UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament de Teoria del Senyal
i Comunicacions

Universitat Politècnica de Catalunya (UPC)

Doctoral Thesis

Towards Zero-Power Wireless Machine-to-Machine Networks

Author:
Francisco Vázquez-Gallego

Supervisors:
Dr. Jesus Alonso-Zarate
Dr. Luis Alonso

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

Department of Signal Theory and Communications

May 2016

Abstract

This thesis aims at contributing to overcome two of the main challenges for the deployment of highly dense wireless M2M networks in data collection scenarios for the Internet of Things: the management of massive numbers of end-devices that attempt to get access to the wireless channel; and the need to extend the network lifetime to reduce maintenance costs. In order to solve these challenges, two complementary strategies are considered. Firstly, the thesis focuses on the design, analysis and performance evaluation of random and hybrid access protocols that can handle abrupt transitions in the traffic load and minimize the energy consumption devoted to communications. And secondly, the use of energy harvesting (EH) is considered in order to provide the network with unlimited lifetime. To this end, the second part of the thesis focuses on the design and analysis of EH-aware MAC protocols, and proposes new performance metrics that take the variability and the unpredictability of the harvested energy into account.

While the Frame Slotted-ALOHA (FSA) protocol has been traditionally adopted in star topology networks for data collection, results show that FSA-based protocols lack of scalability and present synchronization problems as the network density increases. Indeed, the frame length of FSA must be adjusted to the number of contenders, which may be complex to attain in dense networks with large and dynamic number of end-devices. In order to overcome these issues, a tree splitting-based random access protocol, referred to as Low Power Contention Tree-based Access (LP-CTA), is proposed in the first part of this thesis. In LP-CTA, the frame length can be very short and fixed, which facilitates synchronization and provides better network scalability than FSA.

While LP-CTA uses data slots for contention, it is possible to use short access requests in minislots, where collisions are resolved using tree splitting, and avoid the contention in data. Since these minislots can be much shorter than the duration of a data packet, the performance can be improved. The Low Power Distributed Queuing (LP-DQ) protocol proposed in this thesis is based on this idea. LP-DQ combines tree splitting with the logic of two distributed queues that manage the contention resolution and the collision-free data transmission. Results show that LP-DQ outperforms LP-CTA and FSA in terms of delay and energy efficiency, and it relaxes the need to know the size of the network and adapts smoothly to any change in the number of end-devices.

The approach of LP-DQ is convenient when the messages transmitted by each end-device fit in one single slot, however, if the end-devices generate long messages that have to be fragmented, it is better to add a reservation mechanism in order to boost the performance. In this sense, the LPR-DQ protocol is proposed as an extension of LP-DQ where the concept of reservation is integrated to allow the end-devices reserve as many collision-free slots as needed.

The second part of the thesis is devoted to the integration of the MAC layer with the use of energy harvesting. While energy harvesters can theoretically provide infinite lifetime, the variability and fluctuations of the harvested energy may cause the end-devices to enter temporarily in energy shortage. This fact is considered for the design of EH-aware MAC protocols and three performance metrics are proposed: the probability of delivery, the data delivery ratio and the time efficiency.

While previous research works on data collection networks with energy harvesting focus on Dynamic FSA (DFSA), the EH-CTA protocol is proposed in this thesis as an adaptation of LP-CTA that takes the energy harvesting process into account. Results show that EH-CTA outperforms DFSA if the energy threshold for an end-device to become active is properly configured. In addition, while DFSA needs to adapt the frame length dynamically, EH-CTA uses a fixed frame length, thus facilitating scalability and synchronization.

Finally, the EH-RDFSA and EH-DQ protocols are proposed for scenarios where data must be fragmented. EH-RDFSA is a combination of RFSA and DFSA, and EH-DQ is an extension of LPR-DQ. Results show that EH-DQ uses a short and fixed frame length, regardless of the number of contenders, and outperforms EH-RDFSA. Taking that into account, EH-DQ is a promising alternative for highly dense data collection networks with EH where end-devices generate bursts of data fragmented in multiple packets.

Dedicated to my family, especially to my wonderful daughters, ***Alicia*** *and* ***Marina****, and my wife,* ***Ruth****, who are an unconditional source of support, love and inspiration.*

Acknowledgements

First of all, I would like to express my sincere gratitude to my advisors, Dr. Jesus Alonso Zarate and Dr. Luis Alonso, for the immense amount of time devoted to discuss on my research. Their vision, inspiring comments and their ability to motivate have been of vital importance for the success of this thesis.

Some parts of the work presented in this thesis could not have been possible without the collaboration of some students: Iván Balboteo, Sergio Luz, Eduardo Sánchez, Oriol Bardina, Ana Maria Mandalari, Orazio Briante, Shuan Wu and Ailan He. I would like to take this opportunity to thank you for all what I have learned from you and for your valuable contributions on analysis, simulations and experimental work. Likewise, I am grateful for the efforts of Giacomo Genovese and Vasilis Karagiannis, whose great works on experimental platforms for the Internet of Things were the seed of the IoT testbed that is being used to experimentally evaluate the protocols proposed in this thesis.

Also, I would like to thank Dr. Stephan Spfletschinger, Dr. Joan Bas, Marc Rietti, Ana Cristina Hernndez and Giovanni Internodato for their collaboration and valuable contributions on the evaluation of Successive Interference Cancellation techniques.

Special thanks go to Dr. Pere Tuset for his collaboration on the experimental evaluation of the LP-DQ and FSA protocols for low-power wireless M2M networks while he was visiting our research group at the M2M department of CTTC. Likewise, I would like to thank Dr. Raúl Palacios for his support and inspiring comments, and also for the good time shared in Sidney and Atlanta.

I would like to extend my gratitude to all my friends, who, directly or not, consciously or not, have contributed to this thesis. Special thanks go to Héctor, Antonio, Daniel and Federico, who are always there to listen and to discuss on research and technology.

As in every professional step in my life, this achievement could not have been possible without my family. I would like to thank my parents for teaching me the meaning of effort and dedication, and for their unconditional support, specially within the last three years of my life.

I would like to dedicate this work to my dear uncle Tomás, who left us a few weeks before the lecture of this thesis.

Finally, I would like to thank my wife, Ruth, for her love, for her strength, and for her support in all my vital and professional challenges and adventures. I would like to finnish these acknowledgments by dedicating this work to Ruth and my lovely daughters, Alicia and Marina. You are the most important persons in my life.

Contents

List of Figures

List of Tables

Abbreviations

ACK	**Ack**nowledgement
AMR	**A**utomatic **M**eter **R**eading
ARS	**A**ccess **R**equest **S**equence
BAN	**B**ody **A**rea **N**etwork
BT	**B**inary **T**ree
CCA	**C**lear **C**hannel **A**ssessment
CRA	**C**ontention **R**esolution **A**lgorithm
CRDSA	**C**ontention **R**esolution **D**iversity **S**lotted-**A**LOHA
CRQ	**C**ollision **R**esolution **Q**ueue
CRI	**C**ontention **R**esolution **I**nterval
CSMA	**C**arrier **S**ense **M**ultiple **A**ccess
DCR	**D**ata **C**ollection **R**ound
DDR	**D**ata **D**elivery **R**atio
DFSA	**D**ynamic **F**rame **S**lotted-**A**LOHA
D-FSA	**D**iversity **F**rame **S**lotted-**A**LOHA
DQ	**D**istributed **Q**ueuing
DQBAN	**D**istributed **Q**ueuing **B**ody **A**rea **N**etworks
DQCA	**D**istributed **Q**ueuing **C**ollision **A**voidance
DQCOOP	**D**istributed **Q**ueuing **Coop**erative
DQMAN	**D**istributed **Q**ueuing for **M**obile **A**d hoc **N**etworks
DQRAP	**D**istributed **Q**ueuing **R**andom **A**ccess **P**rotocol
DTQ	**D**ata **T**ransmission **Q**ueue
D2D	**D**evice-to-**D**evice
EG-DFSA	**E**nergy **G**roup **D**ynamic **F**rame **S**lotted-**A**LOHA
EH	**E**nergy **H**arvesting
EH-CTA	**E**nergy **H**arvesting-aware **C**ontention **T**ree-based **A**ccess

EH-DFSA	**E**nergy **H**arvesting-aware **D**ynamic **F**rame **S**lotted-**A**LOHA
EH-DQ	**E**nergy **H**arvesting-aware Low Power **D**istributed **Q**ueuing with Reservation
EH-RDFSA	**E**nergy **H**arvesting-aware **R**eservation **D**ynamic **F**rame **S**lotted-**A**LOHA
ESD	**E**nergy **S**torage **D**evice
ETSI	**E**uropean **T**elecommunications **S**tandards **I**nstitute
FSA	**F**rame **S**lotted-**A**LOHA
FBP	**F**eed**b**ack **P**acket
HTC	**H**uman **T**ype-**C**ommunication
IC	**I**nterference **C**ancellation
ID	**Id**entification
IFS	**I**nter **F**rame **S**pace
IoT	**I**nternet **o**f **T**hings
IRSA	**I**rregular **R**epetition **S**lotted-**A**LOHA
LP-CTA	**L**ow **P**ower **C**ontention **T**ree-based **A**ccess
LP-DQ	**L**ow **P**ower **D**istributed **Q**ueuing
LPR-DQ	**L**ow **P**ower **D**istributed **Q**ueuing with **R**eservation
LPWA	**L**ow **P**ower **W**ide **A**rea
LTN	**L**ow **T**hroughput **N**etwork
MAC	**M**edium **A**ccess **C**ontrol
M2M	**M**achine-to-**M**achine
MTC	**M**achine **T**ype-**C**ommunication
QT	**Q**uery **T**ree
RFD	**R**equest **F**or **D**ata
RFID	**R**adio **F**requency **Id**entification
RFSA	**R**eservation **F**rame **S**lotted-**A**LOHA
R-CRDSA	**R**eservation **C**ontention **R**esolution **D**iversity **S**lotted-**A**LOHA
SIC	**S**uccessive **I**nterference **C**ancellation
SHM	**S**tructural **H**ealth **M**onitoring
TDMA	**T**ime **D**ivision **M**ultiple **A**ccess
WLAN	**W**ireless **L**ocal **A**rea **N**etwork
WPAN	**W**ireless **P**ersonal **A**rea **N**etwork
WSN	**W**ireless **S**ensor **N**etwork

Chapter 1

Introduction

1.1 Overview of Wireless Machine-to-Machine Networks

The Internet of Things (IoT) [1][2] is a paradigm which emphasizes the vision of a global infrastructure of networked physical objects. The IoT, or the Internet of Everything, will connect things with things, people with people, and people with things, facilitating a rich set of revolutionary services and applications such as Smart Grids, Smart Cities, Smart Buildings, and eHealth, among many others. The IoT will provide new opportunities, experiences, and businesses for people and countries around the world, and it will create a revolution in the way we run industries, cities, etc.

In order to facilitate the development of the IoT, the next generation of wireless networks (5G and beyond) will need to encompass highly demanding Human-Type Communications (HTC) with Machine-to-Machine (M2M) or Machine-Type Communications (MTC). In 2020, it is envisioned that a factor of 10 to 100 more MTC-devices than personal mobile devices will be connected to cellular networks [3], thus posing severe challenges in the design of communication networks. While current communication standards were conceived for human-based data traffic, with relatively few devices at high data rates, MTC-devices generate small amounts of data with very diverse traffic patterns (e.g., triggered by events, periodically, or sporadic). The massive deployment of MTC-devices may cause congestion in the access networks. Consequently, it is necessary to offload networks and ensure that different types of users (humans and machines) can coexist. An alternative is to investigate solutions based on primary cellular networks which are complemented with short-range radio technologies such as low-power WiFi,

Zigbee, etc. In this sense, the M2M architecture proposed by ETSI [4] considers that connectivity between MTC-devices (also referred to as end-devices) and the access network must be provided by a combination of long range wireless networks, with direct connection using a cellular radio interface at the end-device, and M2M area networks, which connect the end-devices through one (or more) gateways that concentrate traffic to and from the access network. The gateways also perform protocol translation, traffic aggregation and authentication functionalities, among others. M2M area networks can be based on a wide range of existing wireless technologies such as Wireless Personal Area Networks (WPAN), Body Area Networks (BAN), Radio Frequency Identification (RFID), Wireless Local Area Networks (WLAN) and Wireless Sensor Networks (WSN).

Lately, the Low Power Wide Area (LPWA) networks are emerging as disruptive solutions to facilitate MTC ensuring low cost, long range and low energy consumption [5]. LPWA networks exploit narrowband communications and attain extremely high sensitivity at the receiver side. Companies like SigFox [6] or initiatives like Weightless [7] and the LoRa Alliance [8] are struggling to define new standards for M2M connectivity over license-free bands. Weigthless and the LoRa Alliance have defined their own proprietary protocol and SigFox has been working with ETSI to define a specification for Low Throughput Networks (LTN) [9] operating in the 868MHz band and using ALOHA and simple modulations (e.g., FSK, GMSK).

Although the use of M2M area networks and LPWA networks are good alternatives to 3GPP Standards, these networks are still not ready to handle thousands of simultaneous end-devices connected to the gateways or access points. They use simple Medium Access Control (MAC) protocols that compromise the scalability of these solutions, and they will suffer from congestion and high energy consumption as the IoT increases its ubiquity.

1.2 Motivation

The work presented in this thesis focuses on data collection applications using wireless M2M area networks and LPWA networks. Some examples of such applications, among many others, are Asset Tracking, using active RFID to accurately track the real-time location of assets; Automatic Meter Reading (AMR), where a gateway collects readings from electricity, water or gas meters; and Structural Health Monitoring (SHM), where a large number of wireless sensors measure vibrations in civil infrastructures. In these

networks, hundreds or thousands of end-devices transmit data to a gateway periodically or triggered by an event. These networks suffer abrupt transitions in the traffic load, which may change from idle-to-saturation conditions (also referred to as delta traffic or batch arrival [10]) when a huge number of end-devices have new data ready to transmit in a given time. Although the amount of data generated by each end-device may be relatively low, the total number of end-devices that can attempt to get access to the wireless channel simultaneously can be potentially larger than the one manageable by traditional MAC protocols.

In addition, the high density of these networks will make very difficult and expensive to physically handle and maintain a massive number of end-devices. The end-devices need to be low-cost and must operate autonomously for years (or even decades) with none or very limited access to energy sources. Therefore, energy efficiency is essential in order to ensure extended lifetime and guarantee zero maintenance costs in battery recharge or replacements. Three complementary strategies to extend the network lifetime consist in reducing the energy consumption required for communications [11], using energy harvesters that collect the needed energy from the environment [12][13], and implementing passive or low-power operation techniques such as wake-up radios [14][15].

It is widely known that the MAC layer plays an important role in the energy consumption needed for communications [11][16]. The MAC layer coordinates the access to a shared medium and determines when the radio transceivers are powered on to transmit, listen, or receive. Therefore, energy efficient MAC protocols are needed to reduce the energy waste due to packet collisions, over-hearing, idle listening, and protocol overheads [16]. There are two main categories of MAC protocols which are relevant for M2M [17]: reservation-based and random access protocols. Reservation-based protocols define a schedule that allows each end-device to transmit data without collisions, e.g., using a specific time-slot in Time Division Multiple Access (TDMA). Unfortunately, these protocols require the knowledge of the network topology, which may be difficult to attain in dense networks with an unknown, huge, and dynamic number of end-devices due to the delay and energy required to update the topology knowledge and to maintain a collision-free schedule. On the contrary, random access protocols [17], such as ALOHA [18] or Carrier Sense Multiple Access (CSMA), do not require knowledge of the network topology, and their simplicity and distributed operation make them ideal for

low-cost and simple end-devices. For this reason, many standards for short-range communications rely on random access, e.g., CSMA in the IEEE 802.11 and IEEE 802.15.4 standards, ALOHA in the EN 13757-4 standard (Wireless M-bus) for AMR, or Frame Slotted-ALOHA (FSA) in the ISO/IEC-18000-7 standard for asset tracking. However, it is well known that conventional random access protocols suffer from congestion and degraded performance [19], in terms of throughput and energy consumption, when the traffic load increases or when the number of end-devices is huge (hundreds or thousands), due to the high probability of collision. Consequently, the design of energy-efficient MAC protocols, which adapt automatically to abrupt transitions in the traffic load and to a variable number of end-devices (10s to 1000s), is one of the key challenges to prolong the lifetime of future M2M networks for data collection. The first part of this thesis is aimed in this direction, focusing on the design, analysis, and performance evaluation of MAC protocols to manage the access of a huge number of end-devices, ensuring high energy efficiency and scalability.

An alternative to improve the performance of random access is to use a Collision Resolution Algorithm (CRA) [20, 21], such as the tree-splitting algorithm [20], which resolve collisions by iteratively splitting a group of contenders into smaller sub-groups in order to reduce the probability of collision. CRAs are typically implemented using data slots to resolve contention [22, 23], thus not attaining all the potential gains that this approach can offer. Instead of using data for contention, it is possible to separate contention from data transmission through the use of contention-based access requests using minislots. Since these minislots can be much shorter than the duration of a data packet, the performance of the network can be boosted. This approach is the foundation of the Distributed Queuing (DQ) access, which is a hybrid access protocol that combines a CRA with the logic of two distributed queues to manage the contention resolution and the collision-free data transmission, respectively. Existing performance analyses of DQ have been done in steady-state conditions, assuming that end-devices generate packets following a Poison distribution. Under these conditions, DQ achieves maximum performance, close to the theoretical maximum capacity of the channel, using 3 minislots regardless of the number of contenders. This makes CRAs and DQ very appealing in dense M2M networks where the number of end-devices is large and is unknown a priori. Therefore, the MAC protocols analyzed and proposed in the first part of the thesis are classified into three groups:

1. Random access protocols based on Frame Slotted-ALOHA.
2. Random access protocols based on Contention Resolution Algorithms.
3. Hybrid access protocols based on Distributed Queuing.

While the lifetime of battery-powered systems is limited by the capacity of the batteries, the use of energy harvesting techniques enables the implementation of fully autonomous systems that collect the needed energy from the environment (e.g., from sunlight, thermoelectric, mechanical, or electromagnetic energy sources), convert it into electrical energy, and store it in a rechargeable battery or capacitor. The development of commercial M2M networks with energy harvesting capabilities has received attention in recent years. As an example, EnOcean [24] manufactures batteryless sensing end-devices based on the ALOHA protocol. Despite energy harvesters theoretically provide infinite lifetime, they may not guarantee fully continuous operation of the end-devices. This is due to the fact that the energy sources are highly variable and unpredictable. Thus, the energy available over a short period of time may not be sufficient to ensure the operation of an end-device, which may enter in temporary energy shortage until it gets enough energy. Consequently, the design and analysis of MAC protocols for future M2M networks need to integrate the novel features of energy harvesting, and new metrics must be defined in order to evaluate the protocols performance over the characteristics of the energy sources, the capacity of the energy storage devices, etc. While the performance of random access has been studied very extensively in many research works related to the MAC layer, the study of MAC protocols on networks with energy harvesting is a much more recent area of research. Previous research works mainly focus on traditional slotted-ALOHA [25], FSA [26], Dynamic FSA (DFSA) [27–29], and CSMA [30][31]. However, there is a need for the design and analysis of innovative MAC protocols for dense M2M wireless networks with energy-harvesting capabilities. The second part of this thesis is aimed in this direction, focusing on the design, analysis, and performance evaluation of energy-harvesting-aware MAC protocols that potentially enable perpetual operation of M2M networks. The EH-aware MAC protocols proposed in the second part of the thesis are classified into three groups:

1. EH-aware random access protocols based on Dynamic Frame Slotted-ALOHA.
2. EH-aware random access protocols based on Contention Resolution Algorithms.
3. EH-aware hybrid access protocols based on Distributed Queuing.

The remainder of this chapter is organized as follows. A review of the state-of-the-art in the field of MAC protocols for dense networks and energy harvesting is presented in Section 1.3. Then, the main contributions and the organization of this thesis are described in detail in Section 1.4. Section 1.5 lists the publications related to the dissemination of the contributions of this thesis, and finally Section 1.6 lists other research contributions which are not included in this thesis.

1.3 State-of-the-Art

The aim of this section is to provide an exhaustive review of the state-of-the-art of the topics related to this thesis so that the contributions herein presented can be properly stated and justified. Firstly, Section 1.3.1 overviews the most representative works related to random and hybrid access protocols for dense wireless M2M networks. And secondly, Section 1.3.2 summarizes the main contributions in the field of MAC protocols for wireless networks with energy harvesters, and the works related to analytical models of the energy harvesting process.

1.3.1 MAC Protocols for Dense Wireless M2M Networks

A vast amount of MAC protocols have been designed within the context of wireless M2M networks. The most representative random and hybrid access protocols related to the work presented in this thesis can be classified into four groups:

1. Protocols based on Frame Slotted-ALOHA.
2. Reservation Frame Slotted-ALOHA.
3. Protocols based on Contention Resolution Algorithms.
4. Protocols based on Distributed Queuing Access.

The four groups of protocols are reviewed in the following sections.

1.3.1.1 Frame Slotted-ALOHA based Access Protocols

The frame slotted ALOHA (FSA) protocol has been identified in the past as a good approach to handle the delta traffic in data collection scenarios due to its simplicity and good performance in terms of delay and throughput. As an example, FSA was adopted

in the ISO/IEC 18000-7 standard, which is targeted at active RFID systems. In FSA [32], the contention process is carried out in a sequence of time frames divided into slots. Each device chooses one of the slots of a frame randomly in order to transmit its data. The process is repeated, frame after frame, until all the devices have transmitted their data successfully. A number of research works in RFID have analysed the performance of FSA in terms of average delay and energy consumption.

Regarding performance in terms of delay, the work in [33] has computed the optimal frame length (i.e., the number of access slots per frame), for different numbers of tags, that reduces the number of frames required to identify all the tags. This work assumes that the number of contending tags is known a priori and is constant during the identification process. The number of frames needed to identify all the tags has been calculated with computer-based simulations using the estimated optimal frame length. The authors in [34] have estimated the average number of frames required to identify a full set of tags using different frame lengths. The results show that with a given frame length, the number of frames to solve the contention increases exponentially with the number of tags.

Assuming that the number of tags cannot be known a priori, the algorithms proposed in [35–37] estimate the number of tags contending in one frame in order to select the optimal frame length to be used in subsequent frames. The works in [38, 39] show that FSA is optimal, in terms of delay or throughput, when the number of slots per frame is equal to the number of contending devices in every frame, which introduces the concept of dynamic FSA.

In terms of energy consumption, the works in [22, 39, 40] compare the energy consumed by the tags and the reader using several variants of FSA. The results show that, as in the case of the average delay, using a fixed frame length the energy consumption in FSA grows exponentially with the number of tags. However, they do not describe how the energy consumption changes using different frame lengths, and they do not determine if there exists an appropriate frame length that may minimize the energy consumption of the identification process. Although these works propose theoretical analysis based on probabilistic methods, they do not demonstrate by any means (e.g., computer-based simulation) the validity and accuracy of their derivations.

Recent research works [41–44] have adopted successive interference cancellation (SIC) techniques to resolve collisions and improve the performance of random access. The

work in [45] proposes diversity frame slotted-ALOHA (D-FSA) as a variation of slotted-ALOHA (SA). In D-FSA, devices are synchronized to a common time frame pattern where every frame is divided into a number of slots in which devices contend to transmit several replicas of each data packet. Results show that D-FSA provides slight throughput enhancement with respect to SA. A more efficient use of the packet replicas was provided by Contention Resolution Diversity Slotted-ALOHA (CRDSA) [41] and Irregular Repetition Slotted-ALOHA (IRSA) [42]. In CRDSA and IRSA, similarly to D-FSA, devices also transmit two or more replicas of each data packet in every frame at random slots. In addition, both protocols implement a SIC algorithm which allows decoding collided packets, thus increasing throughput with respect to D-FSA. The works in [43] and [44] also proposed variants of FSA with SIC for M2M and Radio Frequency Identification (RFID), respectively.

1.3.1.2 Reservation Frame Slotted-ALOHA

The reservation FSA protocol (RFSA) was proposed in [46] to improve the performance of FSA when devices generate messages which require fragmentation. A number of research works have proposed the use of RFSA in satellite communications [47–50] and inter-vehicle communication [51–53]. The contention process in RFSA is composed of a sequence of time frames divided into slots. When a device successfully transmits a packet in a slot of a given frame, that particular slot is reserved for the same device in subsequent frames until the last packet of the message is sent. As a result, the device has the equivalent of an assigned Time Division Multiple Access (TDMA) channel for as long as it has packets to send. The slots which are either empty or where a collision occurs remain available in subsequent frames for contention of other devices, just as in FSA.

The work in [49] has studied the performance of RFSA in terms of average message delay and channel throughput in satellite communications. For the purpose of the analysis, authors consider that each user (or device) generates messages according to a stationary Poisson process with rate λ messages per second, where each message consists of υ packets of fixed length, and υ is a random variable geometrically distributed with mean $\overline{\upsilon}$. Results show that RFSA adapts to the nature of the traffic. The maximum channel throughput of RFSA ranges from that of slotted-ALOHA when $\overline{\upsilon} = 1$, to that of fixed assigned TDMA channels when $\overline{\upsilon} \to \infty$.

The works in [48] and [50] have also analysed the dynamic behaviour of RFSA with multi-packet messages. The author assumes that each user generates messages according to a geometrically distributed process, and each message consists of a group of packets whose number is geometrically distributed. A Markovian model is formulated to analyse the throughput versus average message delay trade-off characteristic. Results show that the performance of RFSA is very similar to slotted-ALOHA when the mean number of packets per message tends to one. However, RFSA outperforms slotted-ALOHA, in terms of throughput and average message delay, when the mean number of packets per message increases.

The work in [53] has studied the performance of the RFSA protocol with priority (PR-ALOHA) proposed for inter-vehicle communication. Priority is introduced in RFSA by reserving specific time slots in the frame of RFSA exclusively for high priority traffic. Traffic is divided into either low or high priority (e.g., safety related messages or multimedia). When a user generates a normal priority message, it may contend for any empty slot which is not high priority. When the message is of high priority, the user may contend for any empty slot, including the slots not reserved for high priority traffic. As it could be expected, the simulation results show that as the number of slots reserved for high priority increases, the delay of the high priority messages decreases. However, the throughput of the low priority messages decreases and their delay increases.

The work in [54] proposes the Reservation-Contention Resolution Diversity Slotted ALOHA protocol (R-CRDSA) for satellite networks. R-CRDSA introduces the basic concept of CRDSA [41] in the RFSA protocol. In CRDSA, each user transmits several replicas of each packet in a frame using different randomly selected slots. The coordinator (e.g., satellite transponder) implements an Interference Cancellation (IC) algorithm that allows decoding data packets that were initially lost due to collision. When a packet is successfully decoded in a given slot, that slot is allocated to the successful user. This user will transmit the rest of the packets of its message in the allocated slot without any replicas in other slots. The authors assume multi-packet message arrivals at each user, where each message consists of a random number of packets with arbitrary distribution. Results show that when the average number of packets per message is one, the performance of R-CRDSA and RFSA are very similar to CRDSA and slotted-ALOHA, respectively. As the average number of packets increases, R-CRDSA achieves the highest

throughput. In addition, the differences in throughput between R-CRDSA and RFSA decrease as the mean number of packets per message increases.

1.3.1.3 Contention Resolution Algorithms

A promising strategy to improve the maximum stable throughput of random access protocols is to use a Collision Resolution Algorithm (CRA) as it was demonstrated in [20] and [21]. CRAs resolve collisions by organizing the retransmission of colliding packets in such a way that all packets are always successfully transmitted with finite delay. The basic CRA is the binary tree-splitting algorithm that was originally proposed in [20] and later analysed in [21, 55]. It iteratively splits a group of contending devices into smaller sub-groups to reduce collisions in an efficient manner. The delay and energy performance of the binary tree-splitting algorithm has been evaluated in different applications such as RFID. The two most widely used protocols in RFID are the Binary Tree (BT) [56] and the Query Tree (QT) protocol [22, 57]. Both protocols work by splitting the RFID tag identifications bitwise using queries from the reader until all the tags are identified. For example, in each request of QT, the reader or coordinator sends a query including a binary prefix. The tags with this binary prefix in their identification (ID) respond back in a single slot. In the case of a collision, the reader adds 1 bit (0 or 1) to the prefix and sends a new query with the longer prefix. The process is repeated until all tags successfully transmit their ID.

The work in [22] compares the energy consumed by the reader and the tags using QT and FSA. For the purpose of the analysis, authors derive the average number of frames required to identify all tags and the average number of tag responses, and they consider only the energy consumed in transmission and reception modes, but not in sleep. Results show that the energy consumed by the reader using QT is lower than using FSA. For the energy consumed by the tags, QT achieves better performance than FSA when the tags contend in all the frames. However, if the identified tags withdraw from contention in subsequent frames, FSA outperforms QT in terms of tags energy consumption.

The works in [58, 59] propose a generic m-ary tree-splitting algorithm to resolve collisions among devices that contend in a sequence of time frames divided into a number m of time slots where they transmit data. They compute the average number of contention frames required to resolve the contention, referred to as Collision Resolution Interval (CRI). The CRI is used in [58] to analyze the network throughput considering

that devices generate packets following a Poison distribution. In the first frame, every device randomly selects one of the slots to transmit. If two or more devices collide, i.e., they transmit in the same slot, a new frame is assigned only to the sub-group of devices that caused the collision in that slot. Therefore, new frames become available after each frame with one or more slots with collision. The process terminates when all devices have been able to successfully transmit their data packet.

The work in [57] presents an analysis of the energy consumed by tags using m-ary tree-splitting in RFID. The analysis neglects the energy consumed by the tags in reception and sleep mode. Results show that the energy consumed by one tag decreases when the frame length increases, and it increases logarithmically when the number of tags increases. The work in [23] evaluates the energy consumption of the reader and tags using an m-ary splitting algorithm by means of computer-based simulations and compute the energy savings with respect to QT.

Up to our knowledge, previous works in RFID focus on the minimization of the delay and compare the energy consumed by the tags and the reader using several variants of FSA [22, 40], binary tree-splitting [22, 56, 57], and m-ary tree-splitting algorithms [23, 57]. The results show that the energy consumption grows exponentially with the number of tags. However, these works do not describe how to maximize the energy efficiency during the identification process, and the analytical energy models are not complete for M2M devices, e.g., some authors consider only the energy consumed by the tags in transmission, but neither in reception nor in sleep states.

1.3.1.4 DQ: Distributed Queuing Access

While the tree-splitting algorithm implemented in [22, 23, 57, 60] uses data slots to resolve contention, it can be also used to handle channel access requests using minislots, which are actually shorter than data packets. This approach was implemented in the Distributed Queuing (DQ) access, which was first introduced in the DQ Random Access Protocol (DQRAP) [61] and DQ Collision Avoidance (DQCA) [62] protocols.

In the DQ access, the collision of data packets is avoided by separating access requests from the transmission of data. Therefore, since the duration of data packets are longer than an access request, DQ achieves greater energy efficiency. The basic idea of DQ is that devices request for channel access in a short contention window divided into

minislots, thus confining collisions to a part of the frame. Then, collisions are resolved by using the tree-splitting algorithm proposed in [20]. When a device succeeds transmitting its access request, it enters into a queue and waits for its turn to transmit data in a collision-free data slot.

Even though DQ was originally designed for cable TV distribution (DQRAP [61]), it has been adapted to different communication networks over the last years; to wired centralized networks (Extended DQRAP [63], Prioritized DQRAP [64]), satellite communications with long propagation delays (Interleaved DQRAP [65]), code-division multiple access for 3G cellular networks (DQRAP/CDMA [66]), Wireless Local Area Networks (DQCA [62]), cooperative communications (DQCOOP [67]), wireless ad hoc networks (DQMAN [68]), and body area networks (DQBAN [69]).

Existing analyses of DQ [61–69] consider that the devices generate packets following a random Poisson distribution and study the steady-state performance of the protocol. In these conditions, results show the following features of DQ:

- DQ provides near-optimum performance in terms of throughput and delay.
- The performance of DQ is independent of the amount of contending devices and the total offered data traffic.
- DQ behaves as a random access scheme under low traffic and switches to a reservation-based access scheme when the traffic load increases.
- DQ achieves maximum performance using 3 access request minislots regardless of the number of devices.

Despite all the benefits provided by DQ, none of the previous works in DQ have studied the delay and energy performance of DQ under delta traffic conditions in M2M networks for data collection scenarios.

1.3.2 Energy Harvesting in Wireless M2M Networks

1.3.2.1 Energy Harvesting-Aware MAC Protocols

Energy harvesters capture energy from the environment and avoid the need to replace or recharge batteries of end-devices periodically [12][13][70]. However, the inclusion of energy harvesters in M2M networks opens new challenges in the design of communication

protocols. For example, since the harvested energy is not guaranteed continuously, end-devices might not operate normally due to temporary energy shortages.

In the past, some research works, e.g., [71–73], have used a Markov chain to model the operation of wireless sensors with energy harvesting capabilities. Unfortunately, these models only consider one device transmitting without taking into account the multiple access to a shared medium in scenarios where multiple devices interact with each other. As an example, the work in [71] shows the packet loss probability due to energy shortage and the average time to fall in energy shortage of a single device.

The work in [30] was one of the first contributions related to the analysis of the MAC layer with energy-harvesting considerations. It studies the data queue stability for CSMA in wireless networks with energy harvesters. The work in [31] proposes a modified version of the CSMA protocol of IEEE 802.11 integrating energy harvesters. It evaluates the throughput and delay performance using a three-dimensional Markov chain model.

The work in [25] analyzes the throughput and packet drop rate in wireless networks with energy-harvesting using slotted-ALOHA. It presents a Markov chain model that represents the energy stored in a device assuming that data packets are generated following a random Poisson distribution and that energy arrives at random time instants in fixed amounts. Results show that throughput increases with the energy harvesting rate, while the packet drop rate decreases.

The works in [27][29] analyze DFSA and FSA using a single frame in data collection networks where devices transmit data periodically to a fusion center if they have enough available energy. They propose a Markov chain to analyze the energy stored in a device and compute the probability of delivery. Results show that the probability of delivery increases with the energy harvesting rate. The work in [28] proposes a variant of DFSA referred to as Energy Group-DFSA (EG-DFSA) which creates groups of devices according to their energy availability. Devices contend in separated instances of the optimal DFSA for each group. Results show that EG-DFSA outperforms DFSA.

The work in [26] optimizes the duty-cycling in data-collection networks with energy harvesting. It considers FSA to coordinate the transmission of data to a coordinator. It proposes an algorithm to estimate the amount of harvestable energy in each device and to predict the optimum sleep period to ensure energy neutral operation, i.e., the energy consumption in each device is always less or equal to the amount of harvested energy.

1.3.2.2 Energy Harvesting Process

Regarding the analytical modeling of the energy harvesting process, several research works [74–76] have used different probability distributions to model energy sources.

