists in the following points:

- Performance comparison of five machine learning techniques, namely SVM, k-NN, ANN, NB, and LR, which have been *optimized* for NLOS.
- Identification of the best subset of UWB signal features for each algorithm.
- Evaluation of the NLOS performance when considering real measurements collected in different indoor environments.
- Analysis of the performance gain obtained by considering the multiple signals collected from different directions by using multi-antenna receivers.

Based on the obtained results, we can draw the following conclusions: i) the detection accuracy can be dramatically improved by using signals received from five (or more) equally-spaced directions; ii) SVM is the best performing algorithm, as also observed in some previous papers, but k-NN achieves very similar performance with lower complexity; and iii) the most useful UWB signal features are kurtosis, and mean and variance of the excess delay.

11.2 Related work

Much work has been dedicated to improve the accuracy of the LOS/NLOS classification task in UWB systems. Many classifiers, however, do not exploit the full potential of machine learning techniques. In [245], the authors use a likelihood ratio test employing the amplitude and delay statistics of the channel in NLOS and LOS scenarios. In [246], the empirical pdf of the range measurements is extracted from many samples collected at the same position, then it is compared to the expected pdf for a LOS scenario using binary hypothesis testing. Similarly, an approach based on distance characterization between probability distributions, using only root mean square delay spread and kurtosis as signal features, is presented in [247]. The work in [248] uses binary hypothesis testing to compare range measurements to NLOS error models, leveraging the fact that the variance of NLOS measurements is larger than that of LOS measurements. An error model is also used in [249] to identify range measurements affected by NLOS errors, based on the a priori knowledge of the standard deviation of the measurement noise. In alternative, NLOS identification and error mitigation can be performed by taking into account the geometry of the environment [250,251]. Another approach consists in performing an exhaustive search over subsets of range measurements using the least-median-of-squares method, with the aim of obtaining a consistent set of LOS range measurements [252].

In [253] and [254], both ToA range measurements and RSS information are used. The intuition is that NLOS conditions are usually associated with high range estimates and low RSS values. Analogously, [255] presents different techniques for NLOS identification based on temporal RSS measurements and applies it to Wi-Fi signals. The techniques are based, respectively, on Least Squares Support Vector Machine (LS-SVM), a Gaussian process classifier, and a hypothesis testing classifier. These techniques, however, do not use features extracted from the whole signal envelope, thus losing potentially useful information.

Multi-antenna receivers make it possible to extract additional information by comparing the signals received by the multiple antennas [256]. The technique introduced in [257] uses this approach by exploiting the distribution of the phase difference of two signals received through two different elements in an antenna array.

Other works use supervised machine learning techniques to differentiate between LOS and NLOS signals. In [258], the authors propose the use of a LS-SVM for this purpose. The feature space is composed by some characteristics of the received signal envelope, like maximum amplitude, energy, and kurtosis. A similar approach is proposed in [259] where Support Vector Data Description (SVDD) is used instead of SVM to perform the NLOS identification. Alternatively, a Relevance Vector Machine can be used as a LOS/NLOS classifier [260]. The

work in [261] presents a NLOS identification technique for on-body area networks using four time-domain features extracted from UWB signals, namely kurtosis, entropy, mean, and variance. In addition, SVM and Gaussian processes regressors are proposed in [262] to mitigate ranging errors in the NLOS case. Further approaches are reported in [241].

As it is apparent from this overview, the NLOS detection problem has been addressed with several approaches and considering different sets of signal features, so that it is not easy to understand which solution is preferable. This work aims at shedding some light on this aspect, by proposing a systematic comparison of five of the most representative data-driven classification algorithms on the same empirical dataset, and for different combinations of the UWB signal features, so as to identify the most promising combinations of features and machine learning approaches. In addition, in this work the feature set is expanded by considering signals received from different directions, which proves to be very informative for the NLOS identification task.

11.3 Signal features

Consider the case of a device receiving a beacon from an anchor. The received signal is usually given by the overlapping of a direct component (LOS) and multiple reflected contributors (NLOS). If the energy of the direct component is larger than that of the other terms, the signal propagation is said to be in LOS conditions, otherwise we refer to NLOS conditions. Compared to the NLOS case, a LOS propagation condition is generally characterized by (i) lower time spreading of the received signal; (ii) more peaky signal amplitude; (iii) higher energy of the first received signal component. In order to capture these characteristics, we now consider the following six features that can be extracted from the received signal $r(t)$ [258, 263]:

- Energy:
$$\mathcal{E}_r = \int_{-\infty}^{+\infty} |r(t)|^2 \mathrm{d}t \,. \tag{11.1}$$
- Maximum amplitude:
$$r_{\max} = \max_t |r(t)| \,. \tag{11.2}$$
- Rise time:
$$t_{\mathrm{rise}} = t_{\mathrm{H}} - t_{\mathrm{L}} \,, \tag{11.3}$$
with $t_{\mathrm{L}} = \min\{t : (r(t))^2 \geq \alpha^2 \sigma_n^2\}$, $t_{\mathrm{H}} = \min\{t : |r(t)| \geq \beta r_{\max}\}$, σ_n^2 being the variance of the thermal noise. The values of $\alpha > 0$ and $0 < \beta \leq 1$ have been set to $\alpha = 6$ and $\beta = 0.6$, as suggested in [258].
- Mean excess delay:
$$\tau_{\mathrm{MED}} = \int_{-\infty}^{+\infty} t\psi(t) \mathrm{d}t \,, \tag{11.4}$$
where $\psi(t) = |r(t)|^2/\mathcal{E}_r$ is the normalized power profile.
- Mean squared delay spread (MS-DS):
$$\tau_{\mathrm{MS}} = \int_{-\infty}^{+\infty} (t - \tau_{\mathrm{MED}})^2 \psi(t) \mathrm{d}t \,. \tag{11.5}$$

Parameter	Value
SVM	
Soft margin	10
Gaussian kernel scale	2
k**-NN**	
Number of neighbors	10
ANN	
Hidden layer size	2.5 * input cardinality
NB	
Gaussian kernel window width	10
LR	
Classification threshold	0.25

Table 11.1: Classifier parameters

- Kurtosis:

$$\kappa = \frac{1}{\sigma_{|r|}^4 T} \int_T \left(|r(t)| - \mu_{|r|}\right)^4 \mathrm{d}t \,, \tag{11.6}$$

where $\mu_{|r|} = \frac{1}{T}\int_T |r(t)|\mathrm{d}t$ is the signal mean and $\sigma_{|r|}^2 = \frac{1}{T}\int_T \left(|r(t)| - \mu_{|r|}\right)^2 \mathrm{d}t$ is the variance of the signal.

We observe that these features have already been used in the past to classify a received UWB signal as LOS/NLOS. However, here a more systematic study is proposed, that aims to test the capability of different machine learning algorithms to correctly and precisely classify a signal as LOS/NLOS based on a subset of these features, as better explained in the next section.

11.4 Machine learning techniques

In this section, the machine learning techniques considered in this study to perform the LOS/NLOS classification task are briefly described. Each of the following classifier has a number of parameters, which is tuned to achieve the best classification performance. The chosen parameters are reported in Tab. 11.1.

Support Vector Machine

SVM [264] is a supervised learning technique that aims to find the optimal hyperplane to linearly separate the input data once mapped into another space via the so-called kernel functions. SVMs proved to be effective even in very high dimensional spaces and are also efficient in terms of memory occupation, due to the use of only a subset of the training point (the *support vectors*) in the decision function.

k-Nearest Neighbors

The k-NN classifier is a non-parametric method that assigns an object[1] to the most common class in its neighborhood in the features space [265]. The neighborhood is composed by the nearest $k > 0$ objects among those used for training. The distance function can be any metric measure, however the standard Euclidean distance is the most common choice and is the one used in this analysis. Being a non-parametric method, this technique can exhibit very good performance even in situations where the decision boundary is very irregular. On the other hand, k-NN should, in principle, keep in memory all the training data to perform the classification, so its memory occupation can be significant.

Artificial Neural Network

ANNs have proved useful for a number of purposes, including classification tasks [266]. Their design has been inspired by the biological structure that constitutes animal brains: nodes (called *neurons*) are connected through *synapses*, usually composing a hierarchical structure. In feed-forward ANNs, input data is given to the network through the input layer, while results are obtained in the output layer. Other neurons are grouped in one or more hidden layers. Each neuron maintains a state and produces an output, both of which depend on the input and on a given activation function. Due to the low number of input features, a feed-forward ANN with a single hidden layer is used for the classification problem.

Naive Bayes

The NB classifier [267] is a simple probabilistic classifier based on the Bayes' theorem. This classifier considers the different features to be independent from one another, which is not the case in our scenario. However, in the literature, NB proved to achieve good performance even when the independence assumption does not hold.

Logistic Regression

LR is a predictive method used to categorically classify objects [268]. This method is one of the many forms of regression analysis and its concept is similar to the linear regression method. However, LR predictions of the outputs are categorical (i.e., picked from a finite and discrete set of categories) rather than continuous. In this work, the binary LR is used to decide between the LOS and NLOS cases.

11.5 Experimental setup

This section describes the way the data have been collected and processed to obtain the experimental results.

[1] In machine learning terminology, the term *object* indicates an element of the input space.

11.5.1 Dataset

Tests are conducted using publicly available data,[2] containing channel responses captured in indoor environments with receivers using circular array antenna [269]. Using an antenna array instead of a single antenna introduces spatial diversity in the system, allowing for measurements of both spatial and temporal properties of the channel. The used dataset contains data from four buildings made of different materials, namely:

- *NIST North*: sheetrock wall with aluminum studs;
- *Child Care*: plaster walls with wooden studs;
- *Sound*: cinder block;
- *Plant*: steel.

For each building, data is available for 50 different transmitter-receiver positions of which 40 are NLOS and 10 are LOS. For each position, 96 channel responses are collected, one for each element of the circular array antenna. Therefore, adjacent measurements are separated by an angle of 3.75 degrees. Frequency responses from the dataset have been combined with a classical impulse used in UWB localization techniques [270] to obtain the UWB signals, from which the six previously described features have been extracted and used either singularly, or in vectors, as explained later.

11.5.2 Machine learning training and testing

To test the performance of each learning technique, an n-fold cross-validation technique has been used [228]. In more detail, the dataset is partitioned into n subsets (folds). The training phase uses $n-1$ folds (training set), and the performance is computed on the left-out fold (validation set). This operation is repeated n times, each time changing the validation set. The experiments have been conducted by setting $n = 5$ and calculating the estimated classification accuracy averaged over all the 5 rounds. As we will see in the next section, measurements collected at a given location for different arrival angles can be correlated. To avoid any bias in the results due to such dependency, the data in the validation set are never taken from locations already considered in the training set. Furthermore, to avoid any bias due to the particular subdivision of the dataset in subsets, the n-fold cross-validation routine is iterated m times and the average of the estimated classification accuracy over all the m iterations is reported.

For the single building scenario, the dataset used is from the NIST North building, while data from all four buildings is employed in the multi-building case.

The classifiers used in this study are the one implemented in MATLAB's Statistics and Machine Learning Toolbox [271]. For each classifier, the input vector is given by a subset $\mathcal{F}$ of the six features described earlier.

In the validation phase, the number of true positive (T_P), false positive (F_P), true negative (T_N), and false negative (F_N) events are counted. Since the final objective of the work is to detect the NLOS condition (e.g., to decrease

[2] `https://www-x.antd.nist.gov/uwb/`

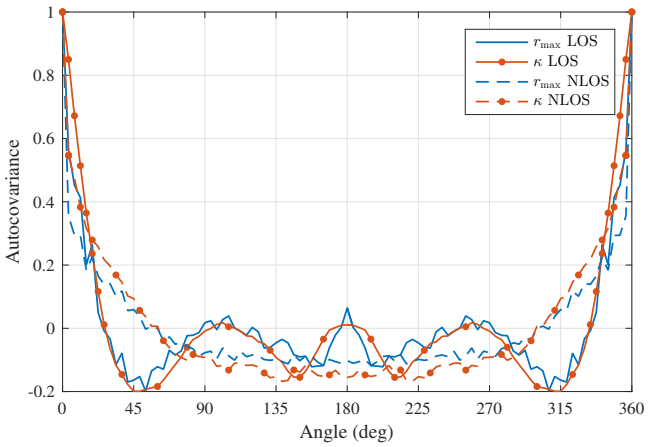


Figure 11.1: Autocovariance of $r_{\max}$ and κ (other features have similar shapes) when varying the direction of the antenna. The values have been averaged over all positions on all four buildings.

the weight of such signals in localization algorithms), the false positive event refers to the detection of a NLOS condition whenever the real condition is LOS. Analogously, the false negative event refers to the detection of a LOS condition in place of NLOS. The F1 score is defined as the harmonic mean between precision $\left(\frac{T_P}{T_P+F_P}\right)$ and recall $\left(\frac{T_P}{T_P+F_N}\right)$, which results in

$$\mathrm{F1} = \frac{2T_P}{2T_P + F_P + F_N} \,. \tag{11.7}$$

The probability of false alarm (p_{FA}) and missed detection (p_{MD}) are, respectively,

$$p_{\mathrm{FA}} = \frac{F_P}{F_P + T_N} \,; \qquad p_{\mathrm{MD}} = \frac{F_N}{T_P + F_N} \,. \tag{11.8}$$

11.6 Experimental results

This section presents the results of the performed experiments, together with a proposal to improve classification performance by exploiting signals simultaneously received by different elements of the circular array antenna.

11.6.1 Analysis of correlation between antenna directions

As mentioned, the measurements collected for different arrival angles at a given location can be correlated. Fig. 11.1 shows the covariance of some features (namely, maximum signal amplitude and kurtosis) when varying the angular lag, both in the LOS and NLOS cases. Since correlated data bring lower information, we expect that the learning phase can be performed by considering only a subset of almost-uncorrelated measurements, without affecting the accuracy of the results. To test this conjecture, a number of reduced datasets is extracted from the whole dataset, each obtained by picking only k measurements with maximum angular distance out of the 96 available for each transmitter-receiver

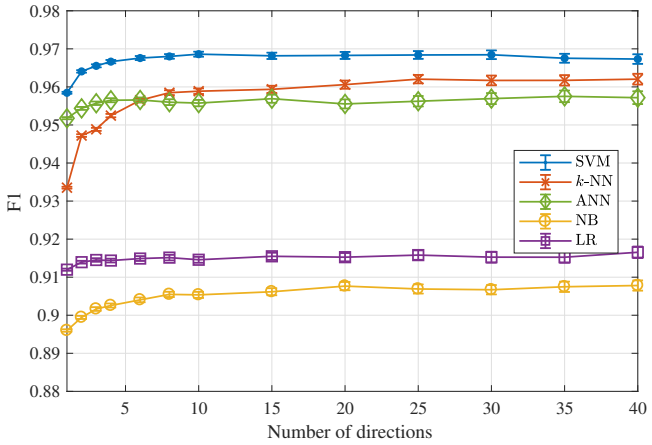


Figure 11.2: Average F1 score (four buildings case) with 99% confidence intervals when using all six features but only a subset of available directions for each location.

f	1	2	3	4	5	6
SVM	$\{\kappa\}$ $\{\kappa\}$	$\{r_{\max}, \tau_{\text{MED}}\}$ $\{\tau_{\text{MS}}, \kappa\}$	$\{r_{\max}, \tau_{\text{MED}}, \tau_{\text{MS}}\}$ $\{r_{\max}, \tau_{\text{MED}}, \kappa\}$	$\{r_{\max}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{r_{\max}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$	$\{r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$	$\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$
k-NN	$\{\kappa\}$ $\{\kappa\}$	$\{r_{\max}, \tau_{\text{MED}}\}$ $\{\tau_{\text{MS}}, \kappa\}$	$\{r_{\max}, \tau_{\text{MED}}, \tau_{\text{MS}}\}$ $\{r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}\}$	$\{r_{\max}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{r_{\max}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$	$\{r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$	$\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$
ANN	$\{\kappa\}$ $\{\kappa\}$	$\{r_{\max}, \tau_{\text{MED}}\}$ $\{\tau_{\text{MS}}, \kappa\}$	$\{\mathcal{E}_r, \tau_{\text{MED}}, \kappa\}$ $\{r_{\max}, \tau_{\text{MED}}, \kappa\}$	$\{\mathcal{E}_r, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \kappa\}$	$\{\mathcal{E}_r, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$	$\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$
NB	$\{\kappa\}$ $\{\kappa\}$	$\{r_{\max}, \kappa\}$ $\{\mathcal{E}_r, \kappa\}$	$\{r_{\max}, \tau_{\text{MED}}, \kappa\}$ $\{t_{\text{rise}}, \tau_{\text{MS}}, \kappa\}$	$\{\mathcal{E}_r, r_{\max}, \tau_{\text{MS}}, \kappa\}$ $\{\mathcal{E}_r, t_{\text{rise}}, \tau_{\text{MED}}, \kappa\}$	$\{r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{\mathcal{E}_r, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$	$\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$
LR	$\{\kappa\}$ $\{\kappa\}$	$\{\tau_{\text{MED}}, \kappa\}$ $\{r_{\max}, \kappa\}$	$\{t_{\text{rise}}, \tau_{\text{MS}}, \kappa\}$ $\{\mathcal{E}_r, \tau_{\text{MED}}, \kappa\}$	$\{t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{\mathcal{E}_r, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$	$\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MS}}, \kappa\}$ $\{\mathcal{E}_r, r_{\max}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$	$\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$ $\{\mathcal{E}_r, r_{\max}, t_{\text{rise}}, \tau_{\text{MED}}, \tau_{\text{MS}}, \kappa\}$

Table 11.2: Feature sets that provide the best F1 score for each feature set size f and learning technique. The first row for each technique refers to the single building scenario, the second to the four buildings scenario.

position. The training-validation analysis of the four machine learning techniques is then performed on such reduced datasets. Note that, to counteract the higher variability in the dataset when k is small, the number m of iterations of the cross-validation routine is adapted to k in order for the product $k \times m$ to remain always equal to 9600.

Fig. 11.2 confirms our premises: the accuracy of the classification improves with the number of angles, but the gain becomes negligible when using more than about 20 directions, though already with 5 angles the performance is close to the maximum. According to this result, in the following analysis only 20 out of 96 measurements is used for each position.

11.6.2 Prediction accuracy for different feature sets

In Fig. 11.3 and Fig. 11.4 the F1 score obtained for each technique is reported when varying the number of features used as input for the single building and the four buildings cases, respectively. For each point, the reported results have been obtained with the best selection of f features as for Tab. 11.2. Since we

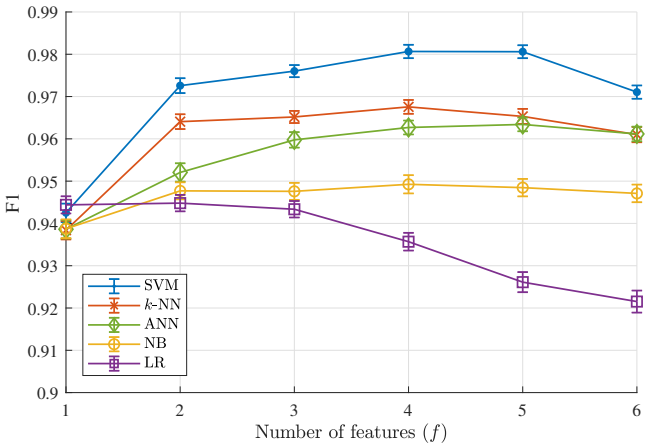


Figure 11.3: Average F1 score with 99% confidence intervals for the different classifiers when varying the number of features in the single building scenario.

set $k = 20$, the number of iterations of each n-fold cross-validation routine is set to $m = 480$.

For all techniques, increasing the number of features improves the accuracy of the predictors up to a certain point, after which the performance slightly decreases. A likely explanation is that the inclusion of any additional feature in the input vector brings diminishing diversity, due to the mutual correlation among the features, while increasing the noise. SVM, k-NN, and ANN provide the best classification accuracy, both for the single building and the four buildings scenarios. In the first scenario, all these techniques reach an excellent accuracy using only $f = 4$ features. When considering four buildings, they reach the best F1 score using 5 features, one more than in the single building case. In all cases, SVM offers marginally superior performance than k-NN, followed by ANN.

In Fig. 11.5 and Fig. 11.6 the false alarm and missed detection probabilities are depicted for the single and four buildings cases, respectively. Almost all classifiers have a false alarm probability higher than the missed detection probability, which is a desirable property in the positioning algorithms since the error introduced by a NLOS signal misinterpreted as LOS is larger than that caused by the exclusion of a LOS signal because misclassified as NLOS. When the number of signals is small, however, a high false alarm probability may lead to the exclusion of most of the valid measurements for the positioning algorithm. In this case, k-NN has an edge over the competing classifiers, since its false alarm probability is lower than its missed detection probability when using at least four features in the single building case. Conversely, ANN has a large false alarm probability, particularly in the four buildings case, thus it should be used only when there is an abundance of received signals.

In Tab. 11.2 the set of features resulting in the best F1 score for the different algorithms are reported. Note that, when considering only one feature, the kurtosis (κ) is the most informative in all the considered cases. Furthermore, the kurtosis feature is also included in the best feature sets for almost all the algorithms, both for the single building and four buildings scenarios. Therefore,

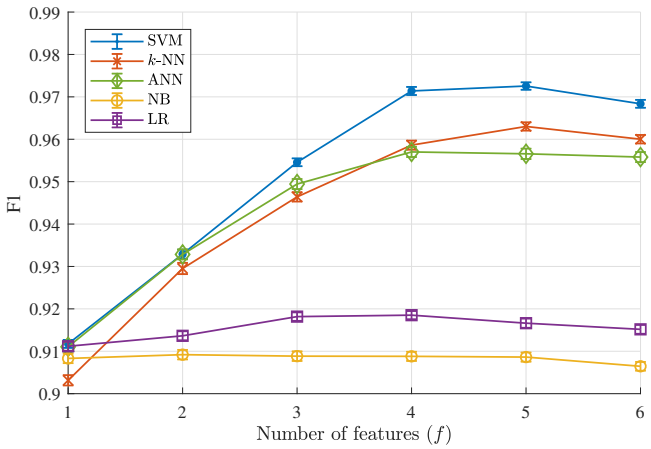


Figure 11.4: Average F1 score with 99% confidence intervals for the different classifiers when varying the number of features in the four buildings scenario.

we can conclude that it is one of the most valuable features for the NLOS identification, in general. Other valid features are the mean excess delay (τ_{MED}) and the mean squared delay (τ_{MS}), which appear in a number of optimal feature sets.

11.6.3 Exploitation of multiple angular directions

As noticed from Fig. 11.2, increasing the input vector of the classifiers by considering multiple signals from different directions can improve the NLOS detection performance. On the other hand, a longer input vector increases the complexity and duration of the training phase, and may also introduce more noise in the classification. In this section a different approach to exploit the availability of multiple signals received from different angles is proposed. More specifically, we can pick a number N of directions with maximum angular distance and classify each of the N received signals independently. In this way, the classifier is fed with the best-set features extracted from each single signal, providing its LOS/NLOSclassification for that signal. Then, we decide the link status by a simple majority voting between the N classifications. In the experiments, an odd value for N is chosen to avoid ties in the majority voting.

In Fig. 11.7, the F1 score for the three best performing classifiers is reported. We observe that this technique is able to notably improve the performance of the classifiers.

To confirm the significance of this improvement, the Wilcoxon signed-rank paired test [272] is used, which determines if two samples have been selected from populations with the same distribution. When considering the samples obtained when using only one direction ($N = 1$) with those obtained for multiple directions ($N > 1$) for any of the three classifiers used in Fig. 11.7, the test returns p-values smaller than $3 \cdot 10^{-5}$, which reject the null hypothesis that the samples belong to the same population. In other words, such a low p-value confirms that the performance gaps observable in Fig. 11.7 when increasing N are significant and not due to stochastic variations.

Furthermore, as also noticed in Fig. 11.2, the use of more than five directions

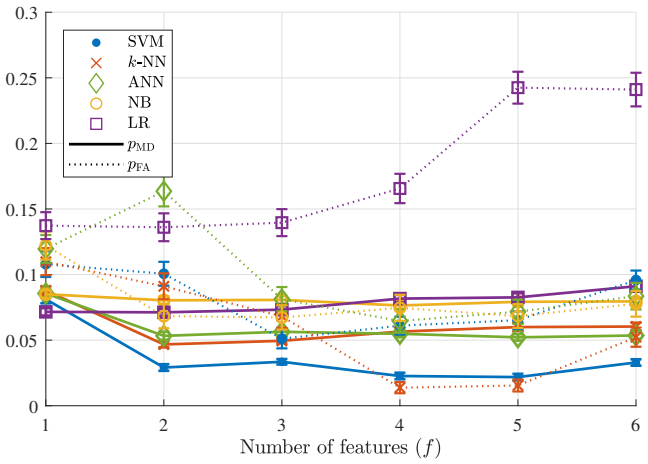


Figure 11.5: Probability of false alarm (dotted lines) and missed detection (solid lines) with 99% confidence intervals for the different classifiers when varying the number of features in the single buildings scenario.

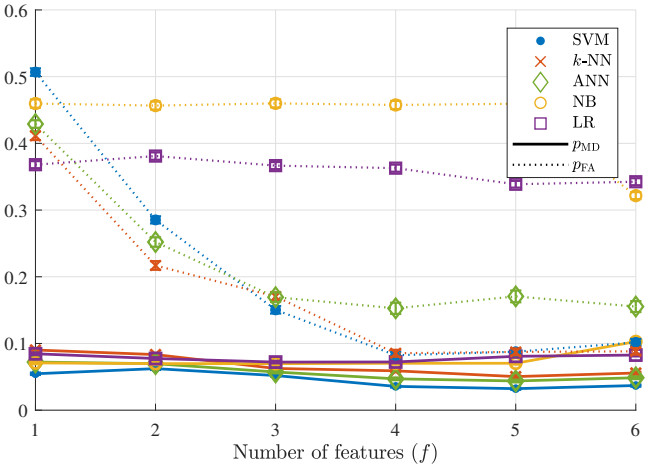


Figure 11.6: Probability of false alarm (dotted lines) and missed detection (solid lines) with 99% confidence intervals for the different classifiers when varying the number of features in the four buildings scenario.

does not bring significant performance improvements because of the correlation between different directions.

11.7 Conclusions

In this study, the LOS/NLOS classification problem has been addressed in a UWB positioning system. The transmission of UWB signals has been considered in four different indoor environments and six features from the received waveforms have been extracted. These features have been used as inputs to five different machine learning algorithms to compare their performance.

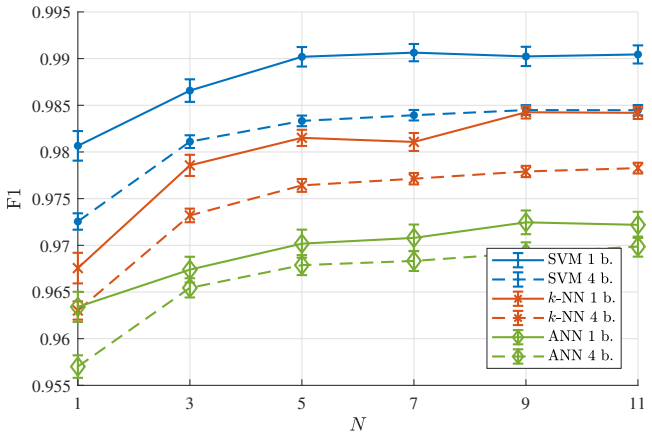


Figure 11.7: Average F1 score with 99% confidence intervals by applying a majority vote among N directions, in the single building (*1 b.*, solid lines) and four buildings (*4 b.*, dashed lines) scenarios.

A preliminary analysis of the data showed that, when using directional antennas to receive signals, the use of five angular directions is already sufficient to get close to the best performance, while with twenty or more directions the gain in terms of classification accuracy is negligible, while the time and complexity to train the networks increase considerably.

When comparing the F1 score obtained for different subsets of the signal features, results show that SVM and k-NN have high accuracy on NLOS detection, while NB and LR have poor classification accuracy. Among the two best performing techniques, SVM performs slightly better than k-NN in all scenarios.

We also observe that, for all the considered cases, the best performance is achieved when selecting a subset of four or five features, rather than using all of them, likely because of the mutual correlation among the features. The most valuable feature is the kurtosis, followed by mean excess delay and MS-DS. Based on these results, we can conclude that a good solution to perform LOS/NLOS classification of UWB signals is to apply SVM on five signal features, namely maximum amplitude, mean excess delay, MS-DS, kurtosis, and possibly rise time.

Finally, an alternative approach has been proposed, which classifies multiple signals received from different angles independently and makes a final classification decision with a majority voting. Significant performance improvement can be obtained by considering five signals simultaneously received at maximum angular distance. By classifying them independently as LOS or NLOS, and then performing a majority voting among the results, we can achieve an F1 score of 0.99 and 0.983 for the single and four buildings cases, respectively, which represent excellent results.

Part III

IoT Security

Chapter 12

Introduction to IoT security

IoT networks are designed to be pervasive and to be applied in he most diverse settings. This pervasivity increases the attack surface in IoT applications and makes them both more vulnerable and more attractive for hackers. Healthcare, logistics, factory automation, automotive, just to name a few, are all application area of IoT that have stringent requirements in terms of security.

In 2014, the Federal Trade Commission investigated on vulnerabilities in TRENDNet security camera, which allowed attackers to access to the video and audio streams of the camera [273, 274]. At the Blackhat conference, in 2015, security researchers explained how to hijack Jeep vehicles via their cellular data connection [275]. This connections is normally used to transmit monitoring data with the car manufacturer, but the researchers were able to obtain full control of the vehicle through this connection, including acceleration and brakes control. In October 2016, the botnet created by a malware called *Mirai* attacked the servers of the service provider Dyn, which led to a large portion of the Internet to be unreachable [276–278]. The malware infected weakly protected IoT devices and started making requests to servers operated by Dyn, causing an overload in a so-called Distributed Denial of Service (DDoS) manner. In 2017, researchers found that implantable cardiac devices produced by St. Jude Medical can be accessed by attackers, who were able to deplete the battery or administer incorrect pacing or shocks [279, 280].

These are just few examples of how IoT security can affect our lives. In the following, the features characterizing a secure system will be described and we will understand why implementing those on IoT systems is difficult. Then, some common IoT protocols will be investigated focusing on security aspects, highlighting their vulnerabilities.

12.1 Classification of security goals and features

Security is composed by a set of *requirements* that the system designer wants to satisfy. Those requirements may be violated by a wide range of *threats*, against which some security *services* are put in place. In the following we explore in more detail the security requirements, threats, and services.

12.1.1 Security requirements

Security requirements are tailored to the system being considered. Some systems may need just a few of the following requirements, depending on who is going to use them and the type of information they store and transmit. For example, a public announcement service does not need confidentiality, since the processed information is public anyway, but integrity and availability are of the greatest importance.

Confidentiality Confidentiality means that the message or information is available only to the intended receiver. This does not deal with the integrity of the message, which is explained next.

Integrity To guarantee integrity, the information must not corrupted, and must be received exactly as transmitted.

Availability This requirements is about guaranteeing that the provided service is always available [281], contrary to what happened with, e.g., Dyn services attacked by Mirai, mentioned previously.

Accountability Accountability guarantees that it is possible to discover the originator of every action or event. This is so to assure that the originator is authorized to perform the considered action and that the responsible for issues in the system can be identified.

Privacy Privacy requires that no private[1] information is made available to non-authorized parties. Also, the access to authorized parties must be limited to the information needed to perform their duties, e.g., a medic is authorized to access to health information of their patients, but not to financial information, or to health information of people who are not their patients [282]. Privacy also deals with *side information*, e.g., the purchase of a particular set of products may indicate a pregnancy condition [281, 283, 284].

Exclusivity This guarantees that only authorized applications can run on a device and transmit data over the network [285]. This requirement has become particularly relevant with the growth of IoT, since, contrary to what happens with traditional PC systems, IoT devices infected by malware may continue to work without issues for long time, before the malware activates simultaneously on all devices and attack a given target, like in the Mirai botnet case [276–278].

12.1.2 Security threats

Threats make a system to not satisfy one or more security requirements. Different threats are often combined together to perform an attack on the system.

[1] The definition of what is "private" is evolving over time and is defined by national and international regulations.

Eavesdropping This threat attempts to undermine the confidentiality and privacy requirements by obtaining confidential information [282,285]. This may be done, e.g., by listening to the messages on the network or even capturing electromagnetic emissions from wired keyboard [284,286].

Message replay Using message replays, an attacker can make the target system to perform again a certain action [285]. Note that the message is not modified, but just sent again some time later its previous transmission. This may be used, for example, for opening radio controlled gates or car locks by simply capturing a legitimate request and retransmitting it later. This breaks the accountability and exclusivity requirements.

Modification Unlike the message replay, modification is about modifying a message in transit. The modified message, therefore, may contain different requests or information than the original one [285]. This threat violates the integrity, accountability, and exclusivity requirements.

Forgery / fabrication In this threat, the attacker creates a message pretending the sender was another entity. Differently from modification, this threat is present even when no legitimate entity is using the network. Forgery violates the accountability and exclusivity requirements.

Masquerading In the masquerading threat, the attacker is able to impersonate someone or something else. This threat is closely related to forgery, but it is more general, dealing not only with messages on the network, but also with actions executed in a single machine.

Repudiation This threat concerns the ability of the sender to deny to have sent or received certain messages. This breaks the accountability requirement, since it makes impossible to deem the users responsible for their actions. Forgery, masquerading, and repudiation are often considered as a whole when designing a secure system, since, e.g., the ability to perform a forgery attack allows users to repudiate the sending of messages declaring they were forged by an attacker.

Profiling This treats undermines the privacy requirement, by collecting information on a single user based on many messages and different information sources. An example is the collection of statistical data by retail chains with the objective of tailoring promotions to specific people [283], a practice also used by online advertiser companies.

Fingerprinting Similar to the previous threat, this one still breaks the privacy requirement, but by gathering information on a user or device based on a single message or a small set of messages. Often it is used as the first step on attacks, in order to understand what type of devices the attacker is dealing with.

Denial of Service In this threat to availability, the service is made unavailable to legitimate entities, often caused by the attacker overusing the network, storage, or computational resources of the service [282]. In the *distributed* version of the attack, the resource overload is caused by a multitude of devices performing the same attack in a coordinated manner. In the IoT scenario, this attack is particularly critical, both performed against the IoT system itself and performed by IoT devices towards data centers. In the first case, the IoT system is usually resource constrained, so it is easy to completely exhaust the network resources or deplete batteries in battery operated devices. In the latter case, while IoT devices are not able to perform a traditional denial of service attack due to their limited resources, the coordination of a massive amount of resource constrained devices is able to easily overload powerful systems and networks, as in the Mirai botnet case [276–278].

Physical damage This threat is about breaking or destroying the devices. This may cause partial malfunction or full denial of service, impairing the availability requirement [282]. This threat is more serious than the previous one because device has also to be repaired or substituted after the attack has ended, but it is more difficult to execute, since it requires physical access to the target device and can not be automated.

Node capture This is again a threat where physical access to the target device is required. Leveraging the physical access the attacker obtains the information stored on the device, by using either destructive or non destructive methods. Capturing the devices can not only cause impairments to availability, as a consequence of the device destruction, but also on confidentiality and privacy [282].

Node controlling This threat includes both the installation of malware on the device or the use of unprotected programming interfaces to control the device [282]. After the attacker has gained control of the target, the latter can be used to gain control of other devices or to perform a coordinated attack. This threat is often linked with the creation of botnets for distributed denial of service attacks. Since the device is completely under the control of the attacker, this is the worst threat and damages all the listed security requirements.

12.1.3 Security services

Security services are ways to protect the system from the security threats in order to satisfy the security requirements.

Secrecy Secrecy deals with ways to hide information, which are readable only by authorized entities. For example, one of the secrecy techniques is cryptography, where authorized entities possess keys able to decipher the message. Secrecy techniques protects against eavesdropping, profiling, fingerprinting, and node capture.[2]

[2]Note that the term "node capture" specifically refers to the ability to gather information from devices which the attacker has physical access to, not to the ability to actually capture the node.

Integrity protection This service, which protects against modification, guarantees that the message or the stored data is not modifiable by an attacker without a clear indication of such modification [285]. Hash functions and signatures are common techniques in this family.

Authentication Authentication refers to the ability to identify the entity that is operating on the system [281,282]. It is able to protect from the forgery, repudiation, masquerading, node controlling, and message replay threats.

Authorization and Access control While authentication deals with securely identifying an entity, authorization and access control concerns with assigning authorization to users and access permissions to resources [282, 285]. Therefore, it is analogous to a map between users and resources. This service stops denial of service and node controlling threats.

Accounting / Notarization This service, often referred together with the previous two as AAA (Authentication, Authorization, and Accounting), records the activity on the system so that it is possible to trace back to the originating entity of every action. It protects from repudiation.

Anonymization This service takes care of removing, in the exchanged messages, information which can be used to identify or profile the originating entity or other entities [282]. While this works against the profiling and fingerprinting threat, it may collide with the authentication and accounting services. A balance between those two seemingly contrasting needs can be found by making identifiable data readable only to the AAA services, with the help of the secrecy service.

Security fault tolerance Fault tolerance in terms of security assumes the form of a group of entities that monitor the system and signal, and potentially react to, active security threats [282].