The work in [74] uses a uniform distribution, geometric distribution, transformed geometric distribution, Poisson distribution, transformed Poisson distribution, and a Markovian model to fit empirical datasets. Results show that no single probability distribution can fit all the empirical datasets of solar energy.

The work in [75] shows that piezo-eletric energy can be modeled by the generalized Markovian model, while solar energy can be modeled by a stationary Markovian model. Consequently, there is no unique statistical model that fits with the same energy source, and empirical data is needed to validate the accuracy of the analytical model.

The work in [76] also proposes to use analytical models in which the energy harvested in a time slot is a random variable D, which represents the energy harvested by a device in a short time slot of several seconds or minutes. For mathematical tractability, the authors assume that D takes one of M discrete values $[d_1, ..., d_M]$ with probability $[p_1, ..., p_M]$. In this thesis, we consider an arbitrary probability distribution of the discrete exponential families (e.g., binomial, geometric) to model the energy harvested in a period of time.

1.4 Thesis Outline and Contributions

The main contributions of this thesis cover the design, analysis and performance evaluation of MAC protocols to build highly dense M2M networks for data collection with unlimited lifetime. In the considered scenario, data is periodically generated by at the end-devices and needs to be collected by a gateway or coordinator in a star topology network. In this context, and taking into account the background presented in the previous section, this thesis is organized in two interrelated parts:

- **Part 1**: Energy-Efficient MAC Protocols for Wireless M2M Networks.
- **Part 2**: Energy Harvesting-Aware MAC Protocols for Wireless M2M Networks.

The contributions of both parts of the thesis are detailed in Section 1.4.1 and Section 1.4.2, respectively.

1.4.1 Energy-Efficient MAC Protocols for Wireless M2M Networks

The first part of the thesis is included in Chapter 2. It focuses on the design and analysis of random and hybrid access protocols for dense networks in star topology for data collection. Two key performance metrics are considered: the delay required to resolve the contention, and the energy consumed by the gateway and the end-devices during the data collection process. The **contributions of the first part of the thesis** are summarized in Table 1.1.

TABLE 1.1: Contributions of the first part of the thesis.

N.	Title of the contribution	Section
1.1	Analysis and performance evaluation of FSA-based access protocols	2.5.1
1.2	Design, analysis and performance evaluation of the LP-CTA protocol based on a tree-splitting algorithm	2.5.2
1.3	Design, analysis and performance evaluation of the LP-DQ protocol based on Distributed Queuing	2.5.2
1.4	Analysis and performance evaluation of the RFSA protocol	2.6.1
1.5	Design, analysis and performance evaluation of the LPR-DQ protocol based on Distributed Queuing with reservation	2.6.2

Each contribution of the first part of the thesis is motivated and described in the following tables.

Contribution 1.1
Analysis and performance evaluation of FSA-based access protocols
State of the Art
The MAC layer for data collection scenarios is typically based on Frame-Slotted ALOHA (FSA) due to its simplicity. Recent research works on FSA mainly focus on the minimization of the delay and some of them also compare the energy consumed by the end-devices and the gateway using several variants of FSA. However, they do not analyze how the energy consumption is affected by the frame length, and do not determine the optimal frame length of FSA to minimize the energy consumed by the gateway and the end-devices during the data collection process.
Beyond the State of the Art
Accurate delay and energy models are formulated for two variants of FSA and Dynamic FSA (DFSA). The theoretical models are validated by means of computer-based simulations and the performance of these protocols is evaluated in terms of delay and energy consumption.

Contribution 1.2
Design, analysis and performance evaluation of the LP-CTA protocol based on a tree-splitting algorithm

State of the Art

An alternative to reduce the delay and the energy consumption in the data collection process is to use an m-ary tree-splitting algorithm to resolve collisions among end-devices that contend in a sequence of time frames similar to FSA. The work in [57] analyzes the energy consumed by the end-devices using an m-ary tree-splitting in RFID, but it neglects the energy consumption in reception and sleep modes. The work in [23] evaluates the energy consumption in RFID by means of computer-based simulations. However, previous works on RFID with m-ary tree splitting do not describe how to maximize the energy efficiency during the data collection process, and the analytical energy models are not complete for typical M2M end-devices.

Beyond the State of the Art

The novel Low Power Contention Tree-based Access (LP-CTA) protocol is proposed for data collection networks. LP-CTA is a random access protocol with the frame structure of FSA and incorporates an m-ary tree splitting algorithm to resolve collisions. The tree-splitting algorithm is implemented using a Collision Resolution Queue to reduce the listening periods of the end-devices, thus reducing their energy consumption. Accurate delay and energy models are formulated for LP-CTA and validated by means of computer-based simulations. The performance of LP-CTA is evaluated in terms of delay and energy consumption and compared with that of FSA and DFSA.

Contribution 1.3
Design, analysis and performance evaluation of the LP-DQ protocol based on Distributed Queuing

State of the Art

The Distributed Queuing (DQ) protocol has been comprehesively analyzed in the literature by assuming that the end-devices generate data following a Poisson distribution or considering that the network is in saturation conditions. Previous works [61–69] demonstrate that under these conditions DQ shows a near-optimum performance in terms of throughput. However, none of the previous works in DQ have studied the delay and energy performance of DQ in the case of delta traffic conditions.

Beyond the State of the Art

The novel Low Power DQ (LP-DQ) protocol is proposed for M2M data collection networks. LP-DQ is a hybrid access protocol based on DQ that combines very short contention periods for access request and collision-free slots for data. The end-devices requiring access to the system transmit access requests using slotted-ALOHA and collisions are resolved with tree splitting. When an end-device succeeds in transmitting an access request, it waits for its turn in sleep mode until it can transmit data in a collision-free slot. The delay and energy models of LP-DQ are formulated and validated by means of computer-based simulations. The performance of LP-DQ is evaluated and compared with that of LP-CTA, FSA and DFSA.

Contribution 1.4
Analysis and performance evaluation of the RFSA protocol

State of the Art

Reservation Frame Slotted-ALOHA (RFSA) was proposed as an extension of FSA to manage the access when end-devices generate long messages fragmented into small data packets. Previous works on RFSA [47–50] focus on throughput analysis assuming that the network is in steady-state conditions and the end-devices generate multi-packet messages following a given random distribution, typically Poisson. Results show that RFSA outperforms FSA in terms of throughput. However, the performance of RFSA in the case of delta traffic conditions remains an open research question which deserves attention for its application in data-collection scenarios.

Beyond the State of the Art

An accurate theoretical model is formulated to calculate the delay and energy consumption for RFSA in data-collection networks with fragmentation of long messages. The theoretical model is validated with extensive computer-based simulations. The performance of RFSA is evaluated and compared with that of FSA.

Contribution 1.5
Design, analysis and performance evaluation of the LPR-DQ protocol based on Distributed Queuing with reservation

State of the Art

The approach of the LP-DQ protocol is convenient when the messages transmitted by each end-device fit in one single data slot. However, if the end-devices generate long data messages that have to be fragmented into a certain number of small packets, it is better to add a reservation mechanism in order to boost the performance of the protocol.

Beyond the State of the Art

The Low Power Distributed Queuing with Reservation (LPR-DQ) is proposed for data collection networks where the end-devices generate long messages that require fragmentation. LPR-DQ constitutes an extension of the LP-DQ protocol. While in LP-DQ only one collision-free slot is reserved for each end-device that succeeds in transmitting an access request, in LPR-DQ an end-device can reserve as many collision-free slots as the number of fragments in each message. The energy model of LRP-DQ is formulated and validated with computer-based simulations. The performance of LPR-DQ is evaluated and compared with that of RFSA and FSA.

1.4.2 Energy Harvesting-Aware MAC Protocols for Wireless M2M Networks

The second part of the thesis is confined to Chapter 3. It is devoted to the design and analysis of energy harvesting-aware MAC protocols for data collection networks in star topology where each end-device is equipped with an energy harvester and an energy storage device (ESD), i.e., battery or super-capacitor. The energy harvested at the end-devices is modeled with a discrete probability distribution. Three key performance metrics are proposed in order to evaluate the trade-offs in the design of EH-aware MAC

protocols: the *probability of delivery* and the *data delivery ratio*, which measure the ability of the MAC protocol to successfully deliver data from the end-devices to the gateway without depleting the ESDs, and the *time efficiency*, which measures the data collection rate at the gateway. The performance metrics are evaluated over the amount of harvested energy per end-device, the number of end-devices, the capacity of the ESDs, and the minimum energy (or threshold) that an end-device needs to transmit data. The **contributions of the second part of the thesis** are summarized in Table 1.2.

TABLE 1.2: Contributions of the second part of the thesis.

N.	Title of the contribution	Section
2.1	Analysis and performance evaluation of the EH-DFSA protocol	3.5.1
2.2	Design, analysis and performance evaluation of the EH-CTA protocol	3.5.2
2.3	Design, analysis and performance evaluation of the EH-RDFSA protocol	3.6.1
2.4	Design, analysis and performance evaluation of the EH-DQ protocol based on Distributed Queuing with reservation	3.6.2

Each contribution of the second part of the thesis is motivated and described in the following tables.

Contribution 2.1
Analysis and performance evaluation of the EH-DFSA protocol
State of the Art
The works in [27–29] analyze the performance of the Dynamic Frame Slotted-ALOHA (DFSA) protocol in terms of the probability of delivery and the data collection rate in networks with energy harvesting probabilities. Results show that both performance metrics increase with the energy harvesting rate. However, these works do not evaluate how the performance is affected by the capacity of the batteries, the minimum energy that an end-device needs to transmit data to the gateway, and the number of end-devices in the network.
Beyond the State of the Art
The performance of the EH-DFSA protocol is analyzed and evaluated in order to demonstrate how it is affected by the capacity of the batteries, the number of end-devices in the network, and the minimum energy (or threshold) considered for an end-device to become active and transmit data to the gateway.

Contribution 2.2
Design, analysis and performance evaluation of the EH-CTA protocol
State of the Art
The design and analysis of random access protocols based on a tree-splitting algorithm for wireless networks with energy-harvesting has never received attention.
Beyond the State of the Art
The novel EH-aware Low Power Contention Tree-based Access (EH-CTA) protocol is proposed as an adaptation of LP-CTA for data collection networks with energy harvesting. The performance of EH-CTA is analyzed and evaluated in terms of the probability of delivery and the data collection rate. The performance of EH-CTA is compared with that of EH-DFSA.

Contribution 2.3
Design, analysis and performance evaluation of the EH-RDFSA protocol
State of the Art
The design and analysis of a DFSA protocol with reservation for networks with energy-harvesting capabilities has never received attention.
Beyond the State of the Art
The novel EH-aware Reservation DFSA (EH-RDFSA) is proposed as a combination of RFSA and DFSA for data collection networks with energy harvesting where the end-devices generate long messages that require fragmentation. The performance of EH-RDFSA is analyzed and evaluated in terms of the data delivery ratio and the data collection rate. The performance of EH-RDFSA is compared with that of EH-DFSA and TDMA.

Contribution 2.4
Design, analysis and performance evaluation of the EH-DQ protocol based on Distributed Queuing with reservation
State of the Art
The design and analysis of a hybrid access protocol based on Distributed Queuing with reservation for networks with energy-harvesting capabilities has never received attention.
Beyond the State of the Art
The EH-aware Low Power Distributed Queuing with Reservation (EH-DQ) is proposed as an extension and adaptation of LPR-DQ for data collection networks with energy harvesting where the end-devices generate long messages that require fragmentation. In EH-DQ, each end-device can reserve a certain number of collision-free slots which depends on the energy available in the energy storage device and the number of data packets in each message. The performance of EH-DQ is analyzed and evaluated in terms of the data delivery ratio and the data collection rate. The performance of EH-DQ is compared with that of EH-RDFSA, EH-DFSA and TDMA.

The thesis is concluded in Chapter 4, which summarizes the main findings, and describes future lines of research. The publications related to the contributions and ideas exposed in this thesis are presented in the next section.

1.5 Publications

The contributions of this thesis have been published in scientific journals and international conferences as detailed in Section 1.5.1 and 1.5.2, respectively. Finally, other research contributions which are not in the scope of this thesis are detailed in Section 1.6.

1.5.1 Journals and Magazines

[J6] F. Vázquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Contention Tree-based Access for Wireless Machine-to-Machine Networks with Energy Harvesting", submitted to the IEEE Transactions on Wireless Communications journal, 2015.

[J5] F. Vázquez-Gallego, J. Alonso-Zarate, P. Tuset, and L. Alonso, "Modelling and Analysis of Reservation Frame Slotted-ALOHA in Wireless Machine-to-Machine Area Networks for Data Collection", MPDI Sensors, 15, pp. 15, 3911-3931; doi:10.3390/s150203911, 9 February 2015.

[J4] F. Vázquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Energy Analysis of Contention Tree-based Access Protocols in Dense Machine-to-Machine Area Networks", Hindawi Journal of Sensors, Article ID. 685164, April 2015. Free Open Access: http://www.hindawi.com/journals/js/2015/685164/

[J3] P. Tuset, F. Vázquez-Gallego, J. Alonso-Zarate, X. Vilajosana, and L. Alonso, "Experimental energy consumption of Frame Slotted ALOHA and Distributed Queuing for data collection scenarios in smart cities", MDPI Sensors journal, 2014.

[J2] P. Tuset, F. Vázquez-Gallego, J. Alonso-Zarate, X. Vilajosana, and L. Alonso, "LPDQ: a self-scheduled TDMA MAC protocol for one-hop dynamic low power wireless networks", ELSEVIER Pervasive and Mobile Computing Journal, special issue on on IoT 2014.

[J1] F. Vázquez-Gallego, J. Alonso-Zarate, L. Alonso, and M. Dohler, "Analysis of Energy Efficient Distributed Neighbour Discovery Mechanisms for Machine-to-Machine Networks," ELSEVIER Ad Hoc Networs journal, special issue on M2M and IoT, 2013.

1.5.2 International Conferences

[C11] F. Vázquez-Gallego, L. Alonso, J. Alonso-Zarate, "Energy Harvesting-aware Distributed Queuing Access for Wireless Machine-to-Machine Networks", submitted to the IEEE GLOBECOM 2016.

[C10] F. Vázquez-Gallego, L. Alonso, J. Alonso-Zarate, "Energy Harvesting-aware Contention Tree-based Access for Wireless Machine-to-Machine Networks", in proc. of the IEEE ICC 2016, Kuala Lumpur, Malaysia, May 2016.

[C9] F. Vázquez-Gallego, J. Alonso-Zarate, L. Alonso, "Reservation Dynamic Frame Slotted-ALOHA for Wireless M2M Networks with Energy Harvesting", in proc. of the IEEE ICC 2015, London, UK, June 2015.

[C8] F. Vázquez-Gallego, S. Wu, Y. Chen, K. K. Chai, J. Alonso-Zarate, "Analysis and Performance Evaluation of Dynamic Frame Slotted-ALOHA in Wireless Machine-to-Machine Networks with Energy Harvesting", IEEE GLOBECOM 2014, Workshop on Green Broadband access: energy efficient wireless and wired network solutions, December 12, 2014.

[C7] F. Vázquez-Gallego, O. Briante, A.M. Mandalari, J. Alonso-Zarate, G. Ruggeri, A. Molinaro, "Performance Evaluation of Reservation Frame Slotted-ALOHA for Data Collection M2M Networks", in proc. of the European Wireless 2014, May 2014, Barcelona.

[C6] O. Briante, F. Vázquez-Gallego, A.M. Mandalari, J. Alonso-Zarate, G. Ruggeri, A. Molinaro, "Duty-Cycle Optimization for Machine-to-Machine Area Networks Based on Frame Slotted-ALOHA with Energy Harvesting Capabilities", in proc. of the European Wireless 2014, May 2014, Barcelona.

[C5] P. Tuset, F. Vázquez-Gallego, J. Alonso-Zarate, X. Vilajosana, and L. Alonso, "Demonstrating Low-Power Distributed Queuing for Active RFID Communications At 433 MHz", in proc. of IEEE INFOCOM 2014, Demo Sessions, April-May 2014, Toronto, Canada. **BEST DEMO RUNNER-UP AWARD INFOCOM 2014**.

[C4] F. Vázquez-Gallego, J. Alonso-Zarate, P. Tuset-Peiró, L. Alonso, "Energy Analysis of a Contention Tree-based Access Protocol for Machine-to-Machine Networks with

Idle-to-Saturation Traffic Transitions", in proc. of the IEEE ICC 2014, Sydney, Australia, June 2014.

[C3] F. Vázquez-Gallego, J. Alonso-Zarate, P. Tuset-Peiró, L. Alonso, "Energy Performance of Distributed Queuing Access in Machine-to-Machine Networks with Idle-to-Saturation Transitions," in proc. of the IEEE GLOBECOM, December, 2013.

[C2] F. Vazquez-Gallego, J. Alonso-Zarate, L. Alonso, "Energy and Delay Analysis of Contention Resolution Mechanisms for Machine-to-Machine Networks based on Low-Power WiFi", in proc. of the IEEE ICC 2013, Budapest, Hungary, June 2013.

[C1] F. Vazquez-Gallego, J. Alonso-Zarate, and L. Alonso, "Energy Analysis of Distributed Neighbour Discovery Algorithms Based on Frame Slotted-ALOHA for Cooperative Networks", in proc. of the IEEE GLOBECOM 2012, Anaheim, California, USA, 3-7 December 2012.

1.6 Other Research Contributions

Besides the main contributions of this thesis outlined in the previous sections, a number of other research works have been carried out before and while this thesis was prepared. Although some of them are not in the scope of the thesis, it is worth mentioning some contributions published in scientific journals:

[J5] A. Laya, C. Kalalas, F. Vázquez-Gallego, L. Alonso and J. Alonso-Zarate, "Goodbye, ALOHA!," in IEEE Access, vol. PP, no. 99, pp. 1-1, doi: 10.1109/ACCESS.2016.2557758, April 2016, ISSN: 2169-3536.

[J4] V. Karagiannis, P. Chatzimisios, F. Vázquez-Gallego, J. Alonso-Zarate, "A Survey on Application Layer Protocols for the Internet of Things," Transactions on Internet of Things and Cloud Computing, vol. 1, no. 1, ISSN: 2331-4753 (Print) 2331-4761 (Online), ICAS Publishing Group, January 2015.

[J3] X. Vilajosana, P. Tuset, F. Vázquez-Gallego, J. Alonso-Zarate, and L. Alonso, "Standardized Low-Power Wireless Communication Technologies For Distributed Sensing Applications," IEEE Sensors journal 2014.

[J2] F. Vázquez-Gallego, J. Alonso-Zarate, C. Liss, and Ch. Verikoukis, "OpenMAC: A New Reconfigurable Experimental Platform for Energy-Efficient Medium Access Control Protocols," IET Science, Measurement and Technology, May 2012, Vol. 6, No. 3, pp. 139148, May 2012.

[J1] F. Vázquez-Gallego, J. Alonso-Zarate, Ch. Verikoukis, and L. Alonso, "A Survey on Prototyping Platforms for the Development and Experimental Evaluation of Medium Access Control Protocols," IEEE Wireless Communications Magazine, vol. 19, no. 1, pp. 74-81, Feb. 2012, ISSN: 1536-1284.

and international conferences:

[C19] C. Kalalas, F. Vázquez-Gallego, J. Alonso-Zarate, "Handling Mission-Critical Communication in Smart Grid Distribution Automation Services through LTE", submitted to the IEEE SmartGridComm 2016.

[C18] R. Muñoz, R. Vilalta, C. Verikoukis, J. Mangues, J. Alonso-Zarate, A. Georgiadis, M. Payaro, R. Casellas, R. Martínez, J. Serra, D. Pubill, J. Núñez, J. Baranda, F. Vázquez-Gallego, A. I. Pérez-Neira, "The CTTC 5G end-to-end experimental platform integrating IoT, SDN, and distributed cloud", in proc. of Wireless World Research Forum Meeting 35 (WWRF), 14-16 October 2015, Copenhagen (Denmark).

[C17] A. Hernández, F. Vázquez-Gallego, L. Alonso, J. Alonso-Zarate, "Performance Evaluation of Frame Slotted ALOHA with Intra-Frame and Inter-Frame Successive Interference Cancellation", in proc. of the IEEE Globecom 2015, December 2015.

[C16] G. Interdonato, S. Pfletschinger, F. Vázquez-Gallego, J. Alonso-Zarate, G. Araniti, "Intra-Slot Interference Cancellation for Collision Resolution in Irregular Repetition Slotted ALOHA", in proc. of IEEE ICC 2015, Workshop on Massive Uncoordinated Access Protocols, June 2015, London, UK.

[C15] R. Palacios, D. Franch, F. Vázquez-Gallego, J. Alonso-Zarate, F. Granelli, "Experimental Evaluation of Reverse Direction Transmissions in WLAN Using the WARP Platform, in proc. of IEEE ICC 2015, June 2015, London, UK.

[C14] F. Vázquez-Gallego, M. Rietti, J. Bas, J. Alonso-Zarate, L. Alonso, "Performance Evaluation of Frame Slotted-ALOHA with Successive Interface Cancellation in

Machine-to-Machine Networks", in proc. of the European Wireless 2014, May 2014, Barcelona.

[C13] L. Di Gregorio, D. Gajic, C. Liss, A. Foglar, F. Vázquez-Gallego, "WAN Optimizations in Vehicular Networking," in proc. of the Wireless Congress 2013, Munich, November 2013.

[C12] J. Bas, F. Vázquez-Gallego, C. Gavrincea, J. Alonso-Zarate, "Energy and Delay Model of Binary BCH Codes for Dense Machine-to-Machine Networks," in proc. of the PIMRC 2013, London, September 2013.

[C11] P. Tuset-Peiró, F. Adelantado, X. Vilajosana, F. Vázquez-Gallego, J. Alonso-Zarate, "On the use of the 433 MHz band to Improve the Energy Efficiency of M2M Communications," in proc. of the PIMRC 2013, London, September 2013.

[C10] F. Vázquez-Gallego, J. Alonso-Zarate, Ch. Verikoukis, D. Gajic, C. Liss, L. Di Gregorio, "Experimental Performance Evaluation of a Cooperative ARQ Scheme using OpenMAC," in proc. of the ISCWCS 2013, Ilmenau, Germany, August 2013.

[C9] F. Vázquez-Gallego, J. Alonso-Zarate, I. Balboteo, L. Alonso, "DPCF-M: a Medium Access Control Protocol for Dense Machine-to-Machine Area Networks with Dynamic Gateways," in proc. of the IEEE SPAWC 2013, Darmstadt, Germany, June 2013.

[C8] F. Vázquez-Gallego, J. Alonso-Zarate, Ch. Verikoukis, D. Gajic, C. Liss, "Fast Prototyping of Medium Access Control Protocols with OpenMAC," in proc. of the 17th IEEE CAMAD, Barcelona, September 2012.

[C7] J. Bas, F.Vázquez Gallego, C. Gavrincea, J. Alonso-Zarate, "Flexible Physical Layer Design subject to Delay and Energy Constraints," in proc. of the 17th IEEE CAMAD, Barcelona, September 2012.

[C6] A. Antonopoulos, J. Bas, M. Katz, H. Lundqvist, T. Moreira, K. Ntontin, F. Vázquez-Gallego, N. Zorba, "Green-T: Enabling Techniques for Energy Efficient Mobile Terminals," in proc. of the 17th IEEE CAMAD, Barcelona, September 2012.

[C5] F. Vázquez-Gallego, J. Alonso-Zarate, D. Gajic, C. Liss, and Ch. Verikoukis, "Implementation and Performance Evaluation of a New Experimental Platform

for Medium Access Control Protocols," in proc. of the ICST Tridentcom 2012, Thessaloniki, Greece, June 2012.

[C4] Omar I. Al-Surkhi, F. Vázquez-Gallego, A. Rodriguez-Jornet, M. Garca, J. Mas, J. Ibeas, and P.J. Riu, "Local Tissue Bioimpedance Measurement for Fluid Shifts Estimation During Hemodialysis", 13th International Conference on Electrical Bioimpedance combined with the 8th Conference on Electrical Impedance Tomography, ICEBI, Gratz (Austria), 2007.

[C3] Omar I. Al-Surkhi, P.J. Riu, F. Vázquez-Gallego, and J. Ibeas, "Monitoring Cole-Cole Parameters during Haemodialysis", 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Lyon (France), August 2007.

[C2] Omar I. Al-Surkhi, P.J. Riu, F. Bogonez, F. Vázquez-Gallego, and J. Ibeas, "Monitoring Fluid Shifts during haemodialisys using electrical bio-impedance techniques", International Biomedical Engineering Conference (CIBEC'06), Cairo (Egypt), Dec. 2006.

[C1] F. Bogónez, F. Vázquez-Gallego, O. Surakhy, J. Sevilla, and P. Riu, "Portable Wireless Bioimpedance Measurement System for Hemodialisis Monitoring", World Congress on Medical Physics and Biomedical Engineering, Seoul (Corea), Sept. 2006.

Chapter 2

Energy-Efficient MAC Protocols for Wireless M2M Networks

2.1 Introduction

The MAC layer for data collection scenarios is typically based on Frame-Slotted ALOHA (FSA) due to its simplicity. Recent research works [22, 39, 40] mainly focus on the minimization of the delay and compare the energy consumed by the end-devices and the gateway (i.e., tags and reader of RFID) using several variants of FSA. However, these works do not analyze how the energy consumption is affected by the frame length, and do not determine the optimal frame length configuration of FSA to minimize the energy consumed by the gateway and the end-devices during the data collection process. In this chapter, accurate delay and energy models are formulated for two variants of FSA and Dynamic FSA (DFSA). The theoretical models are validated by means of computer-based simulations and the performance of these protocols is evaluated in terms of delay and energy consumption.

An alternative to reduce the delay and the energy consumption in the data collection process is to use an m-ary tree-splitting algorithm to resolve collisions among end-devices that contend in a sequence of time frames similar to FSA. The work in [57] analyzes the energy consumed by the end-devices using an m-ary tree-splitting in RFID, but it neglects the energy consumption in reception and sleep modes. The work in [23] evaluates the energy consumption in RFID by means of computer-based simulations. Up to our knowledge, previous works on RFID with m-ary tree splitting do not describe how

to maximize the energy efficiency during the data collection process, and the analytical energy models are not complete for typical M2M end-devices. In this chapter, the novel Low Power Contention Tree-based Access (LP-CTA) protocol is proposed for data collection networks. LP-CTA is based on the frame structure of FSA and incorporates an m-ary tree splitting algorithm to resolve contention. A Collision Resolution Queue (CRQ) is used to implement tree splitting and to reduce the energy consumed by the end-devices. Accurate delay and energy models are formulated for LP-CTA and validated by means of computer-based simulations. The performance of LP-CTA is evaluated in terms of delay and energy consumption and compared with that of FSA and DFSA.

The Distributed Queuing (DQ) protocol has been comprehesively analyzed in the literature by assuming that the end-devices generate data following a Poisson distribution or considering that the network is in saturation conditions. Previous works [61–69] demonstrate that under these conditions DQ shows a near-optimum performance in terms of throughput. However, to the best of our knowledge, none of the previous works in DQ have studied the delay and energy performance of DQ in the case of delta traffic conditions. In this chapter, the novel Low Power DQ (LP-DQ) protocol is proposed as an adaptation of DQ that is tailored for M2M data collection scenarios. In LP-DQ, end-devices are in low power listening mode and wake up periodically to transmit data. During the contention process, the end-devices only need to listen to the feedback packets of those frames where they transmit data or access requests, and sleep in the rest of the frames in order to reduce energy consumption. The delay and energy models of LP-DQ are formulated and validated by means of computer-based simulations. The performance of LP-DQ is evaluated and compared with that of LP-CTA, FSA and DFSA.

Reservation Frame Slotted-ALOHA (RFSA) was proposed in the past to manage the access to the wireless channel when end-devices generate long messages fragmented into small data packets. To the best of our knowledge, previous works [47–50] focus on the throughput analysis of RFSA assuming that the network is in steady-state conditions and the end-devices generate multi-packet messages following a given random distribution, typically Poisson. Results show that RFSA outperforms FSA in terms of throughput. However, the performance of RFSA in the case of delta traffic conditions remains an open research question which deserves attention for its application in data-collection scenarios. In this chapter, an accurate theoretical model is formulated to calculate the

delay and energy consumption for RFSA in data-collection networks with fragmentation of long messages. The theoretical model is validated with extensive computer-based simulations. The performance of RFSA is evaluated and compared with that of FSA.

Finally, the Low Power Distributed Queuing with Reservation (LPR-DQ) is proposed in this chapter for data collection networks where the end-devices generate long multi-packet messages. In LPR-DQ, each end-device can reserve a certain number of collision-free slots which depends on the number of data packets in each message. The energy model of LRP-DQ is formulated and validated with computer-based simulations. The performance of LPR-DQ is evaluated and compared with that of RFSA and FSA.

2.2 Chapter Outline

The remainder of this chapter is organized as follows. The system model and the performance metrics used in this chapter are introduced in Section 2.3 and Section 2.4, respectively.

Two different data collection scenarios (I and II) have been considered. In scenario I, each end-device has exactly 1 new data packet to send to the gateway in every data collection round, e.g., identification, measurement, alarm, etc. In scenario II, each end-device has 1 or more new data packets to send in every data collection round, e.g., set of measurements, images, etc.

Section 2.5 focuses on the design, analysis and performance evaluation of MAC protocols for scenario I. In particular, two variants of FSA and DFSA protocols are evaluated, and the LP-CTA and LP-DQ protocols, based on tree splitting and DQ, are introduced and evaluated.

Section 2.6 focuses on the design, analysis and performance evaluation of MAC protocols for scenario II. In particular, the RFSA and FSA protocols are evaluated, and the LPR-DQ protocol is introduced and evaluated.

Finally, Section 2.7 concludes the chapter by summarizing the most relevant results and the key contents of the chapter.

2.3 System Model

2.3.1 Network and Data Model

We consider a wireless network in star topology formed by one coordinator (or gateway) and n end-devices in the communication range of the coordinator, as shown in Figure 2.1. The coordinator collects data from the end-devices by periodically initiating data collection rounds (DCR), once every T_R seconds, as depicted in Figure 2.2. Each DCR is initiated when the coordinator broadcasts a request for data (RFD) packet. In the k-th DCR, each end-device has a number $l(k)$ of data packets ready to be transmitted. The data process $l(k)$ can be modeled as a discrete random variable with probability mass function $p_j = \Pr\{l(k) = j\}$ with $j \in \{1, 2, ...\}$. The value of $l(k)$ is considered to be identically and independently distributed (i.i.d.) over all end-devices and DCRs. We assume that the data packets have a common and constant length.

We assume that the end-devices are in a low-power listening mode [77] in which they periodically wake up and turn on the radio transceiver for a short period of time to detect communication requests from the coordinator. We assume that all of the end-devices are synchronized and listen to the channel when the coordinator transmits an RFD. After decoding an RFD, the end-devices start contending for the channel to transmit their $l(k)$ data packets. The k-th DCR is organized into a sequence of F_k time frames. The end-devices are synchronized to a common time frame pattern and transmit their data packets according to the rules of the adopted MAC protocol, which defines the structure of the frames. The coordinator broadcasts a feedback packet (FBP) or beacon at the end of each frame to enable the synchronization of the end-devices. The contents of the FBP payload depend on the adopted MAC layer. When a data packet is successfully

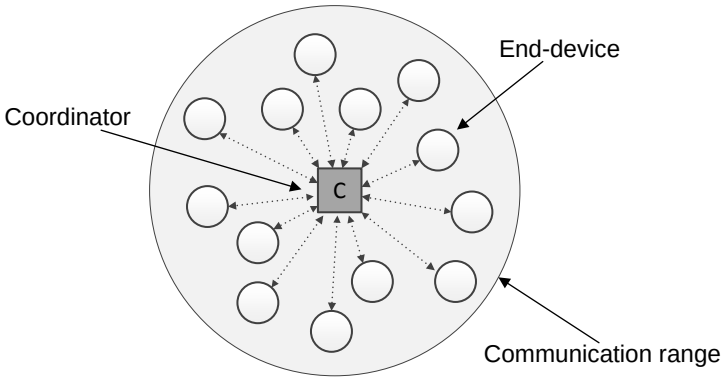


FIGURE 2.1: Wireless network in star topology.

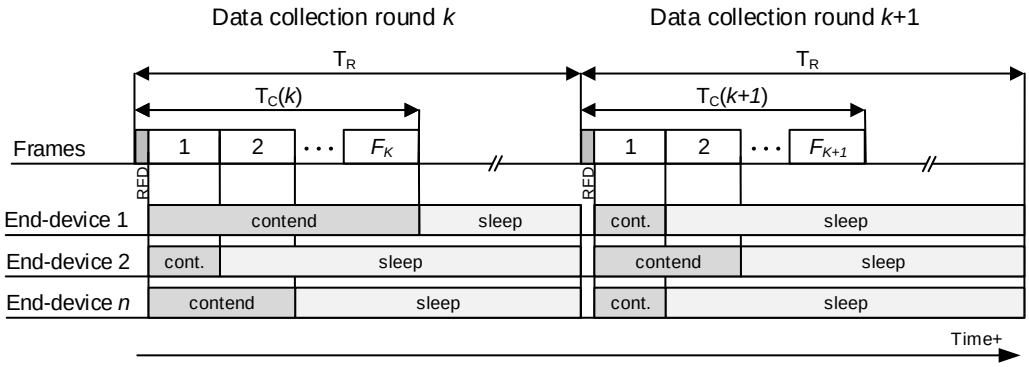


FIGURE 2.2: Sequence of data collection rounds.

received by the coordinator in any of the F_k frames, then an acknowledgement is sent by the coordinator. Once an end-device succeeds in transmitting its $l(k)$ data packets, it switches to sleep mode and saves energy until the next DCR starts. All end-devices wake up again to listen to the channel when the coordinator sends the RFD of every DCR. We assume that the duration $T_C(k)$ of the k-th DCR is shorter than the time T_R between any two consecutive DCRs (i.e., $T_C(k) \ll T_R$ for all k) to ensure that successive DCRs do not overlap.

In order to focus on the contention process, we consider a communication scenario where all packets are always transmitted without transmission errors induced by the wireless channel. In addition, we assume that when two or more data packets collide, none of them can be decoded by the coordinator, *i.e.*, there is no capture effect. The inclusion of the capture effect and transmission errors, and their impact on the overall network performance, constitutes part of our future work.

2.3.2 Energy Consumption Model

Regarding energy consumption, the coordinator and the end-devices can be in four different modes of operation: (i) transmitting a packet; (ii) receiving; (iii) idle listening; or (iv) sleeping. The associated power consumptions are ρ_{tx}, ρ_{rx}, ρ_{σ}, or ρ_{sleep}, respectively. The model is general for any power-saving mechanism and any value of the power consumptions in different modes of operation. We assume that the energy required to switch between sleep and active modes (*i.e.*, transmitting, receiving, idle listening) is negligible. In sleep mode, the radio interface is fully disabled, and thus, the end-devices consume the lowest power consumption.

2.4 Performance Metrics

The following sections describe the criteria that are considered in this chapter to measure the performance of the MAC protocols.