Network fault tolerance This service, instead, deals with network status, in order to assign network resources and prioritize traffic [281]. Other than providing protection against denial of service, it can help the security fault tolerance service to assess if an attack is running, by detecting, e.g., unusual network traffic to a group of devices [282].

12.2 Tailoring security solutions to IoT scenarios

The characteristics of IoT systems make the implementation of security measures difficult. First of all, a large part of IoT is composed by embedded and resource constrained devices. Limitations on memory, computational power, and low-power communication all contribute to put a limit on the level of auxiliary information that can be exchanged to implement security functions, particularly those based on cryptography [284].

Diversity and heterogeneity also cause issues, because security features must be designed and implemented separately for each of the many types of devices that compose the system [281, 282, 287]. For example, smartphones are capable of executing complex cryptographic routines, while wearable devices may need to collaborate with more powerful devices in order to run such routines. Automatic updates are a partial solutions to this problem, and, in any case, they are dependent on timely update distribution by device manufacturer, which is not always guaranteed, sometimes requiring intervention from governmental entities [274]. Moreover, keeping such a heterogeneous set of devices up to date with security updates and updated credentials is very complex. Not having a centralized AAA service makes users keep the default authentication credentials in IoT devices to avoid managing a large amount of credentials. Even in the case where users customize access credentials, they may be not periodically changed, since it would be a very time consuming task. A centralized AAA system would provide easier credentials and authorization management [281], but would require a homogeneous system.

Different devices also have different security requirements, so not every part of the network is covered by the same set of security services [284]. However, vulnerabilities in non-critical devices, like a smart kettle, may be used to access to a critical network that device is connected to, like a home or corporate Wi-Fi network [285, 287, 288]. Additionally, diversity increases the attack surface: an attacker has a greater possibility to find a vulnerability in at least one of the devices part of the same system. Isolation between critical and non-critical part of the system is therefore advisable, but difficult, particularly for pervasive IoT systems.

> *As a designer, you have to find all the flaws in the system, while the attacker only one. You will fail, so you have to plan for that failure.*
>
> — M. Muller, CTO, ARM [289]

Furthermore, the large amount of data that this massive amount of devices exchange makes it difficult to protect against privacy issues [281, 282]. In particular, these systems are prone to provide unintended side information, e.g., the presence of an active baby monitor signal outside a house discloses when a baby is present. Also, cryptanalysis attacks against cryptographic algorithms, particularly when using a short key length because of the devices limitations, are facilitated when the attacker has a larger set of previous messages to analyse [284].

A closely related security issue concerns *trust*: consider an IoT system that manipulates data of user A using devices manufactured by company B that uses a third party provider C for data communication. This data is then analysed in the cloud owned by company D to provide advanced services to the user. In this scenario there must be a strong trust relationship between all parties, that often do not know each other. In this scenario, for example, trust is related to the important topics of data integrity, legitimate use of the service, and privacy [282]. If a trust relationship can not be created, complex and multi-layered security measures must be put in place, exponentially increasing the complexity of an already intricate system.

IoT devices are often installed in public places but they are still connected to private networks (e.g., a doorbell connected to the home wireless network). This increase the importance of threats that require physical access to the device

(physical damage, node capture, and node controlling) compared to non-IoT networks, since, in the latter case, the devices composing the network are usually kept in private area.

> *I'm less worried about the data being stolen, I'm actually worried that the device is stolen, because if you lose control of an IoT device, it's game over. Loss of the data is easier to protect than loss of the device.*
>
> — M. Muller, CTO, ARM [289]

To conclude, IoT raise a number of issues never seen before with traditional systems. Implementation of security features is challenging because of resource constraints, heterogeneity, and large amount of data. In addition, in IoT a security issue may become a safety issue, so care must be taken as to design the system to fail in a safe way [287, 289].

12.3 Security in IoT communication protocols

Due to the peculiarity of IoT networks and the limited amount of resources they have, a number of protocols have been specifically designed for IoT use cases. In this section, we analyse three of the most widely used protocols in this category, focusing on their security features.

12.3.1 ZigBee

Protocol description

ZigBee [290] is a two-way, wireless communication standard developed by the ZigBee Alliance. It provides the application and network layers, while the link and physical layers used are from IEEE 802.15.4 [291]. Thanks to its low cost and low power consumption, ZigBee is one of the most used technology to connect IoT devices. The ZigBee stack is composed by the following layers:

- ***Application Layer* (APL).** The application layer is divided into two more specialized sub-layers:
  - *Application Support Sub-Layer* (APS) provides the data transmission service, security services, and allows the binding of devices between two or more application entities located on the same network.
  - *ZigBee Device Objects* (ZDO) is responsible for the initialization of the APS, network layer, and security provider.
- ***Network Layer* (NWK).** Some of the functionalities provided by this layer are routing, security, and configuration of new devices. This layer also manages the establishment of new connections, the joining and leaving procedures, and the addressing and neighbor discovery services.

ZigBee includes different *application profiles*. These establish agreements for messages, message formats, and processing actions that developers have to respect in order to create interoperable and heterogeneous applications. In such a way, devices from various vendors are able to seamlessly communicate in a ZigBee network [290].

For what concerns security, ZigBee allows for message encryption and authentication using the *Advanced Encryption Standard* (AES) in the *counter with cipher block chaining message authentication code* (CCM) mode. This provides the secrecy and authentication security services described above. Integrity protection is provided by a 128 bit message integrity code (MIC) and replay protection is based on a 4 Byte frame counter.

A ZigBee network also includes a *Trust Center*, a device trusted by all the other nodes in the network and that usually corresponds to the network coordinator. The Trust Center is responsible to authenticate devices that request to join the network, decide whether to accept or deny the join request, maintain and distribute network keys, and enable end-to-end security between devices.

The cryptographic routines used in ZigBee employ two different types of key: *link key* and *network key*, each of them 128 bit long. The first type of key is used to secure unicast communications between APL entities and is known only by the pair of entities that use it, while the second key type is used for broadcast communications and is shared amongst all devices in the same network. Keys can be acquired in three ways [292]:

- *Pre-installation*: they key is installed in the device during the manufacturing process.

- *Key transport*: the key is generated elsewhere (usually by the Trust Center) and then communicated to the device. The standard suggests to load the key using an out-of-band technique, however it includes the possibility to send the key in-band. In the latter case, the key may be sent in clear text or encrypted using a pre-shared key specific for each application profile and known to every device. For example, for Home Automation devices, the pre-shared key is defined in the ZigBee standard and is publicly available. For ZigBee Light Link (ZLL) devices, instead, the pre-shared key "will be distributed only to certified manufacturers and is bound with a safekeeping contract", as the ZLL specification affirms [293]. However, it has been leaked on the Internet in 2015, so it is now publicly known. [292, 294]

- *Key establishment*: this technique is only used for link keys. Key establishment allows to exchange a link key L_i between the Trust Center and another device of the network for securing communications between these two entities. The procedure is started by the exchange of a trusted information, the *master key*, pre-installed during the manufacturing process. The master key, different for each application profile, is provided by the ZigBee Alliance to its members. After this phase, the devices exchange ephemeral data that are used to derive L_i. When two devices i and j need to communicate with each other, the Trust Center provides them with a link key $L_{i,j}$, protecting it using the link keys L_i and L_j, respectively.

The process through which a new ZigBee network is set up or a new ZigBee device is added to an existing network is called *commissioning*. Commissioning procedures are defined by the different application profiles. In addition to that, there is also a common procedure specified in the ZigBee standard that allows the connection between devices of different application profiles.

Attack surface

A possible attack vector is trying to discover the keys used to secure the network. In particular, ZigBee has been shown to be vulnerable to plaintext attacks [295]. This technique enables the recovery of a cryptographic key by having access to both the encrypted and decrypted messages. For example, the repeated encryption of publicly known (e.g., because defined in the specification) and fixed messages, makes the system vulnerable to plaintext attacks. Therefore, to ensure a high security level, the network key needs to be changed periodically.

Other attack vectors are specific to ZLL installations. In 2012, LIFX and Philips presented their first smart lights solutions and, afterwards, many other companies developed similar connected light systems. Many vendors, such as Philips, use the ZLL application profile. A number of security investigations on smart light systems have disclosed that designers and manufacturers tend to implement only the essential security measurements that are necessary to obtain ZigBee Alliance's certification [294]. At a first analysis, it may seems unnecessary to implement many security precautions in a light system, since they do not elaborate confidential information and can still be operated manually in case the network does not work. However, as explained above, we have to remember that attackers may use these devices to relay an attack to the rest of the home or corporate network, bringing more critical devices at risk.

In [294], the authors investigate the state of the art security in three different ZigBee smart light systems: *Osram Lightify*, *GE Link* and *Philips Hue*. Vulnerabilities of both bulbs and interconnected devices are evaluated, and, as a consequence, seven different types of attack are reported. The attacks are based on the *inter-PAN frames*, the frames used to transmit touchlink commissioning commands such as scan request, scan response, and so on. These frames are neither secured nor authenticated: an attacker can send the same commands pretending to belong to the network.

- *Active device scan.* The scan searches for ZLL devices in the range of the attacker, sending scan requests in different channels. Listening on the correpsonding scan responses, the attacker can obtain a complete overview of the devices connected to the network. The behavior of the three systems analysed is different one to another. All light bulbs and controller from Lightify respond to the attacker's scan request, the Link controller does not respond, and, finally, the Hue controller responds only if its Touchlink commissioning button has been pushed within the last 30 seconds.

- *Blink attack.* This attack can be activated after a device scan, by sending to the attacked device the inter-PAN command *identify request*. The device, then, starts to blink for a default period. Such a command is implemented to allow the owner of the devices to understand which device has a certain address. All the three light bulb types are vulnerable to this attack.

- *Reset attack.* The attacker resets all setting of the ZLL device to the factory state, by sending the inter-PAN command *reset to factory new request* after having completed the device scan. All devices are vulnerable to this attack.

- *DoS attack* and *hijack attack*. In these attacks the end user loses control on the target device. Two approaches are possible in order to make a DoS attack. The first is to force a device to change the transmission channel, sending a *network update request* inter-PAN command including the new channel. The second is to make the device join a non existing network, changing its network key with arbitrary bytes. Hijack attack is similar to this second attack, with the difference that it forces the device to join a new network chosen by the attacker. In this case, the network key of the desired network is used. These two attacks are executed by sending the inter-PAN command *network join end device request*: at the reception of the command, the device leaves its current network, changing its parameters according to the new configuration. All the evaluated smart light systems are vulnerable to DoS and hijack attacks but all of them integrate functions to regain control over attacked devices.

- *Network key extraction attack*. This attack allows the attacker to extract the current network key, by eavesdropping the messages exchanged by the devices during the touchlink commissioning procedure. A preliminary DoS attack is needed to disconnect a device from the network. After that, the attacked device will start a commissioning procedure in order to regain access to the network. Therefore, the attacker can extract the network key from the *network join end device request*. As previously seen, the network key is encrypted using the well known master key. Only Philips Hue devices can be attacked in this way because the touchlink commissioning procedure is not enabled in the other devices.

- *Inject commands attack*. This attack makes it possible to send commands to the devices in order to control their actions. The knowledge of the current network key is needed, via, e.g., the execution of the previous attack. All the three smart light systems are vulnerable to this attack.

12.3.2 Bluetooth Low Energy

Protocol description

Bluetooth is a widely used short range connectivity protocol [62]. Its low energy and IoT-tailored version, initially named Bluetooth Low Energy (BLE), has been first introduced in the Bluetooth Core Spec version 4.0. BLE is a wireless protocol operating in the unlicensed 2.4 GHz ISM band and uses 40 channels with 2 MHz spacing. Its physical layer uses a Gaussian Frequency Shift Keying modulation with index around 0.5: this scheme allows the use of fewer advertising channel and enables lower power consumption. The physical layer data rate is 1 Mbps. The coverage range is typically over various tens of meters [296]. The BLE MAC Layer is split into two parts, *advertising* and *data communication* [297]: 37 of the available channels are used during the transmission of data and the remaining 3 are used by unconnected devices to broadcast device information and establish connections.

As ZigBee, BLE uses AES-CCM with 128 bit keys for encryption and authentication purposes. The agreement on a symmetric key is part of the pairing procedure, which is executed as follows:

1. Devices exchange their capabilities and authentication requirements. This phase is completely unencrypted.

2. Devices generate or exchange a Temporary Key (TK) using one of the available pairing methods, then exchange some values to confirm that the TK is the same for both devices. After that, a Short Term Key (STK) is generated from the TK and will be used to encrypt the data exchange.

3. Optionally, devices exchange transport specific keys if bonding requirements are present.

The available pairing methods are three:

- **Just Works:** in this case, the TK is 0. Of course, this does not provide any level of security.

- **Out of Band:** the TK is exchanged out-of-band, e.g., using near field communication. This method provides a security level that is as high as that of the out-of-band method used to exchange the key. However, it can be inconvenient for the user.

- **Passkey:** the TK is a six digit number that the user passes between the devices. For example, one of the devices may generate the number, show it in a display, and make the user enter it into the other device. The security level is still high, but requires devices to have a way to make the user read and input the TK, which may be impossible for miniaturized IoT devices.

Starting from the 4.2 version of BLE, a new pairing procedure has been put in place, using elliptic curve cryptography:

1. Each device generates an Elliptic Curve Diffie Hellman (ECDH) public-private key pair. Then, they exchange the public key with each other and derive a key, called *DHKey*, from their own secret key and the public key of the other device, using elliptic curve functions.

2. The devices use one of the available pairing methods to confirm that *DHKey* is the same for both of them and to generate a Long Term Key (LTK) that will be used to symmetrically encrypt the data exchange.

3. Optionally, devices perform a final step, equal to the one for 4.1 BLE devices.

Also the pairing methods have been changed, with the introduction of a new option and the hardening of the methods in the previous version.

- **Just Works:** the non-initiating device generates a nonce and a confirmation value C_b, function of the nonce and the public keys. C_b and the nonce are then sent to the initiating device. The latter generates its own nonce and sends it to the non-initiating device. It also uses the non-initiating device's nonce and the public keys to generate its own confirmation value C_a, which should match C_b.

- **Numeric Comparison:** this method is equal to Just Works, but also generates a value which is function of the public keys and the nonces. This value must be displayed to the user, which must manually confirm that the shown number is the same in both devices.

- **Out of Band:** with this method, random numbers and commitment values, which are function of the random numbers and public keys, are exchanged in an out-of-band fashion, e.g., using near field communication.

- **Passkey:** in this method, the user first inputs a k bit long secret passkey to both devices (or reads it from one of the devices and inputs it to the other). Then, for each bit $i = 1, \ldots, k$ of the passkey, the device pair must perform a two-step procedure: (i) Each device generates a nonce and computes a commitment value, which is function of the nonce, the passkey, and the public keys. Commitments and nonces are then exchanged between devices. (ii) After that, each device recalculates the commitments as before, but exchanging the order of the two public keys, and using the nonce of the other device. If the passkey is the same, the commitment value must be equal to the one found before.

Attack surface

The pairing methods just described have some important security issues. In BLE 4.0 and 4.1, there is no resistance to eavesdropping or man-in-the-middle attacks during the pairing phase, except for Out of Band pairing). In fact, in the Just Works pairing method the key is known, while in the passkey method the key is easily brute-forced (and in some cases brute-force is not even required) [298–302]. BLE 4.2 is affected by similar problems too, in particular for the passkey pairing, since the passkey is verified one bit at a time [302, 303]. When the attacker is interested in eavesdropping, the attacker can try to match the confirmation value considering the current bit r_i of the key equal to 0. If the confirmation value does not match, $r_i = 1$. When trying to directly connect to a device, instead, the attacker can consider $r_i = 0$. If the other device aborts the procedure, then $r_i = 1$. This procedure can be repeated for bit $i = 1, \ldots, k$, learning, therefore, the entire key.

Another issue is linked to the advertise mode of BLE devices. In [297], authors found that the analysed fitness trackers are almost always in advertise mode. This is because the master device frequently disconnects from the tracker in order to preserve energy. Therefore, when the smartphone app for the fitness tracker is not running, the tracker closes its communication link, remaining in advertise mode until the next connection establishment. Also, they found that most of the analysed devices expose always the same MAC address. This makes possible to capture exchanged messages and correlate over a long period of time the BLE traffic between a pair of devices. As an example, an attacker may be able to track the movements of the BLE device owner or even just verify its presence in an area. As a security feature, the BLE specification allows a device to use random MAC addresses and to frequently change them. For example, the *Apple Watch* randomizes the MAC address both when it is rebooted and during normal usage at an approximately 10 minutes interval [304].

12.3.3 6LoWPAN and CoAP

Protocol description

To make IoT devices support IP networks and traditional upper layer protocols, like HTTP, two protocols, published by the IETF, are usually implemented: 6LoWPAN and CoAP. The first is an IPv6 adaptation protocol that defines mechanisms to make IP connectivity viable for tightly resource constrained devices that communicate over low power and lossy links such as IEEE 802.15.4 [305, 306], leveraging compression and fragmentation mechanisms. CoAP is a RESTful protocol at the application layer, transported over UDP. It has been designed to be easy to map to HTTP via proxies, to support retransmissions, sleepy devices, and resource discovery. On the downsides, the usage of UDP instead of TCP does not allow message reordering and retransmission of lost packets. Often the physical and MAC layers employed in networks using these protocols are those from IEEE 802.15.4.

6LoWPAN routing is based on the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) defined in RFC 6550 [307]; it has been mainly designed for multi-point to point communications, such as those in WSNs. However, it also supports point to multi-point (sink broadcast) and point-to-point (leaf nodes communicating with each other). RPL builds a Direct Acyclic Graph (DAG) based on a root node called Low power and lossy Border Router (LBR), usually being the device responsible for the management of a group of nodes and representing the border between two networks. From the DAG, RPL creates a Destination Oriented Direct Acyclic Graph (DODAG) tree, which contains only one root and excludes any network loop. Starting from the DODAG root, devices broadcasts their DODAG Information Objective (DIO) message, which contains device and link metrics. The Global repair and Local repair mechanisms are used in case of a broken link: the first recalculates the whole topology, while the second operates locally, by informing all the children of a node that they need to update their parent.

Attack surface

We are now going to describe some attacks against 6LoWPAN devices. A hypothetical attacker can act on the RPL, at the application layer, or at the adaptation layer, based on the type of control of the network that it wants to achieve.

Attacks against the RPL Many of the attacks on 6LoWPAN focus on redirecting traffic and disrupting the routing tree. In the following we report some of the attacks against the RPL [308–310].

- *Clone ID and sybil attacks.* In the clone ID attack, the malicious node clones the identity of another node. Similarly, in the sybil attack, the attacker uses the identity of several entities at the same time. In this way, the attacker can redirect and access a large amount of network traffic. These types of attack can be detected by keeping track of the number of instances of each identity or monitoring the geographical location of the devices.

- *Sinkhole attack.* The malicious node attracts to it a lot of traffic, by declaring very efficient routing paths. Following this, the attacker may modify or drop the packets that pass through it.

- *Selective forwarding and black hole attacks.* These attacks take place when a malicious node of the network, that is supposed to forward the packets along the correct routing path, discards some of the traffic (selective forwarding) or all of it (black hole) that passes through it. Possible solutions may be the creation of disjoint or dynamic paths inside the DAG.

- *Hello flooding attack.* The *Hello* message is used by a node in a 6LoWPAN network to announce its presence. If a node receives a Hello message, it deduces that the sender node is in its neighbor, so a direct link between them is available. An attacker can exploit this mechanism by broadcasting a Hello message using a larger than permitted transmission power. In this way, a large number of nodes consider the attacker a neighbor. However, when a node tries to use the new link, the sent packets will be lost, since the permitted power level is used. This type of attack can be avoided using link layer acknowledgments to check the message reception.

Contrary to the previous ones, the following attacks are based on the RPL service messages [308–310].

- *Local repair attack.* In this attack, a node without any connectivity problem continuously sends local repair messages. This forces an update of the network topology, which is costly both in terms of computational resources and in energy, causing service degradation and early energy depletion for battery operated devices.

- *Version number attack.* The *version number* is a field of DIO messages incremented every time that a rebuilding of the DODAG has to be done. If an attacker forwards DIO messages where the version number has been forcefully increased, the whole DODAG is going to be unnecessarily rebuilt. Again, this causes service degradation and energy depletion.

Attacks from the Internet side Neither 6LoWPAN nor CoAP provide secrecy, authentication, or integrity protection. The use of 6LoWPAN and CoAP without additional security measures, therefore, makes the devices fully accessible from the Internet. A proposal has been made to add extensions to CoAP in order to provide built-in security, but the proposal did not become an actual standard [311]. The CoAP specification, instead, suggests using Datagram TLS (DTLS) to provide secrecy, authentication, and integrity protection [312, 313]. Alternatively, IPsec can be used to provide authentication and encryption at the IP level.

Attacks at the Adaptation Layer The translation of the packets between Internet and the 6LoWPAN network is implemented at the border router. The lack of authentication and the computational resources of the device that perform the adaptation make this mechanism vulnerable. Two attacks that can be achieved at this level, *fragment duplication* and *buffer reservation*, are presented in [314]. The fragment duplication attack is based on the fact that a node cannot

verify at the 6LoWPAN layer if a received fragment belongs to the same IPv6 packet of the previous ones (in fact, this control is managed by higher layers). Therefore, a malicious node inside the network can inject fragments with the same header of the legitimate 6LoWPAN packets. The target node cannot decide which fragments to use during packet reassembly procedure, since it cannot distinguish between legitimate and spoofed IPv6 fragments. This causes corruption in IPv6 packets, which are consequently dropped. The buffer reservation attack leverages the scarce memory of the network nodes. In the 6LoWPAN network, receiving nodes must reserve buffer space to reassembly the fragments that belong to the same IPv6 packet. When the reassembly buffer is assigned to one IPv6 packet, received fragments of other IPv6 packets are dropped out. Since, buffer space reservation is kept for 60 seconds, if an arbitrary fragment is transmitted by the attacker to the target node, its communication will be blocked for the following minute. Consecutive repetitions of this attack causes a long term DoS to the targeted device, while employing just a small amount of the attacking node resources.

In order to protect 6LoWPAN networks from the attackers, Intrusion Detection Systems (IDSs) specifically tailored to IoT networks have been studied [308, 309, 315]. An IDS monitors network parameters and is able to identify signs of intrusions or attacks. IDSs for 6LoWPAN networks are optimized in order to save the largest amount of network resources. Due to the vast attack surface, IDSs should operate both at the adaptation, RPL, and application layers. Therefore, a hybrid architecture is needed, in which a centralized module installed on the border router cooperate with distributed modules installed on internal nodes.

12.4 Conclusions

We have seen that there are still many critical security issues in current IoT protocols, which are made even more dangerous by the pervasivity of these type of networks and their use in mission-critical systems. In the following chapter, a new authentication technique is proposed, which aims at solving some of the issues with current authentication strategies by employing a completely different approach, based on physical channel features.

Chapter 13

A Stochastic Geometry Analysis of Distributed Physical Layer Authentication in 5G Systems

13.1 Introduction

Message authentication is the security service which allows a device to verify that some received message is actually coming from the claimed source. It typically encompasses two steps: a) an identity acquisition step, by which some special feature operating as a fingerprint (e.g., a key or some characteristics of the transmitter) is acquired by the receiver and associated to the user, and b) an identity verification step by which, upon reception of a message, the special feature is checked on the received message in order to confirm the sender identity. Typically authentication is performed by cryptographic techniques, however, since the seminal works of the 1980's [316] authentication can also be performed by using features of the physical channel, either through the use of a key exchanged among the users or directly using the physical transmission properties. Typically, physical-layer authentication (PLA) techniques may find application as a complement to cryptographic mechanisms, or as the only solutions for devices with limited energy resources or communication capabilities, such as in the IoT and fifth-generation (5G) cellular systems [317]. Challenges and developments of PLA have been recently highlighted in [318].

Here we focus on key-less approaches to PLA (see [319,320] for recent literature surveys), where the special feature is some characteristic of the communication channel between the authentic transmitter and the receiver, and the verification procedure consists in checking if the channel over which the message was transmitted exhibits this feature. Due to the different transmission scenarios, and channel features (for example, time-variations of the channel), a variety of

practical solutions has been proposed in the literature, including PLA based on impulse responses in wideband channel transmissions [321], frequency responses of a multicarrier signal [322,323], multiple-input-multiple-output (MIMO) channel responses [324,325], and carrier frequency offsets [326]. In order to overcome the difficulties of a proper channel estimation due to either synchronization errors or time-variations of the channel, further refinements of these techniques are continuously studied as shown by recent results on the use of channel quantization [327] and authentication without phase detection [328]. An analysis of attacks and countermeasures in PLA systems has been considered in [325,329], in terms of false alarm and missed detection probabilities.

When applied to 5G cellular contexts, PLA techniques can be implemented in a distributed fashion over multiple BSs [330]. Leaving aside the simple scenario in which the receiver authenticates the transmitter without cooperation by any other nodes and considering a more complex architecture in which multiple nodes cooperate opens a number of issues. When many cooperating nodes with some performance limitation (e.g., in terms of energy or data rate) are available, a suitable selection of cooperating nodes should be considered to make the network efficient. In this respect, when energy consumption is concerned, nodes selection has been proposed in [331], while when communication overhead is a concern, compressed sensing techniques can be applied [332]. Indeed, authentication performance is significantly affected not only by the number of nodes, but also by the relative position between the legitimate cooperating nodes and the attacker. Therefore it is essential to have a better understanding of scenarios in which nodes are randomly placed, such as a cellular 5G networks. In other contexts outside authentication, analysis and design of networks has been performed using various approaches, among which one of the most promising is *stochastic geometry* [103, 104, 106] that studies random point patterns. In particular, when applied to wireless networks stochastic geometry allows the mathematical analysis of random channel access, single- and multi-tier cellular networks, and cognitive networks (see [333] for a survey). Typically, the distribution of nodes in the network considered by stochastic geometry follows a Poisson process model and semi-analytical results involving solutions of integrals can be obtained. Stochastic geometry has also been applied to the security context, for example security connectivity was studied in [332], while scaling laws for secure communications in large networks have been analyzed in [334]. However, no study of distributed PLA in large networks has been performed in the literature, while its investigation would provide insight into both performance and design criteria for the authentication network.

This work analyzes distributed PLA in a cellular 5G system using stochastic geometry tools. In particular, the authentication feature is the channel between the terminal and the BSs, modeled as a set of parallel additive white Gaussian noise (AWGN) channels, which corresponds for example to an orthogonal frequency division multiplexing (OFDM) or a MIMO system, as typically encountered in cellular systems. The decision process compares the observed channel in the initialization and verification phases and a maximum likelihood decision is taken. Two channel models are considered: in the first case the channel is characterized by path-loss and affected by an AWGN estimation error or log-normal shadowing; in the second case the channel is affected by path-loss and shadowing. For both cases, stochastic geometry tools allows us to obtain the statistics of the hypothesis testing problem, being able to derive the aver-

age missed detection and false alarm probabilities of this authentication system under the assumption that BSs are distributed according to a PPP on a plane.

The rest of the chapter is organized as follows. In Sec. 13.2 a physical layer authentication strategy is proposed and modeled. The following Sec. 13.3 will analyse in depth the two cases of (i) channel with path loss and noisy estimation, and (ii) channel with path loss and shadowing. In Sec. 13.4, the theoretical performance of the proposed strategies will be extracted leveraging some stochastic geometry tools. Numerical results will be presented in Sec. 13.5, while some final remarks will be given in Sec. 13.6.

13.2 System Model

Consider a legitimate node positioned at $\boldsymbol{u}_{\mathrm{L}} = (u_{\mathrm{L}}, v_{\mathrm{L}})$ and a random number N_{BS} of base stations located at $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$, with $\boldsymbol{x}_k = (x_k, y_k)$. Each time a message is sent in the uplink, all the N_{BS} base stations receive the message and estimate the corresponding channel response. An attacker positioned at $\boldsymbol{u}_{\mathrm{A}} = (u_{\mathrm{A}}, v_{\mathrm{A}})$ wants to forge messages by impersonating the legitimate node. The objective of the authentication system is to accept all the legitimate messages and to reject the forged messages. A smart attacker may also change the transmitted signal so that the estimated channel response estimated upon reception of the forged message is similar to that for messages from the legitimate node.

The channel between any transmitter-receiver pair is represented by multiple coefficients, in general (e.g., MIMO, OFDM), and their number is denoted by N_c. Also, let $h_{k,n}$ be the n-th coefficient of the channel between the legitimate device and the k-th base station.

At the network deployment (or at a proper time) an association procedure allows the system to learn the channels $h_{k,n}$, $k = 1, \ldots, N_{\mathrm{BS}}$, $n = 1, \ldots, N_c$. We suppose this association procedure is secure. When receiving a message, the system compares, the maximum likelihood (ML) estimated channel responses $\hat{h}_{k,n}$ for each of the base stations with the channel response previously learned in the association phase. Given a tunable threshold θ, the message is deemed authentic and accepted if

$$\sum_k \sum_n \left| \hat{h}_{k,n} - h_{k,n} \right|^2 < \Theta \,, \tag{13.1}$$

while it is marked as forged and discarded otherwise.

The channels in our scenario are affected by exponential path loss and log-normal shadowing. Thus, the channel gain between two nodes located in $\boldsymbol{y}$ and $\boldsymbol{z}$ are given by

$$g(\boldsymbol{y}, \boldsymbol{z}) = \frac{e^{\xi}}{\ell(\boldsymbol{y}, \boldsymbol{z}) + \varepsilon} \,, \tag{13.2}$$

where: e^{ξ} is the log-normal shadowing component, with $\xi \sim \mathcal{N}(0, \sigma_\xi^2)$; $\ell(\boldsymbol{y}, \boldsymbol{z}) = (\|\boldsymbol{y} - \boldsymbol{z}\|/d_0)^{\alpha/2}$ is the path loss, d_0 a normalization factor, and α the corresponding (power) loss exponent; $\varepsilon \ll 1$ is an arbitrary constant, introduced to avoid numerical integration issues.

Using this model, the n-th coefficient for the channel between the legitimate user and the k-th base station is $h_{k,n} = g(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x}_k)$, where $\boldsymbol{u}_{\mathrm{L}}$ and $\boldsymbol{x}_k$ are the location of the legitimate user and of the k-th base station, respectively.

The channel estimation is considered noisy, so that $\hat{h}_{k,n} = h_{k,n} + w_{k,n}$, with each $w_{k,n}$ distributed according to a normal distribution, i.e., $w_{k,n} \sim \mathcal{N}(0, \sigma^2)\ \forall\, k, n$.

The base stations are randomly distributed over a compact region $S \subset \mathbb{R}^2$ according to a spatially uniform PPP Φ. Let $\lambda = E[N_{\mathrm{BS}}]/|S|$ be the constant intensity of Φ in S, where $E[N_{\mathrm{BS}}]$ is the mean number of base stations in S. The intensity of Φ over $\mathbb{R}^2$ is therefore $\lambda'(\boldsymbol{x}) = \lambda\, \mathbf{1}_S(\boldsymbol{x})$, where $\mathbf{1}_S(\boldsymbol{x})$ is the indicator function of S. Also, the intensity measure of Φ is $\Lambda(A) = \lambda |A \cap S|$ for any $A \subset \mathbb{R}^2$.

We want to determine the false alarm and missed detection probabilities of the proposed physical layer authentication scheme, taking into account the noise in the estimation of channel coefficients and the inherent randomness in the shadowing component.

Expressions for false alarm and missed detection probabilities will be given as a function of the distance between the legitimate user and the attacker. If the attacker's position is known only statistically or through some bounds on its distance, the results must be correspondingly averaged, interpreted as outage values or deterministic bounds.

13.3 Analysis for fixed base station positions

13.3.1 Channel with path loss and noisy estimation

In this scenario, the shadowing component is ignored, i.e., $\xi \equiv 0$. In case of attack, the estimated channel responses are those between each base station and the attacker, affected by noise, that is $\hat{h}_{k,n} = g(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x}_k) + w_{k,n}$. The missed detection probability is, thus,

$$p_{\mathrm{MD}} = \Pr_{\{\boldsymbol{x}_k, w_{k,n}\}} \left[\sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \left| \hat{h}_{k,n} - h_{k,n} \right|^2 < \Theta \right] \tag{13.3}$$

$$= \Pr_{\{\boldsymbol{x}_k, w_{k,n}\}} \left[\sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \left| g\left(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x}_k\right) + w_{k,n} - g\left(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x}_k\right) \right|^2 < \Theta \right] . \tag{13.4}$$

Analogously, in case of an authentic message, the estimated channel responses $\hat{h}_{k,n}$ are the one between each base station and the legitimate node, thus the probability of false alarm is

$$p_{\mathrm{FA}} = \Pr_{\{\boldsymbol{x}_k, w_{k,n}\}} \left[\sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \left| \hat{h}_{k,n} - h_{k,n} \right|^2 > \Theta \right]$$

$$= \Pr_{\{w_{k,n}\}} \left[\sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} w_{k,n}^2 > \Theta \right] . \tag{13.5}$$

13.3.2 Channel with path loss and shadowing

The second scenario ignores the estimation noise ($w_{k,n} \equiv 0 \; \forall \, k, n$) but considers the log-normal shadowing. Note that, by taking the logarithm of channel response $g(\boldsymbol{y}, \boldsymbol{z})$, we have

$$\log(g(\boldsymbol{y}, \boldsymbol{z})) = -\log(\varepsilon + \ell(\boldsymbol{y}, \boldsymbol{z})) + \xi \,, \tag{13.6}$$

where $\ell(\boldsymbol{y}, \boldsymbol{z}) = (d(\boldsymbol{y}, \boldsymbol{z})/d_0)^{-\alpha/2}$.

By using maximum likelihood detection techniques, the missed detection probability is

$$\begin{aligned} p_{\mathrm{MD}} &= \Pr_{\{\boldsymbol{x}_k, \xi^A_{k,l}, \xi^0_{k,l}\}} \left[\sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \left| \log \hat{h}_{k,n} - \log h_{k,n} \right|^2 < \Theta \right] \\ &= \Pr_{\{\boldsymbol{x}_k, \xi^A_{k,l}, \xi^0_{k,l}\}} \left[\sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \left(-\log\left(\varepsilon + \ell\left(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x}_k\right)\right) + \xi^A_{k,l} \right. \right. \\ &\qquad \left. \left. + \log\left(\varepsilon + \ell\left(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x}_k\right)\right) - \xi^0_{k,l} \right)^2 < \Theta \right] \\ &= \Pr_{\{\boldsymbol{x}_k, \xi^{(l)}_{k,\mathrm{MD}}\}} \left[\sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \left(\log\left(\frac{\varepsilon + \ell\left(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x}_k\right)}{\varepsilon + \ell\left(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x}_k\right)} \right) + \xi^{(l)}_{k,\mathrm{MD}} \right)^2 < \Theta \right] , \end{aligned} \tag{13.7}$$

where the shadowing exponent for the legitimate user ($\xi^0_{k,l}$) and the attacker ($\xi^A_{k,l}$) are both distributed according to $\mathcal{N}(0, \sigma^2_\xi)$, and $\xi^{(l)}_{k,\mathrm{MD}} = \xi^A_{k,l} - \xi^0_{k,l}$ is distributed according to $\mathcal{N}(0, \sigma^2_{\xi,\mathrm{MD}})$. Suppose $\sigma^2_{\xi,\mathrm{MD}} = 2\sigma^2_\xi(1 - e^{-\delta_{\mathrm{MD}}/X_c})$, where δ_{MD} is the attacker–legitimate user distance. This derives from the Gudmundson's model [335], which describes the correlation of the shadowing component between two points. In this model, the *decorrelation distance* X_c is the distance at which the signal correlation equals $1/e$ of its maximum value. For outdoor systems, X_c typically ranges from 50 m to 100 m [336, 337]. Also, empirical studies show that σ_ξ ranges from 4 to 13 dB [338]. This maximum likelihood criterion makes sense because the information on the phase of the channel is not reliable because of synchronization errors, and the use of the module is equivalent to the use of the power, or its logarithm.

Analogously, the probability of false alarm is

$$\begin{aligned} p_{\mathrm{FA}} &= \Pr_{\{\boldsymbol{x}_k, \xi^0_{k,l}\}} \left[\sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \left| \hat{h}_{k,n} - h_{k,n} \right|^2 > \Theta \right] \\ &= \Pr_{\left\{\xi^{(l)}_{k,\mathrm{FA}}\right\}} \left[\sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \left(\xi^{(l)}_{k,\mathrm{FA}} \right)^2 > \Theta \right] . \end{aligned} \tag{13.8}$$

$\xi^{(l)}_{k,\mathrm{FA}} \sim \mathcal{N}(0, \sigma^2_{\xi,\mathrm{FA}})$ is the difference between two realizations of the random variable $\xi^0_{k,l}$. The legitimate node is able to slightly move after network setup, subject to the condition that the movement is small compared to its distance from the nearest base station. In this way, the path loss component of the channel is not influenced by this movement, however the shadowing component is influenced by it, according to the Gudmundson's model. Hence, we

have $\sigma^2_{\xi,\mathrm{FA}} = 2\sigma^2_\xi(1 - e^{-\delta_{\mathrm{FA}}/X_c})$, where δ_{FA} is the distance between the current position of the legitimate user and its position at network setup.