2.4.1 Energy Consumption

The *energy consumption of the coordinator* is the average energy consumed by the coordinator in the time elapsed since a DCR starts until all the end-devices have succeeded in transmitting their data packets, i.e., the DCR finishes.

The *energy consumption per end-device* is the average energy consumed by an end-device since a DCR starts until the end-device succeeds in transmitting its data packets.

2.4.2 Delay

The *delay* is the average time elapsed since a DCR starts until all the end-devices have succeeded in transmitting their data packets.

2.5 Scenario I: Single Data-Packet per Data Collection Round

This section focuses on MAC protocols for data collection applications where each end-device has exactly one new data packet to transmit in every DCR, i.e., $l(k) = 1$. The structure of this section is organized as follows. Section 2.5.1 focuses on the analysis and performance evaluation of Frame Slotted-ALOHA based Access Protocols, more specifically, of two variants of FSA and DFSA. Section 2.5.2 focuses on the design, analysis and performance evaluation of random access and hybrid protocols based on an m-ary tree splitting algorithm to resolve the contention. In particular, the LP-CTA and LP-DQ protocols are proposed in this section.

2.5.1 Frame Slotted-ALOHA based Access Protocols

This section focuses on the analysis and performance evaluation of FSA and DFSA. The operation of the protocols is decribed in Section 2.5.1.1. Section 2.5.1.2 presents the analysis based on absorbing Markov chains which is required to formulate the analytical models derived later to calculate the delay and the energy consumption. Section 2.5.1.3 is devoted to validate the analytical models and to evaluate the performance of the FSA and DFSA protocols.

2.5.1.1 MAC Protocols Description

2.5.1.1.1 Frame Slotted-ALOHA (FSA-ACK, FSA-FBP)

We consider two different versions of the Frame Slotted-ALOHA protocol: FSA-ACK and FSA-FBP. The frame structures of FSA-ACK and FSA-FBP are shown in Figure 2.3 and Figure 2.4, respectively. Each frame is divided into a fixed number m of contention slots. In the contention process using FSA-ACK and FSA-FBP, each end-device randomly selects one of the slots of a frame to transmit its data packet with no Clear Channel Assessment (CCA), i.e., the end-devices do not assess the channel status (busy or idle) before transmission. Consequently, a given slot can be in one of three states: empty, i.e., no data packet has been transmitted, success, i.e., only one data packet has been transmitted, or collision, i.e., more than one end-device has transmitted in that slot.

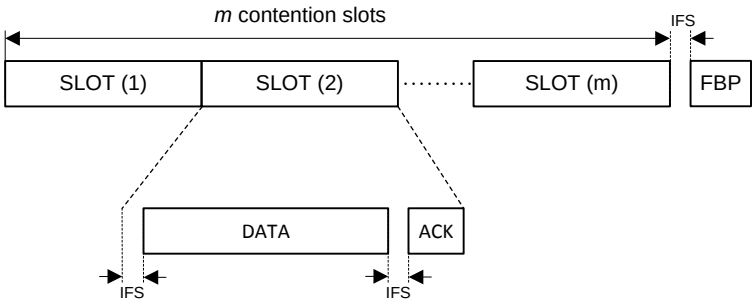


FIGURE 2.3: Frame structure of FSA-ACK and DFSA.

In FSA-ACK, the coordinator responds with an ACK to each data packet decoded successfully in each slot. As shown in Figure 2.3, the duration of a slot in FSA-ACK is adjusted to fit the transmission time of a data packet and an ACK, with the necessary guard times, called Inter Frame Spaces (IFS), between reception and transmission modes to compensate propagation and processing delays and the time required to switch the radio transceivers between reception and transmission. The coordinator transmits a short FBP packet at the end of each frame to enable the synchronization of the end-devices.

In FSA-FBP, the coordinator does not respond with an ACK to each data packet decoded successfully. As shown in Figure 2.4, the duration of a slot in FSA-FBP is adjusted to fit the transmission time of a data packet. The coordinator transmits a FBP packet at the end of each frame to inform of the state of every slot.

In FSA-ACK and FSA-FBP, the end-devices that have not succeeded yet in frame F_i will transmit again in frame F_{i+1}, and the end-devices that have succeeded in a given frame will enter into sleep mode and stop contending in subsequent frames. Therefore, the number of contending end-devices decreases in consecutive frames until the contention process finishes.

Figure 2.5 shows an example of operation of FSA-ACK and FSA-FBP. All the end-devices contend in frame 1: end-devices 1 and 2 succeed, and end-devices 3 and 4 collide.

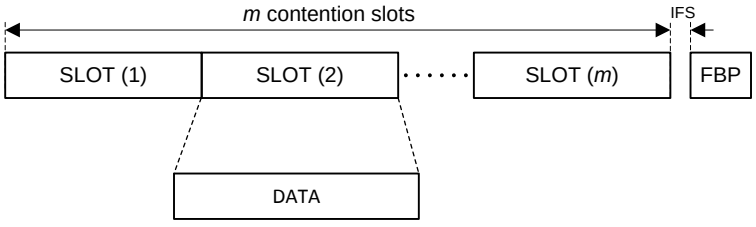


FIGURE 2.4: Frame structure of FSA-FBP.

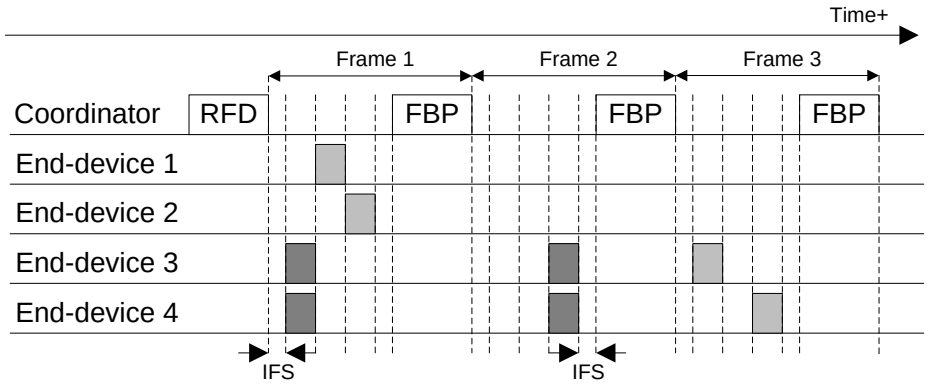


FIGURE 2.5: Example of operation of FSA-ACK and FSA-FBP with 4 end-devices and 3 contention slots per frame. The ACK packets of FSA-ACK have been omitted.

Thus, end-devices 3 and 4 contend again in frame 2, where they collide. Finally, end-devices 3 and 4 succeed in frame 3 and the process ends.

2.5.1.1.2 Dynamic Frame Slotted-ALOHA (DFSA)

The frame structure of DFSA is shown in Figure 2.3. Each frame F_i is divided into a variable number m_i of contention slots. The number of slots is dynamically adjusted to be proportional to the estimated number $\hat{n}_i$ of end-devices that contend in every frame as $m_i = \lceil \rho \cdot \hat{n}_i \rceil$, where ρ is a positive real number. In the contention processs using DFSA an end-device randomly selects one of the slots in every frame to transmit its data packet with no CCA. The coordinator responds with an ACK to each data packet decoded successfully in each slot. As shown in Figure 2.3, the duration of a slot is adjusted to fit the transmission time of a data packet from an end-device and of an ACK from the coordinator, with the necessary guard times and considering both packet length and data transmission rates. When an end-device succeeds in transmitting its data packet in a given frame, it enters into sleep mode and stops contending again in subsequent frames. Otherwise, it reattempts transmission of its data packet in the next frame. At the end of each frame, we assume that the coordinator is able to estimate the number of end-devices that will contend in the next frame and broadcasts the number of contention slots in the FBP. Some examples of estimation algorithms have been proposed in [78].

2.5.1.2 Analysis of Performance Metrics

In this section, we formulate the delay and energy consumption models for the FSA-ACK, FSA-FBP and DFSA protocols. Towards this end, the operation of the protocols is modeled with an absorbing Markov chain where the transition probabilities depend on the probability $P_s(m_c, m_1, n_c)$ that exist a number m_1 of successful slots in a given frame with m_c contention slots, having n_c contending end-devices. Note that the expression of $P_s(m_c, m_1, n_c)$ was derived in the analysis presented in [35].

2.5.1.2.1 Absorbing Markov Chain Model

The FSA-ACK, FSA-FBP and DFSA protocols can be modeled with the absorbing Markov chain depicted in Figure 2.6. Each state of the chain represents the number of end-devices that have succeeded to send their data packet to the coordinator since the contention process started. The Markov chain is characterized by a transition matrix P = $[p_{ij}]$, where each element p_{ij} represents the probability that the number of successful end-devices changes from i to j after one frame, with $i, j = \{0, 1, ..., n\}$. The process finishes, i.e., it is absorbed, when all the end-devices have succeeded, i.e., $i = n$. Once this absorbing state is reached, the probability that the process changes from this state to any other is zero. The not absorbing states are called 'transient' states.

In the next sections, we first formulate the transition matrix for FSA-ACK, FSA-FBP and DFSA. Secondly, we compute the average number of frames in which the process keeps in the same transient state (i.e., with a constant number of successful end-devices), and the average number of frames until the process is absorbed. Finally, we use these values to formulate the delay and energy models of the FSA-ACK, FSA-FBP and DFSA protocols.

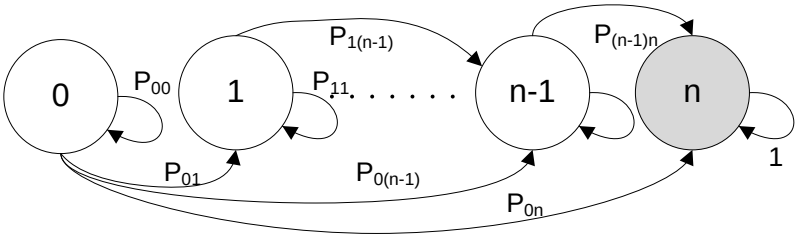


FIGURE 2.6: Generalized state transition diagram of the Markov chain that models the evolution of the numbers of end-devices that succeed in transmitting data using FSA-ACK, FSA-FBP and DFSA.

2.5.1.2.2 Transition Matrix

In the contention process of a DCR using FSA-ACK, FSA-FBP or DFSA, the probability that exist a number m_1 of successful slots in a given frame with m_c contention slots, having n_c contending end-devices, is denoted as $P_s(m_c, m_1, n_c)$. According to [35], this probability can be expressed as

$$P_s\left(m_c, m_1, n_c\right) = \frac{\binom{m_c}{m_1} \prod_{k=0}^{m_1-1} \left(n_c - k\right) G\left(m_c - m_1, n_c - m_1\right)}{{m_c}^{n_c}}, \tag{2.1}$$

where

$$G\left(T, t\right) = T^t + \sum_{k=1}^{t} \left\{ (-1)^k \cdot \prod_{j=0}^{k-1} \left\{ \left(t - j\right)\left(T - j\right) \right\} \left(T - k\right)^{t-k} \frac{1}{k!} \right\}, \tag{2.2}$$

with $T = m_c - m_1$ and $t = n_c - m_1$.

The transition probabilities p_{ij} for FSA-ACK and FSA-FBP can be defined as

$$p_{ij} = \begin{cases} P_s\left(m, j-i, n-i\right), & if\ \ (i < j \leq i + m) \\ 0, & if\ \ (j < i)\ \ or\ \ (j > i + m) \\ 1 - \sum\limits_{k=i+1}^{i+m} p_{ik}, & if\ \ (i = j)\ \ and\ (i < n) \\ 1, & if\ \ (i = j)\ \ and\ (i = n) \end{cases}. \tag{2.3}$$

The rationale for the transition probabilities of FSA-ACK and FSA-FBP is the following. In the first condition, $j - i$ end-devices succeed and the number of contending end-devices is $n - i$, being i the number of end-devices that have succeeded in previous frames. In the second condition, the number of successful end-devices decreases, or is greater than the number of slots, which are both impossible. In the third condition, no new end-devices succeed and the process is not yet absorbed. Finally, in the fourth condition the process is absorbed.

The transition probabilities p_{ij} for DFSA can be defined as

$$p_{ij} = \begin{cases} P_s\left(m_i, j-i, n-i\right), & if\ \ (i < j \leq i + m_i) \\ 0, & if\ \ (j < i)\ \ or\ \ (j > i + m_i) \\ 1 - \sum\limits_{k=i+1}^{i+m_i} p_{ik}, & if\ \ (i = j)\ \ and\ (i < n) \\ 1, & if\ \ (i = j)\ \ and\ (i = n) \end{cases}. \tag{2.4}$$

The rationale for the transition probabilities of DFSA is identical to the one of FSA-ACK and FSA-FBP, but the number m_i of contention slots per frame changes dynamically as the number i of successful end-devices increases, i.e., $m_i = \lceil \rho\,(n-i) \rceil$.

2.5.1.2.3 Number of Contention Frames

Being Q a sub-matrix of P that contains the probabilities in the transient states (i.e., $i < n$ and $j < n$), the fundamental matrix N of the process can be defined as

$$\mathrm{N} = (\mathrm{I} - \mathrm{Q})^{-1}, \tag{2.5}$$

where each element n_{ij} of N is the average number of frames that the process remains in the transient state X_j, i.e., no new end-devices succeed for n_{ij} consecutive frames, given that it started in the transient state X_i. The 'time of absorption' t_0 is defined as the average number of frames until the process is absorbed and can be expressed as

$$t_0 = n_{01} + n_{02} + ... + n_{0n-1} = \sum_{j=0}^{n-1} n_{0j}, \tag{2.6}$$

where recall that n is the total number of end-devices.

2.5.1.2.4 Delay and Energy Consumption

The *average delay*, expressed in seconds, required to resolve the contention process using FSA-ACK and FSA-FBP can be formulated as

$$\overline{T}_C^{\text{FSA-ACK}} = t_0 \cdot T_{frame}^{\text{FSA-ACK}}, \tag{2.7}$$

$$\overline{T}_C^{\text{FSA-FBP}} = t_0 \cdot T_{frame}^{\text{FSA-FBP}}, \tag{2.8}$$

where t_0 is the average number of contention frames required to resolve the contention process, $T_{frame}^{\text{FSA-ACK}}$ and $T_{frame}^{\text{FSA-FBP}}$ are the duration of a frame in FSA-ACK and FSA-FBP, respectively, which can be expressed as

$$T_{frame}^{\text{FSA-ACK}} = m\,(T_{data} + T_{ack} + 2T_{IFS}) + T_{IFS} + T_{FBP}^{\text{FSA-ACK}}, \tag{2.9}$$

$$T_{frame}^{\text{FSA-FBP}} = mT_{data} + 2T_{IFS} + T_{FBP}^{\text{FSA-FBP}}, \tag{2.10}$$

where m is the number of contention slots per frame, T_{data}, T_{ack}, T_{IFS}, $T_{FBP}^{\text{FSA-ACK}}$ and $T_{FBP}^{\text{FSA-FBP}}$ are the time of transmission of a data packet, an ACK, the duration of an IFS, and the time of transmission of a FBP in FSA-ACK and FSA-FBP, respectively.

The *average delay*, expressed in seconds, required to resolve the contention process using DFSA can be formulated as

$$\overline{T}_C^{\text{DFSA}} = \sum_{j=0}^{n-1} n_{0j} \cdot T_{frame}^{\text{DFSA}}(j). \tag{2.11}$$

where j is the number of end-devices that have succeeded in a given point in time, n_{0j} is the average number of frames for which the contention process remains with j end-devices with success, and $T_{frame}^{\text{DFSA}}(j)$ is the duration of a frame, which can be expressed as

$$T_{frame}^{\text{DFSA}}(j) = m_j \left(T_{data} + T_{ack} + 2T_{IFS}\right) + T_{IFS} + T_{FBP}^{\text{DFSA}}, \tag{2.12}$$

where $m_j = \lceil \rho\,(n-j) \rceil$ is the number of contention slots in a frame where $n-j$ end-devices have not succeeded yet, and T_{FBP}^{DFSA} is the time of transmission of a FBP in DFSA.

The *average energy consumed by the coordinator* in a contention process using FSA-FBP is denoted by $\overline{E}_{coord}^{\text{FSA-FBP}}$. In FSA-FBP, the coordinator executes the following operations repeatedly in every frame: *(i)* listens to the channel for the m slots of each frame in order to receive incoming data packets, *(ii)* keeps in idle state for the duration of two IFS, and *(iii)* transmits a FBP packet at the end of the frame. Therefore, the average energy consumption of the coordinator using FSA-FBP can be formulated as

$$\overline{E}_{coord}^{\text{FSA-FBP}} = t_0 \left(m\rho_{rx}T_{data} + 2\rho_\sigma T_{IFS} + \rho_{tx}T_{FBP}^{\text{FSA-FBP}}\right). \tag{2.13}$$

The *average energy consumed by the coordinator* in a contention process using FSA-ACK and DFSA are denoted by $\overline{E}_{coord}^{\text{FSA-ACK}}$ and $\overline{E}_{coord}^{\text{DFSA}}$, respectively. In FSA-ACK and DFSA, the coordinator executes the following operations repeatedly in every frame: *(i)* listens to the channel for the slots of each frame in order to receive incoming data packets, *(ii)* responds with an ACK to every data packet decoded or sleeps for the time of transmission of an ACK, *(iii)* keeps in idle state for the duration of an IFS, and *(iv)* transmits a FBP packet at the end of the frame. Therefore, the average energy

consumption of the coordinator using FSA-ACK and DFSA can be formulated as

$$\overline{E}_{coord}^{\text{FSA-ACK}} = t_0\big(m\left(\rho_{rx}T_{data} + 2\rho_{sleep}T_{IFS} + \rho_{sleep}T_{ack}\right) + \rho_\sigma T_{IFS} + \rho_{tx}T_{FBP}^{\text{FSA-ACK}}\big) + n\big(\left(\rho_{tx} - \rho_{sleep}\right)T_{ack} + 2\left(\rho_\sigma - \rho_{sleep}\right)T_{IFS}\big), \quad (2.14)$$

$$\overline{E}_{coord}^{\text{DFSA}} = \sum_{j=0}^{n-1} n_{0j}\big(m_j\left(\rho_{rx}T_{data} + 2\rho_{sleep}T_{IFS} + \rho_{sleep}T_{ack}\right) + \rho_\sigma T_{IFS} + \rho_{tx}T_{FBP}^{\text{DFSA}}\big) + n\big(\left(\rho_{tx} - \rho_{sleep}\right)T_{ack} + 2\left(\rho_\sigma - \rho_{sleep}\right)T_{IFS}\big). \quad (2.15)$$

The *average energy consumed by all the end-devices* in a contention process using FSA-FBP, FSA-ACK and DFSA is denoted by $\overline{E}_{devices}^{\text{FSA-FBP}}$, $\overline{E}_{devices}^{\text{FSA-ACK}}$ and $\overline{E}_{devices}^{\text{DFSA}}$, respectively, which can be expressed as

$$\overline{E}_{devices}^{\text{FSA-FBP}} = \sum_{j=0}^{n-1} n_{0j}\big((n-j)E_{data}^{\text{FSA-FBP}} + jE_{sleep}^{\text{FSA-FBP}}\big), \quad (2.16)$$

$$\overline{E}_{devices}^{\text{FSA-ACK}} = \sum_{j=0}^{n-1} n_{0j}\big((n-j)E_{data}^{\text{FSA-ACK}} + jE_{sleep}^{\text{FSA-ACK}}\big), \quad (2.17)$$

$$\overline{E}_{devices}^{\text{DFSA}} = \sum_{j=0}^{n-1} n_{0j}\big((n-j)E_{data}^{\text{DFSA}} + jE_{sleep}^{\text{DFSA}}(j)\big), \quad (2.18)$$

where j is the number of end-devices that have succeeded in a given point in time, n_{0j} is the average number of frames for which the process remains with j end-devices with success, $n-j$ is the number of end-devices that have not succeeded yet, $E_{data}^{\text{FSA-FBP}}$, $E_{data}^{\text{FSA-ACK}}$ and E_{data}^{DFSA} are the energy consumed by an end-device in a frame where it transmits its data packet using FSA-FBP, FSA-ACK and DFSA, and $E_{sleep}^{\text{FSA-FBP}}$, $E_{sleep}^{\text{FSA-ACK}}$ and $E_{sleep}^{\text{DFSA}}(j)$ are the energy consumed by an end-device in a frame where it sleeps using FSA-FBP, FSA-ACK and DFSA, respectively.

In FSA-FBP, an end-device that has not succeeded yet to transmit its data packet executes the following operations in every frame: *(i)* it transmits the data packet in one of the m slots, *(ii)* sleeps for the other $m-1$ slots, *(iii)* keeps in idle state for two IFS,

and *(iv)* receives the FBP. Thus, the value of $E_{data}^{\text{FSA-FBP}}$ can be obtained as

$$E_{data}^{\text{FSA-FBP}} = \rho_{tx} T_{data} + (m-1)\,\rho_{sleep} T_{data} + 2\rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}^{\text{FSA-FBP}}. \quad (2.19)$$

In FSA-ACK and DFSA, an end-device that has not succeeded yet to transmit its data packet executes the following operations in every frame: *(i)* it transmits the data packet in one of the m slots, *(ii)* receives an ACK packet, *(iii)* sleeps for the other $m-1$ slots, *(iv)* keeps in idle state for two IFS, and *(v)* receives the FBP. Thus, the value of $E_{data}^{\text{FSA-ACK}}$ and $E_{data}^{\text{DFSA}}(j)$ can be obtained as

$$E_{data}^{\text{FSA-ACK}} = \rho_{tx} T_{data} + \rho_{rx} T_{ack} + 2\rho_{\sigma} T_{IFS} + (m-1)\,\rho_{sleep}\,(T_{data} + T_{ack} + 2T_{IFS}) + \rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}^{\text{FSA-ACK}}, \quad (2.20)$$

$$E_{data}^{\text{DFSA}}(j) = \rho_{tx} T_{data} + \rho_{rx} T_{ack} + 2\rho_{\sigma} T_{IFS} + (m_j-1)\,\rho_{sleep}\,(T_{data} + T_{ack} + 2T_{IFS}) + \rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}^{\text{DFSA}}. \quad (2.21)$$

The end-devices that have succeeded in transmitting their data in a given frame will not contend in subsequent frames. Then, the successful end-device sleeps for the duration of the frame. The energy consumed by an end-device in a frame where it sleeps using FSA-FBP, FSA-ACK and DFSA can be expressed as

$$E_{sleep}^{\text{FSA-FBP}} = \rho_{sleep} \cdot T_{frame}^{\text{FSA-FBP}}, \quad (2.22)$$

$$E_{sleep}^{\text{FSA-ACK}} = \rho_{sleep} \cdot T_{frame}^{\text{FSA-ACK}}, \quad (2.23)$$

$$E_{sleep}^{\text{DFSA}}(j) = \rho_{sleep} \cdot T_{frame}^{\text{DFSA}}(j). \quad (2.24)$$

In the following sections, the delay and energy models are validated with MATLAB simulations. The analytical results have been compared to the simulation within a range of frame lengths for different numbers of end-devices.

2.5.1.3 Model Validation and Performance Evaluation

In this section, we use the theoretical models formulated in Section 2.5.1.2 to evaluate the delay and the energy consumption for FSA-ACK, FSA-FBP and DFSA. We first describe the considered scenario. Then, we analyze how the delay and the energy consumption are influenced by the number m of slots per frame in FSA-ACK and FSA-FBP, the value of ρ in DFSA, and the number of end-devices in the network. We discuss the numerical results and determine the criteria to minimize the delay and the energy consumption.

2.5.1.3.1 Scenario

The system parameters used to validate the analytical models and to evaluate the performance are summarized in Table 2.1. They have been selected according to the IEEE 802.15.4 standard [79] and from the specifications of the CC2520 radio transceiver [80]. We have considered that each end-device has one data packet with a payload of 114 bytes ready to transmit in every DCR.

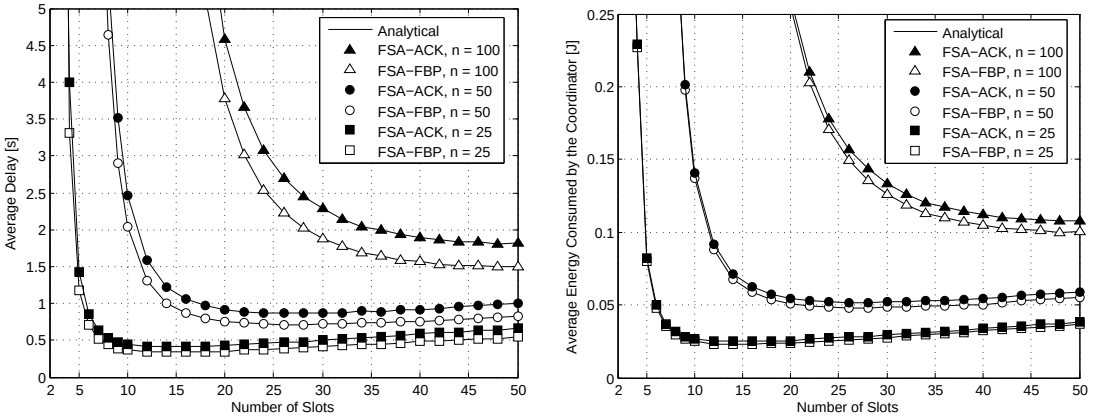
In FSA-FBP, the length of the FBP payload has been set to attach 2 bits per slot that inform about the status of each slot. In DFSA and FSA-ACK, the coordinator responds with an ACK to each data packet decoded successfully in each slot and it transmits a short FBP packet at the end of each frame to enable the synchronization of the end-devices.

All the packets are composed of a physical layer preamble, a MAC header, a payload and a cyclic redundancy code (CRC) of 2 bytes for error control.

We have validated the proposed theoretical models by means of computer-based simulations. The results of 1000 simulation samples have been averaged for each test case. Results show a tight match between analysis and simulation in all tested cases, thus validating the correctness of the theoretical analyses.

TABLE 2.1: System Parameters for FSA-ACK, FSA-FBP and DFSA

Parameter	**Value**	**Parameter**	**Value**
MAC header	8 bytes	Data-rate	250 kbps
Data payload	114 bytes	T_{data}	4.1 ms
$T_{preamble}$	160 μs	T_{IFS}	192 μs
ρ_{tx}	100.8 mW	T_{ACK}	512 μs
$\rho_{rx} = \rho_{\sigma}$	66.9 mW	ρ_{sleep}	60 nW



(a) Average delay

(b) Average energy consumed by the coordinator

FIGURE 2.7: (a) Average delay (in seconds) required to terminate a DCR over the frame length using FSA, and (b) Average energy consumed by the coordinator in a DCR over the frame length using FSA-ACK and FSA-FBP.

2.5.1.3.2 Configuration of FSA-ACK and FSA-FBP

The delay and the energy consumption of the coordinator using FSA-ACK and FSA-FBP are represented in Figure 2.7(a) and Figure 2.7(b), respectively, as a function of the number m of slots per frame and considering $n \in \{25, 50, 100\}$.

Results show that the delay and the energy consumption of the coordinator have a minimum at a certain value of m which represents the optimal point of operation. The minimum value can be found when the number of slots per frame is $m = n/2$. Therefore, the number of slots in FSA-ACK and FSA-FBP has to be adjusted according to the number of end-devices in order to minimize the delay and the energy consumption of the coordinator.

As it can be observed, when the value of m is below the optimum, the delay and the energy consumption in FSA-ACK and FSA-FBP tend to infinite if $m < n/4$ (approximately). Indeed, the shorter the frame length, the higher the probability of collision, the more frames required to complete contention process, and thus the higher the delay and the energy consumption.

When the number of slots per frame is above the optimum, the delay and the energy consumption of the coordinator increase almost linearly and proportionally to the number of slots. Indeed, although the average number of frames needed to complete a DCR tends to a constant value when the frame length increases, as shown in Figure 2.8, a higher number of slots per frame leads the coordinator to listen to the channel during longer time periods, yielding greater delay and energy consumption. However, since the

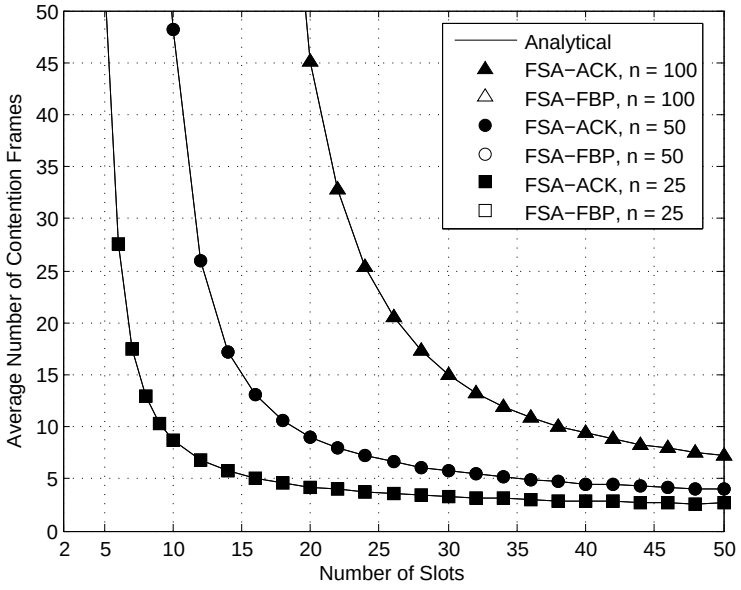


FIGURE 2.8: Average delay (in frames) required to terminate a DCR over the frame length using FSA-ACK and FSA-FBP.

delay and the energy consumption of the coordinator increase with a small slope after the minimum, it is better to over-dimension the number of slots when the number of end-devices is unknown in order to ensure that the condition $m > n/4$ holds.

The energy consumption per end-device in a DCR using FSA-ACK and FSA-FBP is represented in Figure 2.9 as a function of the number of slots per frame and considering $n \in \{25, 50, 100\}$.

Results show that the energy consumption per end-device tends to a minimum value when $m \geq n$ in FSA-ACK and FSA-FBP. Indeed, the higher the frame length, the lower the average number of frames where an end-device has to contend, and the lower the energy consumption of the end-devices. In addition, although a higher number of slots leads to longer frames and longer periods in sleep, the use of a very low power sleep mode yields reduced energy consumption when the value of m increases.

There is a trade-off between the delay, the energy consumption of the coordinator, and the energy consumption per end-device in FSA-ACK and FSA-FBP. When the number of slots per frame increases above $n/2$, the end-devices consume less energy in a DCR, at the cost of increasing the delay and the energy consumption of the coordinator.

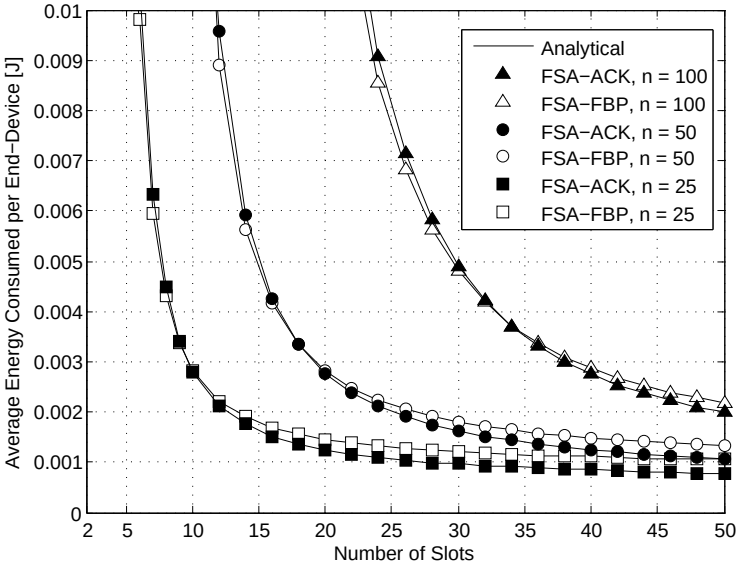
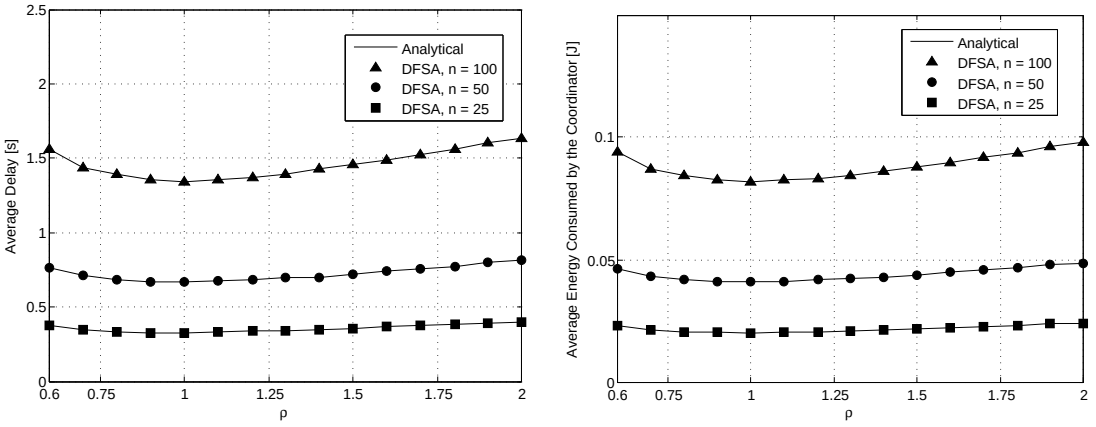


FIGURE 2.9: Average energy consumed per end-device in a DCR over the frame length using FSA-ACK and FSA-FBP.



(a) Average delay

(b) Average energy consumed by the coordinator

FIGURE 2.10: (a) Average delay (in seconds) required to terminate a DCR over the value of ρ in DFSA, and (b) Average energy consumed by the coordinator in a DCR over the value of ρ in DFSA.

2.5.1.3.3 Configuration of DFSA

The delay and the energy consumption of the coordinator using DFSA are represented in Figure 2.10(a) and Figure 2.10(b), respectively, as a function of ρ and considering $n \in \{25, 50, 100\}$. Recall that in DFSA the number m_i of slots per frame is proportional to the number $\hat{n}_i$ of end-devices that contend in every frame, i.e., $m_i = \lceil \rho \cdot \hat{n}_i \rceil$, where ρ is a positive real number.