13.4 Stochastic Geometry Analysis

In this section, the results for missed detection and false alarm probabilities from the previous section are generalized to the random distribution of the BSs, according to the system model. In doing so, a fundamental result in the analysis of point processes, the Campbell's theorem for PPPs, is used [106, §4], described in Sec. 4.3.2.

13.4.1 Channel with path loss and noisy estimation

In the specific scenario where the shadowing is ignored, the characteristic function (CF) of the random variable $\gamma'_{\mathrm{MD}} = \sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \left(g\left(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x}_k\right) + w_{k,n} - g\left(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x}_k\right)\right)^2$, representing the sum in (13.3), becomes

$$\varphi_{\mathrm{MD}}(t) = \exp\left(\int_S \mathrm{E}_{\left\{w^{(n)}\right\}}\left[\exp\left(jt\sigma^2 \sum_{n=1}^{N_c} f\left(\boldsymbol{x}, w^{(n)}\right)\right) - 1\right] \lambda \,\mathrm{d}\boldsymbol{x}\right) \tag{13.9}$$

where

$$f(\boldsymbol{x}, w) = \left(\frac{g\left(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x}\right) - g\left(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x}\right)}{\sigma} + \frac{w}{\sigma}\right)^2 . \tag{13.10}$$

For fixed $\boldsymbol{x}$, the sum $\sum_{n=1}^{N_c} f\left(\boldsymbol{x}, w_n\right)$ is a noncentral chi-squared random variable with N_c degrees of freedom and noncentrality parameter

$$\rho_{\mathrm{BS}}(\boldsymbol{x})^2 = \sum_{n=1}^{N_c} \frac{(g(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x}) - g(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x}))^2}{\sigma^2} = N_c \frac{(g(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x}) - g(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x}))^2}{\sigma^2} . \tag{13.11}$$

We can thus use the explicit form of its CF to calculate the expectation in (13.9):

$$\mathrm{E}_{\left\{w^{(l)}\right\}}\left[\exp\left(jt\sigma^2 \sum_{n=1}^{N_c} f\left(\boldsymbol{x}, w_n\right)\right) - 1\right] = \frac{\exp\left(\frac{j\rho_{\mathrm{BS}}(\boldsymbol{x})^2 t\sigma^2}{1-j2t\sigma^2}\right)}{\left(1 - j2t\sigma^2\right)^{N_c/2}} - 1 . \tag{13.12}$$

The CF of γ'_{MD} is then

$$\varphi_{\mathrm{MD}}(t) = \exp\left(\lambda \int_S \left[\frac{\exp\left(\frac{j\rho_{\mathrm{BS}}(\boldsymbol{x})^2 t\sigma^2}{1-j2t\sigma^2}\right)}{\left(1 - j2t\sigma^2\right)^{N_c/2}} - 1\right] \mathrm{d}\boldsymbol{x}\right) . \tag{13.13}$$

Instead, the CF of the random variable $\gamma'_{\mathrm{FA}} = \sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} w_{k,n}^2$, representing the sum in (13.5), can be derived from the CF of γ'_{MD} by setting $\boldsymbol{u}_{\mathrm{A}}$ to the value of $\boldsymbol{u}_{\mathrm{L}}$. The noncentrality parameter of the noncentral chi-squared distribution becomes, thus, zero and the CF of γ'_{FA} simplifies to

$$\varphi_{\mathrm{FA}}(t) = \exp\left(\lambda \int_S \left[\left(1 - j2t\sigma^2\right)^{-N_c/2} - 1\right] \mathrm{d}\boldsymbol{x}\right) . \tag{13.14}$$

13.4.2 Channel with path loss and shadowing

The stochastic geometry analysis in this scenario is analogous to that carried on in the previous one, therefore only the final derivations are reported. For that, we now consider two cases: (i) the shadowing is independent for each channel coefficient of a single user–base station pair; (ii) the instantaneous shadowing is the same for all channel coefficients of a user–base station pair, i.e., $\xi_{k,\mathrm{MD}}^{(1)} = \xi_{k,\mathrm{MD}}^{(2)} = \ldots = \xi_{k,\mathrm{MD}}^{(N_c)} = \xi_{k,\mathrm{MD}}$ and $\xi_{k,\mathrm{FA}}^{(1)} = \xi_{k,\mathrm{FA}}^{(2)} = \ldots = \xi_{k,\mathrm{FA}}^{(N_c)} = \xi_{k,\mathrm{FA}}$.

Independent shadowing

The probability of missed detection is

$$\Pr_{\{\boldsymbol{x}_k, \xi_{k,\mathrm{MD}}^{(l)}\}} \left[\sigma_{\xi,\mathrm{MD}}^2 \sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \breve{f}\left(\boldsymbol{x}_k, \xi_{k,\mathrm{MD}}^{(n)}\right) < \Theta \right], \tag{13.15}$$

where

$$\breve{f}(\boldsymbol{x}, \xi) = \left(\frac{1}{\sigma_{\xi,\mathrm{MD}}} \log\left(\frac{\varepsilon + \ell(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x})}{\varepsilon + \ell(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x})} \right) + \frac{\xi}{\sigma_{\xi,\mathrm{MD}}} \right)^2. \tag{13.16}$$

As previously done, by exploiting Campbell's theorem, we can write the CF of $\gamma''_{\mathrm{MD}} = \sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \breve{f}\left(\boldsymbol{x}_k, \xi_{k,\mathrm{MD}}^{(n)}\right)$, which is

$$\varphi_{\mathrm{MD}}(t) = \exp\left(\int_S \mathrm{E}_{\left\{\xi_{\mathrm{MD}}^{(n)}\right\}} \left[\exp\left(jt\sigma_{\xi,\mathrm{MD}}^2 \sum_{n=1}^{N_c} \breve{f}\left(\boldsymbol{x}, \xi_{\mathrm{MD}}^{(n)}\right) \right) - 1 \right] \lambda \, \mathrm{d}\boldsymbol{x} \right) \tag{13.17}$$

For fixed $\boldsymbol{x}$, $\sum_{n=1}^{N_c} \breve{f}(\boldsymbol{x}, \xi_{\mathrm{MD}}^{(n)})$ is a noncentral chi-squared distributed variable with N_c degrees of freedom and noncentrality parameter

$$\breve{\rho}_{\mathrm{BS}}(\boldsymbol{x})^2 = \frac{N_c}{\sigma_{\xi,\mathrm{MD}}^2} \left(\log\left(\frac{\varepsilon + \ell(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x})}{\varepsilon + \ell(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x})} \right) \right)^2. \tag{13.18}$$

From there, we can calculate the expected value of this noncentral chi-squared variable and, therefore, we can explicit the CF of γ''_{MD}:

$$\varphi_{\mathrm{MD}}(t) = \exp\left(\lambda \int_S \left[\frac{\exp\left(\frac{j\breve{\rho}_{\mathrm{BS}}(\boldsymbol{x})^2 t\sigma_{\xi,\mathrm{MD}}^2}{1 - j2t\sigma_{\xi,\mathrm{MD}}^2} \right)}{\left(1 - j2t\sigma_{\xi,\mathrm{MD}}^2\right)^{N_c/2}} - 1 \right] \mathrm{d}\boldsymbol{x} \right) \tag{13.19}$$

As for the CF of $\gamma''_{\mathrm{FA}} = \sum_{k=1}^{N_{\mathrm{BS}}} \sum_{n=1}^{N_c} \left(\xi_{k,\mathrm{FA}}^{(n)}\right)^2$, we follow the same reasoning as before, but replacing $\sigma_{\xi,\mathrm{MD}}$ with $\sigma_{\xi,\mathrm{FA}}$. Also, the noncentrality parameter results zero, since the path loss terms can be simplified, resulting in

$$\varphi_{\mathrm{FA}}(t) = \exp\left(\lambda \int_S \left[\left(1 - j2t\sigma_{\xi,\mathrm{FA}}^2\right)^{-N_c/2} - 1 \right] \mathrm{d}\boldsymbol{x} \right). \tag{13.20}$$

Identical shadowing for channel coefficients

Note that in this case, varying N_c is equivalent to varying Θ. So, doubling N_c is the same as halving Θ, both in p_{MD} and p_{FA}. Performance of the authentication scheme is then the same as the single channel per link case (i.e., the independent shadowing case with $N_c = 1$). For the sake of completeness, though, we explicitly perform the analysis also in this case.

Since the shadowing is identical for the different channel coefficients, the CF of γ''_{MD} is just

$$\varphi_{\mathrm{MD}}(t) = \exp\left(\int_S \mathrm{E}_{\xi_{\mathrm{MD}}}\left[\exp\left(jt\sigma^2_{\xi,\mathrm{MD}} N_c \breve{f}\left(\boldsymbol{x}, \xi_{\mathrm{MD}}\right)\right) - 1\right] \lambda \, \mathrm{d}\boldsymbol{x}\right), \quad (13.21)$$

with $\breve{f}\left(\boldsymbol{x}, \xi_{\mathrm{MD}}\right)$ being a noncentral chi-squared variable with one degree of freedom and noncentrality parameter

$$\breve{\rho}_{\mathrm{BS}}(\boldsymbol{x})^2 = \frac{1}{\sigma^2_{\xi,\mathrm{MD}}}\left(\log\left(\frac{\varepsilon + \ell(\boldsymbol{u}_{\mathrm{L}}, \boldsymbol{x})}{\varepsilon + \ell(\boldsymbol{u}_{\mathrm{A}}, \boldsymbol{x})}\right)\right)^2 \quad (13.22)$$

The CF of γ''_{MD} is, therefore,

$$\varphi_{\mathrm{MD}}(t) = \exp\left(\lambda \int_S \left[\frac{\exp\left(\frac{jN_c\breve{\rho}_{\mathrm{BS}}(\boldsymbol{x})^2 t\sigma^2_{\xi,\mathrm{MD}}}{1-j2N_c t\sigma^2_{\xi,\mathrm{MD}}}\right)}{\left(1 - j2N_c t\sigma^2_{\xi,\mathrm{MD}}\right)^{1/2}} - 1\right] \mathrm{d}\boldsymbol{x}\right) \quad (13.23)$$

Analogously, the CF of γ''_{FA} is

$$\varphi_{\mathrm{FA}}(t) = \exp\left(\lambda \int_S \left[\left(1 - j2N_c t\sigma^2_{\xi,\mathrm{FA}}\right)^{-1/2} - 1\right] \mathrm{d}\boldsymbol{x}\right). \quad (13.24)$$

13.4.3 CF inversion

From the CF $\varphi(t)$, the CDF is given by [339]

$$F(y) = \frac{1}{2} - \frac{1}{\pi}\int_0^\infty \Im\left(\frac{e^{-jty}\varphi(t)}{t}\right) \mathrm{d}t\,, \quad (13.25)$$

where $\Im(s)$ denotes the imaginary part of s. Usually, this integral can not be explicitly solved, so we resort to its numerical integration, using the SciPy scientific library for Python.

13.5 Results

In this section, performance results of the proposed authentication scheme, in terms of false alarm and missed detection probabilities, are reported. The parameter values used to obtain the numerical results are reported in Tab. 13.1.

Fig. 13.1 shows the performance of the scheme when no shadowing is considered. We can see that it is possible to have both a false alarm and missed detection probability values of less than 10^{-3} even in such a simple case. When shadowing is considered but the channel estimation noise is ignored, we can reach similar results, as shown in Fig. 13.2.

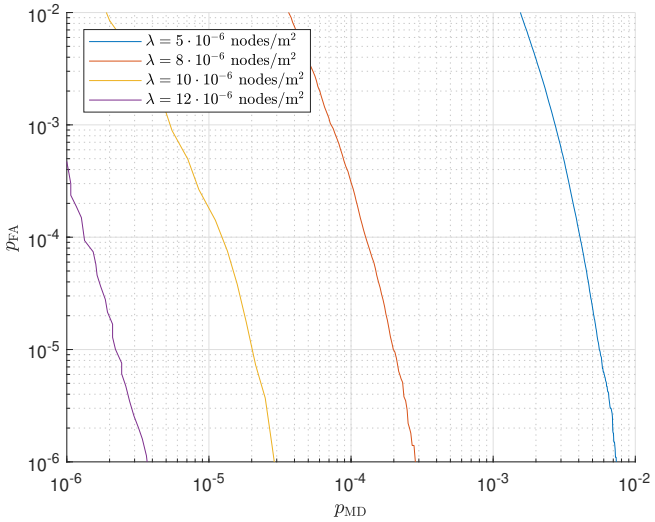


Figure 13.1: p_{FA} vs p_{MD} for different base station densities, when ignoring the shadowing component.

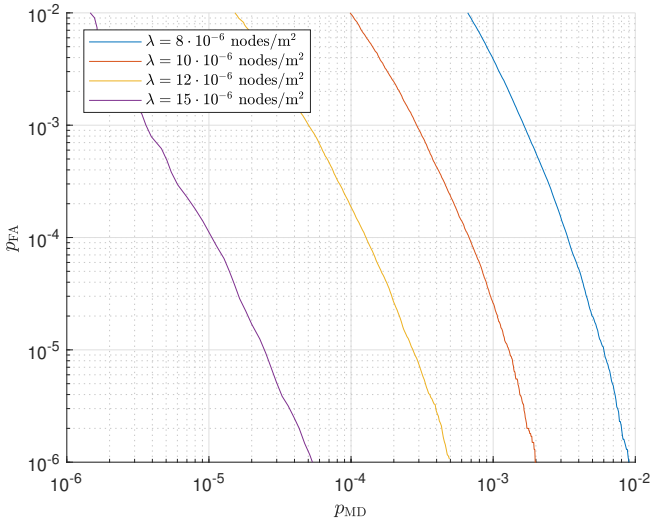


Figure 13.2: p_{FA} vs p_{MD} for different base station densities, when ignoring the channel estimation noise.

Parameter		Value
Path loss exponent	β	3
Path loss normalization factor	d_0	1
Channel gain stability constant	ε	1
Number of channel coefficients	N_c	1
Estimation noise power	σ^2	$2 \cdot 10^{-14}$ W
Variance of the shadowing in dB	σ_ξ^2	8 dB
Decorrelation distance	X_c	75 m
Attacker–legitimate user distance	δ_{MD}	5 m
Legitimate user movement after setup	δ_{FA}	0.5 m

Table 13.1: Value of the parameters used for the numerical evaluation.

In both cases, increasing the base station density, or raising the amount of channel coefficients considered, improves the performance of this scheme, though increasing the size and complexity of the infrastructure. System designers, therefore, have the ability to choose the best tradeoff between complexity and performance level that is adequate to their applications.

13.6 Conclusions

In this chapter, a new authentication scheme, based on physical channel features, has been proposed. The channel considered in this study consisted of the gain and shadowing components, in addition to the estimation noise. A mathematical model based on stochastic geometry allowed us to analyze the performance of this scheme in terms of probability of false alarm and missed detection. Numerical results showed that performance, in the considered scenario, can be tuned based on the specific needs of the application. This means that the proposed authentication scheme can either be set up to have high performance, by, e.g., increasing the base station density, and then used as a standalone mechanism or can be employed, with a more relaxed performance level and, therefore, lower complexity, as an addition to a traditional authentication scheme in order to increase its security level.

Chapter 14

Final considerations

In this thesis we explored current issues and proposed innovative solutions for IoT systems. Part I focused on CPSs. The work gravitated towards the communication technologies, since communication reliability and network lifetime play a critical role in these systems. To understand the characteristic of CPSs, we analysed the communication patterns of MTDs and a new model for them has been proposed. The new model has proved to be general enough to capture the characteristics of a wide variety of MTDs, enabling its use in contexts with heterogeneous traffic sources. Then, we concentrated on channel access schemes. We reached the conclusion that exploiting innovative rate adaptation techniques enables a significant performance improvement over state of the art protocols, both in terms of transmission success probability and energy efficiency. These improvements are more significant the more dense the network is, enabling the realization of massive and ubiquitous IoT. Also, knowing the content of the messages, in our case sensing data from WSNs, allowed us to reach important improvement over state of the art protocols, particularly in extending the lifetime of battery-powered devices.

In Part II the use of machine learning techniques for network and service optimization has been introduced. In fact, prediction and optimization of the traffic flows in the IoT infrastructure is important to guarantee a high QoS, in particular when deploying services that require a large bandwidth. Video streaming towards mobile devices, like smartphones and tablets, is such an example. In this part, a technique for predicting cell load has been presented, followed by an investigation on dynamic video streaming techniques. For these, a machine learning technique to infer the quality-rate characteristic of a video has been presented, together with resource management and video admission control strategies. Additionally, a predictive proxy that prefetches video segments to improve the quality experienced by the user has been proposed. These techniques proved effective to increase the video QoE and can be used together for maximum effectiveness: the cell traffic predictor estimates the future load of the network links, then the resource management algorithm, based on the predicted load and the estimated quality-rate curves, can assign resources to different users to increase the QoE. At last, the prefetching proxy provides further quality improvements, while wasting less bandwidth and reducing the server load compared to a traditional non-predictive proxy. Machine learning can also be applied to positioning services, which have to provide the system

with accurate information on the location of the devices composing it. We have seen that learning techniques have the ability to enhance the performance of NLOS detection algorithms, particularly when combined with multidirectional receivers. This improves the accuracy of the positioning system, allowing for its use in critical scenarios.

In the third part, we focused on security aspects of IoT. While the factors analysed in the previous part are *technological* enabler, security is a *legal* and *psychological* enabler for ubiquitous IoT. In fact, users[1] will not fully embrace the IoT paradigm until sufficient security level will be assured. Therefore, we investigated the security issues of standard and commonly used protocols in IoT. Then, we concentrated on authentication mechanism, and a new technique using physical layer features has been proposed. This authentication strategy proved to be effective as a stand-alone mechanism or as an additional security layer for legacy systems.

In conclusion, we have seen that, while currently available technologies are not ready for massive and secure deployments, the proposed strategies allow us to unleash the full potential of IoT. In fact, when properly designed, these systems are able to securely interconnect powerful devices, like smartphones and PCs, and resource constrained devices, e.g., sensors and automation controllers, through the Internet. The virtual and physical world can therefore be integrated seamlessly, realizing the CPS vision. The resulting data flows, composed by multimedia streams and short control and data messages, have to be efficiently and securely managed. Large attention must be given to innovative security solutions, which will play a predominant role in fostering the acceptance of these systems by the users and will enable their use for strictly regulated and critical applications.

[1] Note that the term *user* denotes also companies or governmental agencies, which have stricter security regulations to comply to than people.

Published papers

[1] C. Pielli, D. Zucchetto, A. Zanella, L. Vangelista, and M. Zorzi, "Platforms and protocols for the internet of things," *EAI Endorsed Transactions on Internet of Things*, vol. 1, no. 1, Oct. 2015.