Results show that the delay and the energy consumption of the coordinator have a minimum at $\rho = 1$. Therefore, the number of slots per frame in DFSA has to be

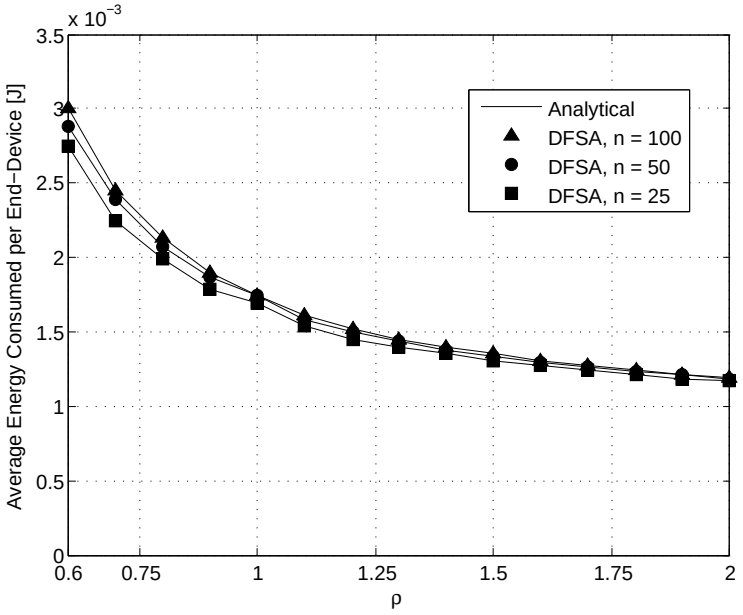


FIGURE 2.11: Average energy consumed per end-device in a DCR over the value of ρ in DFSA.

adjusted to be equal to the number of end-devices that contend in every frame in order to minimize the delay and the energy consumption of the coordinator.

As it can be observed, the delay and the energy consumption of the coordinator slightly increase when $\rho \neq 1$. Indeed, if $\rho < 1$, the number of slots per frame becomes lower, the probability of collision in a given slot increases, and the number of frames required to complete a DCR increases. Therefore, the coordinator has to transmit more FBP packets and its energy consumption in transmission mode increases. If $\rho > 1$, the number of slots per frame increases, the probability of collision decreases, and the number of frames required to complete a DCR decreases. However, the delay and the energy consumption of the coordinator increase because the number of slots per frame is higher, thus leading the coordinator to waste more energy in listening mode.

The energy consumption per end-device in a DCR using DFSA is represented in Figure 2.11 as a function of ρ and considering $n \in \{25, 50, 100\}$.

Results show that the energy consumption per end-device tends to a minimum value when ρ increases. Indeed, the higher the frame length, the lower the energy wasted in collisions. It is worth noting that the energy consumption per end-device is almost insensitive to the number of end-devices. Indeed, for a given ρ, the average number of retransmissions of an end-device in DFSA does not change with the number of end-devices because the probability that an end-device succeeds in a given frame is constant

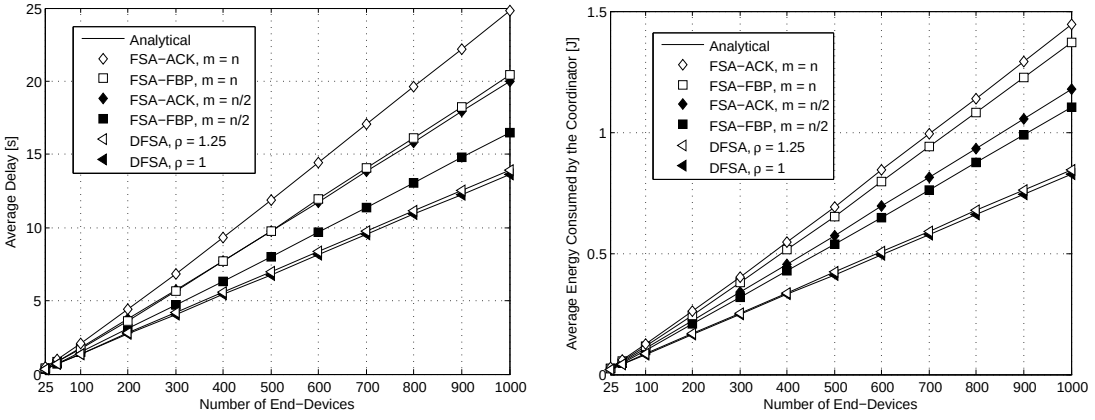
for any number of end-devices in the network thanks to the dynamic reconfiguration of the frame length.

2.5.1.3.4 Delay and Energy Performance in Dense Networks

The delay, the energy consumption of the coordinator and the energy consumption per end-device are represented in Figure 2.12(a), Figure 2.12(b) and Figure 2.13, respectively, for FSA-ACK, FSA-FBP and DFSA as a function of the number of end-devices (from 25 to 1000 end-devices). For each protocol, we have considered the number of slots per frame that minimizes the delay and the energy consumption of the coordinator (i.e., $m = n/2$ in FSA-ACK and FSA-FBP, and $\rho = 1$ in DFSA) and the number of slots per frame that minimize the energy consumption per end-device (i.e., $m = n$ in FSA-ACK and FSA-FBP, and $\rho = 1.25$ in DFSA).

As it can be observed in Figure 2.12(a) and Figure 2.12(b), the delay and the energy consumption of the coordinator increase almost linearly with the increasing value of n in FSA-ACK, FSA-FBP and DFSA. As it could be expected, the delay and the energy consumption of the coordinator are minimum in each protocol when $m = n/2$ in FSA-ACK and FSA-FBP, and $\rho = 1$ in DFSA.

Results show that FSA-FBP outperforms FSA-ACK in terms of delay. This is due to the fact that the duration of a frame in FSA-ACK is longer than in FSA-FBP. Indeed, although the duration of the FBP packets is shorter in FSA-ACK, the duration



(a) Average delay

(b) Average energy consumed by the coordinator

FIGURE 2.12: (a) Average delay (in seconds) required to terminate a DCR over the number of end-devices using FSA-ACK, FSA-FBP and DFSA, and (b) Average energy consumed by the coordinator in a DCR over the number of end-devices using FSA-ACK, FSA-FBP and DFSA.

of the slots in FSA-ACK is much longer because it is adjusted to fit a data packet and an ACK. Similarly, FSA-FBP outperforms FSA-ACK in terms of energy consumption of the coordinator. Indeed, in FSA-ACK the coordinator consumes more energy in transmission mode due to the transmission of ACK packets. The differences in delay and energy consumption of the coordinator between the two protocols increase as the number of end-devices increases. It is worth noting that DFSA outperforms FSA-ACK and FSA-FBP in terms of delay and energy consumption of the coordinator thanks to the adaptation of the frame length to the number contenders in every frame.

If the protocols are configured with their optimal number of slots that minimize the delay and the energy consumption of the coordinator, DFSA provides delay reductions of a 17% with respect to FSA-FBP and 32% with respect to FSA-ACK; and FSA-FBP provides delay reductions of a 18% with respect to FSA-ACK. In terms of energy consumption of the coordinator, DFSA provides energy savings of a 23% with respect to FSA-FBP and 28% with respect to FSA-ACK; and FSA-FBP provides energy savings of a 6% with respect to FSA-ACK. Therefore, the use of DFSA and FSA-FBP can improve considerably the delay and the energy consumption of the coordinator in M2M networks where the number of end-devices can be large.

As it can be observed in Figure 2.13, the energy consumption per end-device using FSA-FBP increases linearly with the number of end-devices. Indeed, since the payload of

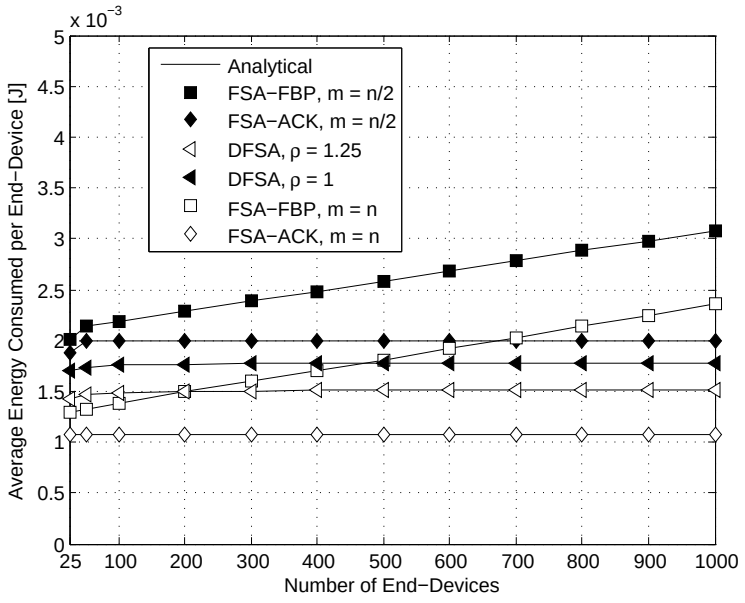


FIGURE 2.13: Average energy consumed per end-device in a DCR over the number of end-devices using FSA-ACK, FSA-FBP and DFSA.

the FBP packet in FSA-FBP is proportional to the number of slots, the energy consumed by an end-device in reception mode also increases with the number of slots. Therefore, if $m \geq n$, the energy consumption per end-device in FSA-FBP increases as the number of end-devices increases. On the contrary, the energy consumption per end-device using DFSA and FSA-ACK is almost insensitive to the number of end-devices. Indeed, in FSA-ACK and DFSA the end-devices receive a very short FBP packet in every frame, with constant length regardless of the number of slots, and listen for the duration of a short ACK in every transmission attempt. It is worth noting that FSA-ACK outperforms FSA-FBP and DFSA in terms of energy consumption per end-device.

As it could be expected, the energy consumption per end-device is minimum in each protocol when $m = n$ in FSA-ACK and FSA-FBP, and $\rho = 1.25$ in DFSA. With these configurations, FSA-ACK provides energy savings of more than a 28% with respect to DFSA and 54% with respect to FSA-FBP when $n = 1000$ end-devices. However, if the frame length of FSA-ACK is not well configured, e.g., $m = n/2$, then DFSA outperforms FSA-ACK in terms of energy consumption per end-device, and DFSA provides energy savings of more than a 50% with respect to FSA-FBP and 24% with respect to FSA-ACK when $n = 1000$ end-devices.

2.5.1.4 Conclusions

In this section, a theoretical analysis has been proposed to compute the average delay and energy consumption in data collection networks where a large group of end-devices transmit data periodically to a coordinator using Frame Slotted-ALOHA (FSA) based Access protocols. In particular, two variants of FSA (named FSA-ACK and FSA-FBP) and the Dynamic FSA (DFSA) protocol have been considered. While the frame length m is constant in FSA-ACK and FSA-FBP, it is adjusted dynamically in DFSA to be proportional to the number n_i of contenders in every frame, i.e., $m = \lceil \rho \cdot n_i \rceil$. In addition, while in FSA-FBP the coordinator transmits a feedback packet (FBP) to inform about the state of the slots in every frame, in FSA-ACK and DFSA the coordinator responds with an ACK to each data packet decoded successfully in each slot.

The proposed theoretical models are based on absorbing Markov chains and have been validated using computer simulations. Results show that there is an optimum frame length in FSA-ACK and FSA-FBP which minimizes the delay and the energy

consumption of the coordinator. This optimum value depends on the number n of end-devices, being $m = n/2$ for FSA-ACK and FSA-FBP. Results also show that the energy consumption per end-device tends to a minimum value when $m \geq n$ in FSA-ACK and FSA-FBP. Therefore, the number of slots in FSA-ACK and FSA-FBP has to be adjusted according to the number of end-devices and there is a trade-off between the delay, the energy consumption of the coordinator, and the energy consumption per end-device. When the number of slots per frame increases above the number of end-devices, each end-device consumes less energy in a data collection round, but the delay and the energy consumption of the coordinator slightly increase.

For DFSA, results show that the number of slots per frame has to be adjusted to be equal to the number of end-devices that contend in every frame in order to minimize the delay and the energy consumption of the coordinator, i.e., $\rho = 1$. In addition, the energy consumption per end-device is almost insensitive to the number of end-devices, thanks to the dynamic reconfiguration of the frame length, and it tends to a minimum value when ρ increases. Therefore, there is a trade-off between the delay, the energy consumption of the coordinator, and the energy consumption per end-device in DFSA. When the value of ρ increases, the end-devices consume less energy, but the delay and the energy consumption of the coordinator slightly increase.

The delay and the energy consumption of the coordinator increase almost linearly with the increasing number of end-devices in FSA-ACK, FSA-FBP and DFSA. Results show that DFSA outperforms FSA-ACK and FSA-FBP in terms of delay and energy consumption of the coordinator, and FSA-FBP outperforms FSA-ACK. The energy consumption per end-device using FSA-FBP increases linearly with the number of end-devices. Contrarily, the energy consumption per end-device using DFSA and FSA-ACK is almost insensitive to the number of end-devices. Results show that FSA-ACK outperforms FSA-FBP and DFSA in terms of energy consumption per end-device.

2.5.2 Tree Splitting-based Access Protocols

This section focuses on the design, analysis and performance evaluation of the LP-CTA and LP-DQ protocols proposed in this thesis. The operation of LP-CTA and LP-DQ is decribed in Section 2.5.2.1. Section 2.5.2.2 presents the theoretical analysis of LP-CTA and LP-DQ and formulates the delay and energy models. Section 2.5.2.3 is devoted to validate the models and to evaluate the performance of both LP-CTA and LP-DQ through comprehensive computer-based simulations. The comparison with the performance of the FSA and DFSA protocols is also presented in this section.

2.5.2.1 MAC Protocols Description

2.5.2.1.1 Low Power Contention Tree-based Access (LP-CTA)

The frame structure of LP-CTA is shown in Figure 2.14. At the beginning of each frame, the end-devices choose randomly one of the slots of the frame to transmit their data packet. However, the way that collisions are resolved in LP-CTA is different to the approach in Frame Slotted-ALOHA. In LP-CTA, the end-devices are organized into sub-groups using an m-ary tree-splitting algorithm [20]. An example of the operation of the m-ary tree-splitting is represented in Figure 2.15.a. Each node of the tree represents a frame of m=3 slots, and the number in each slot denotes the number of end-devices that transmit in that slot. At the first frame, all the end-devices transmit their data packet. When two or more end-devices collide in a specific slot, a new frame is assigned only to the end-devices that caused the collision in order to reattempt access, and they are queued into a logical queue referred to as Collision Resolution Queue (CRQ). Therefore, if there are k slots with collision, then k new frames are scheduled after the current frame, and k new sub-groups of end-devices are queued into CRQ. The process is repeated, frame after frame, leading to the formation of a tree whose expansion stops at frames which contain only empty (0) and/or successful slots (1).

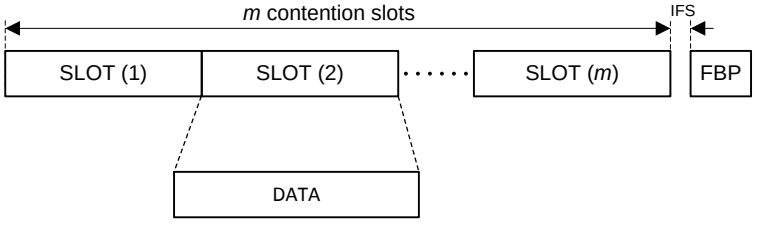
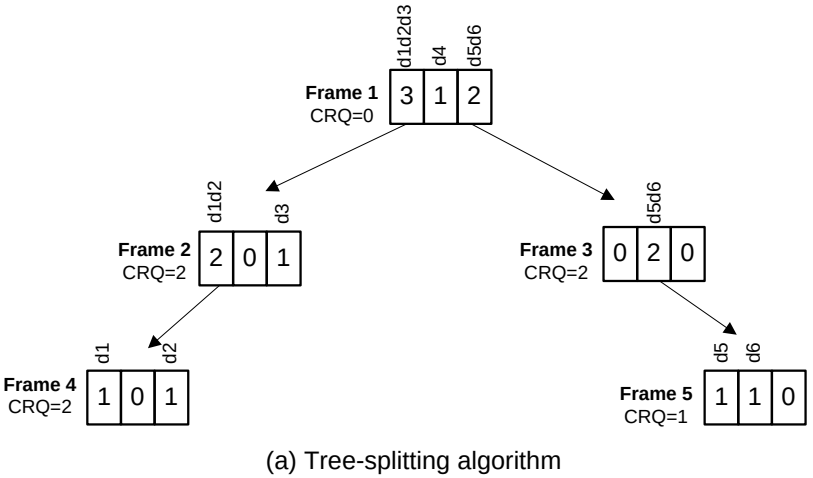


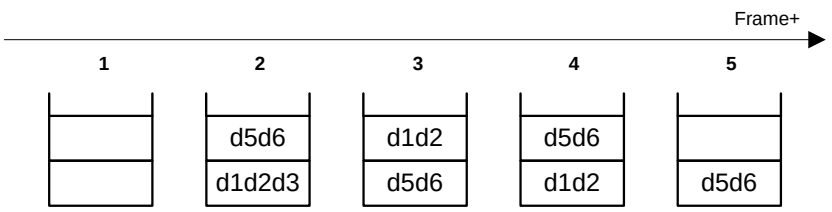
FIGURE 2.14: Frame structure of LP-CTA.

There are two basic ways of implementing the m-ary tree-splitting algorithm: *(i)* with serial search, or *(ii)* with parallel search. In the serial search, all the frames in each branch of the tree are executed before proceeding to the next branch. This means that the collision of a group of end-devices in a certain slot is solved before starting to solve the collision of other end-devices in a different slot. On the other hand, in the parallel search, all the frames in each level of the tree are executed before proceeding to the next level. This means that the resolution of collisions in different slots is done interleaved in time. In LP-CTA, we have implemented the parallel search by using the logical CRQ, which allows reducing the listening periods of the end-devices, thus reducing their energy consumption.

The logical CRQ is represented at each end-device by two integer numbers: *(i)* the position of the end-device in CRQ, and *(ii)* the length of CRQ, i.e., number of sub-groups of end-devices waiting to retransmit in another frame. The length of CRQ is updated by the coordinator at the end of each frame according to the following rules: *1)* it is incremented by the number of slots with collision in the current frame; and *2)* if the



(a) Tree-splitting algorithm



(b) Contents of CRQ in each frame of the process

FIGURE 2.15: Example of LP-CTA with 6 end-devices (d1 to d6) and 3 slots: (a) tree-splitting algorithm, and (b) time diagram with the contents of CRQ.

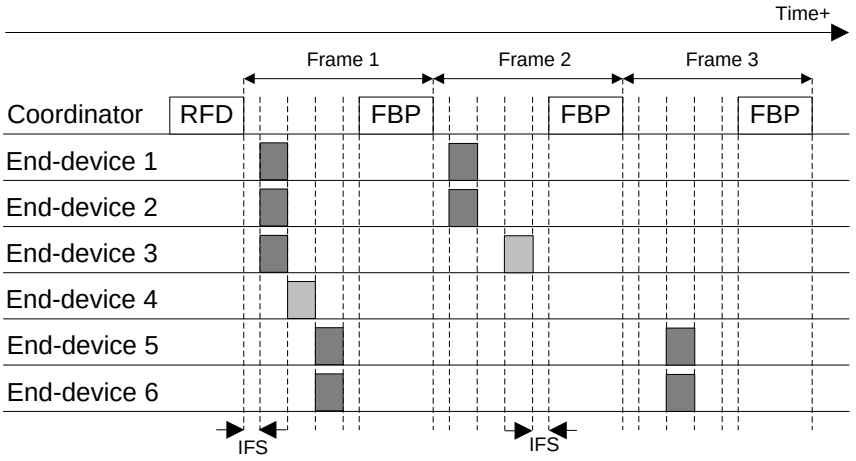


FIGURE 2.16: Example of the time diagram of LP-CTA with 6 end-devices and 3 contention slots.

length of CRQ>0 (i.e., CRQ is not empty), it is decremented by one after the current frame. The coordinator broadcasts the length of CRQ and the state of the m contention slots (success, empty, or collision) in every FBP. With this information, an end-device that collided in a given frame can compute its new position in CRQ. The position of an end-device in CRQ is always decremented by one at the end of every frame. When an end-device occupies the first position, it transmits again in the next frame. Therefore, the end-devices must only receive the FBP in those frames where they contend in order to know whether they succeed (to leave CRQ) or collide (to enter again in CRQ), and they switch to sleep mode during those frames where they do not contend.

The contents of CRQ in every frame are shown in the example illustrated in Figure 2.15.b. Figure 2.16 shows the time diagram for the realization of the tree depicted in Figure 2.15. For the sake of simplicity, only the first three frames of the parallel search implementation have been represented. At frame 1, all the end-devices (d1 to d6) contend: d1, d2 and d3 collide in slot 1; d4 succeeds in slot 2; d5 and d6 collide in slot 3. Thus, d1, d2 and d3 enter in the first position of CRQ; d5 and d6 enter in the second position. At frame 2 only d1, d2 and d3 contend (because they occupy the first position in CRQ): d1 and d2 collide and enter in CRQ again; d3 succeeds and leaves CRQ; d5 and d6 move to the first position of CRQ. At frame 3, d5 and d6 contend, collide, and enter in the second position of CRQ again; d1 and d2 move to the first position of CRQ. At frame 4, d1 and d2 contend and succeed; d5 and d6 move to the first position of CRQ. At frame 5, d5 and d6 contend and succeed.

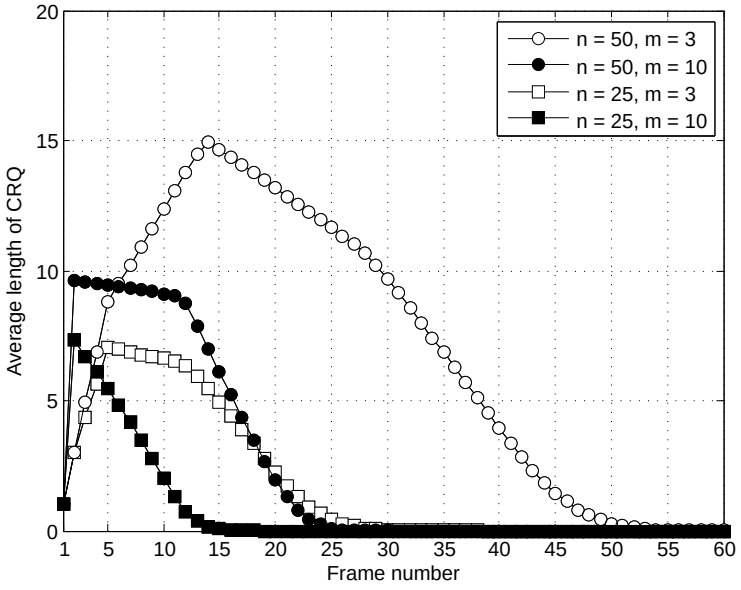


FIGURE 2.17: Evolution of the average length of CRQ over the number of frame in a DCR.

Figure 2.17 shows the evolution of the average length of CRQ over the number of frame in a DCR by considering $n \in \{25, 50\}$ and $m \in \{3, 10\}$. The results have been obtained through computer-based simulations using MATLAB. The results of 1000 simulation samples have been averaged for each test case. At the beginning of a DCR, the length of CRQ grows very rapidly due to the collisions of data packets in the contention slots because all the end-devices attempt to trasmit simultaneously. The length of CRQ increases up to a certain value which depends on the number m of contention slots and the number n of end-devices. The higher the value of m, the lower the probability of collision in a given frame, and thus the lower the maximum length of CRQ. On the contrary, the higher the value of n, the higher the number of collisions per frame, and thus the higher the maximum length of CRQ. The m-ary tree splitting algorithm splits collisions into sub-groups and allows to resolve them in frames allocated in subsequent levels of the contention tree. Consequently, the length of CRQ decreases and converges to zero when all collisions have been resolved.

2.5.2.1.2 Low Power Distributed Queuing (LP-DQ)

The basic idea of the LP-DQ protocol is to concentrate access requests in a short contention window while data transmission is kept collision-free. When an end-device succeeds in transmitting its access request, it waits for a collision-free data slot to transmit its data packet. The frame structure of LP-DQ is divided in three parts as shown in

Figure 2.18: *(i)* m contention slots devoted to the transmission of access requests, *(ii)* one collision-free data slot, and *(iii)* a feedback packet (FBP). A guard time called Inter Frame Space (IFS) is left between reception and transmission modes to compensate propagation and processing delays and the time required to switch the radio transceivers between reception and transmission.

At the beginning of every DCR, the end-devices randomly select one of the m contention slots to transmit an Access Request Sequence (ARS) packet. A given contention slot can be in one of three states: empty, i.e., no ARS has been transmitted, success, i.e., only one ARS has been transmitted, or collision, i.e., more than one end-device has transmitted in that slot. Depending on whether the ARS packet collides or is successfully decoded by the coordinator, every end-device is queued into one of two logical and distributed queues:

1) The end-devices that collide transmitting their ARS are queued into the Collision Resolution Queue (CRQ). The length of CRQ and the position of the end-devices in CRQ are updated by executing the m-ary tree-splitting algorithm represented in the example of Figure 2.19.a. Each node of the tree represents a frame of $m = 3$ contention slots, and the number in each slot denotes the number of end-devices that transmit an ARS in that slot. At frame 1, all the end-devices contend. If two or more end-devices collide in a slot, a new frame is assigned only to the end-devices that caused the collision in order to reattempt access, and they are queued into CRQ. Therefore, if there are k slots with collision, then k new frames are scheduled after the current frame, and k new sub-groups of end-devices are queued in CRQ. The end-devices in the first position of CRQ always contend in the next frame by selecting an access slot at random. The process is repeated leading to the formation of a tree whose expansion stops at frames which contain only empty and/or successful slots.

2) The end-devices that succeed in transmitting their ARS are queued into the Data Transmission Queue (DTQ). Even though any queue management strategy could be applied, we consider that the end-devices transmit their data packet in the collision-free data slot of subsequent frames according to a first-in first-out (FIFO) mechanism.

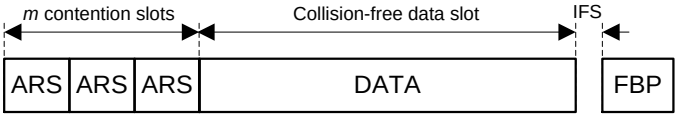
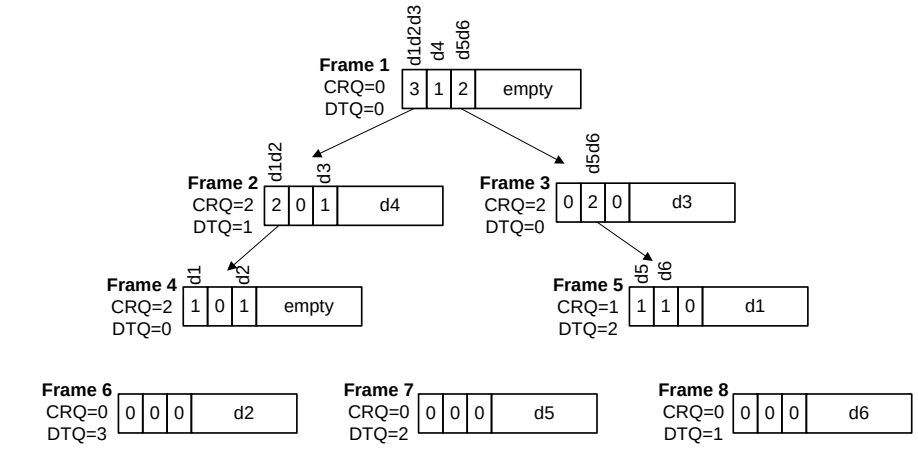
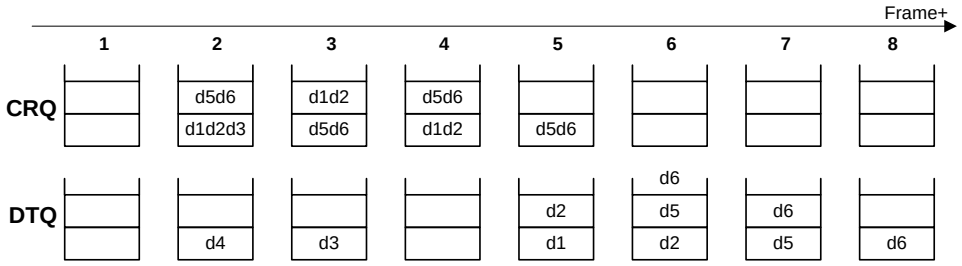


FIGURE 2.18: Frame structure of LP-DQ.



(a) Tree-splitting algorithm



(b) Contents of CRQ and DTQ in each frame of the process

FIGURE 2.19: Example of LP-DQ with 6 end-devices (d1 to d6) and 3 contention slots: (a) tree-splitting algorithm, and (b) contents of CRQ and DTQ in each frame.

When an end-device occupies the first position of DTQ, it transmits its data packet in the collision-free data slot of the next frame.

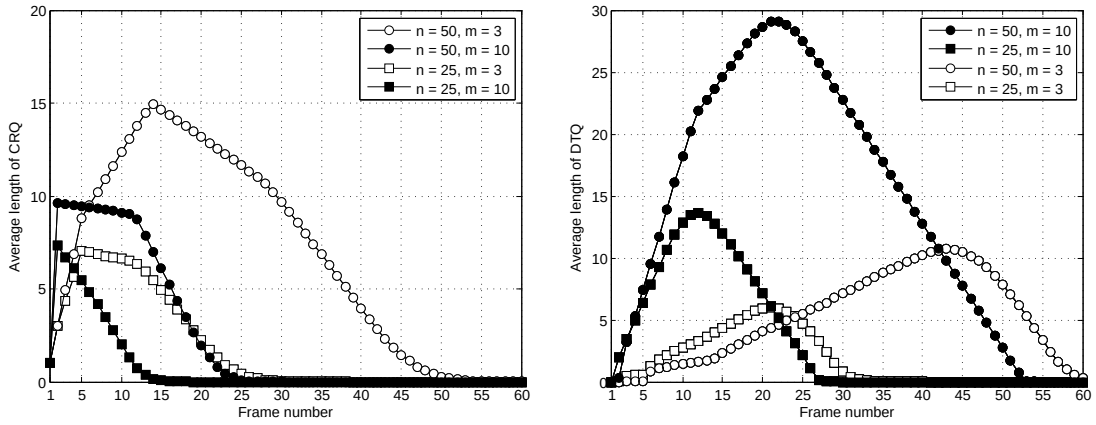
CRQ and DTQ are characterized at every end-device by 2 integer numbers: *i)* the position of the end-device in the queue, and *ii)* the length of the queue, i.e., total number of elements in the queue. The length of CRQ represents the number of sub-groups of end-devices waiting to retransmit an ARS. The length of DTQ represents the number of end-devices that have succeeded in transmitting an ARS and wait for a data transmission slot. The coordinator updates the length of CRQ and DTQ at the end of every frame according to the following rules: *1)* the length of CRQ is incremented by the number of contention slots with collision in the previous frame; *2)* if the length of CRQ>0, then it is decremented by one after the current frame; *3)* the length of DTQ is incremented by the number of contention slots with success in the previous frame; and *4)* if the length of DTQ>0, it is decremented by one if there was a data packet transmitted in the previous frame. In the example of Figure 2.19, the contents of the slots and the lengths of CRQ

and DTQ in each frame are shown in Figure 2.19.a. The contents of CRQ and DTQ are shown in Figure 2.19.b.

The coordinator broadcasts in every FBP the length of the two queues and the state of the m contention slots. With this information, an end-device which transmitted an ARS can compute its position in CRQ when it collided, or its position in DTQ when succeeded. The position of an end-device in CRQ and DTQ is always decremented by one at the end of each frame. Therefore, the end-devices receive the FBP in those frames where they transmit either ARS or data, and they switch to sleep mode in order to save energy in those frames where they do not transmit either ARS or data.

Figure 2.19 shows an example of LP-DQ with 6 end-devices (d1 to d6) and 3 contention slots. The contents of the ARS contention slots and the data slot of each frame are shown in Figure 2.19.a. The contents of CRQ and DTQ are shown in Figure 2.19.b. At frame 1, all the end-devices contend (d1, d2 and d3 collide in slot 1; d4 succeeds in slot 2; and d5 and d6 collide in slot 3). Thus, d1, d2 and d3 enter in the first position of CRQ; d5 and d6 enter in the second position of CRQ; and d4 enters in the first position DTQ. At frame 2, only d1, d2 and d3 contend (they are in the first position of CRQ), and d4 transmits data. d1 and d2 collide in slot 1 and enter in CRQ again; d3 succeeds in slot 3 and enters in DTQ. At frame 3: d5 and d6 contend (they collide and enter in CRQ again); and d3 transmits data. At frame 4: d1 and d2 contend (they succeed and enter in DTQ); since DTQ was empty, no end-device transmits data. At frame 5: d5 and d6 contend (they succeed and enter in DTQ); and d1 transmits data. At frame 6, 7 and 8: since CRQ is empty, no end-device contends; and d2, d5, and d6 transmit data, respectively.

Figure 2.20(a) and Figure 2.20(b) show the evolution of the average length of CRQ and DTQ, respectively, over the number of frame in a DCR by considering $n \in \{25, 50\}$ and $m \in \{3, 10\}$. The results have been obtained through computer-based simulations using MATLAB. The results of 1000 simulation samples have been averaged for each test case. At the beginning of a DCR, the length of CRQ grows very rapidly due to the collisions of ARS packets in the contention slots because all the end-devices attempt to get access to the channel. The length of CRQ increases up to a certain value which depends on the number m of contention slots and the number n of end-devices. The higher the value of m, the lower the probability of collision in a given frame, and thus the lower the maximum length of CRQ. On the contrary, the higher the value of n, the higher



(a) Length of CRQ

(b) Length of DTQ

FIGURE 2.20: (a) Evolution of the average length of CRQ over the number of frame in a DCR, and (b) Evolution of the average length of DTQ in a DCR.

the number of collisions per frame, and thus the higher the maximum length of CRQ. The m-ary tree splitting algorithm splits collisions into sub-groups and allows to resolve them in frames allocated in subsequent levels of the contention tree. Consequently, the length of CRQ decreases and converges to zero when all collisions have been resolved. When an end-device succeeds in transmitting its ARS packet in a given frame, it enters into DTQ, which grows almost linearly frame after frame until it reaches a certain value which depends on the number of contention slots and the number of end-devices. The higher the value of m, the higher the number of end-devices that succeed in a given frame, and the more rapidly increases the length of DTQ. Finally, the length of DTQ decreases linearly and converges to zero when all the end-devices have transmitted their data packet.

2.5.2.2 Analysis of Performance Metrics

In this section, we analyze the delay and energy consumption for the LP-CTA protocol, and the energy consumption for LP-DQ. Towards this end, we first have to obtain the average number of frames $\overline{d_n}$ required for an end-device to contend until it succeeds in transmitting a data packet in LP-CTA, or an access request in LP-DQ. Secondly, we need to calculate the total average number of frames $\overline{L_n}$ required to resolve the contention using LP-CTA, i.e., all the end-devices have sent their data packet successfully. Note that the particular expressions of $\overline{d_n}$ and $\overline{L_n}$ were derived in [59]. Then, using these parameters, we compute the average delay and the average energy consumed by the

coordinator using LP-CTA, and the average energy consumed per end-device in a DCR using LP-CTA and LP-DQ.

2.5.2.2.1 Number of Contention Frames per End-Device

The number of frames in which an end-device contends until it succeeds in transmitting a data packet in LP-CTA, or an access request in LP-DQ, is equivalent to the number d of tree levels required by an end-device to succeed. In the examples of Figure 2.15 and Figure 2.19, d is 1, 2, or 3 levels depending on the end-device.

According to the analysis developed in [59], the probability distribution of d, when the number of end-devices is n and the number of slots per frame is m, can be formulated as

$$\Pr(d|n) = \begin{cases} 0, & if \ \ (n = 1) \ \ and \ \ (d \neq 1) \\ 1, & if \ \ (n = 1) \ \ and \ \ (d = 1) \\ p_s(d) - p_s(d-1), & if \ \ (n \geq 2) \end{cases}, \tag{2.25}$$

where $p_s(d)$ is the probability that a slot in level d selected by a random end-device is not selected by any of the other $n-1$ end-devices, assuming that there are no transmission errors, and is given by

$$p_s(d) = m^d \frac{1}{m^d} \left(1 - \frac{1}{m^d}\right)^{n-1} = \left(1 - \frac{1}{m^d}\right)^{n-1}, \tag{2.26}$$

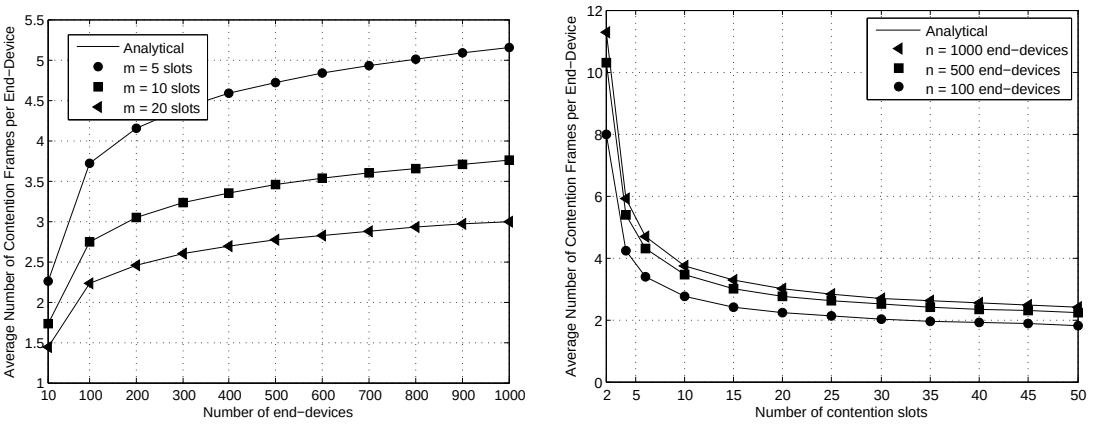
where $1/m^d$ is the probability that an end-device selects one of the slots in level d. The difference between $p_s(d)$ and $p_s(d-1)$ is the probability that the random end-device requires precisely d levels to be the only occupant of a contention slot.

The average number $\overline{d_n}$ of levels (or frames) required for an end-device to succeed in transmitting a data packet in LP-CTA, or an access request in LP-DQ, can be expressed as

$$\overline{d_n} = \sum_{d=1}^{\infty} d \Pr(d|n), \tag{2.27}$$

which is derived in [59] and can be formulated as

$$\overline{d_n} \simeq \log_m(n-1) + \left(\frac{1}{2} + \frac{\gamma}{\log m}\right) + \frac{1}{2n \log m}, \tag{2.28}$$



(a) (b)

FIGURE 2.21: Average number of contention frames per end-device (a) over the number of end-devices, and (b) over the number of contention slots per frame.

where the Euler's constant $\gamma \approx 0.5772$, and the logarithm base m is given by $\log_m n \equiv \log n / \log m$, with $\log n \equiv \ln n$.

Figure 2.21(a) shows the value of $\overline{d_n}$ as a function of the number of end-devices (from 10 to 1000) and considering $m = 5$, 10 and 20 contention slots per frame. It can be observed that $\overline{d_n}$ increases logarithmically with n for a given value of m. Figure 2.21(b) shows the value of $\overline{d_n}$ as a function of the number m of slots (from 2 to 50), and considering n =100, 500, and 1000 end-devices. The value of $\overline{d_n}$ decreases exponentially when m increases. It is worth noting that the value of $\overline{d_n}$ is finite and less than 12 frames when the number of slots is minimum ($m = 2$) and $n \leq 1000$ end-devices. In addition, the values of $\overline{d_n}$ are very similar when the number n of end-devices is either low or high regardless of the number m of slots.

2.5.2.2.2 Number of Frames in LP-CTA

The average number of frames required to resolve the contention using LP-CTA, denoted by $\overline{L_n}$, with n end-devices and m slots per frame, can be expressed as

$$\overline{L_n} = 1 + \sum_{k=1}^{\infty} c_n(m^k), \tag{2.29}$$

where $c_n(m^k)$ is the average number of contention slots with collisions in level k, i.e., the average number of frames generated in the next level $k+1$, and can be expressed as

$$c_n(m^k) = m^k \left[1 - [1 - p_1(k)]^n\right] - n[1 - p_1(k)]^{n-1}. \tag{2.30}$$

After some algebra and using classical Fourier analysis, the expression of $\overline{L_n}$ was formulated in [59] as

$$\overline{L_n} \simeq \frac{n}{\log m} - \frac{1}{m-1}, \tag{2.31}$$

which increases linearly with n for a given value of m.

2.5.2.2.3 Delay and Energy Consumption using LP-CTA

The *average delay* to resolve the contention using LP-CTA can be calculated as

$$\overline{T}_C^{\text{LP-CTA}} = \overline{L_n} \cdot T_{frame}^{\text{LP-CTA}}, \tag{2.32}$$

where $\overline{L_n}$ (2.31) is the average number of contention frames, and $T_{frame}^{\text{LP-CTA}}$ is the duration of a LP-CTA frame, which is given by

$$T_{frame}^{\text{LP-CTA}} = mT_{data} + 2T_{IFS} + T_{FBP}^{\text{LP-CTA}}, \tag{2.33}$$

where m is the number of contention slots and T_{data}, T_{IFS}, and $T_{FBP}^{\text{LP-CTA}}$ are the duration of a contention slot, an IFS, and the time of transmission of a FBP packet, respectively.

The *average energy consumed by the coordinator* is denoted by $\overline{E}_{coord}^{\text{LP-CTA}}$. In LP-CTA, the coordinator executes the following operations repeatedly: (i) listens to the channel for the m slots of each frame in order to receive incoming data packets, (ii) keeps in idle state for the duration of two IFS, and (iii) transmits a FBP packet after each frame. Therefore, the average energy consumption of the coordinator can be formulated as

$$\overline{E}_{coord}^{\text{LP-CTA}} = \overline{L_n}\left(m\rho_{rx}T_{data} + 2\rho_\sigma T_{IFS} + \rho_{tx}T_{FBP}^{\text{LP-CTA}}\right). \tag{2.34}$$

The time T_R between any two consecutive DCRs using LP-CTA can be expressed as

$$T_R = \overline{T}_{\text{LP-CTA}} + \overline{T}_{sleep}, \tag{2.35}$$

where $\overline{T}_{\text{LP-CTA}}$ is the average time elapsed since the DCR starts until an end-device succeeds in transmitting its single data packet to the coordinator, and $\overline{T}_{sleep}$ is the remaining time of the DCR in which the end-device sleeps until the next round starts.

The average time $\overline{T}_{\text{LP-CTA}}$ can be defined as

$$\overline{T}_{\text{LP-CTA}} = \left(\overline{CRQ}_{data} + \overline{CRQ}_{sleep}\right) T_{frame}^{\text{LP-CTA}}, \tag{2.36}$$

where $\overline{CRQ}_{data}$ is the average number of frames where an end-device contends (i.e., it transmits the data packet) until it succeeds; and $\overline{CRQ}_{sleep}$ is the average number of frames where an end-device is in CRQ without contending (i.e., it sleeps until it has to contend in a future frame).

The average energy consumed by one end-device in a DCR using LP-CTA, denoted by $\overline{E}_R^{\text{LP-CTA}}$, can be expressed as

$$\overline{E}_R^{\text{LP-CTA}} = \overline{E}_{\text{LP-CTA}} + \rho_{sleep}\overline{T}_{sleep}, \tag{2.37}$$

where $\overline{E}_{\text{LP-CTA}}$ is the average energy consumed by an end-device in the LP-CTA contention resolution process and can be expressed as

$$\overline{E}_{\text{LP-CTA}} = \overline{CRQ}_{data} E_{data}^{\text{LP-CTA}} + \overline{CRQ}_{sleep}\rho_{sleep} T_{frame}^{\text{LP-CTA}}. \tag{2.38}$$

where $E_{data}^{\text{LP-CTA}}$ is the energy consumed by the end-device in a frame where it contends. The end-device executes the following operations: (*i*) transmits data in 1 slot selected randomly, (*ii*) remains in sleep mode in the other m-1 slots, and (*iii*) listens to the channel to receive the FBP. Then, E_{data} can be formulated as

$$E_{data}^{\text{LP-CTA}} = \rho_{tx} T_{data} + (m-1)\,\rho_{sleep} T_{data} + 2\rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}^{\text{LP-CTA}}. \tag{2.39}$$

By substituting $\overline{T}_{sleep}$ from (2.35), $\overline{T}_{\text{LP-CTA}}$ (2.36), and $\overline{E}_{\text{LP-CTA}}$ (2.38) in (2.37), and after some basic algebra, the average energy consumed by an end-device in a DCR using LP-CTA can be expressed as

$$\overline{E}_R^{\text{LP-CTA}} = \overline{CRQ}_{data} E_{data}^{\text{LP-CTA}} + \overline{CRQ}_{sleep}\rho_{sleep} T_{frame}^{\text{LP-CTA}} + \rho_{sleep}\left(T_R - \overline{T}_{\text{LP-CTA}}\right). \tag{2.40}$$

Substituting (2.36) in (2.40), and after some basic algebra, $\overline{E}_R^{\text{LP-CTA}}$ can be formulated as

$$\overline{E}_R^{\text{LP-CTA}} = \overline{CRQ}_{data} E_{data}^{\text{LP-CTA}} + \rho_{sleep}\left(T_R - \overline{CRQ}_{data} T_{frame}^{\text{LP-CTA}}\right). \tag{2.41}$$

Since we assume that there are neither transmission errors nor capture effect, then $\overline{CRQ}_{data} = \overline{d_n}$ and $\overline{E}_R^{\text{LP-CTA}}$ can be expressed as

$$\overline{E}_R^{\text{LP-CTA}} = \overline{d_n}\left(E_{data}^{\text{LP-CTA}} - \rho_{sleep} T_{frame}^{\text{LP-CTA}}\right) + \rho_{sleep} T_R. \tag{2.42}$$

2.5.2.2.4 Energy Consumption using LP-DQ

The time T_R between any two consecutive DCRs using LP-DQ can be expressed as

$$T_R = \overline{T}_{\text{LP-DQ}} + \overline{T}_{sleep}, \tag{2.43}$$

where $\overline{T}_{\text{LP-DQ}}$ is the average time elapsed since the DCR starts until one end-device is capable of successfully transmitting a single data packet to the coordinator, which can be defined as

$$\overline{T}_{\text{LP-DQ}} = \left(\overline{CRQ}_{ARS} + \overline{CRQ}_{sleep} + \overline{DTQ}_{sleep} + \overline{DTQ}_{listen} + \overline{DTQ}_{data}\right) T_{frame}^{\text{LP-DQ}}, \tag{2.44}$$

where $\overline{CRQ}_{ARS}$ is the average number of frames where an end-device contends (i.e., it transmits an ARS); $\overline{CRQ}_{sleep}$ is the average number of frames where an end-device is in CRQ without contending (i.e., it sleeps until it has to contend); $\overline{DTQ}_{sleep}$ is the average number of frames where an end-device is in DTQ waiting for its collision-free data slot (i.e., it sleeps); $\overline{DTQ}_{listen}$ is the average number of frames where an end-device is in DTQ and listens to the feedback packet of the frame before the one where it has to transmit data in order to check whether the end-device in the previous position of DTQ has transmitted data successfully; $\overline{DTQ}_{data}$ is the average number of frames where an end-device is in DTQ and has to transmit data; and $T_{frame}^{\text{LP-DQ}}$ is the duration of a frame, which is given by

$$T_{frame}^{\text{LP-DQ}} = mT_{ARS} + T_{data} + 2T_{IFS} + T_{FBP}^{\text{LP-DQ}}, \tag{2.45}$$

where m is the number of contention slots and T_{ARS}, T_{data}, T_{IFS}, and $T_{FBP}^{\text{LP-DQ}}$ are the duration of a contention slot (for access requests), a collision-free data slot, an IFS, and the time of transmission of a FBP packet, respectively.

The average energy consumed by an end-device in a DCR, denoted by $\overline{E}_R^{\text{LP-DQ}}$, can be expressed as

$$\overline{E}_R^{\text{LP-DQ}} = \overline{E}_{\text{LP-DQ}} + \rho_{sleep}\overline{T}_{sleep}, \tag{2.46}$$

where $\overline{E}_{\text{LP-DQ}}$ is the average energy consumed by an end-device in the LP-DQ contention resolution process, which can be expressed as

$$\overline{E}_{\text{LP-DQ}} = \overline{CRQ}_{ARS}E_{ARS}^{\text{LP-DQ}} + \left(\overline{CRQ}_{sleep} + \overline{DTQ}_{sleep}\right)\rho_{sleep}T_{frame}^{\text{LP-DQ}} + \\ + \overline{DTQ}_{listen}E_{listen}^{\text{LP-DQ}} + \overline{DTQ}_{data}E_{data}^{\text{LP-DQ}}. \tag{2.47}$$

$E_{ARS}^{\text{LP-DQ}}$ is the energy consumed by an end-device in a frame where it contends. The end-device executes the following operations: (i) transmits an ARS in 1 contention slot selected randomly, (ii) keeps in sleep mode in the other $m-1$ slots and in the collision-free data slot, and (iii) listens to the channel to receive a FBP at the end of the frame. Then, $E_{ARS}^{\text{LP-DQ}}$ can be formulated as

$$E_{ARS}^{\text{LP-DQ}} = \left(\rho_{tx} + (m-1)\,\rho_{sleep}\right)T_{ARS} + \rho_{sleep}T_{data} + 2\rho_{\sigma}T_{IFS} + \rho_{rx}T_{FBP}^{\text{LP-DQ}}. \tag{2.48}$$

$E_{listen}^{\text{LP-DQ}}$ is the energy consumed by an end-device in a frame where it only listens to the feedback packet of the frame before it has to transmit data. The end-device executes the following operations: (i) remains in sleep mode in the m contention slots and in the collision-free data slot, and (ii) listens to the channel to receive the FBP. Then, $E_{listen}^{\text{LP-DQ}}$ can be formulated as

$$E_{listen}^{\text{LP-DQ}} = m\rho_{sleep}T_{ARS} + \rho_{sleep}T_{data} + 2\rho_{\sigma}T_{IFS} + \rho_{rx}T_{FBP}^{\text{LP-DQ}}. \tag{2.49}$$

$E_{data}^{\text{LP-DQ}}$ is the energy consumed by an end-device in a frame where it transmits a data packet. The end-device executes the following operations: (i) remains in sleep mode in the m contention slots, (ii) transmits data in the collision-free data slot, and (iii) listens to the channel to receive the FBP. Then, $E_{data}^{\text{LP-DQ}}$ can be formulated as

$$E_{data}^{\text{LP-DQ}} = m\rho_{sleep}T_{ARS} + \rho_{tx}T_{data} + 2\rho_{\sigma}T_{IFS} + \rho_{rx}T_{FBP}^{\text{LP-DQ}}. \tag{2.50}$$

By substituting $\overline{T}_{sleep}$ from (2.43), $\overline{T}_{\text{LP-DQ}}$ (2.44), and $\overline{E}_{\text{LP-DQ}}$ (2.47) in (2.46), and after some basic algebra, the average energy consumed by an end-device in a DCR using LP-DQ can be expressed as

$$\overline{E}_R^{\text{LP-DQ}} = \overline{CRQ}_{ARS} E_{ARS}^{\text{LP-DQ}} + \overline{DTQ}_{listen} E_{listen}^{\text{LP-DQ}} + \overline{DTQ}_{data} E_{data}^{\text{LP-DQ}} + \rho_{sleep} T_R - \rho_{sleep} \left(\overline{CRQ}_{ARS} + \overline{DTQ}_{listen} + \overline{DTQ}_{data} \right) T_{frame}^{\text{LP-DQ}}. \quad (2.51)$$

Since we assume that there are no transmission errors, then $\overline{CRQ}_{ARS} = \overline{d_n}$ (2.28), $\overline{DTQ}_{listen} = 1$, and $\overline{DTQ}_{data} = 1$. Therefore, $\overline{E}_R^{\text{LP-DQ}}$ can be expressed as

$$\overline{E}_R^{\text{LP-DQ}} = \overline{d_n} \left(E_{ARS}^{\text{LP-DQ}} - \rho_{sleep} T_{frame}^{\text{LP-DQ}} \right) + E_{listen}^{\text{LP-DQ}} + E_{data}^{\text{LP-DQ}} + \rho_{sleep} \left(T_R - 2T_{frame}^{\text{LP-DQ}} \right). \quad (2.52)$$

2.5.2.3 Model Validation and Performance Evaluation

In this section, we use the theoretical models formulated in Section 2.5.2.2 to evaluate the delay and the energy consumption for LP-CTA and LP-DQ. We first describe the considered scenario. Then, we analyze how the delay and the energy consumption are influenced by the number of slots per frame and the number of end-devices. We discuss the numerical results and determine the criteria to minimize the delay and the energy consumption. In addition, we compare the performance of LP-CTA, LP-DQ, DFSA, FSA-ACK and FSA-FBP.

2.5.2.3.1 Scenario

The system parameters used to validate the analytical models and to evaluate the performance are summarized in Table 2.2. They have been selected according to the IEEE 802.15.4 standard [79] and from the specifications of the CC2520 radio transceiver [80]. We have considered that the coordinator initiates one DCR every hour, i.e., $T_R = 3600$s. Each end-device has one data packet with a payload of 114 bytes ready to transmit in every DCR.

In LP-DQ and LP-CTA, the FBP payload includes 2 bits to inform about the status of each contention slot (i.e., empty, success, or collision) and 2 bytes for the length of CRQ. In addition, in LP-DQ the FBP payload includes 2 bytes for the length of DTQ. In LP-DQ, we consider that the payload of the ARS packets is empty in order to

TABLE 2.2: System Parameters for LP-CTA and LP-DQ

Parameter	**Value**	**Parameter**	**Value**
MAC header	8 bytes	Data-rate	250 kbps
Data payload	114 bytes	T_{data}	4.1 ms
$T_{preamble}$	160 μs	T_{IFS}	192 μs
ρ_{tx}	100.8 mW	T_{ARS}	480 μs
$\rho_{rx} = \rho_{\sigma}$	66.9 mW	ρ_{sleep}	60 nW

minimize the time and the energy wasted due to collisions in the access requests from the end-devices.

In FSA-FBP, the length of the FBP payload has been set to attach 2 bits per slot that inform about the status of each slot. In DFSA and FSA-ACK, the coordinator responds with an ACK to each data packet decoded successfully in each slot and it transmits a short FBP packet at the end of each frame to enable the synchronization of the end-devices.

All the packets are composed of a physical layer preamble, a MAC header, a payload and a cyclic redundancy code (CRC) of 2 bytes for error control.

We have validated the proposed theoretical models by means of computer-based simulations. The results of 1000 simulation samples have been averaged for each test case. Results show a tight match between analysis and simulation for both protocols in all tested cases, thus validating the correctness of the theoretical analyses. Results for DFSA, FSA-ACK and FSA-FBP have been obtained through computer-based simulations also with MATLAB.

2.5.2.3.2 Configuration of LP-CTA and LP-DQ

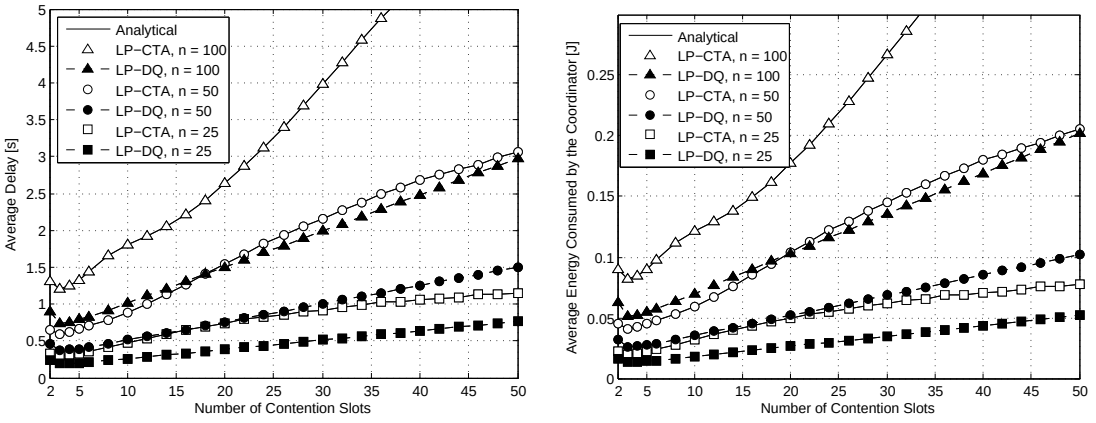
The delay and the energy consumption of the coordinator are represented in Figure 2.22(a) and Figure 2.22(b), respectively, as a function of the number m of contention slots per frame and considering $n \in \{25, 50, 100\}$. Results show a convex shape with a minimum at a certain value of m which represents the optimal point of operation.

Results show that the minimum value of the delay and the energy consumption of the coordinator for LP-DQ and LP-CTA can be found when the number of slots is $m = 3$ regardless of the number of end-devices. Contrarily, as it was demonstrated in Section 2.5.1.3, the number of contention slots in DFSA, FSA-ACK and FSA-FBP has to be adjusted according to the number of end-devices (using $\rho = 1$ in DFSA and $m = n/2$

in FSA-ACK and FSA-FBP) to minimize the delay and the energy consumption of the coordinator.

As it can be observed, when the number of contention slots is very low, LP-DQ and LP-CTA still yield low values of delay and energy consumption of the coordinator. On the contrary, as it was demonstrated in Section 2.5.1.3, the delay and the energy consumption of the coordinator using DFSA, FSA-ACK and FSA-FBP tend to infinite when the number of slots per frame is very low. Indeed, since the tree splitting algorithm of LP-DQ and LP-CTA splits the end-devices into sub-groups in order to reduce the probability of collision, the number of frames to finish a DCR in LP-DQ and LP-CTA is much lower than in DFSA, FSA-ACK and FSA-FBP when the value of m is small.

When the number of contention slots is above the optimum, the delay and the energy consumption of the coordinator increase almost linearly with the number of contention slots in LP-DQ and LP-CTA. Indeed, the higher the number of contention slots, the lower the probability of collision in a given frame, and thus the lower the number of frames required to resolve a DCR. However, as it can be observed in Figure 2.23, the average number of frames required to resolve a DCR in LP-CTA and LP-DQ tends to a minimum value when m is above a certain value. Therefore, since the delay and the energy consumption of the coordinator are proportional to the number of frames, they both increase as the number of contention slots increases because a higher number of



(a) Average delay

(b) Average energy consumed by the coordinator

FIGURE 2.22: (a) Average delay (in seconds) required to terminate a DCR over the number of contention slots using LP-CTA and LP-DQ, and (b) Average energy consumed by the coordinator in a DCR over the number of contention slots using LP-CTA and LP-DQ.

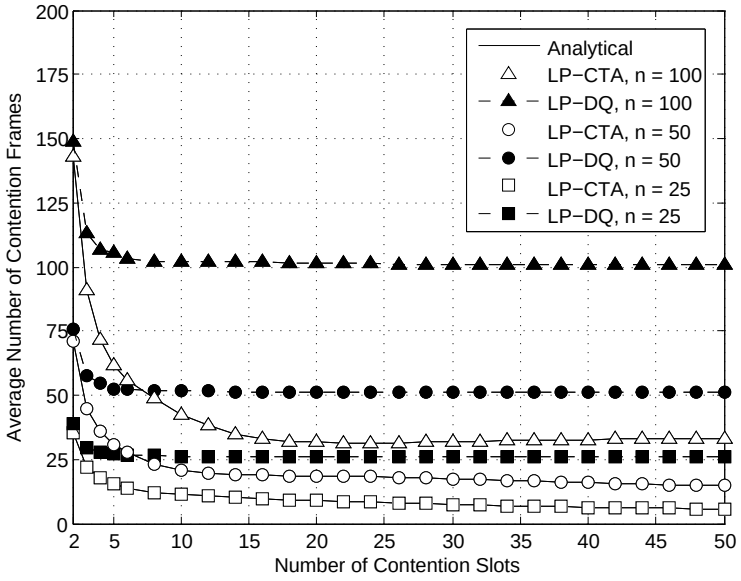


FIGURE 2.23: Average delay (in frames) required to terminate a DCR over the number of contention slots using LP-CTA and LP-DQ.

contention slots leads to longer frames and longer periods of sleep and idle listening, yielding greater delay and energy consumption of the coordinator.

The energy consumption per end-device in a DCR using LP-DQ and LP-CTA is represented in Figure 2.24 as a function of the number m of contention slots per frame and considering $n \in \{25, 50, 100\}$.

Results show that the average energy consumed per end-device tends to a minimum value when $m \geq 10$ and $m \geq 20$ in LP-DQ and LP-CTA, respectively, regardless of the number n of end-devices. The independency of the results with n relaxes the need to know the size of the network. Contrarily, as it was demonstrated in Section 2.5.1.3, the number of contention slots in DFSA, FSA-ACK and FSA-FBP has to be adjusted according to the number of end-devices (using $\rho > 1$ in DFSA and $m \geq n$ in FSA-ACK and FSA-FBP) in order to minimize the energy consumed per end-device.

As it can be observed, when the number of contention slots decreases, the average energy consumed per end-device in a DCR using LP-CTA and LP-DQ increases exponentially. This is due to the fact that the probability of collision among end-devices becomes higher when m decreases, thus leading the average number of frames $\overline{d_n}$ (2.28) where an end-device has to contend to increase exponentially. It can be also observed that the energy consumed per end-device when $m = 2$ is 1.5 and 3 times bigger than the respective minimum values of LP-DQ and LP-CTA for higher values of m. On the

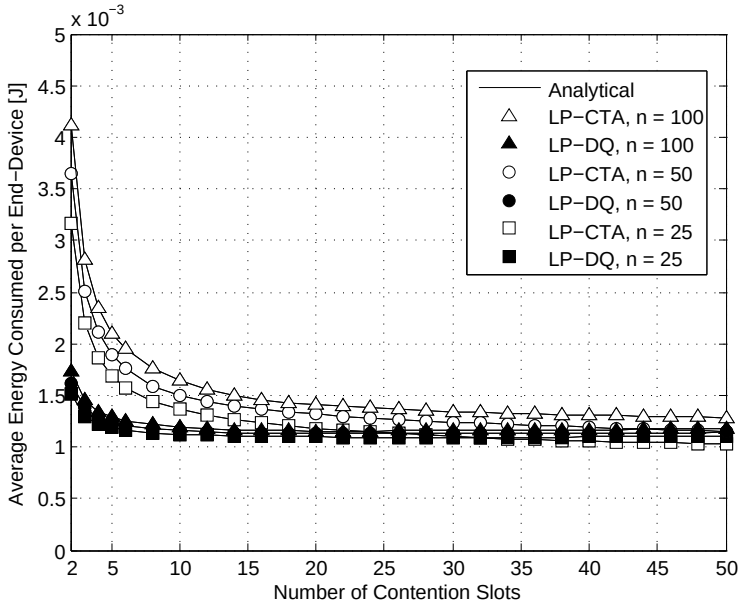


FIGURE 2.24: Average energy consumed per end-device in a DCR over the number of contention slots using LP-CTA and LP-DQ.

contrary, as it was demonstrated in Section 2.5.1.3, the energy consumed per end-device using DFSA, FSA-ACK and FSA-FBP tends to infinite when the number of slots per frame is very low. Indeed, when the value of m is low, the tree splitting algorithm of LP-DQ and LP-CTA reduces the number of frames where an end-device has to contend with respect to DFSA, FSA-ACK and FSA-FBP.

Finally, it is worth mentioning that the average energy consumed per end-device in a DCR using LP-DQ and LP-CTA tends to the minimum value when the value of m is very high. Indeed, although a higher value of m leads to longer sleep periods in the end-devices, the use of an ultra-low power sleep mode (e.g., 90nW [80]) yields reduced energy consumption in the end-devices.

There is a trade-off between the delay, the energy consumption of the coordinator, and the energy consumption per end-device in LP-DQ and LP-CTA. When the number of contention slots per frame increases, the end-devices consume less energy in a DCR, at the cost of increasing the delay and the energy consumption of the coordinator.

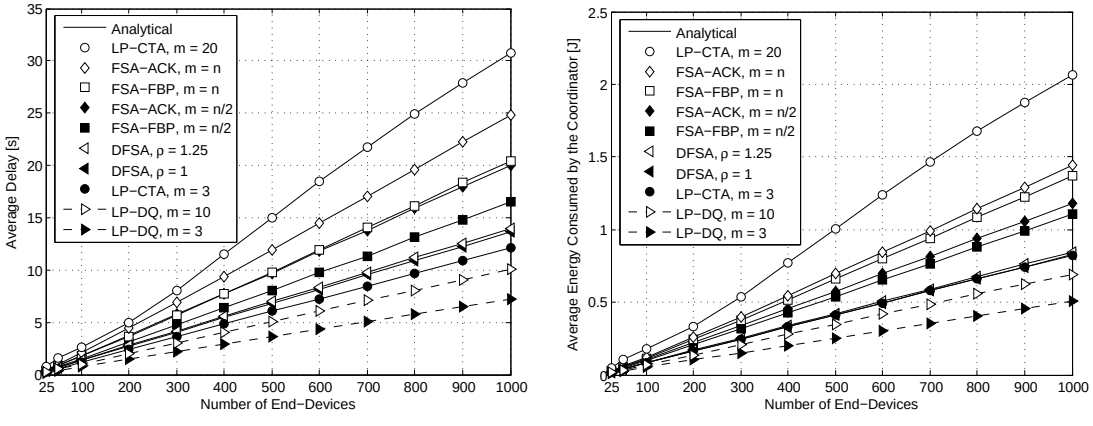
While the number of contention slots needs to be optimized in DFSA, FSA-ACK and FSA-FBP according to the expected number of end-devices in order to minimize the delay and energy consumption, the optimum number of contention slots in LP-DQ and LP-CTA does not depend on the number of end-devices, which relaxes the need

to know the size of the network and adapts smoothly to any change in the number of contending end-devices.

2.5.2.3.3 Delay and Energy Performance in Dense Networks

The delay, the energy consumption of the coordinator and the energy consumption per end-device are represented in Figure 2.25(a), Figure 2.25(b) and Figure 2.26, respectively, for LP-DQ, LP-CTA, DFSA, FSA-ACK and FSA-FBP as a function of the number n of end-devices (from 25 to 1000 end-devices). For each protocol, we have considered the number of contention slots per frame that minimizes the delay and the energy consumption of the coordinator (i.e., $m = 3$ in LP-DQ and LP-CTA, $m = n/2$ in FSA-ACK and FSA-FBP, and $\rho = 1$ in DFSA) and the number of contention slots per frame that minimize the energy consumption per end-device (i.e., $m = 10$ in LP-DQ, $m = 20$ in LP-CTA, $m = n$ in FSA-ACK and FSA-FBP, and $\rho = 1.25$ in DFSA).

As it can be observed in Figure 2.25(a) and Figure 2.25(b), the delay and the energy consumption of the coordinator increase almost linearly with the increasing value of n in LP-DQ, LP-CTA, DFSA, FSA-ACK and FSA-FBP. As it could be expected, the delay and the energy consumption of the coordinator are minimum in each protocol when $m = 3$ in LP-DQ and LP-CTA, $m = n/2$ in FSA-ACK and FSA-FBP, and $\rho = 1$ in DFSA. It is worth noting that LP-DQ outperforms LP-CTA, DFSA, FSA-ACK and FSA-FBP in terms of delay and energy consumption of the coordinator. Indeed, the contention process in LP-CTA, DFSA, FSA-ACK and FSA-FBP is executed using long



(a) Average delay

(b) Average energy consumed by the coordinator

FIGURE 2.25: (a) Average delay (in seconds) required to terminate a DCR over the number of end-devices using FSA-ACK, FSA-FBP, DFSA, LP-CTA and LP-DQ, and (b) Average energy consumed by the coordinator in a DCR over the number of end-devices using FSA-ACK, FSA-FBP, DFSA, LP-CTA and LP-DQ.

contention slots where the end-devices transmit data packets. On the contrary, LP-DQ uses very short contention slots to make requests for data transmission using ARS packets. The resolution of collisions is separated from the transmission of data in LP-DQ, thus completely avoiding the collision of long data packets and leading to a faster and more energy efficient process.

If the protocols are configured with their optimal number of contention slots that minimize the delay and the energy consumption of the coordinator, LP-DQ provides delay reductions of a 40% with respect to LP-CTA, 47% with respect to DFSA, 56% with respect to FSA-FBP, and 64% with respect to FSA-ACK; and LP-CTA provides delay reductions of a 11% with respect to DFSA, 27% with respect to FSA-FBP, and 40% with respect to FSA-ACK. In terms of energy consumption of the coordinator, LP-DQ provides energy savings of a 40% with respect to LP-CTA and DFSA, 54% with respect to FSA-FBP, and 57% with respect to FSA-ACK; and LP-CTA provides energy savings of a 2.5% with respect to DFSA, 25% with respect to FSA-FBP, and 30% with respect to FSA-ACK. Therefore, the use of LP-DQ and LP-CTA can improve considerably the delay and the energy consumption of the coordinator in M2M networks where the number of end-devices can be large and unpredictable.