[2] E. Lovisotto, E. Vianello, D. Cazzaro, M. Polese, F. Chiariotti, D. Zucchetto, A. Zanella, and M. Zorzi, "Cell traffic prediction using joint spatio-temporal information," in *Proceedings of the 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, May 2017, pp. 1–4.

[3] D. Zucchetto and A. Zanella, "Uncoordinated access schemes for the IoT: Approaches, regulations, and performance," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 48–54, Sep. 2017.

[4] O. N. Østerbø, D. Zucchetto, K. Mahmood, A. Zanella, and O. Grøndalen, "State modulated traffic models for machine type communications," in *29th International Teletraffic Congress (ITC 29)*, vol. 1, Sep. 2017, pp. 90–98.

[5] D. Zucchetto, C. Pielli, A. Zanella, and M. Zorzi, "A random access scheme to balance energy efficiency and accuracy in monitoring applications," in *Proceedings of the Information Theory and Applications Workshop*, Feb. 2018.

[6] M. D. F. D. Grazia, D. Zucchetto, A. Testolin, A. Zanella, M. Zorzi, and M. Zorzi, "QoE multi-stage machine learning for dynamic video streaming," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 1, pp. 146–161, Mar. 2018.

[7] D. Zucchetto and A. Zanella, "Multi-rate ALOHA protocols for machine-type communication," in *Proceedings of the International Conference on Computing, Networking and Communications (ICNC)*, Mar. 2018, pp. 524–530.

[8] D. Zucchetto, C. Pielli, A. Zanella, and M. Zorzi, "Random access in the IoT: an adaptive sampling and transmission strategy," in *Proc. of the 2018 IEEE International Conference on Communications (ICC 2018)*, May 2018.

[9] R. Coutinho, F. Chiariotti, D. Zucchetto, and A. Zanella, "Just-in-time proactive caching for DASH video streaming," in *Proceedings of thr 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Jun. 2018, pp. 1–6.

[10] M. Sansoni, G. Ravagnani, D. Zucchetto, C. Pielli, A. Zanella, and K. Mahmood, "Comparison of M2M traffic models against real world data sets," in *Proceedings of the IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks*, Sep. 2018, pp. 1–6.

Bibliography

[11] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[12] C. Anton-Haro and M. Dohler, *Machine-to-Machine (M2M) Communications: Architecture, Performance and Applications*, 1st ed. Woodhead Publishing Ltd., Jan. 2015.

[13] W. Wolf, "Cyber-physical systems," *Computer*, vol. 42, no. 3, pp. 88–89, Mar. 2009.

[14] Y. Zhang, F. Xie, Y. Dong, G. Yang, and X. Zhou, "High fidelity virtualization of cyber-physical systems," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 04, no. 02, Jun. 2013.

[15] F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng, "From wireless sensor networks towards cyber physical systems," *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 397–413, Aug. 2011.

[16] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, "Cyber-physical systems: A new frontier," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, Jun. 2008, pp. 1–9.

[17] L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia CIRP*, vol. 17, pp. 9–13, Jul. 2014.

[18] L. Wang, M. Törngren, and M. Onori, "Current status and advancement of cyber-physical systems in manufacturing," *Journal of Manufacturing Systems*, vol. 37, pp. 517–527, May 2015.

[19] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber–physical system security for the electric power grid," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 210–224, Jan. 2012.

[20] "IoT Connections to Grow 140Accelerates ROI," Juniper Research, Tech. Rep., Jun. 2018. [Online]. Available: https://www.juniperresearch.com/press/press-releases/iot-connections-to-grow-140-to-hit-50-billion

[21] "Ericsson mobility report," Ericsson, Tech. Rep., Jun. 2018. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-june-2018.pdf

[22] "The Internet of Things: How to capture the value of IoT," McKinsey&Company, Tech. Rep., May 2018. [Online]. Available: https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/The%20Internet%20of%20Things%20How%20to%20capture%20the%20value%20of%20IoT/How-to-capture-the-value-of-IoT.ashx

[23] 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, Mar. 2015.

[24] L. Tan and N. Wang, "Future Internet: The Internet of Things," in *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 5, 22–10 Aug. 2010, pp. 376–380.

[25] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, Oct. 2016.

[26] Vodafone Group Plc., "New Study Item on Cellular System Support for Ultra Low Complexity and Low Throughput Internet of Things," 3GPP TSG GERAN, Tech. Rep. GP-140421, May 2014.

[27] A. Rico-Alvarino, M. Vajapeyam, H. Xu, X. Wang, Y. Blankenship, J. Bergman, T. Tirronen, and E. Yavuz, "An overview of 3GPP enhancements on machine to machine communications," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 14–21, Jun. 2016.

[28] Y.-E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3GPP Narrowband Internet of Things," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 117–123, Mar. 2017.

[29] "M2M and IoT redefined through cost effective and energy optimized connectivity," SIGFOX, Tech. Rep. [Online]. Available: http://www.sigfox.com/static/media/Files/Documentation/SIGFOX_Whitepaper.pdf

[30] F. Sforza, "Communications system," U.S. Patent 8,406,275, Mar., 2013.

[31] "LoRaWAN™Specification," LoRa™Alliance, Feb. 2016, V1.0 rev. B.

[32] T. Myers, "Random phase multiple access communication interface system and method," U.S. Patent 7,782,926, Aug., 2010.

[33] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol–HTTP/1.1," IETF RFC 2616, 1999.

[34] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[35] (2014, Oct.) OASIS: MQTT Version 3.1.1. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

[36] (2015) ISO/IEC DIS 20922 Information technology–Message Queuing Telemetry Transport (MQTT) v3.1.1. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=69466

[37] (2012, Oct.) OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0. [Online]. Available: http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf

[38] (2014) ISO/IEC 19464:2014 Information technology–Advanced Message Queuing Protocol (AMQP) v1.0 specification. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=64955

[39] P. Patierno, "Internet of Things: protocols war!" in *Betterembedded 2014*, Florence, Italy, 4–5 Jul. 2014.

[40] A. Foster, "Messaging Technologies for the Industrial Internet and the Internet of Things," PrismTech, Tech. Rep., Nov. 2013.

[41] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," IETF RFC 5246, 2008.

[42] G. Fersi, "Middleware for Internet of Things: a study," in *IEEE International Conference on Distributed Computing in Sensor Systems*, Jun. 2015.

[43] S. Bandyopadhyay, M. Sengputa, S. Maiti, and S. Dutta, "Role of Middleware for Internet of Things," *International Journal of Computer Science & Engineering Survey*, vol. 2, no. 3, pp. 94–105, Aug. 2011.

[44] *Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1000 MHz frequency range with power levels ranging up to 500 mW*, ETSI EN 300 220, ETSI European Standard, Rev. 2.4.1, May 2012, (accessed on Nov 28, 2016). [Online]. Available: http://www.etsi.org/deliver/etsi_en/300200_300299/30022001/02.04.01_60/en_30022001v020401p.pdf

[45] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the Internet of Things: A Review," in *2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)*, vol. 3, Mar. 2012, pp. 648–651.

[46] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," *IEEE Network*, vol. 44, no. 9, pp. 51–58, Sep. 2011.

[47] *IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks*, IEEE Std., 2012.

[48] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Large-scale measurement and characterization of cellular machine-to-machine traffic," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1960–1973, Dec. 2013.

[49] N. Nikaein, M. Laner, K. Zhou, P. Svoboda, D. Drajic, M. Popovic, and S. Krco, "Simple traffic modeling framework for machine type communication," in *Proceedings of the Tenth International Symposium on Wireless Communication Systems (ISWCS)*, Aug. 2013, pp. 1–5.

[50] M. Centenaro and L. Vangelista, "A study on M2M traffic and its impact on cellular networks," in *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Dec. 2015, pp. 154–159.

[51] "Study on RAN improvements for machine-type communications," 3GPP TR 37.868 V11.0.0, 3GPP, Tech. Rep., Sep. 2011.

[52] G. C. Madueño, C. Stefanović, and P. Popovski, "Reliable and efficient access for alarm-initiated and regular M2M traffic in IEEE 802.11ah systems," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 673–682, Oct. 2016.

[53] O. Al-Khatib, W. Hardjawana, and B. Vucetic, "Traffic modeling for machine-to-machine (M2M) last mile wireless access networks," in *Proceedings of the 2014 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2014, pp. 1199–1204.

[54] E. Grigoreva, M. Laurer, M. Vilgelm, T. Gehrsitz, and W. Kellerer, "Coupled markovian arrival process for automotive machine type communication traffic modeling," in *Proceedings of the IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[55] H. Thomsen, C. N. Manchon, and B. H. Fleury, "A traffic model for machine-type communications using spatial point processes," in *28th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct. 2017, pp. 1–6.

[56] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, "Traffic models for machine type communications," in *The Tenth International Symposium on Wireless Communication Systems*, Aug. 2013.

[57] T. W. Liao, "Clustering of time series data — a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, May 2005.

[58] D. R. Cox, *Renewal theory*, ser. Methuen's monographs on applied probability and statistics. Methuen and Co. Ltd, 1962.

[59] A. M. Kshirsagar and Y. P. Gupta, "Asymptotic values of first two moments in Markov renewal processes," *Biometrika*, vol. 54, no. 3-4, pp. 597–603, 1967.

[60] J. J. Hunter, "On the moments of Markov renewal processes," *Advances in Applied Probability*, vol. 1, no. 2, pp. 188–210, 1969.

[61] "The zettabyte era: Trends and analysis," CISCO, White Paper, Jun. 2016, (accessed on February 27, 2017). [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.pdf

[62] "Bluetooth Core Specification 4.2," Bluetooth SIG, Dec. 2014, (accessed on Nov 28, 2016). [Online]. Available: https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=286439

[63] *Short range narrow-band digital radiocommunication transceivers - PHY, MAC, SAR and LLC layer specifications*, ITU-T G.9959, International Telecommunication Union Recommendation, Jan. 2015, (accessed on Nov 28, 2016). [Online]. Available: http://handle.itu.int/11.1002/1000/12399

[64] "Cisco visual networking index: Global mobile data traffic forecast update, 2015–2020," Cisco, White Paper, Feb. 2016, (accessed on Nov 15, 2016). [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[65] Code of Federal Regulations, Title 47, Ch. I, Part 15, Federal Communications Commission, (accessed on Nov 10, 2016). [Online]. Available: http://www.ecfr.gov/cgi-bin/text-idx?node=pt47.1.15

[66] T. Li, H. Wang, and L. Tong, "Hybrid Aloha: A novel medium access control protocol," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 4, May 2006, pp. IV–IV.

[67] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05)*, Jul. 2005, pp. 166–172.

[68] A. Zanella, "Adaptive batch resolution algorithm with deferred feedback for wireless systems," *IEEE Transactions on Wireless Communications*, vol. 11, no. 10, pp. 3528–3539, Oct. 2012.

[69] J. Yu and L. Chen, "Stability analysis of frame slotted aloha protocol," in *23rd IEEE International Symposium on Quality of Service (IWQoS)*, Jun. 2015, pp. 329–338.

[70] F. Vázquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Modeling and analysis of Reservation Frame Slotted-ALOHA in wireless Machine-to-Machine area networks for data collection," *Sensors*, vol. 15, no. 2, pp. 3911–3931, Feb. 2015.

[71] S. Tasaka, "Stability and performance of the R-Aloha packet broadcast system," *IEEE Transactions on Computers*, vol. C-32, no. 8, pp. 717–726, Aug. 1983.

[72] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, "Packet reservation multiple access for local wireless communications," *IEEE Transactions on Communications*, vol. 37, no. 8, pp. 885–890, Aug. 1989.

[73] S. Nanda, D. J. Goodman, and U. Timor, "Performance of PRMA: a packet voice protocol for cellular systems," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 3, pp. 584–598, Aug. 1991.

[74] G. Pierobon, A. Zanella, and A. Salloum, "Contention-TDMA protocol: performance evaluation," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 4, pp. 781–788, Jul. 2002.

[75] A. Ephremides and O. A. Mowafi, "Analysis of a hybrid access scheme for buffered users-probabilistic time division," *IEEE Transactions on Software Engineering*, vol. SE-8, no. 1, pp. 52–61, Jan. 1982.

[76] H. K. Lee and S. L. Kim, "Network coded ALOHA for wireless multihop networks," in *2009 IEEE Wireless Communications and Networking Conference*, Apr. 2009, pp. 1–5.

[77] C. Stefanovic and P. Popovski, "ALOHA random access that operates as a rateless code," *IEEE Transactions on Communications*, vol. 61, no. 11, pp. 4653–4662, Nov. 2013.

[78] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Transactions on Information Theory*, vol. 61, no. 12, pp. 6815–6832, Dec. 2015.

[79] "Radio equipment directive," Directive 2014/53/EU, Apr. 2014, (accessed on Nov 28, 2016). [Online]. Available: http://data.europa.eu/eli/dir/2014/53/oj

[80] "Radio and telecommunications terminal equipment," Directive 1999/5/EC, Mar. 1999, (accessed on Nov 28, 2016). [Online]. Available: http://data.europa.eu/eli/dir/1999/5/oj

[81] *Short Range Devices (SRD) operating in the frequency range 25 MHz to 1 000 MHz*, ETSI EN 300 220, ETSI Draft European Standard, Rev. 3.1.0, May 2016, (accessed on Nov 28, 2016). [Online]. Available: http://www.etsi.org/deliver/etsi_en/300200_300299/30022002/03.01.01_30/en_30022002v030101v.pdf

[82] B. Błaszczyszyn, P. Mühlethaler, and S. Banaouas, "Comparison of Aloha and CSMA in wireless ad-hoc networks under different channel conditions," in *Wireless Ad-Hoc Networks*, H. Zhou, Ed. InTech, 2012, pp. 3–22.

[83] M. Kaynia and N. Jindal, "Performance of ALOHA and CSMA in spatially distributed wireless networks," in *Proceedings of the 2008 IEEE International Conference on Communications*, May 2008, pp. 1108–1112.

[84] I. Ramachandran and S. Roy, "On the impact of clear channel assessment on MAC performance," in *Proceedings of IEEE Globecom 2006*, Nov. 2006, pp. 1–5.

[85] L. Negri, M. Sami, Q. D. Tran, and D. Zanetti, "Flexible power modeling for wireless systems: Power modeling and optimization of two Bluetooth implementations," in *Proceedings of the sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, Jun. 2005, pp. 408–416.

[86] N. Abramson, "The ALOHA system: Another alternative for computer communications," in *Proceedings of the 1970 Fall Joint Computer Conference*, ser. AFIPS '70 (Fall), Nov. 1970, pp. 281–285.

[87] M. Ferguson, "A bound and approximation of delay distribution for fixed-length packets in an unslotted ALOHA channel and a comparison with time division multiplexing (TDM)," *IEEE Transactions on Communications*, vol. 25, no. 1, pp. 136–139, Jan. 1977.

[88] M. Ferguson, "An approximate analysis of delay for fixed and variable length packets in an unslotted ALOHA channel," *IEEE Transactions on Communications*, vol. 25, no. 7, pp. 644–654, Jul. 1977.

[89] P. R. Jelenković and J. Tan, "Is ALOHA causing power law delays?" in *Managing Traffic Performance in Converged Networks: Proceedings of the 20th International Teletraffic Congress (ITC20)*, L. Mason, T. Drwiega, and J. Yan, Eds., Jun. 2007, pp. 1149–1160.

[90] S. Bellini and F. Borgonovo, "On the throughput of an ALOHA channel with variable length packets," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1932–1935, Nov. 1980.

[91] A. Dziech and A. R. Pach, "Bounds on the throughput of an unslotted ALOHA channel in the case of a heterogeneous users' population," *Kybernetika*, vol. 25, no. 6, pp. 476–485, 1989.

[92] N. Abramson, "The throughput of packet broadcasting channels," *IEEE Transactions on Communications*, vol. 25, no. 1, pp. 117–128, Jan. 1977.

[93] R. Nelson and L. Kleinrock, "The spatial capacity of a slotted ALOHA multihop packet radio network with capture," *IEEE Transactions on Communications*, vol. 32, no. 6, pp. 684–694, Jun. 1984.

[94] C. Namislo, "Analysis of mobile radio slotted ALOHA networks," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 4, pp. 583–588, Jul. 1984.

[95] D. J. Goodman and A. A. M. Saleh, "The near/far effect in local ALOHA radio communications," *IEEE Transactions on Vehicular Technology*, vol. 36, no. 1, pp. 19–27, Feb. 1987.

[96] J. Arnbak and W. van Blitterswijk, "Capacity of slotted ALOHA in Rayleigh-fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 261–269, Feb. 1987.

[97] I. M. I. Habbab, M. Kavehrad, and C. E. W. Sundberg, "ALOHA with capture over slow and fast fading radio channels with coding and diversity," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 1, pp. 79–88, Jan. 1989.

[98] D. Dardari, V. Tralli, and R. Verdone, "On the capacity of slotted Aloha with Rayleigh fading: the role played by the number of interferers," *IEEE Communications Letters*, vol. 4, no. 5, pp. 155–157, May 2000.

[99] R. Clark Robertson and T. T. Ha, "Effect of capture on throughput of variable length packet Aloha systems," *Computer Communications*, vol. 17, no. 12, pp. 836–842, Dec. 1994.

[100] B. Błaszczyszyn and P. Mühlethaler, "Stochastic analysis of non-slotted Aloha in wireless ad-hoc networks," in *Proceedings of IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[101] N. Jindal, J. G. Andrews, and S. Weber, "Bandwidth partitioning in decentralized wireless networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 5408–5419, Dec. 2008.

[102] S. Weber, J. G. Andrews, and N. Jindal, "An overview of the transmission capacity of wireless networks," *IEEE Transactions on Communications*, vol. 58, no. 12, pp. 3593–3604, Dec. 2010.

[103] F. Baccelli and B. Blaszczyszyn, *Stochastic Geometry and Wireless Networks, Volume I - Theory*, ser. Foundations and Trends in Networking Vol. 3. NoW Publishers, 2009, vol. 1. [Online]. Available: https://hal.inria.fr/inria-00403039

[104] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 7, pp. 1029–1046, Sep. 2009.

[105] J. Kingman, *Poisson Processes*, ser. Oxford Studies in Probability. Clarendon Press, 1993.

[106] M. Haenggi, *Stochastic Geometry for Wireless Networks*. Cambridge University Press, 2013.

[107] A. Zanella and M. Zorzi, "Theoretical analysis of the capture probability in wireless systems with multiple packet reception capabilities," *IEEE Transactions on Communications*, vol. 60, no. 4, pp. 1058–1071, 2012.

[108] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[109] A. Ponsich and C. C. Coello, "Differential evolution performances for the solution of mixed-integer constrained process engineering problems," *Applied Soft Computing*, vol. 11, no. 1, pp. 399–409, Jan. 2011.

[110] P. Padhy, R. K. Dash, K. Martinez, and N. R. Jennings, "A utility-based sensing and communication model for a glacial sensor network," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, May 2006, pp. 1353–1360.

[111] C. Alippi, G. Anastasi, C. Galperti, F. Mancini, and M. Roveri, "Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications," in *IEEE International Conference on Mobile Adhoc and Sensor Systems*, Oct. 2007, pp. 1–6.

[112] X. Wu and M. Liu, "In-situ soil moisture sensing: Measurement scheduling and estimation using compressive sensing," in *ACM/IEEE 11th International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2012, pp. 1–11.

[113] P. Y. Chen, S. Yang, and J. A. McCann, "Distributed real-time anomaly detection in networked industrial sensing systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3832–3842, Jun. 2015.

[114] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis, "Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the north sea," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, ser. SenSys '05, 2005, pp. 64–75.

[115] 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.

[116] A. Jain and E. Y. Chang, "Adaptive sampling for sensor networks," in *Proceeedings of the 1st International Workshop on Data Management for Sensor Networks*, Aug. 2004, pp. 10–16.

[117] 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.

[118] J. Kho, A. Rogers, and N. R. Jennings, "Decentralized control of adaptive sampling in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, pp. 19:1–19:35, May 2009.

[119] 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.

[120] B. Srbinovski, M. Magno, F. Edwards-Murphy, V. Pakrashi, and E. Popovici, "An energy aware adaptive sampling algorithm for energy harvesting WSN with energy hungry sensors," *Sensors*, vol. 16, no. 4, Mar. 2016.

[121] R. Willett, A. Martin, and R. Nowak, "Backcasting: Adaptive sampling for sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, ser. IPSN '04, Apr. 2004, pp. 124–133.

[122] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, vol. 30, Aug. 2004, pp. 588–599.

[123] S. Knorn, S. Dey, A. Ahlen, and D. E. Quevedo, "Distortion minimization in multi-sensor estimation using energy harvesting and energy sharing," *IEEE Trans. on Signal Processing*, vol. 63, no. 11, pp. 2848–2863, Jun. 2015.

[124] 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.

[125] V. Raghunathan, S. Ganeriwal, and M. Srivastava, "Emerging techniques for long lived wireless sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 108–114, Apr. 2006.

[126] 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.

[127] B. Gedik, L. Liu, and P. S. Yu, "ASAP: An adaptive sampling approach to data collection in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 12, pp. 1766–1783, Dec. 2007.

[128] 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.

[129] 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.

[130] 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.

[131] 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, Sep. 2011.

[132] 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.

[133] 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, Jun. 2017.

[134] G. Quer, R. Masiero, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "On the interplay between routing and signal representation for compressive sensing in wireless sensor networks," in *Information Theory and Applications Workshop*, Feb. 2009, pp. 206–215.

[135] 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.

[136] 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.

[137] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. J. Goldsmith, "Power scheduling of universal decentralized estimation in sensor networks," *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 413–422, Feb. 2006.

[138] 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.

[139] 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.

[140] *Information technology–Generic coding of moving pictures and associated audio information*, ISO/IEC Std. 13 818-2:2013, Oct. 2013.

[141] 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.

[142] 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, pp. 6894–6908, Apr. 2017.

[143] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 65–75, Nov. 2014.

[144] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "Anticipatory networking in future generation mobile networks: a survey," *submitted to IEEE Communications Survey and Tutorials*, Jun. 2016. [Online]. Available: https://arxiv.org/abs/1606.00191

[145] F. Chiariotti, D. Del Testa, M. Polese, A. Zanella, G. M. Di Nunzio, and M. Zorzi, "Learning methods for long-term channel gain prediction in wireless networks," in *International Conference on Computing, Networking and Communications (ICNC)*. IEEE, Jan. 2017.

[146] Y. Jiang, D. C. Dhanapala, and A. P. Jayasumana, "Tracking and prediction of mobility without physical distance measurements in sensor networks," in *International Conference on Communications (ICC)*. IEEE, Jun. 2013, pp. 1845–1850.

[147] R. Li, Z. Zhao, X. Zhou, and H. Zhang, "Energy savings scheme in radio access networks via compressive sensing-based traffic load prediction," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 4, pp. 468–478, Nov. 2012.

[148] S. E. Hammami, H. Afifi, M. Marot, and V. Gauthier, "Network planning tool based on network classification and load prediction," in *2016 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, Apr. 2016, pp. 1–6.

[149] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das, "Understanding traffic dynamics in cellular data networks," in *IEEE INFOCOM 2011 - The 30th Annual IEEE International Conference on Computer Communications*. IEEE, Apr. 2011, pp. 882–890.

[150] R. Li, Z. Zhao, X. Chen, J. Palicot, and H. Zhang, "TACT: a transfer actor-critic learning framework for energy saving in cellular radio access networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 2000–2011, Apr. 2014.

[151] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell, "NextPlace: a spatio-temporal prediction framework for pervasive systems," in *International Conference on Pervasive Computing*. Springer, May 2011, pp. 152–169.

[152] H. Gao, J. Tang, and H. Liu, "Mobile location prediction in spatio-temporal context," in *Nokia Mobile Data Challenge Workshop*, Jun. 2012.

[153] W.-S. Soh and H. S. Kim, "QoS provisioning in cellular networks based on mobility prediction techniques," *IEEE Communications Magazine*, vol. 41, no. 1, pp. 86–92, Jan. 2003.

[154] O. Ohashi and L. Torgo, "Wind speed forecasting using spatio-temporal indicators," in *20th European Conference on Artificial Intelligence (ECAI'12)*. IOS Press, Aug. 2012, pp. 975–980.

[155] J. Ma and J. C. Cheng, "Estimation of the building energy use intensity in the urban scale by integrating GIS and big data technology," *Applied Energy*, vol. 183, pp. 182–192, Dec. 2016.

[156] F. Herrema, V. Treve, R. Curran, and H. Visser, "Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach," in *International Conference for Research in Air Transportation*, Jun. 2016.

[157] D. F. Andrews, "A robust method for multiple linear regression," *Technometrics*, vol. 16, no. 4, pp. 523–531, Nov. 1974.

[158] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, Feb. 1970.

[159] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, Jan. 1996.

[160] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, Apr. 2005.

[161] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203–224, Oct. 2007.

[162] U. Grömping, "Variable importance assessment in regression: linear regression versus random forest," *The American Statistician*, vol. 63, no. 4, pp. 308–319, Sep. 2008.

[163] D. F. Specht, "A general regression neural network," *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568–576, Nov. 1991.

[164] N. R. Draper and H. Smith, *Applied regression analysis*. John Wiley & Sons, May 1998.

[165] T. J. Barnett, A. Sumits, S. Jain, and U. Andra, "Cisco Visual Networking Index (VNI) Update Global Mobile Data Traffic Forecast," *Vni*, pp. 2015–2020, 2015. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[166] N. Amram, B. Fu, G. Kunzmann, T. Melia, D. Munaretto, S. Randriamasy, B. Sayadi, J. Widmer, and M. Zorzi, "QoE-based transport optimization for video delivery over next generation cellular networks," in *IEEE ISCC*. IEEE, 2011, pp. 19–24.

[167] "ISO/IEC 23009-1:2014: Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats," International Organization for Standardization, Standard, May 2014.

[168] M. Pantos, "IETF Draft: Apple HTTP Live Streaming (HLS)," Internet Engineering Task Force, Standard, 2013.

[169] "Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH)," 3GPP TS 26.247 v15.3.0, Jun. 2018.

[170] D. Munaretto, F. Giust, G. Kunzmann, and M. Zorzi, "Performance analysis of dynamic adaptive video streaming over mobile content delivery networks," in *IEEE ICC 2014 - Communication QoS, Reliability and Modeling Symposium (ICC'14 CQRM)*, Sydney, Australia, Jun. 2014, pp. 1059–1064.

[171] M. Zanforlin, D. Munaretto, A. Zanella, and M. Zorzi, "SSIM-based video admission control and resource allocation algorithms," in *WiVid Workshop of IEEE WiOpt*, Hammamet, Tunisia, May 2014.

[172] A. Testolin, M. Zanforlin, M. D. F. D. Grazia, D. Munaretto, A. Zanella, M. Zorzi, and M. Zorzi, "A machine learning approach to QoE-based video admission control and resource allocation in wireless systems," in *13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, Jun. 2014, pp. 31–38.

[173] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel distributed processing: Explorations on the microstructure of cognition. Volume 1: Foundations*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986.

[174] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600–612, Apr. 2004.

[175] *Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference*, ITU-T Recommendation J.144, Mar. 2004.

[176] K. Seshadrinathan and A. Bovik, "Motion tuned spatio-temporal quality assessment of natural videos," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 335–350, Feb. 2010.

[177] M. Saad and A. Bovik, "Blind quality assessment of videos using a model of natural scene statistics and motion coherency," in *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Nov. 2012, pp. 332–336.

[178] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," in *8th International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, Nice, France, Dec. 2012, pp. 97–108.

[179] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, 2014.

[180] S. Petrangeli, J. Famaey, M. Claeys, and F. De Turck, "A QoE-driven rate adaptation heuristic for enhanced adaptive video streaming," Ghent University-iMinds, Department of Information Technology, Tech. Rep., 2014.

[181] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 187–198, Aug. 2014.

[182] D. Stohr, A. Frömmgen, J. Fornoff, M. Zink, A. Buchmann, and W. Effelsberg, "Qoe analysis of dash cross-layer dependencies by extensive network emulation," in *Proceedings of the 2016 Workshop on QoE-based Analysis and Management of Data Communication Networks*, ser. Internet-QoE '16, 2016, pp. 25–30.

[183] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 325–338, 2015.

[184] A. Bokani, M. Hassan, and S. Kanhere, "HTTP-based adaptive streaming for mobile clients using Markov Decision Process," in *20th International Packet Video Workshop*, San Jose, CA, USA, Dec 2013, pp. 1–8.

[185] C. Zhou, C.-W. Lin, and Z. Guo, "mDASH: A Markov Decision-based rate adaptation approach for dynamic HTTP streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 4, pp. 738–751, 2016.

[186] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, "D-DASH: a deep Q-learning framework for DASH video streaming," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 703–718, Dec. 2017.

[187] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. De Turck, "Design of a Q-learning-based client quality selection algorithm for HTTP adaptive video streaming," in *Adaptive and Learning Agents Workshop (ALA-2013)*, Saint Paul, Minnesota, USA, May 2013, pp. 30–37.

[188] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. De Turck, "Design and optimisation of a (FA) Q-learning-based HTTP adaptive streaming client," *Connection Science*, vol. 26, no. 1, pp. 25–43, 2014.

[189] V. Martín, J. Cabrera, and N. García, "Q-learning based control algorithm for HTTP adaptive streaming," in *International Conference on Visual Communications and Image Processing (VCIP)*, Singapore, Singapore, Dec 2015, pp. 1–4.

[190] L. Yu, T. Tillo, and J. Xiao, "Qoe-driven dynamic adaptive video streaming strategy with future information," *IEEE Transactions on Broadcasting*, vol. 63, no. 3, pp. 523–534, Sep. 2017.

[191] J. Chen, M. Ammar, M. Fayed, and R. Fonseca, "Client-driven network-level qoe fairness for encrypted 'dash-s'," in *Proceedings of the 2016 Workshop on QoE-based Analysis and Management of Data Communication Networks*, ser. Internet-QoE '16, 2016, pp. 55–60.

[192] A. S. Abdallah and A. B. MacKenzie, "A cross-layer controller for adaptive video streaming over ieee 802.11 networks," in *IEEE International Conference on Communications (ICC)*, Jun. 2015, pp. 6797–6802.

[193] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for Dynamic Adaptive Streaming over HTTP," *IEEE Communications Surveys & Tutorials*, 2017.

[194] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 107–123, Jul. 2003.

[195] S. Latré and F. De Turck, "Joint in-network video rate adaptation and measurement-based admission control: Algorithm design and evaluation," *Journal of Network and Systems Management*, vol. 21, no. 4, pp. 588–622, Dec. 2013.

[196] S. Qadir and A. A. Kist, "Video-aware measurement-based admission control," in *Proc. of the Australasian Telecommunication Networks and Applications Conference (ATNAC)*, Nov. 2013, pp. 178–182.

[197] B. Feitor, P. Assuncao, J. Soares, L. Cruz, and R. Marinheiro, "Objective quality prediction model for lost frames in 3D video over TS," in *IEEE ICC*, Budapest, Hungary, Jun. 2013.

[198] M. Naccari, M. Tagliasacchi, and S. Tubaro, "No-reference video quality monitoring for H.264/AVC coded video," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 932–946, Aug. 2009.

[199] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: a tutorial," *IEEE Communications Surveys and Tutorials*, vol. 6, pp. 58 – 78, Oct-Dec 2004.

[200] M. Katsarakis, R. C. Teixeira, M. Papadopouli, and V. Christophides, "Towards a causal analysis of video qoe from network and application qos," in *Proceedings of the 2016 Workshop on QoE-based Analysis and Management of Data Communication Networks*, ser. Internet-QoE '16, 2016, pp. 31–36.

[201] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki, "Measuring video qoe from encrypted traffic," in *Proceedings of the 2016 Internet Measurement Conference*, ser. IMC '16, 2016, pp. 513–526.

[202] I. Orsolic, D. Pevec, M. Suznjevic, and L. Skorin-Kapov, "A machine learning approach to classifying youtube qoe based on encrypted network traffic," *Multimedia Tools and Applications*, May 2017.

[203] M. Ries, M. Slanina, and D. M. Garcia, "Reference free SSIM estimation for full HD video content," in *Proceedings of 21st International Conference Radioelektronika*, Apr. 2011, pp. 1–4.

[204] T.-L. Lin, N.-C. Yang, R.-H. Syu, C.-C. Liao, W.-L. Tsai, C.-C. Chou, and S.-L. Chen, "NR-Bitstream video quality metrics for SSIM using encoding decisions in AVC and HEVC coded videos," *Journal of Visual Communication and Image Representation*, vol. 32, pp. 257–271, Oct. 2015.

[205] T. Shanableh, "Prediction of structural similarity index of compressed video at a macroblock level," *IEEE Signal Processing Letters*, vol. 18, no. 5, pp. 335–338, May 2011.

[206] P. Goudarzi, "A no-reference low-complexity QoE measurement algorithm for H.264 video transmission systems," *Scientia Iranica*, vol. 20, no. 3, pp. 721–729, Jun. 2013.

[207] A. Rossholm and B. Lövströem, "A new low complex reference free video quality predictor," in *IEEE 10th Workshop on Multimedia Signal Processing*, Oct. 2008, pp. 765–768.

[208] S. Antani, R. Kasturi, and R. Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video," *Pattern recognition*, vol. 35, no. 4, pp. 945–965, Apr. 2002.

[209] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[210] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 24. MIT, Dec. 2012, pp. 1–9.

[211] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," *Advances in neural information processing systems*, vol. 19, p. 1345, Jan. 2007.

[212] A. Testolin, I. Stoianov, A. Sperduti, and M. Zorzi, "Learning orthographic structure with sequential generative neural networks," *Cognitive Science*, vol. 40, no. 3, pp. 579–606, Apr. 2016.

[213] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T SG16/Q6 - Video Coding Experts Group (VCEG), Technical Report VCEG-M33 from the 13th meeting in Austin, Texas, USA, Apr. 2001.

[214] P. Hanhart and T. Ebrahimi, "Calculation of average coding efficiency based on subjective quality scores," *Journal of Visual Communication and Image Representation*, vol. 25, no. 3, pp. 555–564, Apr. 2014.

[215] J. M. Libert, C. P. Fenimore, and P. Roitman, "Simulation of graded video impairment by weighted summation: validation of the methodology," in *Proc. SPIE*, vol. 3845, Multimedia Systems and Applications II, Nov. 1999, pp. 254–265.

[216] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld, "Impact of frame rate and resolution on objective QoE metrics," in *Workshop on Quality of Multimedia Experience (QoMEX)*, Trondheim, Norway, Jun. 2010.

[217] "Test media repository." [Online]. Available: http://media.xiph.org/video/derf/

[218] "Joint scalable video model - reference software." [Online]. Available: http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm

[219] "x264 encoder." [Online]. Available: http://www.videolan.org/developers/x264.html

[220] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Network*, vol. 24, no. 2, Mar. 2010.

[221] M. Zorzi, A. Zanella, A. Testolin, M. D. F. D. Grazia, and M. Zorzi, "Cognition-based networks: A new perspective on network optimization using learning and distributed intelligence," *IEEE Access*, vol. 3, pp. 1512–1530, Aug. 2015.

[222] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, Jan. 1985.

[223] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[224] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984.

[225] M. Zorzi, A. Testolin, and I. P. Stoianov, "Modeling language and cognition with deep unsupervised learning: a tutorial overview," *Frontiers in Psychology*, vol. 4, Aug. 2013.

[226] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 599–619.

[227] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[228] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95) - Volume 2*, Aug. 1995, pp. 1137–1143.

[229] A. Testolin, I. Stoianov, M. De Filippo De Grazia, and M. Zorzi, "Deep unsupervised learning on a desktop PC: a primer for cognitive scientists," *Frontiers in Psychology*, vol. 4, May 2013.

[230] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," *Journal of Machine Learning Research - Proceedings Track*, vol. 27, pp. 17–36, 2012.

[231] K. Liang, J. Hao, R. Zimmermann, and D. K. Y. Yau, "Integrated prefetching and caching for adaptive video streaming over HTTP," *Proceedings of the 6th ACM Multimedia Systems Conference on - MMSys '15*, pp. 142–152, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=2713168.2713181

[232] D. K. Krishnappa, S. Khemmarat, L. Gao, and M. Zink, "On the feasibility of prefetching and caching for online TV services: A measurement study on hulu," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6579 LNCS, pp. 72–80, 2011.

[233] F. Bronzino, D. Stojadinovic, C. Westphal, and D. Raychaudhuri, "Exploiting network awareness to enhance dash over wireless," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan. 2016, pp. 1092–1100.

[234] M. Claeys, N. Bouten, D. De Vleeschauwer, W. Van Leekwijck, S. Latre, and F. De Turck, "An Announcement-based Caching Approach for Video-on-Demand Streaming," *Network and Service Management (CNSM), 2015 11th International Conference on*, 2015.

[235] C. Zhang, J. Liu, F. Chen, Y. Cui, and E. C. H. Ngai, "Dependency-aware caching for HTTP Adaptive Streaming," *2016 Digital Media Industry and Academic Forum, DMIAF 2016 - Proceedings*, pp. 89–93, 2016.

[236] J. De Vriendt, D. De Vleeschauwer, and D. Robinson, "Model for estimating QoE of video delivered using HTTP adaptive streaming," in *IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, Ghent, Belgium, May 2013, pp. 1288–1293.