As it can be observed in Figure 2.26, the energy consumption per end-device using LP-DQ and LP-CTA increases logarithmically with the value of n. Indeed, according

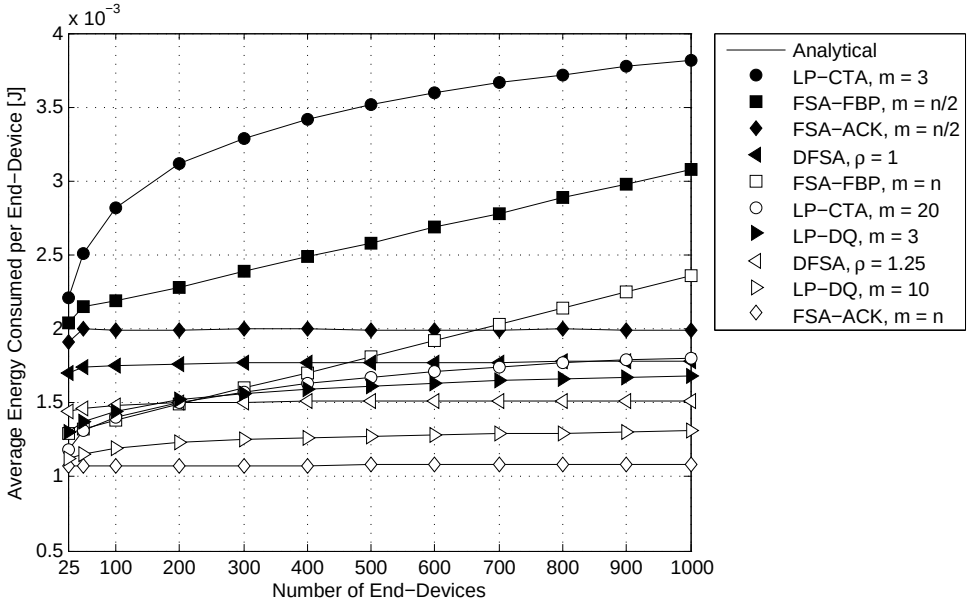


FIGURE 2.26: Average energy consumed per end-device in a DCR over the number of end-devices using FSA-ACK, FSA-FBP, DFSA, LP-CTA and LP-DQ.

to (2.52), this is due to the logarithmic nature of $\overline{d_n}$ (2.28), and also to the use of a very low power sleep mode in those frames where an end-device does not transmit data or ARS packets. On the contrary, as it was demonstrated in Section 2.5.1.3, the energy consumption per end-device using FSA-FBP increases linearly with n, and is much higher than in LP-DQ and LP-CTA. The energy consumption per end-device using DFSA and FSA-ACK is almost insensitive to the number of end-devices. This is due to the fact that in FSA-FBP the length of the FBP increases linearly with n, but in DFSA and FSA-ACK the FBP packets are very short and do not depend on the number of end-devices, thus reducing the energy consumption of the end-devices in reception mode.

As it could be expected, the energy consumption per end-device is minimum in each protocol when $m = 10$ in LP-DQ, $m = 20$ in LP-CTA, $m = n$ in FSA-ACK and FSA-FBP, and $\rho = 1.25$ in DFSA. With these configurations, LP-DQ provides energy savings of more than a 27% with respect to LP-CTA, 13% with respect to DFSA, and 45% with respect to FSA-FBP when $n = 1000$ end-devices. However, if the frame length of FSA-ACK is over-dimensioned, i.e., $m \geq n$, FSA-ACK outperforms LP-DQ in terms of energy consumption per end-device. Contrarily, if the frame length of FSA-ACK is not well configured, e.g., $m = n/2$, then LP-DQ and LP-CTA provide savings of a 35% and 10% with respect to FSA-ACK, respectively, in the energy consumption per end-device.

2.5.2.4 Conclusions

In this section, the Low Power Contention Tree-based Access (LP-CTA) and the Low Power Distributed Queuing (LP-DQ) protocols have been proposed for data collection networks. LP-CTA combines the frame structure of FSA with an m-ary tree splitting algorithm to resolve contention. In LP-DQ, the end-devices request access in a short contention window, where packet collisions are resolved using tree splitting, and data transmissions are kept collision-free. A theoretical model based on probabilistic analysis has been formulated to calculate the average delay and energy consumption for LP-CTA and LP-DQ. Results show that there exists an optimum number m of contention slots in LP-CTA and LP-DQ that minimizes the value of the average delay and energy consumed by the gateway. This optimum value is $m = 3$ regardless of the number of end-devices. Results also show that the energy consumed per end-device tends to a minimum value when $m \geq 10$ and $m \geq 20$ in LP-DQ and LP-CTA, respectively, regardless of the number

n of end-devices. The independency of the results with n relaxes the need to know the size of the network and adapts smoothly to any change in the number of contending end-devices. There is a trade-off between the delay, the energy consumption of the gateway, and the energy consumption per end-device. As the number of contention slots increases, the end-devices consume less energy, but the delay and the energy consumption of the coordinator increase.

The delay and the energy consumption of the gateway in LP-DQ and LP-CTA increase almost linearly with the number of end-devices. Results show that LP-CTA and LP-DQ outperform DFSA, FSA-ACK and FSA-FBP in terms of delay and energy consumption of the coordinator. In particular, LP-DQ provides delay reductions of a 47% with respect to DFSA, 56% with respect to FSA-FBP, and 64% with respect to FSA-ACK; and LP-CTA provides delay reductions of a 11% with respect to DFSA, 27% with respect to FSA-FBP, and 40% with respect to FSA-ACK. In terms of energy consumption of the coordinator, LP-DQ provides energy savings of a 40% with respect to DFSA, 54% with respect to FSA-FBP, and 57% with respect to FSA-ACK; and LP-CTA provides energy savings of a 2.5% with respect to DFSA, 25% with respect to FSA-FBP, and 30% with respect to FSA-ACK.

The energy consumption per end-device in LP-DQ and LP-CTA increases logarithmically with the number of end-devices. Results show that LP-DQ outperforms LP-CTA, DFSA and FSA-FBP in terms of energy consumption per end-device. In particular, LP-DQ provides energy savings of more than a 27% with respect to LP-CTA, 13% with respect to DFSA and 45% with respect to FSA-FBP. Contrarily, only if the frame length of FSA-ACK is over-dimensioned, i.e., $m \geq n$, FSA-ACK outperforms LP-DQ in terms of energy consumption per end-device.

The use of LP-DQ and LP-CTA can improve considerably the energy efficiency of dense M2M networks when the number of end-devices is large and unpredictable. While DFSA, FSA-ACK and FSA-FBP require to adjust the frame length according to the number of end-devices, yielding to very long frames, LP-CTA and LP-DQ use a short and fixed number of contention slots per frame, thus facilitating scalability and synchronization. Taking that into account, the LP-CTA and LP-DQ protocols are interesting alternatives for data collection scenarios.

2.6 Scenario II: Multiple Data-Packets per Data Collection Round

This section focuses on MAC protocols for data collection applications where each end-device has a number $l(k) \geq 1$ of data packets ready to transmit to the coordinator in every DCR k, where $l(k)$ is an exponentially distributed random variable. The value of $l(k)$ is independent of other end-devices and has a new realization in every DCR. The assumption of exponentially-distributed message lengths is common in the analysis of communication networks [81] and has been traditionally used to model random data processes related to different sources of data.

The structure of this section is organized as follows. Section 2.6.1 focuses on the performance evaluation of Frame Slotted-ALOHA based Access Protocols, more specifically, of RFSA and two variants of FSA. Section 2.6.2 focuses on the design, analysis and performance evaluation of a hybrid protocol, named LPR-DQ, for networks where the end-devices generate long multi-packet messages.

2.6.1 Frame Slotted-ALOHA based Access Protocols

This section focuses on the analysis of RFSA and the performance evaluation of RFSA and FSA. The operation of the RFSA and FSA protocols is decribed in Section 2.6.1.1. Section 2.6.1.2 presents the analysis of RFSA based on absorbing Markov chains and formulates the analytical models to calculate the delay and the energy consumption for RFSA. Section 2.6.1.3 is devoted to validate the analytical models and to evaluate the performance of RFSA. The comparison with the performance of the FSA protocols is also presented in this section.

2.6.1.1 MAC Protocols Description

2.6.1.1.1 Frame Slotted-ALOHA (FSA-ACK, FSA-FBP)

We consider the two versions of Frame Slotted-ALOHA described in Section 2.5.1.1.1: the FSA-ACK and FSA-FBP protocols. An example of the k-th DCR using FSA-ACK and FSA-FBP is depicted in Figure 2.27a. In every frame, each end-device randomly selects one of the m slots to transmit one of its $l(k)$ data packets. Recall that in FSA-ACK, the coordinator responds with an ACK to each data packet decoded successfully

in each slot. In FSA-FBP, the coordinator transmits a FBP packet at the end of each frame to inform about the state of the m slots. The frame structures of FSA-ACK and FSA-FBP are shown in Figure 2.3 and Figure 2.4, respectively. When an end-device succeeds in transmitting a data packet, it randomly selects one of the m slots of subsequent frames to transmit the next data packet in its sequence of $l(k)$ packets. This process is repeated, frame after frame, until all of the data packets are successfully transferred to the coordinator, thus contending independently for each data packet.

The example of Figure 2.27a shows the operation of the FSA-ACK and FSA-FBP protocols with $n = 3$ contending end-devices and $m = 3$ slots per frame. In Frame 1, all end-devices transmit their first data packet. End-devices 1 and 3 collide in Slot 1, and End-device 2 succeeds in transmitting its first packet in Slot 3 of Frame 1. In Frame 2, End-devices 1 and 3 transmit again their first packet, and End-device 2 transmits its second packet in Slot 1. All end-devices succeed in Frame 2. In Frame 3, End-devices 1 and 3 transmit their second packet, and End-device 2 transmits its third packet: End-device 1 succeeds in Slot 1, and End-devices 2 and 3 collide in Slot 2. The process continues after Frame 3, until all end-devices succeed in transmitting their $l(k)$ packets to the coordinator.

2.6.1.1.2 Reservation Frame Slotted-ALOHA (RFSA)

An example of the k-th DCR using RFSA is depicted in Figure 2.27b. Each frame of RFSA is composed of a fixed number m of slots and one FBP. In every frame, each end-device randomly selects one slot to transmit the first of its $l(k)$ data packets without CCA. When an end-device succeeds in transmitting its first data packet, the slot with success is reserved to that end-device for the next $l(k) - 1$ frames to transmit its other data packets. At the end of each frame, the coordinator broadcasts an FBP to inform about the state of the m slots: empty, success or failure; and free or reserved. The end-devices that are still contending to transmit their first data packet can only select a slot among the free slots. When an end-device has transmitted the complete sequence of $l(k)$ data packets, the end-device enters into sleep mode to save energy, and its reserved slot becomes free again in order to be used by other contending end-devices in subsequent frames. An end-of-message flag in the header of the last packet indicates that an end-device releases his reserved slot. These processes are repeated, frame after frame, until the coordinator is able to decode the $l(k)$ data packets from every end-device.

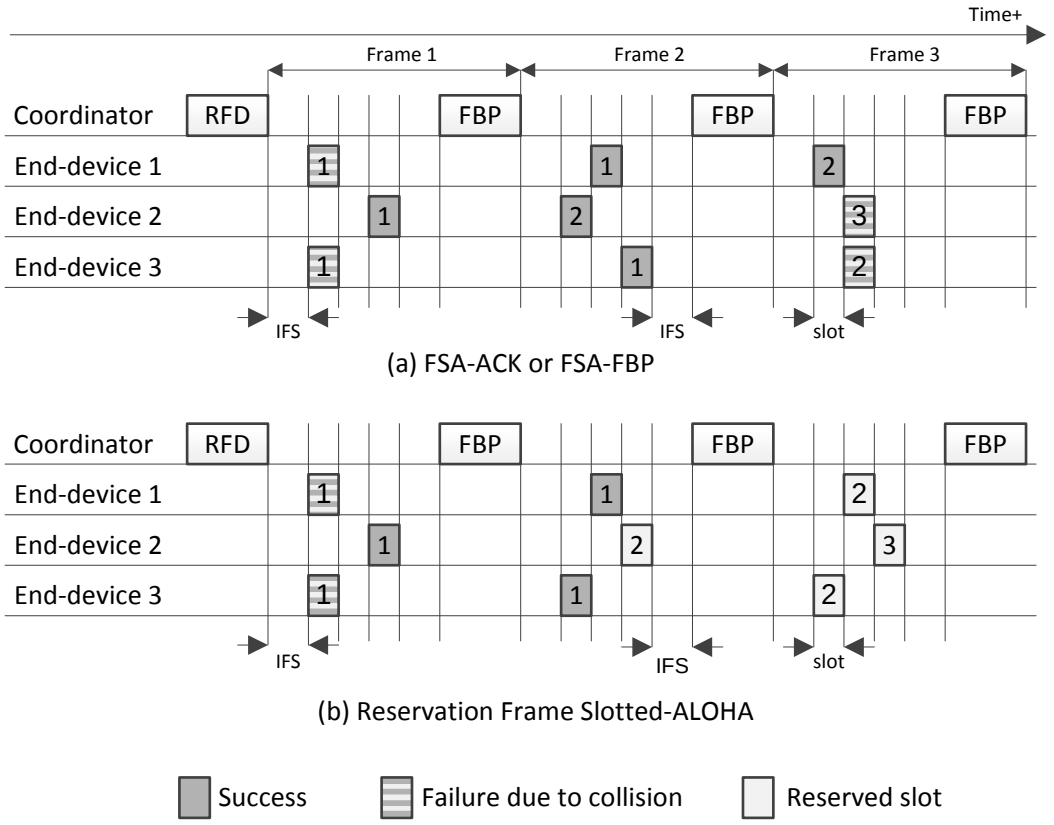


FIGURE 2.27: Example of a DCR considering $n = 3$ end-devices and $m = 3$ slots per frame using: (**a**) FSA-ACK or FSA-FBP; and (**b**) Reservation Frame Slotted-ALOHA. The numbers inside the slots indicate the sequence number of the data packet transmitted by each end-device. The ACK packets of FSA-ACK have been omitted.

The example of Figure 2.27b shows RFSA with three contending end-devices and three slots per frame. In Frame 1, End-device 2 succeeds in transmitting its first packet in Slot 3. Therefore, Slot 3 is reserved for End-device 2 for the transmission of its sequence of data packets in subsequent frames. Since End-devices 1 and 3 collide in Frame 1, they randomly select a new slot in Frame 2 (among the free Slots 1 and 2) to transmit their first packet. End-devices 1 and 3 succeed in transmitting their first data packet in Slots 2 and 1 of Frame 2, respectively. Therefore, Slots 1, 2 and 3 are reserved for End-devices 3, 1 and 2, respectively, for the transmission of their data packets. The process continues after Frame 3, until every end-device has transmitted its $l(k)$ packets to the coordinator.

As shown in Figure 2.27, a guard time called the inter-frame space (IFS) is left between reception and transmission to compensate for propagation, processing and turn-around times to switch the end-devices radio transceiver between reception and transmission modes.

2.6.1.2 Analysis of Performance Metrics

In this section, we analyze the contention process in a DCR using RFSA. We propose an absorbing Markov chain to model the evolution of the number of contending end-devices and the number of free slots within the contention process. The proposed Markov chain is inspired by existing works in data collection scenarios [35, 82]. In order to build the transition matrix that characterizes the Markov chain, we derive the probability that one or more end-devices succeed in transmitting their first data packet and the probability that a reserved slot becomes free in a given frame. Finally, we derive the average number of frames in which the process remains in every state, which is used later in Section 2.6.1.2.4 to formulate the delay and energy models for RFSA.

2.6.1.2.1 Absorbing Markov Chain Model

The contention process of RFSA can be modeled with a discrete-time absorbing Markov chain. The generalized state transition diagram is depicted in Figure 2.28 and Figure 2.29, respectively, for the case where the number n of end-devices is greater than the number m of slots and for the case where the number n of end-devices is lower or equal to the number m of slots. Some transitions have been intentionally omitted for ease of understanding of the figures.

Each state of the chain is defined by $\{c(t), f(t)\}$, where $c(t) \in \{0, 1, ..., n\}$ and $f(t) \in \{0, 1, ..., m\}$ are stochastic processes that represent the number of end-devices that are contending to transmit their first data packet and the number of free slots at time t, respectively. Then, the number of reserved slots at time t can be represented by $r(t) = m - f(t)$. A discrete time scale is adopted, *i.e.*, t and $t+1$ correspond to the beginning of two consecutive frames.

The process starts at $t = 1$, where $c(t) = n$ and $f(t) = m$, *i.e.*, the initial state is (n, m), and it finishes at the absorbing state $(0, m)$. By observing the state transition diagrams depicted in Figure 2.28 and Figure 2.29, the total number K of states can be calculated as:

$$K = \begin{cases} \left(\frac{m}{2}\right)(m+1) + (n-m+1)\,m + 1, & if\ (n > m) \\ \left(\frac{n+1}{2}\right)(n+2), & if\ (n \leq m) \end{cases} \tag{2.53}$$

The process changes from state (c_i, f_i) to state (c_j, f_j) in one-step, with $i, j \in \{1, 2, ..., K\}$, when the following two conditions are met:

Condition 1: A number of contending end-devices succeed in transmitting their first data packet, denoted by S_{ij}, which can be expressed as the difference between the number of contending end-devices in both states,

$$S_{ij} = \begin{cases} c_i - c_j, & if\,(c_i > c_j) \\ 0, & otherwise \end{cases} \tag{2.54}$$

and $S_{ij} \in \{0, 1, ..., \max(S_{ij})\}$, where $\max(S_{ij})$ is the maximum number of contending end-devices that can succeed in one frame. Formally, $\max(S_{ij})$ can be defined as:

$$\max(S_{ij}) = \begin{cases} 0, & if\,(f_i = 1)\,and\,(c_i > 1) \\ f_i - 1, & if\,(c_i > f_i) \\ c_i, & if\,(c_i \leq f_i) \end{cases} \tag{2.55}$$

Condition 2: A number of reserved slots become free, denoted by $F_{ij} \in \{0, 1, ..., (m - f_i)\}$, which can be expressed as the sum of the difference between the number of free slots in both states $(f_j - f_i)$, plus the difference between the number of contending end-devices in both states (*i.e.*, S_{ij}),

$$F_{ij} = \begin{cases} (f_i - f_j) + S_{ij}, & if\,(f_i < m)\,and\,(f_i \leq f_j) \\ 0, & otherwise \end{cases} \tag{2.56}$$

Then, we can formulate the one-step transition probability p_{ij} from state (c_i, f_i) to state (c_j, f_j) as follows:

$$p_{ij} = \begin{cases} P_s(S_{ij}, c_i, f_i) \begin{pmatrix} m - f_i \\ F_{ij} \end{pmatrix} {p_r}^{F_{ij}} (1 - p_r)^{m - f_i - F_{ij}}, \\ \qquad if\,(c_i - \max(S_{ij}) \leq c_j \leq c_i)\,and\,(f_j \leq m - c_i + c_j) \\ 0, \quad if\,(c_j > c_i)\,or\,(c_j < c_i - \max(S_{ij}))\,or\,(f_j > m - c_i + c_j) \\ 1 - \sum_{i \neq j} p_{ij}, \quad if\,(c_i = c_j \neq 0)\,and\,(f_i = f_j \neq m) \\ 1, \quad if\,(c_i = c_j = 0)\,and\,(f_i = f_j = m) \end{cases} \tag{2.57}$$

where $P_s(S_{ij}, c_i, f_i)$ is the probability that there exists a number S_{ij} of end-devices that succeed in transmitting their first data packet in a given frame with f_i free slots having

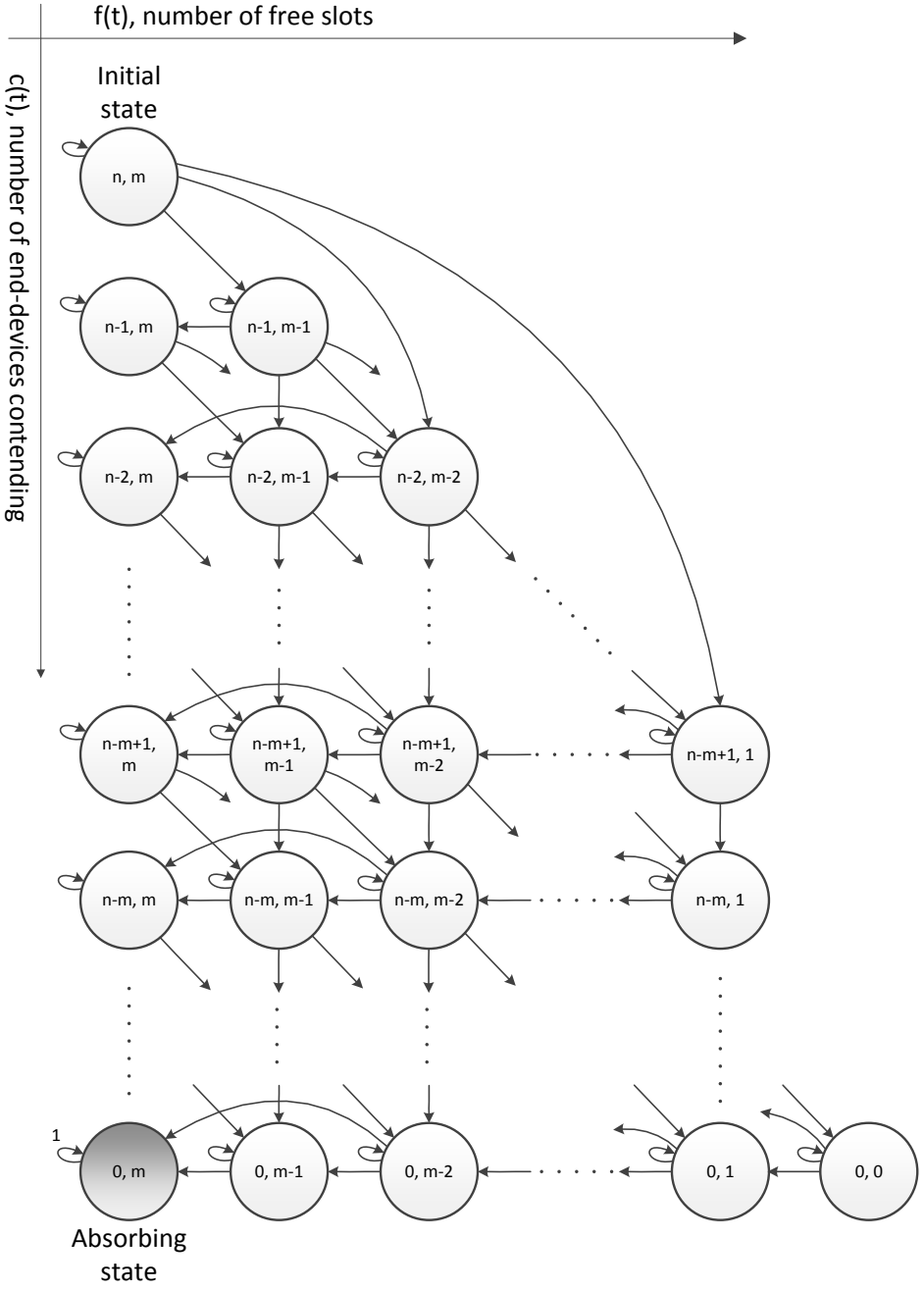


FIGURE 2.28: Generalized state transition diagram of the absorbing Markov chain of RFSA for the case in which the number n of end-devices is greater than the number m of slots.

c_i contending end-devices, and p_r is the probability that a reserved slot is released (*i.e.*, a slot becomes free) when an end-device finishes transmitting its $l(k)$ data packets. We derive both probabilities $P_s(S_{ij}, c_i, f_i)$ and p_r later in Section 2.6.1.2.2.

The rationale for the transition probability is the following. In the first condition, S_{ij} end-devices succeed in transmitting their first data packet to the coordinator, and F_{ij} slots become free among the $(m - f_i)$ slots reserved. In the second condition, the

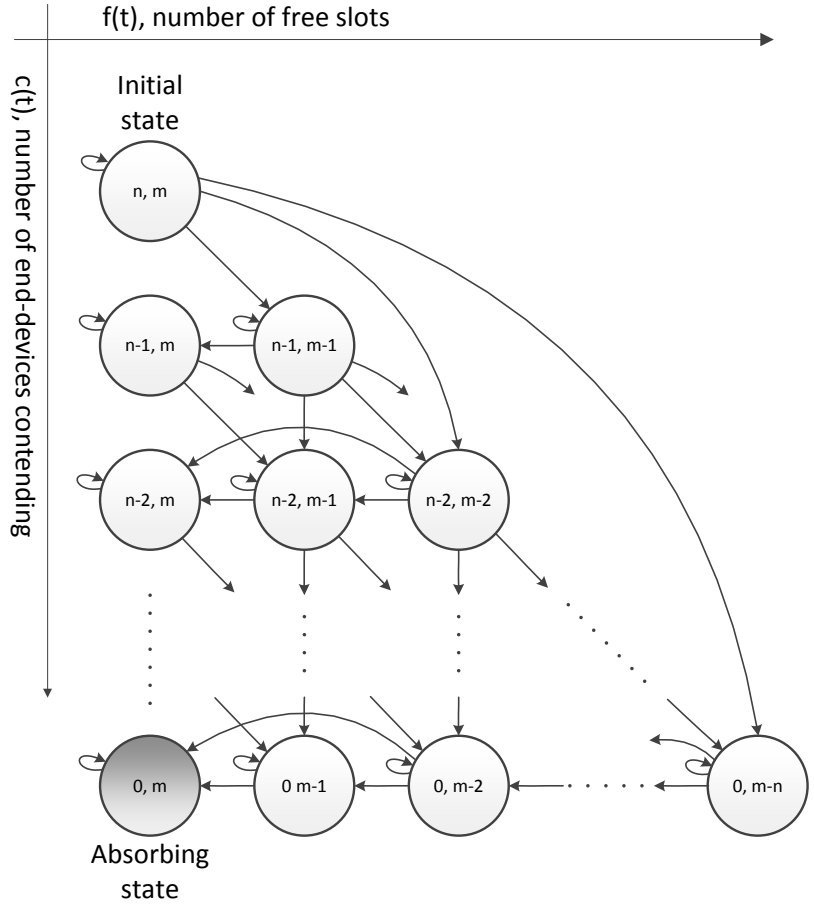


FIGURE 2.29: Generalized state transition diagram of the absorbing Markov chain of RFSA for the case in which the number n of end-devices is lower or equal to the number m of slots.

number of contending end-devices increases, *i.e.*, $c_j > c_i$, it decreases below its lower bound, *i.e.*, $c_j < c_i - \max(S_{ij})$, or the number of free slots increases above its upper bound, *i.e.*, $f_j > m - c_i + c_j$. The three options are impossible and have therefore zero probability. In the third condition, no new end-devices succeed in transmitting their first data packet, *i.e.*, $S_{ij} = 0$, and the process is not yet absorbed, *i.e.*, $c_i = c_j \neq 0$ and $f_i = f_j \neq m$. In the fourth condition, the transition probability is one, because the process has finished and remains in the absorbing state $(0, m)$.

2.6.1.2.2 Probabilities of Success and Release

According to [35], the probability $P_s(S_{ij}, c_i, f_i)$ that S_{ij} contending end-devices succeed in transmitting their first data packet in a frame, having f_i free slots and c_i contending

end-devices, can be formulated as:

$$P_s\left(S_{ij}, c_i, f_i\right) = \frac{\begin{pmatrix} f_i \\ S_{ij} \end{pmatrix} \prod_{k=0}^{S_{ij}-1} \left(c_i - k\right) G\left(f_i - S_{ij}, c_i - S_{ij}\right)}{{f_i}^{c_i}} \tag{2.58}$$

where:

$$G\left(T, t\right) = T^t + \sum_{k=1}^{t} \left\{ \left(-1\right)^k \cdot \prod_{j=0}^{k-1} \left\{\left(t-j\right)\left(T-j\right)\right\} \left(T-k\right)^{t-k} \frac{1}{k!} \right\} \tag{2.59}$$

with $T = f_i - S_{ij}$ and $t = c_i - S_{ij}$.

A reserved slot becomes free when one end-device has transmitted all of its $l(k)$ data packets to the coordinator. Then, since $l(k)$ is an exponentially-distributed random variable, the probability that a reserved slot is released in a given frame can be expressed as:

$$p_r = \frac{1}{\overline{l(k)}} \tag{2.60}$$

where $\overline{l(k)}$ is the mean of the exponential distribution of $l(k)$.

2.6.1.2.3 Number of Frames in Transient States

In order to obtain the average time (in frames) in which the process remains in every transient state of the chain, we first order the K states as $\{X_1, X_2, ..., X_K\}$, where $X_i = (c_i, f_i)$. Being $X_K = (0, m)$ the absorbing state and the other the transient states, with $X_1 = (n, m)$, we can represent the absorbing Markov chain as a $K \times K$ transition matrix P defined as:

$$\mathrm{P} = \begin{bmatrix} \mathrm{Q} & \mathrm{R} \\ 0 & 1 \end{bmatrix} \tag{2.61}$$

where Q is a $(K-1) \times (K-1)$ matrix that contains the transition probabilities between transient states (*i.e.*, $p_{ii} \neq 1$ with $i, j < K$), R is a $(K-1) \times 1$ non-zero column vector that contains the transition probabilities from the transient states to the absorbing state (*i.e.*, $i < K$ and $j = K$), 1 is the transition probability at the absorbing state (*i.e.*, $p_{ii} = 1$ with $i = K$) and 0 is a $1 \times (K-1)$ row vector of zeros.

The fundamental matrix N of P can be defined as:

$$\mathrm{N} = (\mathrm{I} - \mathrm{Q})^{-1} \tag{2.62}$$

where each element n_{ij} of N is equal to the average number of frames that the process remains in the transient state X_j (*i.e.*, with c_j contending end-devices and f_j free slots for n_{ij} consecutive frames) given that it started in the transient state X_i with $i, j = \{1, ..., (K-1)\}$.

The time of absorption, denoted by t_0, is defined as the average number of frames until the process is absorbed (*i.e.*, it reaches state X_K) when it started in state X_1. The time of absorption can be expressed as:

$$t_0 = \mathrm{N} \cdot \mathrm{c} = \sum_{\mathrm{j}=1}^{\mathrm{K}-1} \mathrm{n}_{1\mathrm{j}} \tag{2.63}$$

where c is a $(K-1) \times 1$ column vector with all ones. The values of n_{1j} with $j = \{1, ..., (K-1)\}$ are used in Section 2.6.1.2.4 to formulate the theoretical models of delay and energy consumption using RFSA.

2.6.1.2.4 Delay and Energy Consumption

In this section, we use the analysis presented in the previous section to formulate the calculation of the average delay and energy consumption in a DCR using the RFSA protocol described in Section 2.6.1.1.2.

The *average delay*, expressed in seconds and defined as the time elapsed since the DCR starts until all end-devices succeed in transmitting their $l(k)$ data packets to the coordinator, can be formulated as:

$$\overline{T}_C^{\mathrm{RFSA}} = t_0 \left(m T_{data} + 2T_{IFS} + T_{FBP}^{\mathrm{RFSA}} \right) \tag{2.64}$$

where T_{data}, T_{IFS} and T_{FBP}^{RFSA} are the time of transmission of a data packet, the duration of an IFS, and the time of transmission of an FBP packet, respectively, and t_0 is the average number of frames that the process needs to be absorbed. Indeed, the process starts at state $X_1 = (n, m)$ and finishes when it reaches state $X_K = (0, m)$.

The *average energy consumed by the coordinator* is denoted by $\overline{E}_{coord}^{\mathrm{RFSA}}$. The coordinator executes the following operations in every frame of the process: (i) listens to

the channel for the m slots of each frame to receive incoming data packets; (ii) remains in idle mode for the duration of two IFS; and (iii) transmits the FBP after each frame. Therefore, the average energy consumption of the coordinator can be formulated as:

$$\overline{E}_{coord}^{\mathrm{RFSA}} = t_0 \left(m\rho_{rx} T_{data} + 2\rho_{\sigma} T_{IFS} + \rho_{tx} T_{FBP}^{\mathrm{RFSA}} \right) \tag{2.65}$$

The *average energy consumed by all of the end-devices* is denoted by $\overline{E}_{devices}^{\mathrm{RFSA}}$. Since we consider a homogeneous network and the MAC rules are identical to all of the end-devices, the average energy consumed per end-device can be considered as symmetrical and be expressed as $\overline{E}_{1-device}^{\mathrm{RFSA}} = \overline{E}_{devices}^{\mathrm{RFSA}}/n$.

An end-device that has not succeeded yet in transmitting all of its $l(k)$ data packets executes the following operations in each frame: (i) transmits a data packet in one of the m slots; (ii) switches to sleep for the other $m-1$ slots; (iii) stays in idle mode for two IFS; and (iv) receives the FBP. Thus, the energy consumed by one end-device in a frame where it transmits a data packet can be expressed as:

$$E_{data}^{\mathrm{RFSA}} = \rho_{tx} T_{data} + (m-1)\, \rho_{sleep} T_{data} + 2\rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}^{\mathrm{RFSA}} \tag{2.66}$$

An end-device that has succeeded in transmitting its $l(k)$ data packets in a given frame will sleep in subsequent frames until the DCR finishes. Thus, the energy consumed by one end-device in a frame where it does not transmit a data packet can be expressed as:

$$E_{sleep}^{\mathrm{RFSA}} = \rho_{sleep} \left(m T_{data} + 2T_{IFS} + T_{FBP}^{\mathrm{RFSA}} \right) \tag{2.67}$$

The average energy consumed by all the end-devices can be formulated as:

$$\overline{E}_{devices}^{\mathrm{RFSA}} = \sum_{j=1}^{K-1} n_{1j} E_{frame,j}^{\mathrm{RFSA}} \tag{2.68}$$

where n_{1j} is the average number of frames that the process remains in each transient state X_j and $E_{frame,j}^{\mathrm{RFSA}}$ is the energy consumed by all end-devices in one frame being in

state X_j. Then, $\overline{E}^{\text{RFSA}}_{devices}$ can be expressed as:

$$\overline{E}^{\text{RFSA}}_{devices} = \sum_{j=1}^{K-1} n_{1j} \left(N_j^{data} E^{\text{RFSA}}_{data} + N_j^{sleep} E^{\text{RFSA}}_{sleep} \right) \tag{2.69}$$

where $N_j^{data} = c_j + (m - f_j)$ and $N_j^{sleep} = \left(n - N_j^{data} \right)$ are the number of end-devices that transmit a data packet and the number of end-devices that sleep, respectively, when the process is in state X_j. Recall that c_j and f_j are the number of contending end-devices and the number of free slots in state X_j, respectively, and n is the total number of end-devices that transmit data to the coordinator.

2.6.1.3 Model Validation and Performance Evaluation

In this section, we use the theoretical models formulated in Section 2.6.1.2 to calculate the average delay and energy consumption for RFSA. We first describe the considered scenario. Then, we analyze how the delay and energy consumption are influenced by the number of slots per frame and the number of end-devices. We discuss the numerical results and determine the criteria to minimize the average delay and energy consumption. In addition, we compare the performance for RFSA, FSA-ACK and FSA-FBP.

2.6.1.3.1 Scenario

The system parameters used to validate the analytical models and to evaluate the performance are summarized in Table 2.3. They have been selected according to the IEEE 802.15.4 standard [79] and from the specifications of the CC2520 radio transceiver [80], typically used in M2M networks. We have considered that each end-device has a number L of data packets with a payload of 114 bytes ready to transmit in every DCR, where L is exponentially distributed with mean $\overline{L} = 50$ data packets per end-device.

The length of the FBP payload in RFSA and FSA-FBP has been set to attach two bits per slot that inform about the status of each slot. In FSA-ACK, the coordinator responds with an ACK to each data packet decoded successfully in each slot and it transmits a short FBP packet at the end of each frame to enable the synchronization of the end-devices. All of the packets are composed of a physical layer preamble, a MAC header, a payload and a cyclic redundancy code (CRC) of two bytes for error control.