[237] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "QoE-driven rate adaptation heuristic for fair adaptive video streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 2, pp. 28:1–28:24, Mar. 2016.

[238] Z. Sahinoglu, S. Gezici, and I. Güvenc, *Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*. Cambridge University Press, Sep. 2008.

[239] B. Alavi and K. Pahlavan, "Modeling of the TOA-based distance measurement error using UWB indoor radio measurements," *IEEE communications letters*, vol. 10, no. 4, pp. 275–277, 2006.

[240] A. Zanella, "Best practice in rss measurements and ranging," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2662–2686, 2016.

[241] J. Khodjaev, Y. Park, and A. Saeed Malik, "Survey of NLOS identification and error mitigation problems in UWB-based positioning algorithms for dense environments," *Annals of Telecommunications*, vol. 65, no. 5, pp. 301–311, Jun. 2010.

[242] D. B. Jourdan, D. Dardari, and M. Z. Win, "Position error bound for UWB localization in dense cluttered environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 613–628, Apr. 2008.

[243] B. Denis, J. Keignart, and N. Daniele, "Impact of NLOS propagation upon ranging precision in UWB systems," in *Proc. of the IEEE Conference on Ultra Wideband Systems and Technologies*, Nov. 2003, pp. 379–383.

[244] J. Schroeder, S. Galler, K. Kyamakya, and K. Jobmann, "NLOS detection algorithms for ultra-wideband localization," in *4th Workshop on Positioning, Navigation and Communication*, Mar. 2007, pp. 159–166.

[245] İ. Güvenç, C.-C. Chong, F. Watanabe, and H. Inamura, "NLOS identification and weighted least-squares localization for UWB systems using multipath channel statistics," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, Aug. 2007.

[246] S. Gezici, H. Kobayashi, and H. V. Poor, "Non-parametric non-line-of-sight identification," in *IEEE 58th Vehicular Technology Conference*, vol. 4, Oct. 2003, pp. 2544–2548.

[247] N. Decarli, D. Dardari, S. Gezici, and A. A. D'Amico, "LOS/NLOS detection for UWB signals: A comparative study using experimental data," in *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, May 2010, pp. 169–173.

[248] J. Borras, P. Hatrack, and N. B. Mandayam, "Decision theoretic framework for NLOS identification," in *Proc. of the 48th IEEE Vehicular Technology Conference*, vol. 2, May 1998, pp. 1583–1587.

[249] X. Lin, L. Ju, and S. Chen, "NLOS error identification and range approximation technique in cellular networks," in *Proc. of the 4th International Conference on Wireless Communications, Networking and Mobile Computing*, Oct. 2008, pp. 1–4.

[250] S. Venkatraman, J. Caffery, and H. R. You, "Location using LOS range estimation in NLOS environments," in *IEEE 55th Vehicular Technology Conference*, vol. 2, May 2002, pp. 856–860.

[251] S. Al-Jazzar and J. Caffery, "NLOS mitigation method for urban environments," in *IEEE 60th Vehicular Technology Conference*, vol. 7, Sep. 2004, pp. 5112–5115.

[252] R. Casas, A. Marco, J. J. Guerrero, and J. Falcó, "Robust estimator for non-line-of-sight error mitigation in indoor localization," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, no. 1, p. 043429, Apr. 2006.

[253] N. Alsindi, C. Duan, J. Zhang, and T. Tsuboi, "NLOS channel identification and mitigation in ultra wideband ToA-based wireless sensor networks," in *Proc. of the 6th Workshop on Positioning, Navigation and Communication*, Mar. 2009, pp. 59–66.

[254] K. Yu and Y. J. Guo, "Statistical NLOS identification based on AOA, TOA, and signal strength," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 274–286, Jan. 2009.

[255] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Non-line-of-sight identification and mitigation using received signal strength," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1689–1702, Mar. 2015.

[256] M. Bocquet, C. Loyez, and A. Benlarbi-Delai, "Using enhanced-TDOA measurement for indoor positioning," *IEEE Microwave and Wireless Components Letters*, vol. 15, no. 10, pp. 612–614, Oct. 2005.

[257] W. Xu, Z. Wang, and S. A. Zekavat, "Non-line-of-sight identification via phase difference statistics across two-antenna elements," *IET Communications*, vol. 5, no. 13, pp. 1814–1822, Sep. 2011.

[258] S. Maranò, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.

[259] S. Tian, L. Zhao, and G. Li, "A support vector data description approach to NLOS identification in UWB positioning," *Mathematical Problems in Engineering*, vol. 2014, May 2014.

[260] T. V. Nguyen, Y. Jeong, H. Shin, and M. Z. Win, "Machine learning for wideband localization," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 7, pp. 1357–1380, Jul. 2015.

[261] M. Tabaa, C. Diou, R. Saadane, and A. Dandache, "LOS/NLOS identification based on stable distribution feature extraction and SVM classifier for UWB on-body communications," *Procedia Computer Science*, vol. 32, Supplement C, pp. 882–887, 2014.

[262] H. Wymeersch, S. Marano, W. M. Gifford, and M. Z. Win, "A machine learning approach to ranging error mitigation for UWB localization," *IEEE Transactions on Communications*, vol. 60, no. 6, pp. 1719–1728, Jun. 2012.

[263] I. Guvenc, C. C. Chong, and F. Watanabe, "NLOS identification and mitigation for UWB localization systems," in *Proc. of the IEEE Wireless Communications and Networking Conference*, Mar. 2007, pp. 1571–1576.

[264] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines And Other Kernel-based Learning Methods*. Cambridge University Press, Mar. 2000.

[265] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[266] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451–462, Nov. 2000.

[267] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, no. 2, pp. 103–130, Nov. 1997.

[268] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 20, no. 2, pp. 215–242, 1958.

[269] C. Gentile, A. J. Braga, and A. Kik, "A comprehensive evaluation of joint range and angle estimation in indoor ultrawideband location systems," *EURASIP Journal on Wireless Communications and Networking*, no. 1, Dec. 2008.

[270] P. J. Vial, B. J. Wysocki, and T. A. Wysocki, "An ultra wide band simulator using MATLAB/Simulink," in *Proc. of the 8th International Symposium on DSP and Communication Systems (DSPCS'2005)*, Dec. 2005, pp. 231–236.

[271] "Matlab statistics and machine learning toolbox," 2017, the MathWorks, Natick, MA, USA.

[272] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, Dec. 1945.

[273] C. Heffner, "Exploiting surveillance camera like a Hollywood hacker," presented at Black Hat USA, Feb. 2013. [Online]. Available: https://media.blackhat.com/us-13/US-13-Heffner-Exploiting-Network-Surveillance-Cameras-Like-A-Hollywood-Hacker-WP.pdf

[274] "In the Matter of TRENDnet, Inc." F.T.C. File No. 122-3090, Federal Trade Commission, Jan. 2014. [Online]. Available: https://www.ftc.gov/system/files/documents/cases/140207trendnetcmpt.pdf

[275] D. Schneider. (2015, Aug.) Jeep hacking 101. [Online]. Available: https://spectrum.ieee.org/cars-that-think/transportation/systems/jeep-hacking-101

[276] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, Jul. 2017.

[277] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai botnet," in *Proceedings of the 26th USENIX Security Symposium*, Aug. 2017.

[278] H. Sinanović and S. Mrdovic, "Analysis of Mirai malicious software," in *25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sep. 2017, pp. 1–5.

[279] (2017, Jan.) Cybersecurity vulnerabilities identified in St. Jude Medical's implantable cardiac devices and Merlin@home transmitter: FDA safety communication. U.S. Food and Drug Administration. [Online]. Available: https://www.fda.gov/MedicalDevices/Safety/AlertsandNotices/ucm535843.htm

[280] (2017, Feb.) St. Jude Merlin@home transmitter vulnerability (Update A). Advisory ICSMA-17-009-01A. Industrial Control Systems Cyber Emergency Response Team. [Online]. Available: https://ics-cert.us-cert.gov/advisories/ICSMA-17-009-01A

[281] S. Babar, P. Mahalle, A. Stango, N. Prasad, and R. Prasad, "Proposed security model and threat taxonomy for the Internet of Things (IoT)," in *Proceedings of the Third International Conference on Network Security and Applications*, Jul. 2010, pp. 420–429.

[282] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, Jul. 2013.

[283] A. Pole, "How Target gets the most out of its guest data to improve marketing ROI," in *Predictive Analytics World conference*, Oct. 2010. [Online]. Available: https://www.predictiveanalyticsworld.com/patimes/how-target-gets-the-most-out-of-its-guest-data-to-improve-marketing-roi/6815/

[284] S. Babar, A. Stango, N. Prasad, J. Sen, and R. Prasad, "Proposed embedded security framework for Internet of Things (IoT)," in *Proceedings of the 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*, Feb. 2011, pp. 1–5.

[285] "Security," TR-0008-V2.0.0, oneM2M, Tech. Rep., Aug. 2016.

[286] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards," *Proceedings of the 18th USENIX Security Symposium*, pp. 1–16, Aug. 2009.

[287] "Understanding the role of connected devices in recent cyber attacks," U.S. Congress hearing, House of Representatives, Subcommittee on Communications and Technology and Subcommittee on Commerce, Manufacturing, and Trade, Nov. 2016. [Online]. Available: https://energycommerce.house.gov/hearings/understanding-role-connected-devices-recent-cyber-attacks/

[288] S. Tenaglia and J. Tanen, "Breaking BHAD: Abusing Belkin home automation devices," presented at Black Hat Europe, Nov. 2016.

[289] M. Muller, "IoT security: The ugly truth," presented at the IoT Security Summit, May 2015.

[290] *ZigBee Specification*, Document 053474r20, ZigBee Alliance Std., Rev. 20, Sep. 2012.

[291] *IEEE Standard for Low-Rate Wireless Networks*, IEEE 802.15.4-2015, IEEE Std., Apr. 2016, (accessed on Nov 28, 2016). [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=7460873

[292] T. Zillner and S. Strobl, "ZigBee exploited: The good, the bad, and the ugly," presented at Black Hat USA, Aug. 2015.

[293] *ZigBee Light Link Standard*, Document 11-0037-10, ZigBee Alliance Std., Rev. 1.0, Apr. 2012.

[294] P. Morgner, S. Mattejat, Z. Benenson, C. Müller, and F. Armknecht, "Insecure to the touch: Attacking ZigBee 3.0 via touchlink commissioning," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, Jul. 2017, pp. 230–240.

[295] T. Zillner, "ZigBee smart homes: a hacker's open house," presented at the CRESTcon Conference, May 2016.

[296] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of Bluetooth Low Energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11 734–11 753, Aug. 2012.

[297] A. K. Das, P. H. Pathak, C.-N. Chuah, and P. Mohapatra, "Uncovering privacy leakage in BLE network traffic of wearable fitness trackers," in *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*. ACM, Feb. 2016, pp. 99–104.

[298] A. Y. Lindell, "Attacks on the pairing protocol of Bluetooth v2.1," presented at Black Hat USA, Jun. 2008.

[299] A. Y. Lindell, "Comparison-based key exchange and the security of the numeric comparison mode in Bluetooth v2.1," in *Topics in Cryptology – The Cryptographers' Track at the RSA Conference 2009*. Springer Berlin Heidelberg, Apr. 2009, pp. 66–83.

[300] J. Barnickel, J. Wang, and U. Meyer, "Implementing an attack on Bluetooth 2.1+ secure simple pairing in passkey entry mode," in *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, Jun. 2012, pp. 17–24.

[301] M. Ryan, "Bluetooth: With low energy comes low security," in *7th USENIX Workshop on Offensive Technologies*. Washington, D.C.: USENIX Association, Aug. 2013. [Online]. Available: https://www.usenix.org/conference/woot13/workshop-program/presentation/Ryan

[302] W. K. Zegeye, "Exploiting Bluetooth Low Energy pairing vulnerability in telemedicine," in *Proceedings of the International Telemetering Conference*. International Foundation for Telemetering, Oct. 2015.

[303] T. Rosa, "Bypassing passkey authentication in Bluetooth Low Energy," Cryptology ePrint Archive, Report 2013/309, Tech. Rep., May 2013. [Online]. Available: https://eprint.iacr.org/2013/309

[304] A. Hilts, C. Parsons, and J. Knockel, "Every step you fake: A comparative analysis of fitness tracker privacy and security," Open Effect, Tech. Rep., Apr. 2016. [Online]. Available: https://openeffect.ca/reports/Every_Step_You_Fake.pdf

[305] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 packets over IEEE 802.15.4 networks," RFC 4944, Sep. 2007.

[306] G. Mulligan, "The 6LoWPAN architecture," in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets)*. ACM, 2007, pp. 78–82.

[307] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Mar. 2012.

[308] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," in *Proceedings of the International Conference on Pervasive Computing (ICPC)*, Jan. 2015, pp. 1–6.

[309] A. Rghioui, A. Khannous, and M. Bouhorma, "Denial-of-Service attacks on 6LoWPAN-RPL networks: Issues and practical solutions," *Journal of Advanced Computer Science and Technology*, vol. 3, no. 2, pp. 143–153, 2014.

[310] A. Mayzaud, R. Badonnel, and I. Chrisment, "A taxonomy of attacks in RPL-based Internet of Things," *International Journal of Network Security*, vol. 18, no. 3, pp. 459–473, May 2016.

[311] A. E. Yegin and Z. Shelby, "CoAP Security Options," Internet Engineering Task Force, Internet-Draft draft-yegin-coap-security-options-00, Oct. 2011. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-yegin-coap-security-options-00

[312] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," RFC 7252, Jun. 2014.

[313] J. Mattsson and F. Palombini, "Comparison of CoAP Security Protocols," Internet Engineering Task Force, Internet-Draft draft-ietf-lwig-security-protocol-comparison-01, Jul. 2018. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-lwig-security-protocol-comparison-01

[314] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, "6LoWPAN fragmentation attacks and mitigation mechanisms," in *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, Apr. 2013, pp. 55–66.

[315] A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo, "6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1189–1212, May 2012.

[316] G. Simmons, "A survey of information authentication," *Proceedings of the IEEE*, vol. 76, no. 5, pp. 603–620, May 1988.

[317] N. Yang, L. Wang, G. Geraci, M. Elkashlan, J. Yuan, and M. D. Renzo, "Safeguarding 5g wireless communication networks using physical layer security," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 20–27, Apr. 2015.

[318] X. Wang, P. Hao, and L. Hanzo, "Physical-layer authentication for wireless security enhancement: current challenges and future developments," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 152–158, Jun. 2016.

[319] E. Jorswieck, S. Tomasin, and A. Sezgin, "Broadcasting into the uncertainty: Authentication and confidentiality by physical-layer processing," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1702–1724, Oct. 2015.

[320] Y. Liu, H. H. Chen, and L. Wang, "Physical layer security for next generation wireless networks: Theories, technologies, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 347–376, 2017.

[321] L. Xiao, L. J. Greenstein, N. Mandayam, and W. Trappe, "Channel-based spoofing detection in frequency-selective Rayleigh channels," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5948–5956, Dec. 2009.

[322] F. He, H. Man, D. Kivanc, and B. McNair, "EPSON: enhanced physical security in OFDM networks," in *Proceeding of the IEEE International Conference on Communications (ICC)*, Jun. 2009, pp. 1–5.

[323] X. Wu and Z. Yang, "Physical-layer authentication for multi-carrier transmission," *IEEE Communications Letters*, vol. 19, no. 1, pp. 74–77, Jan. 2015.

[324] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Mimo-assisted channel-based authentication in wireless networks," in *2008 42nd Annual Conference on Information Sciences and Systems*, Mar. 2008, pp. 642–646.

[325] P. Baracca, N. Laurenti, and S. Tomasin, "Physical layer authentication over mimo fading wiretap channels," *IEEE Transactions on Wireless Communications*, vol. 11, no. 7, pp. 2564–2573, Jul. 2012.

[326] W. Hou, X. Wang, J. Y. Chouinard, and A. Refaey, "Physical layer authentication for mobile systems with time-varying carrier frequency offsets," *IEEE Transactions on Communications*, vol. 62, no. 5, pp. 1658–1667, May 2014.

[327] J. Liu and X. Wang, "Physical layer authentication enhancement using two-dimensional channel quantization," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 4171–4182, Jun. 2016.

[328] S. Rumpel, A. Wolf, and E. A. Jorswieck, "Physical layer based authentication without phase detection," in *2016 50th Asilomar Conference on Signals, Systems and Computers*, Nov. 2016, pp. 1675–1679.

[329] A. Ferrante, N. Laurenti, C. Masiero, M. Pavon, and S. Tomasin, "On the error region for channel estimation-based physical layer authentication over rayleigh fading," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 941–952, May 2015.

[330] G. Caparra, M. Centenaro, N. Laurenti, S. Tomasin, and L. Vangelista, "Wireless physical-layer authentication for the Internet of Things," in *Information Theoretic Security and Privacy of Information Systems*, R. F. Schaefer, H. Boche, A. Khisti, and H. V. Poor, Eds. Cambridge University Press, 2017, pp. 390–418.

[331] G. Caparra, M. Centenaro, N. Laurenti, S. Tomasin, and L. Vangelista, "Energy-based anchor node selection for IoT physical layer authentication," in *Proceedings of the IEEE International Conference on Communications (ICC)*, May 2016.

[332] P. C. Pinto, J. Barros, and M. Z. Win, "Secure communication in stochastic wireless networks;part i: Connectivity," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 125–138, Feb. 2012.

[333] H. ElSawy, E. Hossain, and M. Haenggi, "Stochastic geometry for modeling, analysis, and design of multi-tier and cognitive cellular wireless networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 996–1019, 2013.

[334] O. O. Koyluoglu, C. E. Koksal, and H. E. Gamal, "On secrecy capacity scaling in wireless networks," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3000–3015, May 2012.

[335] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electronics Letters*, vol. 27, no. 23, pp. 2145–2146, Nov. 1991.

[336] M. J. Marsan, G. C. Hess, and S. S. Gilbert, "Shadowing variability in an urban land mobile environment at 900 mhz," *Electronics Letters*, vol. 26, no. 10, pp. 646–648, May 1990.

[337] J. Weitzen and T. J. Lowe, "Measurement of angular and distance correlation properties of log-normal shadowing at 1900 mhz and its application to design of pcs systems," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 2, pp. 265–273, Mar. 2002.

[338] A. Goldsmith, *Wireless Communications*. New York, NY, USA: Cambridge University Press, 2005.

[339] V. Witkovský, "Numerical inversion of a characteristic function: An alternative tool to form the probability distribution of output quantity in linear measurement models," *Acta IMEKO*, vol. 5, no. 3, pp. 32–44, Nov. 2016.