We have validated the proposed theoretical models by means of computer-based simulations. A DCR is simulated in MATLAB using a while loop in which all of the

TABLE 2.3: System parameters for RFSA, FSA-ACK and FSA-FBP.

Parameter	**Value**	**Parameter**	**Value**
MAC header	8 bytes	Data-rate	250 kbps
Data payload	114 bytes	FBP payload	m·2 bits/slot
$T_{preamble}$	160 μs	T_{IFS}	192 μs
T_{data}	4.1 ms	$\rho_{rx} = \rho_{\sigma}$	66.9 mW
ρ_{tx}	100.8 mW	ρ_{sleep}	90 nW

operations performed within one contention frame are simulated in each loop iteration. We use a $1 \times n$ vector of packet counters to simulate the number of data packets that must be transmitted by each end-device. Each counter is set at the beginning of the simulation to the number L of data packets for each end-device and decreases when a new data packet is transmitted successfully in one loop iteration. We use a $1 \times n$ vector to simulate the number of slot where each end-device transmits a data packet in one frame. In every iteration of RFSA, we generate a uniformly-distributed random number for each contending end-device to simulate the transmission of the first of its L data packets in a randomly-selected slot among the non-reserved slots. We use a $1 \times m$ vector to simulate the transmission of a feedback packet from the coordinator that informs about the state of the m slots in every frame: (i) empty, if the slot has not been selected by any end-device; (ii) success (reserved), if the slot has been selected by one end-device and is reserved until the packet counter of the end-device is equal to zero, *i.e.*, it finishes the transmission of its L data packets; (iii) failure (collision), if the slot has been selected by two or more end-devices; and (iv) released, if the slot becomes free again in order to be selected by other contending end-devices in subsequent iterations. The while loop is repeated until all of the packet counters are equal to zero.

The analytical results have been compared to the simulation within a range of frame lengths (*i.e.*, number of slots per frame) and for different numbers of end-devices. The results show a tight match between analysis and simulation in all tested cases, thus validating the correctness of the analyses. Results for FSA-ACK and FSA-FBP have also been obtained through computer-based simulations using MATLAB. The results of 1000 simulation samples have been averaged for each test case.

2.6.1.3.2 Configuration of RFSA

The average delay (t_0), expressed in frames, required to terminate a DCR using RFSA,

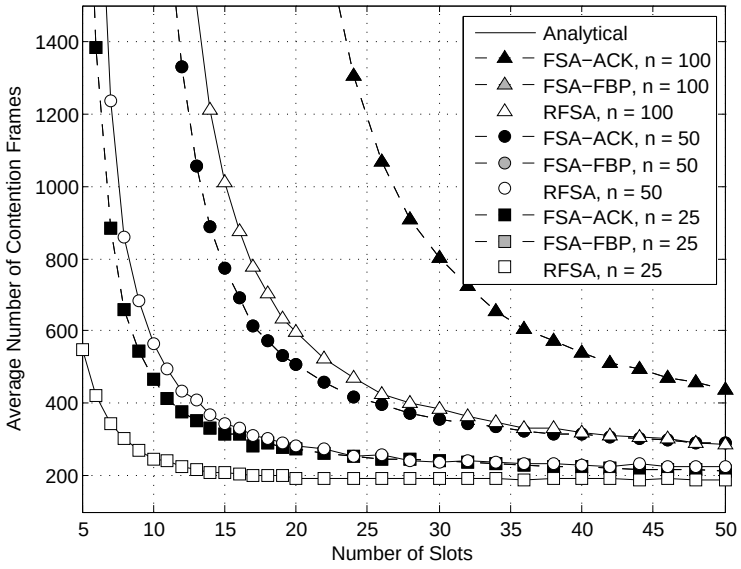


FIGURE 2.30: Average delay (in frames) required to terminate a DCR over the frame length using RFSA, FSA-ACK and FSA-FBP.

FSA-ACK and FSA-FBP is represented in Figure 2.30 as a function of the frame length (from five to 50 slots per frame). We have considered $n = 25$, 50 and 100 end-devices.

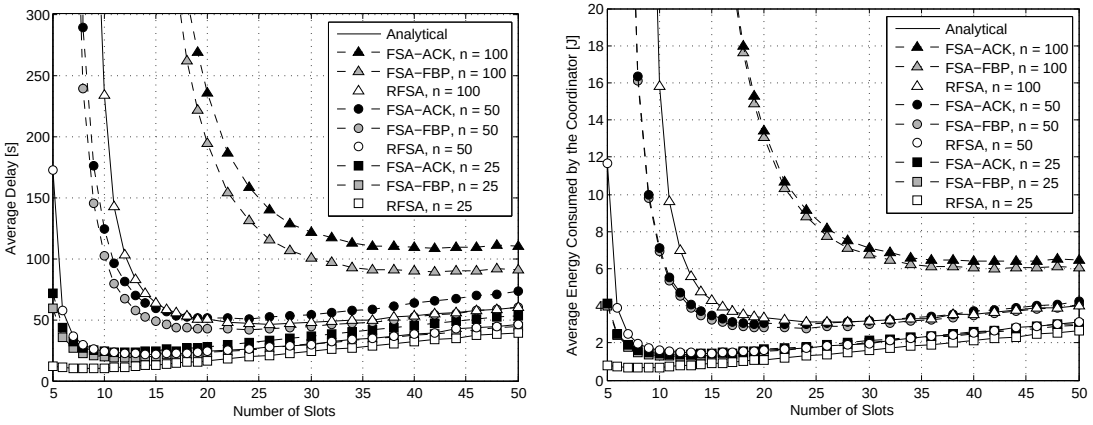
As could be expected, the results show that the value of t_0 decreases exponentially and tends to a constant value when the frame length increases. Indeed, the longer the frame length, the lower the probability of collision among end-devices and the higher the number of successful transmissions per frame. As can be observed, the value of t_0 is identical for FSA-ACK and FSA-FBP, and it is considerably greater than in RFSA. The fact that, in RFSA, an end-device that succeeds in transmitting its first data packet reserves the slot with success for the next $L - 1$ frames leads to a lower probability of collision in subsequent frames, since the number of contending end-devices is also reduced, yielding a shorter delay. The differences between the value of t_0 using RFSA and FSA-ACK (or FSA-FBP) become more apparent when the number of end-devices is higher or when the frame length is shorter, and thus, contention is greater in each frame.

The average delay, expressed in seconds, and the average energy consumed by the coordinator in a DCR are represented in Figure 2.31(a) and Figure 2.31(b), respectively. They have been evaluated as a function of the frame length (from 5 to 50 slots) for RFSA, FSA-ACK and FSA-FBP considering $n = 25$, 50 and 100 end-devices. It is worth noting that, as could be expected from the delay and energy models of RFSA, the

values of $\overline{T}_C^{\text{RFSA}}$ in Equation (2.64) and $\overline{E}_{coord}^{\text{RFSA}}$ in Equation (2.65) are highly correlated, showing a similar trend.

As can be observed in Figure 2.31(a) and Figure 2.31(b), the delay and the energy consumption of the coordinator tend to infinity in RFSA, FSA-ACK and FSA-FBP, when the frame length is lower than $n/8$, $n/4$ and $n/4$, respectively. This is a direct consequence of the very high value of t_0, due to the high probability of collision among the end-devices when the frame length is very low. In addition, the values of delay and energy consumption of the coordinator in FSA-ACK and FSA-FBP are always greater than in RFSA. This is due to the fact that the values of t_0 are considerably greater than in RFSA, as shown in Figure 2.30.

There exists an optimum frame length m_{opt} that minimizes the value of both delay and energy consumption of the coordinator. The value of the optimum frame length has been obtained numerically. First, we have computed the average delay and the average energy consumption over the frame length, and then, we have searched for the frame length that provides the lowest values of delay and energy consumption. We have repeated the process for different values of the number of end-devices in order to confirm the validity of the optimum value, which depends on the number of end-devices, being $m_{opt} \simeq n/5$ for RFSA and $m_{opt} \simeq n/2$ for FSA-ACK and FSA-FBP. For example, when $n = 100$ end-devices, we have minimum average delay, $\overline{T}_C^{\text{RFSA}} \simeq 50$ s, and minimum average energy consumption of the coordinator, $\overline{E}_{coord}^{\text{RFSA}} \simeq 3.2$ J, at $m \simeq 20$



(a) Average delay

(b) Average energy consumed by the coordinator

FIGURE 2.31: (a) Average delay (in seconds) required to terminate a DCR over the frame length using RFSA, FSA-ACK and FSA-FBP, and (b) Average energy consumed by the coordinator in a DCR over the frame length using RFSA, FSA-ACK and FSA-FBP.

slots when using RFSA; and $\overline{T}_C^{\text{FSA-FBP}} \simeq 90$ s and $\overline{E}_{coord}^{\text{FSA-FBP}} \simeq 6$ J at $m \simeq 50$ slots when using FSA-FBP. Therefore, RFSA provides a delay reduction and energy savings of approximately 45% with respect to FSA-FBP when using optimum frame lengths in both access protocols.

For frame lengths greater than m_{opt}, the values of delay and energy consumption of the coordinator increase proportionally to the frame length in RFSA, FSA-ACK and FSA-FBP. Indeed, even though the value of t_0 tends to a constant value when the frame length increases, a higher number of slots per frame yields a greater delay. Since $\overline{T}_C^{\text{RFSA}}$ in Equation (2.64) is proportional to t_0 (constant) and to the frame length (increasing), the value of $\overline{T}_C^{\text{RFSA}}$ increases linearly with the frame length, as shown in Figure 2.31(a). In addition, the coordinator has to listen to the channel in every frame for longer periods of time, yielding greater energy consumption.

Therefore, the frame length has to be tuned in RFSA, FSA-ACK and FSA-FBP to its optimum value, depending on the number of end-devices, taking into account that: (i) a frame length greater than the optimum has a cost in terms of average delay and energy consumed by the coordinator; and (ii) both the average delay and energy consumption increase dramatically when the frame length is much lower than the optimum value. Consequently, since the increase in delay and energy after the optimal point show a very small slope, when the number of end-devices is not known *a priori*, it is better to over-dimension the frame length in order to guarantee that the delay and energy consumption are kept low. Of course, synchronization issues may arise when the frame length is very long. This deserves further research to improve the accuracy of the internal clocks in wireless transceivers so that very long frames can be executed.

The average energy consumed per end-device in a DCR is represented in Figure 2.32 as a function of the frame length (from five to 50 slots) for RFSA, FSA-ACK and FSA-FBP considering $n = 25$, 50 and 100 end-devices. As could be expected, the average energy consumed per end-device using RFSA, FSA-ACK and FSA-FBP tends to infinity when the frame length is very small. Indeed, the more frames required to complete a DCR, the higher the energy consumed per end-device. In addition, the average energy consumed per end-device in FSA-ACK and FSA-FBP is always greater than in RFSA. This is due to the fact that in FSA-ACK and FSA-FBP, an end-device has to contend independently to transmit each one of its L data packets, and contrarily, in RFSA, an end-device has only to contend to transmit its first data packet and uses a reserved slot to

transmit its other $L-1$ packets, yielding less energy consumption per end-device. Results show that the average energy consumed per end-device tends to a minimum value when $m \geq n/2$ and $m \geq n$ using RFSA, and FSA-ACK or FSA-FBP, respectively. Indeed, although a longer frame length leads to longer sleep periods in the end-devices, the use of a low power sleep mode (as shown in Table 2.3) yields reduced energy consumption. Therefore, the frame length has to be adjusted according to the number of contending end-devices in order to minimize the average energy consumed per end-device.

2.6.1.3.3 Delay and Energy Performance in Dense Networks

In this section, we evaluate and compare the performance of RFSA, FSA-ACK and FSA-FBP, in terms of delay and energy consumption, with different numbers of end-devices in dense M2M networks.

The average delay and the average energy consumed by the coordinator in a DCR using RFSA, FSA-ACK and FSA-FBP are represented in Figure 2.33(a) and Figure 2.33(b), respectively, over the number n of end-devices (from 25 to 1000 end-devices). We have considered three different cases in the configuration of the frame lengths of the protocols: *(i)* when the frame length is equal to the total number of end-devices (*i.e.*, $m = n$); *(ii)* when the frame length is lower than the total number of end-devices and close to the optimal point that minimizes the delay and the energy consumption of the coordinator

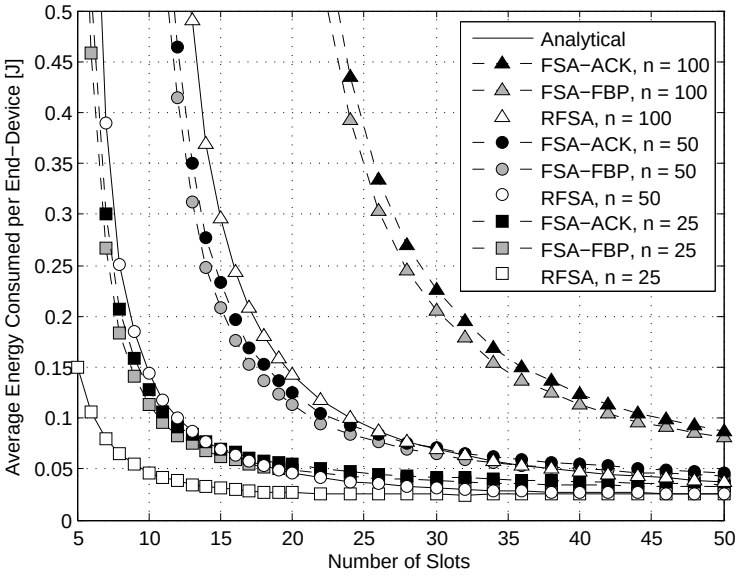
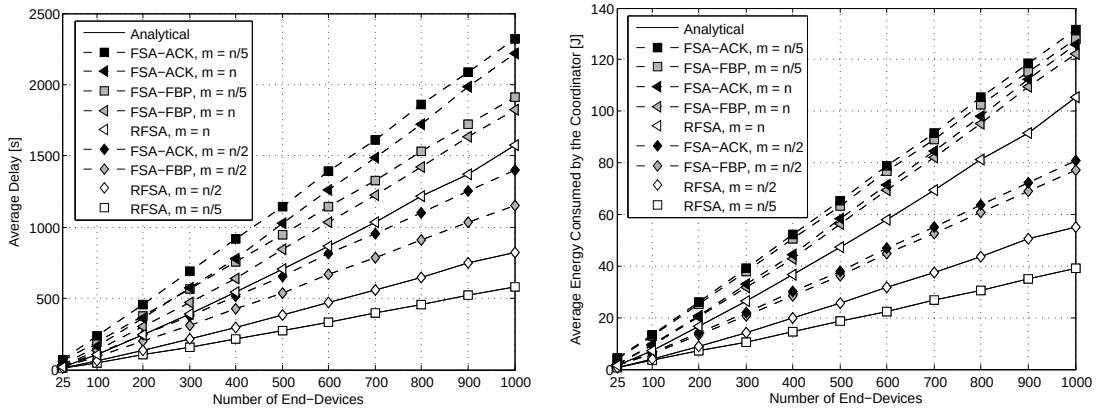


FIGURE 2.32: Average energy consumed per end-device in a DCR over the frame length using RFSA, FSA-ACK and FSA-FBP.

in FSA-ACK and FSA-FBP (*i.e.*, $m = n/2$); and *(iii)* when the frame length is close to the optimal in RFSA (*i.e.*, $m = n/5$).

Results show that the average delay using RFSA, FSA-ACK and FSA-FBP increase linearly with the number of end-devices, and the delay is much higher using FSA-ACK and FSA-FBP than RFSA. When $m = n$, RFSA provides delay reductions of 29% and 14% with respect to FSA-ACK and FSA-FBP, respectively. These delay reductions increase up to a 40% and 28.5% when $m = n/5$, and up to a 75% and 70% when $m = n/5$ with respect to FSA-ACK and FSA-FBP, respectively. Indeed, the delay reduction provided by RFSA is higher when its optimum frame length is used. If the protocols are configured with their optimal frame length to minimize delay, i.e., $m = n/5$ for RFSA and $m = n/2$ for FSA-ACK and FSA-FBP, RFSA provides delay reductions of 58% and 50% with respect to FSA-ACK and FSA-FBP, respectively.

Results show that the average energy consumed by the coordinator using RFSA, FSA-ACK and FSA-FBP increase linearly with the number of end-devices, and the energy consumption of the coordinator is much higher using FSA-ACK and FSA-FBP than RFSA. When $m = n$, RFSA provides energy savings of 16% and 14% with respect to FSA-ACK and FSA-FBP, respectively. These energy saving increase up to a 32% and 28.5% when $m = n/5$, and up to a 75% and 70% when $m = n/5$ with respect to FSA-ACK and FSA-FBP, respectively. Indeed, the energy saving provided by RFSA is higher when its optimum frame length is used. If the protocols are configured with



(a) Average delay

(b) Average energy consumed by the coordinator

FIGURE 2.33: (a) Average delay (in seconds) required to terminate a DCR over the number of end-devices using RFSA, FSA-ACK and FSA-FBP, and (b) Average energy consumed by the coordinator in a DCR over the number of end-devices using RFSA, FSA-ACK and FSA-FBP.

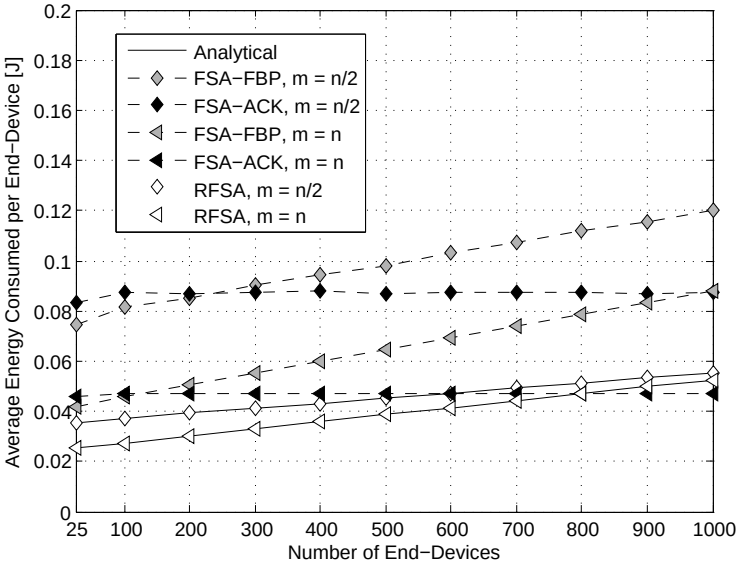


FIGURE 2.34: Average energy consumed per end-device in a DCR over the number of end-devices using RFSA, FSA-ACK and FSA-FBP.

their optimal frame length to minimize the energy consumption of the coordinator, i.e., $m = n/5$ for RFSA and $m = n/2$ for FSA-ACK and FSA-FBP, RFSA provides energy savings of 50% with respect to FSA-ACK and FSA-FBP.

The average energy consumed per end-device in a DCR using RFSA, FSA-ACK and FSA-FBP is represented in Figure 2.34 over the number n of end-devices (from 25 to 1000 end-devices). We have considered two different cases in both access protocols: *(i)* when the frame length is equal to the number of end-devices (*i.e.*, $m = n$); and *(ii)* when the frame length is lower than the total number of end-devices and close to the minimum value that guarantees that the energy consumed per end-device in RFSA is minimum (*i.e.*, $m = n/2$).

Results show that the average energy consumed per end-device using RFSA and FSA-FBP increases linearly with the number n of end-devices. However, in FSA-ACK the average energy consumed per end-device is almost insensitive to the number of end-devices. This is due to the fact that in RFSA and FSA-FBP the length of the FBP increases with the number of end-devices, and thus the energy consumption of an end-device increases because its energy consumption in reception mode increases linearly with the value of n. On the contrary, in FSA-ACK the length of the FBP is fixed and does not depend on the number of end-devices.

Results show that the energy consumption per end-device is much higher using FSA-ACK and FSA-FBP than RFSA. When $m = n$, RFSA provides energy savings of a 40%

with respect to FSA-FBP and FSA-ACK, and it increases up to a 55% when $m = n/2$. If the protocols are configured with their minimum frame length to minimize the energy consumption of the end-devices, i.e., $m = n/2$ for RFSA and $m = n$ for FSA-ACK and FSA-FBP, RFSA provides energy savings of 45% and 40% with respect to FSA-ACK and FSA-FBP, respectively. However, the energy savings of RFSA with respect to FSA-ACK decrease as the number of end-devices increases.

2.6.1.4 Conclusions

In this section, the use of Reservation FSA (RFSA) has been proposed for data collection networks where the end-devices generate fragmented data messages that consist of L packets of fixed length and L is an exponentially-distributed random variable. A theoretical model based on an absorbing Markov chain has been formulated to calculate the average delay and energy consumption for RFSA. Analytical and simulated results show that there exists an optimum frame length m that minimizes the value of the average delay and energy consumed by the gateway. This optimum value depends on the number n of end-devices, being $m \simeq n/5$ for RFSA and $m \simeq n/2$ for FSA-ACK and FSA-FBP. Results also show that the energy consumed per end-device tends to a minimum value when $m \geq n/2$ using RFSA, and $m \geq n$ using FSA-ACK and FSA-FBP. Existing algorithms for the estimation of the number of end-devices could be executed by the coordinator in order to adjust the frame length periodically to its optimal value. The average delay and the average energy consumption in RFSA, FSA-ACK and FSA-FBP increase linearly with the total number of end-devices. RFSA outperforms FSA-ACK and FSA-FBP in terms of delay and energy consumption. In particular, the results show that RFSA can reduce the average delay and the coordinator's energy consumption by more than 25% and the energy consumed per end-device by more than 50%. Therefore, the use of RFSA can considerably improve the energy efficiency of M2M networks for data collection.

2.6.2 Low Power Distributed Queuing with Reservation (LPR-DQ)

The LPR-DQ protocol is proposed in this section as an extension of LP-DQ for networks where the end-devices generate long multi-packet messages. While in LP-DQ only 1 collision-free slot can be reserved for an end-device that succeeds in transmitting an ARS, in LPR-DQ an end-device can reserve a number $l(k) \geq 1$ of collision-free slots in consecutive frames. The operation of LPR-DQ is described in Section 2.6.2.1. Section 2.6.2.2 presents the theoretical analysis of LPR-DQ and formulates the energy model. Section 2.6.2.3 is devoted to validate the model and to evaluate the performance of LPR-DQ through comprehensive computer-based simulations. The comparison with the performance of the RFSA and FSA protocols is also presented in this section.

2.6.2.1 MAC Protocol Description

In LPR-DQ, the end-devices request access to the channel in a short contention window at the beginning of each frame, thus confining collisions to a specific part of the frame. Collisions are resolved by using a tree-splitting algorithm [20] that organizes the end-devices into sub-groups to reduce the probability of collision per transmission attempt. When an end-device succeeds in transmitting its access request, it waits for its turn to transmit data in collision-free slots. Each frame of LPR-DQ is divided in three parts as shown in Figure 2.35: *(i)* m contention slots devoted to the transmission of access request (ARS) packets, *(ii)* one collision-free slot for the transmission of a data packet, and *(iii)* a feedback packet (FBP). A guard time called Inter Frame Space (IFS) is left between reception and transmission modes to compensate propagation and processing delays and the time required to switch the radio transceivers between reception and transmission.

In the k-th DCR, every end-device randomly selects one of the contention slots in every frame to transmit an ARS. Each ARS only contains one field (e.g., 1 byte) that indicates the number $l(k)$ of data packets that must be transmitted by the end-device in the k-th DCR, i.e., the number $l(k)$ of collision-free slots to be reserved. Note that the

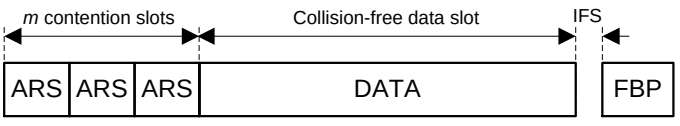


FIGURE 2.35: Frame structure of LPR-DQ.

ARS does not need to identify the end-device. Depending on whether the ARS collides or is successfully decoded by the coordinator, an end-device is queued into one of two logical and distributed queues:

1) The end-devices that have collided in a given contention slot when transmitting their ARS are queued into the *Collision Resolution Queue* (CRQ), sharing the same position in the queue. Note that after every frame, at most m new entries enter into the CRQ, being each one associated to each of the collisions that occurred in the last contention window. The length of the CRQ and the position of the end-devices in the CRQ is updated by executing the tree-splitting algorithm represented in Figure 2.36.a. Each node of the tree represents a frame of m contention slots (m=3 in the example), and the number in each contention slot denotes the number of end-devices that transmit an ARS in that slot. In every level of the tree, an end-device transmits its ARS in only one frame until it succeeds in one level. The algorithm works as follows. At frame 1, all the active end-devices contend. If two or more end-devices collide in a slot, a new frame is assigned only to the end-devices that caused the collision in order to reattempt access, and they are queued into the CRQ. Therefore, if there are j slots with collision in one frame of level d, then j new frames are scheduled in level $d+1$, and j sub-groups of end-devices are queued into the CRQ. Once an end-device has entered in the CRQ, it will retransmit its ARS in a given frame only if it occupies the first position in the CRQ; otherwise, the end-device enters in sleep mode and waits until it reaches the first position of the CRQ.

2) The end-devices that succeed in transmitting their ARS are queued into the *Data Transmission Queue* (DTQ). In principle, any queue discipline could be used to this end. For example, end-devices could enter into the DTQ following the same chronological order of the contention slots. Contrarily to the CRQ, in this case, every position of the DTQ is occupied by just one end-device. Indeed, an end-device occupies a number of positions in the DTQ that is equal to the number $l(k)$ of collision-free data slots reserved by the end-device for this particular DCR. When an end-device reaches the first position of the DTQ, it transmits its data packets in the collision-free slot of successive frames.

The CRQ and DTQ are represented at each end-device by 2 integer numbers per queue representing: *1)* the position of the end-device in the queue, and *2)* the total length of the queue. The length of the CRQ represents the number of sub-groups of end-devices waiting to retransmit an ARS. The length of the DTQ represents the total number of

collision-free slots reserved by the end-devices that have succeeded in transmitting their ARS and wait for their first collision-free slot.

The coordinator updates the length of the CRQ and DTQ at the end of each frame according to the following rules: *1)* the length of the CRQ is incremented by the number of contention slots with collision; *2)* if the CRQ was not empty in the previous frame, then its length is decremented by one; *3)* the length of the DTQ is incremented by the total number of collision-free slots reserved in each frame; and *4)* if the DTQ was not empty in the previous frame, then its length is decremented by one.

The coordinator broadcasts in every FBP: *(i)* the length of the CRQ (2 bytes); *(ii)* the length of the DTQ (2 bytes); and *(iii)* the state of the m contention slots (empty, success, or collision) and the number of collision-free slots reserved in every slot with one successful ARS (1 byte per contention slot). Using the information of the FBP, an end-device that transmitted an ARS can compute its position in the CRQ if it collides, or its position in the DTQ if it succeeds. The positions in the CRQ and DTQ are always decremented by one at the end of each frame. Therefore, the end-devices only receive the FBP in those frames where they transmit either an ARS or a data packet, and they enter into sleep mode in those frames where they do not transmit either ARS or data, in order to save energy.

Figure 2.36 shows an example of the operation of LPR-DQ. The contents of the slots and the lengths of the CRQ and DTQ in every frame are shown in Figure 2.36.a. The contents of both queues are shown in Figure 2.36.b. At frame 1, all the end-devices (d1 to d6) transmit an ARS: d1, d2 and d3 collide in slot 1; d4 succeeds in slot 2; d5 and d6 collide in slot 3. Thus, d1, d2 and d3 enter in the first position of the CRQ; d4 enters in the first position of the DTQ reserving 1 collision-free slot; d5 and d6 enter in the second position of the CRQ. At frame 2, d4 transmits its data packet (because it occupies the first position of the DTQ), and d1, d2 and d3 transmit an ARS (because they occupy the first position of the CRQ): d1 and d2 collide and enter in the CRQ again; d3 succeeds and enters in the DTQ reserving 2 collision-free slots; d5 and d6 move to the first position of the CRQ. At frame 3, d5 and d6 transmit an ARS, collide, and enter in the second position of the CRQ again; d1 and d2 move to the first position of the CRQ; and d3 transmits its first data packet. At frame 4, d3 transmits its second data packet; d1 and d2 transmit an ARS, succeed, and enter in the DTQ reserving 1

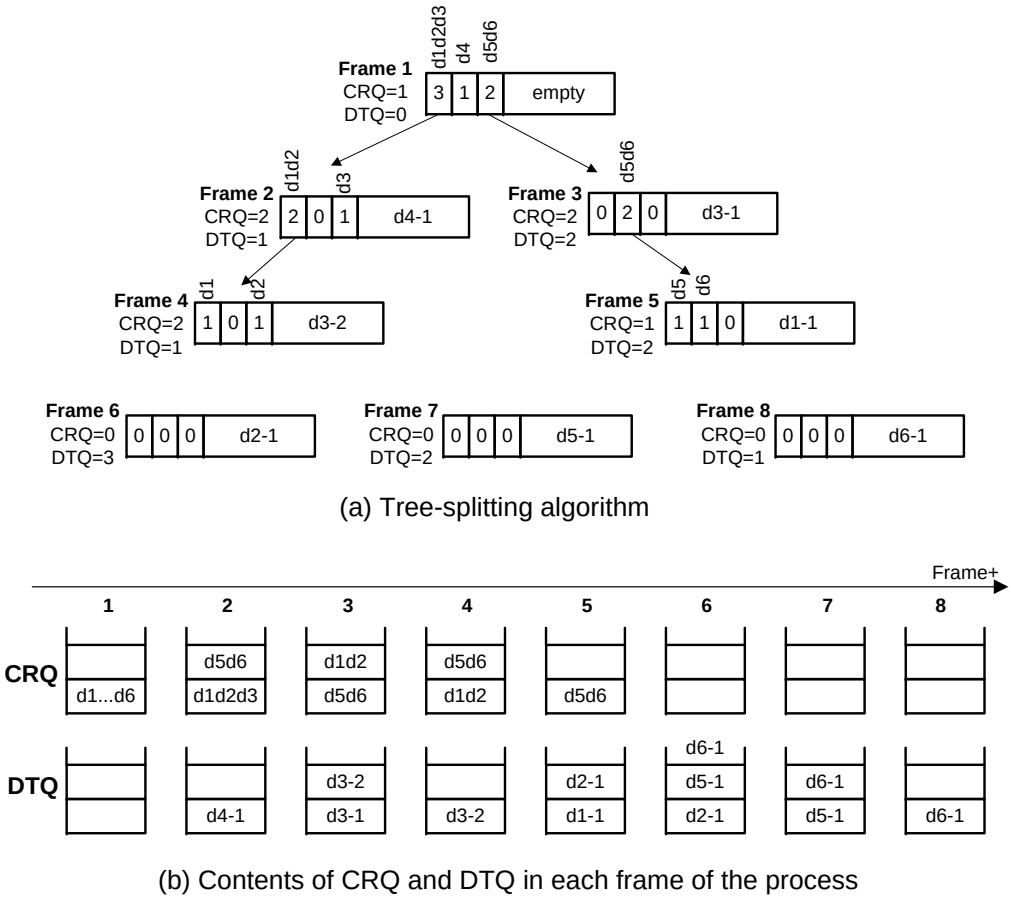


FIGURE 2.36: Example of LPR-DQ with 6 end-devices (d1 to d6) and 3 contention slots: (a) tree-splitting algorithm, and (b) contents of the CRQ and DTQ in each frame.

collision-free slot each; and d5 and d6 move to the first position of the CRQ. The process continues until the end of the DCR.

2.6.2.2 Analysis of Performance Metrics

In this section, we formulate an expression to compute the average energy consumed per end-device in a DCR using LPR-DQ.

2.6.2.2.1 Energy Consumption using LPR-DQ

The time T_R between any two consecutive DCRs using LPR-DQ can be expressed as

$$T_R = \overline{T}_{\text{LPR-DQ}} + \overline{T}_{sleep}, \tag{2.70}$$

where $\overline{T}_{\text{LPR-DQ}}$ is the average time elapsed since the DCR starts until one end-device is capable of successfully transmitting all its data packets to the coordinator, which can

be defined as

$$\overline{T}_{\text{LPR-DQ}} = \left(\overline{CRQ}_{ARS} + \overline{CRQ}_{sleep} + \overline{DTQ}_{sleep} + \overline{DTQ}_{listen} + \overline{DTQ}_{data}\right) T_{frame}^{\text{LPR-DQ}}, \quad (2.71)$$

where $\overline{CRQ}_{ARS}$ is the average number of frames where an end-device contends (i.e., it transmits an ARS); $\overline{CRQ}_{sleep}$ is the average number of frames where an end-device is in CRQ without contending (i.e., it sleeps until it has to contend); $\overline{DTQ}_{sleep}$ is the average number of frames where an end-device is in DTQ waiting for its collision-free data slot (i.e., it sleeps); $\overline{DTQ}_{listen}$ is the average number of frames where an end-device is in DTQ and listens to the feedback packet of the frame before the one where it has to transmit data in order to check whether the end-device in the previous position of DTQ has transmitted all its data packets successfully; $\overline{DTQ}_{data}$ is the average number of frames where an end-device is in DTQ and has to transmit data packets; and $T_{frame}^{\text{LPR-DQ}}$ is the duration of a frame, which is given by

$$T_{frame}^{\text{LPR-DQ}} = mT_{ARS} + T_{data} + 2T_{IFS} + T_{FBP}^{\text{LPR-DQ}}, \quad (2.72)$$

where m is the number of contention slots and T_{ARS}, T_{data}, T_{IFS}, and $T_{FBP}^{\text{LPR-DQ}}$ are the duration of a contention slot (for access requests), a collision-free data slot, an IFS, and the time of transmission of a FBP packet, respectively.

The average energy consumed by an end-device in a DCR, denoted by $\overline{E}_R^{\text{LPR-DQ}}$, can be expressed as

$$\overline{E}_R^{\text{LPR-DQ}} = \overline{E}_{\text{LPR-DQ}} + \rho_{sleep}\overline{T}_{sleep}, \quad (2.73)$$

where $\overline{E}_{\text{LPR-DQ}}$ is the average energy consumed by an end-device in the LPR-DQ contention resolution process, which can be expressed as

$$\overline{E}_{\text{LPR-DQ}} = \overline{CRQ}_{ARS}E_{ARS}^{\text{LPR-DQ}} + \left(\overline{CRQ}_{sleep} + \overline{DTQ}_{sleep}\right)\rho_{sleep}T_{frame}^{\text{LPR-DQ}} + \\ + \overline{DTQ}_{listen}E_{listen}^{\text{LPR-DQ}} + \overline{DTQ}_{data}E_{data}^{\text{LPR-DQ}}. \quad (2.74)$$

$E_{ARS}^{\text{LPR-DQ}}$ is the energy consumed by an end-device in a frame where it contends. The end-device executes the following operations: (i) transmits an ARS in 1 contention slot selected randomly, (ii) keeps in sleep mode in the other $m-1$ slots and in the

collision-free data slot, and (iii) listens to the channel to receive a FBP at the end of the frame. Then, $E_{ARS}^{\text{LPR-DQ}}$ can be formulated as

$$E_{ARS}^{\text{LPR-DQ}} = \left(\rho_{tx} + (m-1)\,\rho_{sleep}\right) T_{ARS} + \rho_{sleep} T_{data} + 2\rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}^{\text{LPR-DQ}}. \quad (2.75)$$

$E_{listen}^{\text{LPR-DQ}}$ is the energy consumed by an end-device in a frame where it only listens to the feedback packet of the frame before it has to transmit data. The end-device executes the following operations: (i) remains in sleep mode in the m contention slots and in the collision-free data slot, and (ii) listens to the channel to receive the FBP. Then, $E_{listen}^{\text{LPR-DQ}}$ can be formulated as

$$E_{listen}^{\text{LPR-DQ}} = m\rho_{sleep} T_{ARS} + \rho_{sleep} T_{data} + 2\rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}^{\text{LPR-DQ}}. \quad (2.76)$$

$E_{data}^{\text{LPR-DQ}}$ is the energy consumed by an end-device in a frame where it transmits one data packet. The end-device executes the following operations: (i) remains in sleep mode in the m contention slots, (ii) transmits a data packet in the collision-free data slot, and (iii) listens to the channel to receive the FBP. Then, $E_{data}^{\text{LPR-DQ}}$ can be formulated as

$$E_{data}^{\text{LPR-DQ}} = m\rho_{sleep} T_{ARS} + \rho_{tx} T_{data} + 2\rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}^{\text{LPR-DQ}}. \quad (2.77)$$

By substituting $\overline{T}_{sleep}$ from (2.70), $\overline{T}_{\text{LPR-DQ}}$ (2.71), and $\overline{E}_{\text{LPR-DQ}}$ (2.74) in (2.73), and after some basic algebra, the average energy consumed by an end-device in a DCR using LPR-DQ can be expressed as

$$\begin{aligned}\overline{E}_{R}^{\text{LPR-DQ}} = {} & \overline{CRQ}_{ARS} E_{ARS}^{\text{LPR-DQ}} + \overline{DTQ}_{listen} E_{listen}^{\text{LPR-DQ}} + \overline{DTQ}_{data} E_{data}^{\text{LPR-DQ}} + \\ & \rho_{sleep} T_R - \rho_{sleep} \left(\overline{CRQ}_{ARS} + \overline{DTQ}_{listen} + \overline{DTQ}_{data}\right) T_{frame}^{\text{LPR-DQ}}. \quad (2.78)\end{aligned}$$

Since we assume that there are no transmission errors, then $\overline{CRQ}_{ARS} = \overline{d_n}$ (2.28), $\overline{DTQ}_{listen} = 1$, and $\overline{DTQ}_{data} = \overline{l(k)}$, where $\overline{l(k)}$ is the average number of data packets per end-device in a DCR. Therefore, $\overline{E}_{R}^{\text{LPR-DQ}}$ can be expressed as

$$\begin{aligned}\overline{E}_{R}^{\text{LPR-DQ}} = {} & \overline{d_n} \cdot \left(E_{ARS}^{\text{LPR-DQ}} - \rho_{sleep} \cdot T_{frame}^{\text{LPR-DQ}}\right) + E_{listen}^{\text{LPR-DQ}} + \\ & \overline{l(k)} \cdot E_{data}^{\text{LPR-DQ}} + \rho_{sleep} \cdot \left(T_R - 2T_{frame}^{\text{LPR-DQ}}\right). \quad (2.79)\end{aligned}$$

2.6.2.3 Model Validation and Performance Evaluation

In this section, we use the theoretical model formulated in Section 2.6.2.2 to evaluate the energy consumption per end-device for LPR-DQ. In addition, we evaluate the delay and the energy consumption of the coordinator for LPR-DQ by means of computer-based simulations. We first describe the considered scenario. Then, we analyze how the delay and the energy consumption are influenced by the number of end-devices. In addition, we compare the performance of LPR-DQ, RFSA, FSA-ACK and FSA-FBP.

2.6.2.3.1 Scenario

The system parameters used to validate the analytical model and to evaluate the performance are summarized in Table 2.4. They have been selected according to the IEEE 802.15.4 standard [79] and from the specifications of the CC2520 radio transceiver [80]. We have considered that the coordinator initiates one DCR every hour, i.e., $T_R = 3600$s. We have considered that each end-device has a number L of data packets with a payload of 114 bytes ready to transmit in every DCR, where L is exponentially distributed with mean $\overline{L} = 50$ data packets per end-device.

In LPR-DQ, the FBP payload includes 2 bytes for the length of CRQ; 2 bytes for the length of DTQ; and 1 byte to inform about the status of each contention slot and the number of collision-free slots reserved in a slot with a successful ARS. We consider that the payload of an ARS packet includes 1 byte to indicate the number of data packets to be transmitted by an end-device.

The length of the FBP payload in RFSA and FSA-FBP has been set to attach two bits per slot that inform about the status of each slot. In FSA-ACK, the coordinator responds with an ACK to each data packet decoded successfully in each slot and it transmits a short FBP packet at the end of each frame to enable the synchronization of the end-devices.

TABLE 2.4: System Parameters for LPR-DQ

Parameter	**Value**	**Parameter**	**Value**
MAC header	8 bytes	Data-rate	250 kbps
Data payload	114 bytes	T_{data}	4.1 ms
$T_{preamble}$	160 μs	T_{IFS}	192 μs
ρ_{tx}	100.8 mW	T_{ARS}	480 μs
$\rho_{rx} = \rho_{\sigma}$	66.9 mW	ρ_{sleep}	60 nW

All of the packets are composed of a physical layer preamble, a MAC header, a payload and a cyclic redundancy code (CRC) of two bytes for error control.

We have validated the proposed theoretical model by means of computer-based simulations. The results of 1000 simulation samples have been averaged for each test case. Results show a tight match between analysis and simulation for LPR-DQ in all tested cases, thus validating the correctness of the theoretical analysis. Results for RFSA, FSA-ACK and FSA-FBP have been obtained through computer-based simulations also with MATLAB.

2.6.2.3.2 Delay and Energy Performance in Dense Networks

The delay, the energy consumption of the coordinator and the energy consumption per end-device in a DCR using LPR-DQ, RFSA, FSA-ACK and FSA-FBP are represented in Figure 2.37, Figure 2.38 and Figure 2.39, respectively, over the number of end-devices (from 25 to 1000 end-devices). For each protocol, we have considered the number of contention slots per frame that minimizes the delay and the energy consumption of the coordinator (i.e., $m = 3$ in LPR-DQ, $m = n/5$ in RFSA, $m = n/2$ in FSA-ACK and FSA-FBP) and the number of contention slots per frame that minimize the energy consumption per end-device (i.e., $m = 10$ in LPR-DQ, $m \geq n/2$ in RFSA, and $m \geq n$ in FSA-ACK and FSA-FBP).

As it can be observed in Figure 2.37 and Figure 2.38, the delay and the energy consumption of the coordinator increase almost linearly with the increasing value of n in LPR-DQ, RFSA, FSA-ACK and FSA-FBP. The delay and the energy consumption of the coordinator are minimum in each protocol when $m = 3$ in LPR-DQ, $m = n/5$ in RFSA, and $m = n/2$ in FSA-ACK and FSA-FBP. It is worth noting that LPR-DQ outperforms RFSA, FSA-ACK and FSA-FBP in terms of delay and energy consumption of the coordinator. Indeed, the contention process in RFSA, FSA-ACK and FSA-FBP is executed using long contention slots where the end-devices transmit data packets. On the contrary, LPR-DQ uses very short contention slots to make requests for data transmission using ARS packets. The resolution of collisions is separated from the transmission of data in LPR-DQ, thus completely avoiding the collision of long data packets and leading to a faster and more energy efficient process.

If the protocols are configured with their optimal number of contention slots that minimize the delay and the energy consumption of the coordinator, LPR-DQ provides

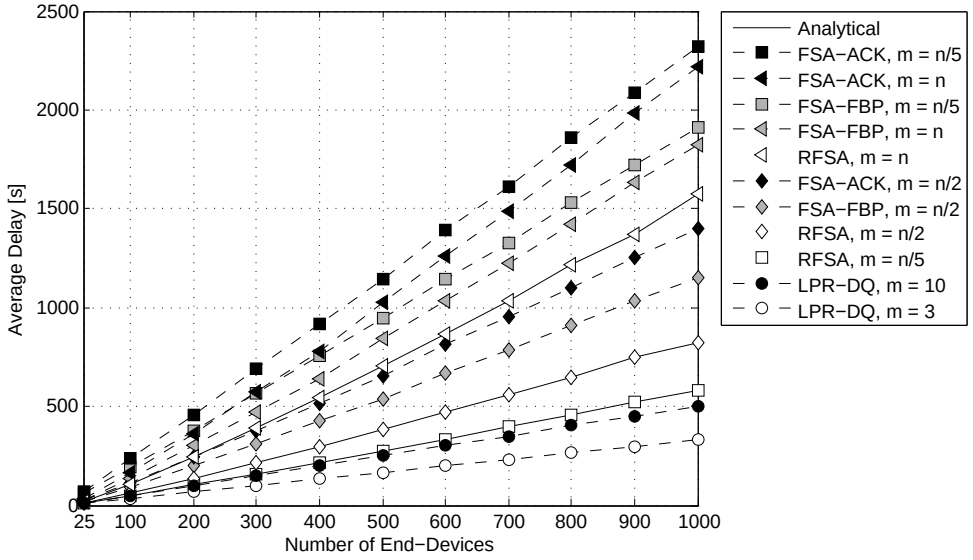


FIGURE 2.37: Average delay (in seconds) required to terminate a DCR over the number of end-devices using LPR-DQ, RFSA, FSA-ACK and FSA-FBP.

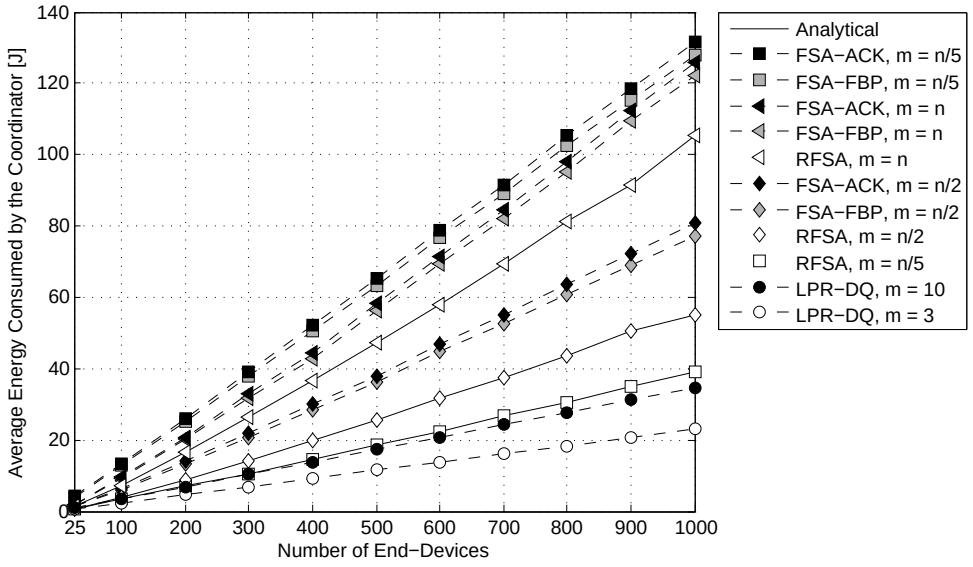


FIGURE 2.38: Average energy consumed by the coordinator in a DCR over the number of end-devices using LPR-DQ, RFSA, FSA-ACK and FSA-FBP.

delay reductions of a 43% with respect to RFSA, 71% with respect to FSA-FBP, and 76% with respect to FSA-ACK. In terms of energy consumption of the coordinator, LPR-DQ provides energy savings of a 41% with respect to RFSA, 70% with respect to FSA-FBP, and 71% with respect to FSA-ACK. Therefore, the use of LPR-DQ can improve considerably the delay and the energy consumption of the coordinator in M2M networks where the number of end-devices can be large and unpredictable.

As it can be observed in Figure 2.39, the energy consumption per end-device using

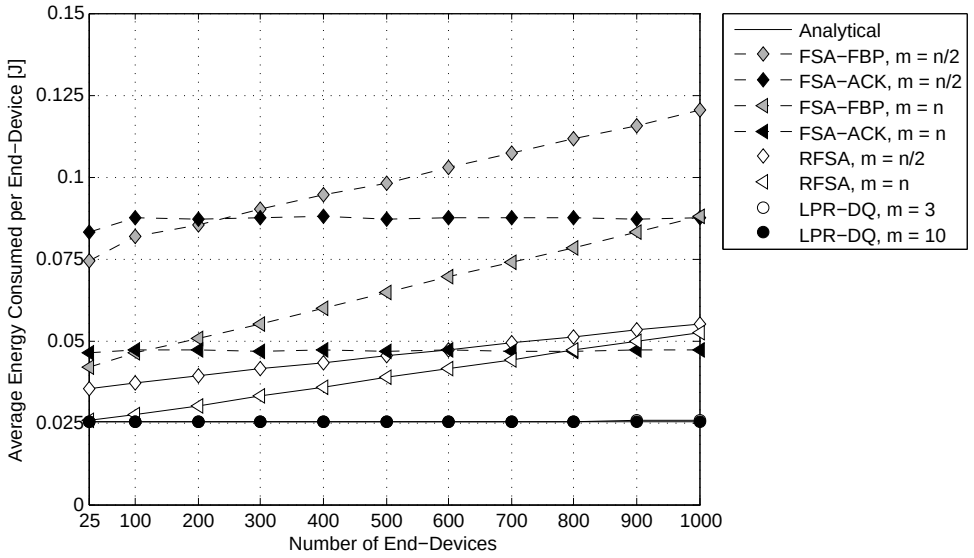


FIGURE 2.39: Average energy consumed per end-device in a DCR over the number of end-devices using LPR-DQ, RFSA, FSA-ACK and FSA-FBP.

LPR-DQ and FSA-ACK is almost insensitive to the number of end-devices. On the contrary, as it was demonstrated in Section 2.6.2.3.2, the energy consumption per end-device using RFSA and FSA-FBP increases linearly with n. The energy consumption per end-device is minimum in each protocol when $m = 10$ in LPR-DQ, $m = n$ in RFSA, FSA-ACK and FSA-FBP. With these configurations, LPR-DQ outperforms RFSA, FSA-ACK and FSA-FBP in terms of energy consumption per end-device, and it provides energy savings of more than a 52% with respect to RFSA, 46% with respect to FSA-ACK, and 71% with respect to FSA-FBP when $n = 1000$ end-devices.

2.6.2.4 Conclusions

In this section, the Low Power Distributed Queuing with Reservation (LPR-DQ) protocol has been proposed for data collection networks where the end-devices generate long messages fragmented in small data packets. In a nutshell, LPR-DQ is an extension of LP-DQ that allows the end-devices to reserve a certain number of collision-free slots in consecutive frames. A theoretical model based on probabilistic analysis has been formulated to calculate the average energy consumed per end-device using LPR-DQ.

Simulation results show that there exists an optimum number m of contention slots in LPR-DQ that minimizes the value of the average delay and energy consumed by the gateway. This optimum value is $m = 3$. The delay and the energy consumed by the gateway using LPR-DQ increase almost linearly with the number of end-devices. Results

show that LPR-DQ outperforms RFSA, FSA-ACK and FSA-FBP in terms of delay and energy consumption of the gateway. In particular, LPR-DQ provides delay reductions of a 43% with respect to RFSA, 71% with respect to FSA-FBP, and 76% with respect to FSA-ACK. In terms of energy consumption of the gateway, LPR-DQ provides energy savings of a 41% with respect to RFSA, 70% with respect to FSA-FBP, and 71% with respect to FSA-ACK.

Analytical and simulation results also show that the energy consumed per end-device tends to a minimum value when $m \geq 10$, and it is almost insensitive to the number of end-devices. Similarly to LP-DQ and LP-CTA, the independency of the results with the number of end-devices relaxes the need to know the size of the network. Finally, Results show that LPR-DQ outperforms RFSA, FSA-ACK and FSA-FBP in terms of energy consumption per end-device, providing energy savings of more than a 52% with respect to RFSA, 46% with respect to FSA-ACK, and 71% with respect to FSA-FBP when $n = 1000$ end-devices.

The use of the LPR-DQ protocol improves considerably the delay and energy efficiency of dense M2M networks for data collection, specially when the number of end-devices is huge and unpredictable. While RFSA, FSA-ACK and FSA-FBP require to adjust the frame length according to the number of end-devices, LPR-DQ can be optimally configured with a short and fixed number of contention slots per frame, and thus facilitates scalability and network synchronization.

2.7 Chapter Conclusions

This chapter focuses on the design, analysis and performance evaluation of random access and hybrid MAC protocols that can handle abrupt traffic transitions (also referred to as delta traffic) and also minimize the energy consumption devoted to communications in star topology wireless M2M networks for data collection. In these networks, a gateway gathers data from a huge and dynamic number of end-devices, which remain in sleep mode to save energy and wake up periodically or triggered by an event to transmit bursts of data to the gateway.

The Frame Slotted-ALOHA (FSA) protocol has been proposed in the past to manage the delta traffic conditions in data collection scenarios. Previous works in FSA and Dynamic FSA (DFSA) mainly focus on the minimization of the delay to resolve the contention. In this chapter, the analytical models to calculate the average delay and the average energy consumption under delta traffic are formulated in Section 2.5.1 for two variants of FSA and DFSA. In addition, their performance is evaluated and compared, and the optimal configuration of the protocols is determined. Results show that there is a trade-off in FSA and DFSA; when the frame length increases above the number of contenders, the end-devices consume less energy, but the delay and the energy consumption of the gateway increase. In FSA, there is a frame length which minimizes delay and the energy consumed by the gateway. This frame length is approximately equal to half the number of end-devices. Regarding the energy consumed per end-device using FSA, it tends to a minimum value when the frame length increases above the number of end-devices. In DFSA, the frame length has to be equal to the number of contenders in every frame to minimize delay and the energy consumed by the gateway; and the energy consumed per end-device tends to a minimum value when the frame length is proportional to the number of contenders. Since the number of end-devices can be very large, the frame length must be large, which may cause synchronization problems. Although the end-devices are synchronized after decoding the feedback (or beacon) packet sent by the gateway at the end of each frame, the time difference between the clocks of the end-devices increases as time advances due to the inaccuracies of crystal oscillators. Therefore, a guard time is needed in every slot in order to facilitate synchronization, which yields longer time frames, thus increasing latency and degrading the energy efficiency of the MAC layer. While DFSA outperforms FSA in terms of delay and energy consumption of the gateway, FSA outperforms DFSA in terms of energy consumed per

end-device if an acknowledgement packet is transmitted by the gateway to each packet successfully decoded.

An alternative to improve the performance of random access is to implement tree splitting algorithms. Tree splitting resolves collisions by organizing the retransmission of colliding packets in such a way that they are always transmitted with finite delay. Previous works in tree splitting-based access protocols for data collection, such as QT or Binary Tree in RFID, evaluate the performance in terms of delay and energy consumption. However, the analytical models are incomplete and do not determine the optimal configuration that minimizes delay and energy consumption. In Section 2.5.2, a new tree splitting-based random access protocol, named LP-CTA, is proposed for data collection networks. In LP-CTA, time is organized in frames further divided into slots where the end-devices contend, and collisions are resolved using a tree splitting algorithm. In addition, in order to reduce the energy consumed by the end-devices in reception mode, the tree splitting algorithm is implemented using a Collision Resolution Queue which allows the end-devices to sleep in complete successive frames until their turn to contend arrives. The analytical models to compute the average delay and the average energy consumption are formulated for LP-CTA.

While LP-CTA uses data slots for contention, it is also possible to use very short access requests transmitted in minislots (or short contention slots) where collisions are resolved with tree splitting. Since the duration of the minislots is much shorter than the duration of a data packet, the performance of the network can be improved. This concept is the foundation of the DQ protocol, which is a hybrid access protocol that combines an m-ary tree splitting algorithm for contention resolution with collision-free data transmission. Previous works in DQ consider homogeneous networks where each end-device generates packets following a random Poisson distribution. In Section 2.5.2, the LP-DQ protocol is proposed for data collection. An analytical model to compute the average energy consumed by the end-devices is formulated, and the performance of LP-DQ is evaluated and compared with respect to LP-CTA, FSA and DFSA in terms of delay and energy consumption. In LP-CTA and LP-DQ there exists a number of contention slots (minislots) that minimizes delay and the energy consumed by the gateway. This optimum value is 3 contention slots regardless of the number of end-devices. Results show that LP-CTA and LP-DQ outperform DFSA and FSA in terms of delay and energy consumed by the gateway. Regarding the energy consumed per

end-device, it tends to a minimum value when 20 or more contention slots are used in LP-CTA, and 10 or more contention slots are used in LP-DQ. Results show that LP-DQ outperforms LP-CTA, DFSA and FSA. Contrarily to FSA and DFSA, in LP-CTA and LP-DQ the gateway does not need to estimate the number of end-devices, and the number of minislots can be very short and fixed, thus facilitating the synchronization of the end-devices and the scalability of the network.

The approach of FSA, DFSA, LP-CTA and LP-DQ is convenient when the data generated by each end-device fits in one single data slot. However, when the end-devices generate long data messages that have to be fragmented, it is better to add a reservation mechanism. The Reservation FSA (RFSA) protocol has been proposed in the past for scenarios where the end-devices transmit long messages with fragmentation. Previous works in RFSA consider networks where each end-device generates messages following a given random Poisson distribution. In Section 2.6.1, an analytical model to calculate the average delay and the average energy consumption under delta traffic is formulated for RFSA, the optimal configuration of the protocol is determined, and the performance of RFSA is compared with respect to FSA. In RFSA, there is a frame length that minimizes delay and the energy consumed by the gateway. This optimum frame length is approximately equal to a fifth of the number of end-devices. Regarding the energy consumed per end-device using RFSA, it tends to a minimum value when the frame length increases above half the number of end-devices. RFSA outperforms FSA in terms of delay and energy consumption. Furthermore, the frame length needed in RFSA is much shorter than in FSA, thus facilitating the synchronization of the end-devices in highly dense networks.

Finally, the LPR-DQ protocol is proposed in Section 2.6.2 as an extension of LP-DQ for data collection networks where the end-devices transmit fragmented data messages. LPR-DQ integrates the concept of reservation to allow each end-device to reserve as many collision-free slots as it needs in a data collection round. The analytical model to calculate the average energy consumed per end-device is formulated, and the performance of LPR-DQ is evaluated and compared with respect to RFSA and FSA. LPR-DQ outperforms RFSA and FSA in terms of delay and energy consumption. Furthermore, the frame structure and the number of contention slots in LPR-DQ is constant regardless of the number of end-devices in the network, which facilitates synchronization and enables scalability. Consequently, the LPR-DQ protocol is an interesting alternative for

data collection networks where each end-device generates bursts of data fragmented in multiple packets.

While the MAC layer for traditional battery-powered networks must be designed to minimize energy consumption in order to guarantee fully continuous operation within a large (but limited) lifetime, the use of energy harvesting could theoretically provide unlimited lifetime. However, due to the high variability and unpredictability of the energy sources, the energy available may not be enough for the operation of the end-devices during certain periods of time, thus causing discontinuous operation of the end-devices. These fluctuations in the available energy are taken into account in the next chapter, which focuses on the design, analysis and performance evaluation of energy harvesting-aware MAC protocols for data collection networks with energy harvesting capabilities.

Chapter 3

Energy Harvesting-Aware MAC Protocols for Wireless M2M Networks

3.1 Introduction

The inclusion of energy harvesting in the operation of the MAC layer is a promising technique to extend the lifetime of M2M networks based on wireless communications. The works in [27–29] were the first contributions related to data collection networks where a group of end-devices equipped with energy harvesters send data periodically to a gateway. These works analyze the performance of the Dynamic Frame Slotted-ALOHA (DFSA) protocol in terms of the probability of delivery and the data collection rate. They derive a simplified Markov chain model of the energy available in an energy storage device (e.g., battery) throughout successive data collection rounds. Results show that both performance metrics increase with the energy harvesting rate. However, these works do not evaluate how the performance is affected by the capacity of the batteries, the minimum energy (or threshold) that an end-device needs to transmit data to the gateway, and the number of end-devices in the network. This chapter aims to fill this gap with a complete analysis and performance evaluation of the energy harvesting-aware DFSA protocol (EH-DFSA) in data collection networks.

A possible strategy to improve the performance of random access based on Frame Slotted-ALOHA is to use a tree-splitting algorithm [20] which resolves the collisions

by organizing the retransmission of colliding data packets in such a way that collisions are split and reduced, and all the packets can be successfully transmitted with finite delay and energy consumption. The performance of tree-splitting algorithms in data collection networks has been evaluated in different applications such as Radio Frequency Identification (RFID) [22, 23, 56]. However, to the best of our knowledge, the design and analysis of random access protocols with tree-splitting for wireless networks with energy-harvesting has never received attention. In this chapter, the novel EH-aware Low Power Contention Tree-based Access (EH-CTA) protocol is proposed as an adaptation of LP-CTA for data collection networks with energy harvesting. The performance of EH-CTA is analyzed and evaluated in terms of the probability of delivery and the data collection rate. The performance of EH-CTA is compared with that of EH-DFSA.

The Reservation FSA (RFSA) protocol was proposed in [46] to improve the performance of FSA when the end-devices generate long messages fragmented into smaller parts. Results show that RFSA outperforms FSA in terms of throughput by letting end-devices reserve the channel for fragmented messages, thus avoiding the need to compete for the channel for each newly generated fragment of the same message. In Section 2.6.1, the RFSA protocol is analyzed and evaluated in data collection networks where the end-devices transmit a burst of data packets upon request from a gateway. Results show that RFSA outperforms FSA and in terms of delay and energy consumption. However, to the best of our knowledge, the design and analysis of a DFSA protocol with reservation for networks with energy-harvesting capabilities has never received attention. In this chapter, the novel EH-aware Reservation DFSA (EH-RDFSA) is proposed as a combination of RFSA and DFSA for data collection networks with energy harvesting. The performance of EH-RDFSA is analyzed and evaluated in terms of the data delivery ratio and the data collection rate. The performance of EH-RDFSA is compared with that of EH-DFSA and TDMA.

Finally, the EH-aware Low Power Distributed Queuing with Reservation (EH-DQ) is proposed in this chapter as an extension and adaptation of LPR-DQ for data collection networks with energy harvesting where the end-devices generate burst of data packets. In EH-DQ, each end-device can reserve a certain number of collision-free slots which depends on the energy available in the energy storage device and the number of data packets in each message. The performance of EH-DQ is analyzed and evaluated in terms

of the data delivery ratio and the data collection rate. The performance of EH-DQ is compared with that of EH-RDFSA, EH-DFSA and TDMA.

3.2 Chapter Outline

The remainder of this chapter is organized as follows. The system model and the performance metrics used in this chapter are introduced in Section 3.3 and Section 3.4, respectively.

Two different data collection scenarios (I and II) have been considered. In scenario I, each end-device has exactly 1 new data packet to send to the gateway in every data collection round, e.g., identification, measurement, alarm, etc. In scenario II, each end-device has 1 or more new data packets to send in every data collection round, e.g., set of measurements, images, etc.

Section 3.5 focuses on the design, analysis and performance evaluation of EH-aware MAC protocols for scenario I. In particular, the EH-DFSA protocol is analyzed and evaluated, and the EH-CTA protocol, based on tree splitting, is proposed, analyzed and evaluated.

Section 3.6 focuses on the design, analysis and performance evaluation of EH-aware MAC protocols for scenario II. In particular, the EH-RDFSA and EH-DQ protocols are proposed, analyzed and evaluated.

Finally, Section 3.7 concludes the chapter by summarizing the most relevant results and the key contents of the chapter.

3.3 System Model

3.3.1 Network and Data Model

We consider a wireless network in star topology formed by one coordinator (or gateway) and n end-devices in the communication range of the coordinator, as shown in Figure 3.1. Each end-device is equipped with a radio-transceiver, a microcontroller, several sensors, an energy harvester and an energy storage device (ESD). As depicted in Figure 3.2, the coordinator gathers data (e.g., measurements) from the end-devices by initiating periodic Data Collection Rounds (DCR). Each DCR starts when the coordinator broadcasts a

Request for Data (RFD) packet, once every T_R seconds. In the k-th DCR, each end-device has a number $l(k)$ of new data packets ready to be transmitted to the coordinator. The data process $l(k)$ can be modeled as a discrete random variable with probability mass function $p_j = \Pr\{l(k) = j\}$ with $j \in \{1, 2, ...\}$. The value of $l(k)$ is considered to be identically and independently distributed (i.i.d.) over all end-devices and DCRs. We assume that the data packets have a common and constant length.

At the beginning of the k-th DCR, an end-device enters into *active mode* to transmit data if the energy available in its ESD is above a predefined energy threshold. Otherwise, the end-device remains in *sleep mode* waiting for the next DCR. In the example of Figure 3.2, end-devices 1, 2, and 4 have enough energy to become active in the k-th DCR, while end-device 3 remains in sleep mode. In the $(k+1)$-th DCR, end-devices 2, 3, and 4 become active, while end-device 1 remains in sleep mode.

The k-th DCR is formed by a sequence of F_k frames where each end-device in active mode transmits data to the coordinator according to the rules of the adopted MAC protocol. The coordinator broadcasts a feedback packet (FBP) or beacon at the end of each frame to enable the synchronization of the active end-devices and to inform about the number of slots of the next frame depending on the MAC layer. An active end-device attempts to transmit its $l(k)$ data packets, one-by-one sequentially in time, as long as it has enough energy in its ESD. During the k-th DCR, an end-device enters into *sleep mode* when it either succeeds in transmitting the last of the $l(k)$ data packets or falls in energy shortage. Therefore, the k-th DCR finishes when all the active end-devices have entered into sleep mode. We assume that the duration $T_C(k)$ of the k-th DCR is much

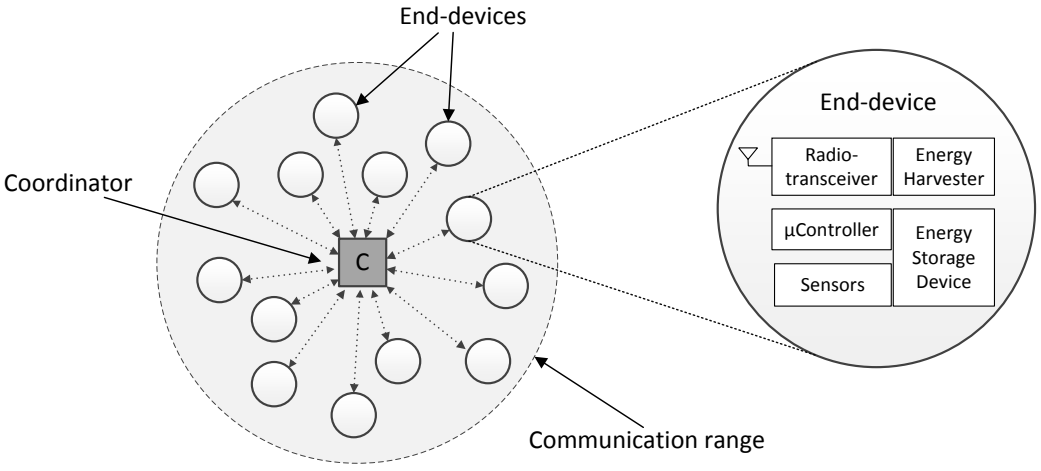


FIGURE 3.1: Wireless network with energy harvesting capabilities in star topology.

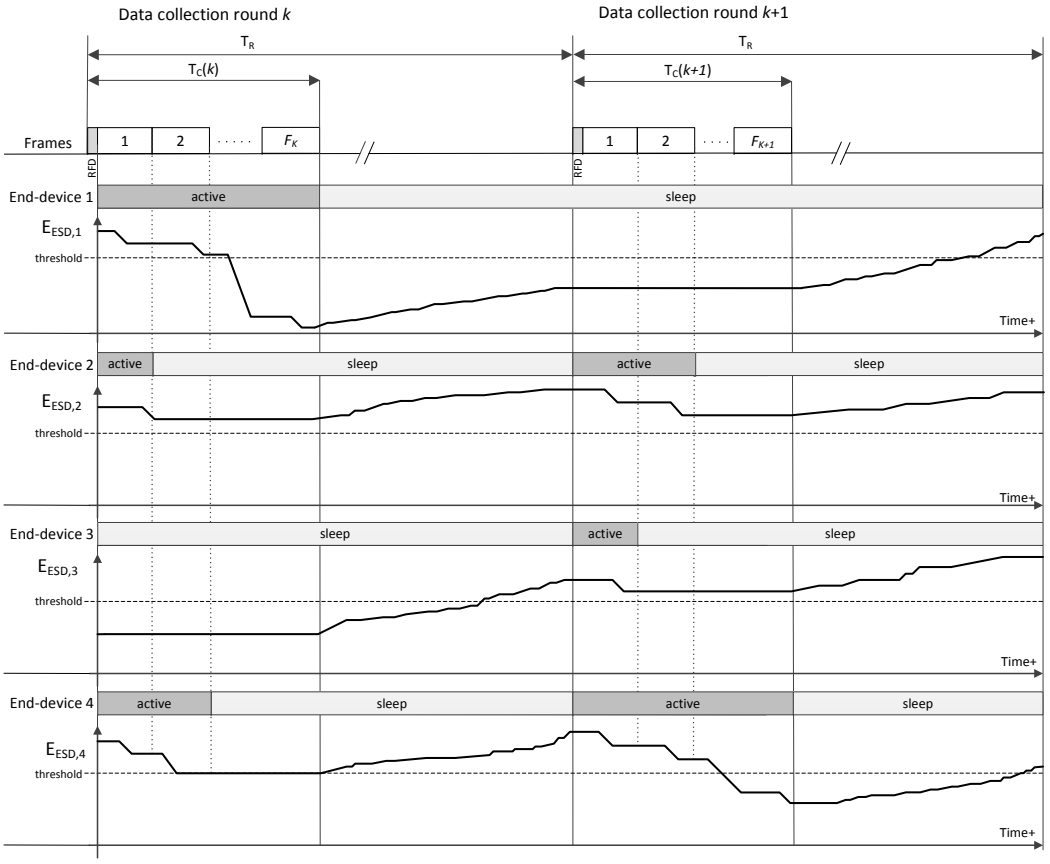


FIGURE 3.2: Sequence of data collection rounds with end-devices equipped with energy harvesters.

shorter than the time T_R between any two consecutive DCRs (i.e., $T_C(k) \ll T_R$ for all k) to ensure that successive DCRs do not overlap.

We consider that if an end-device fails to transmit one or more data packets due to energy shortage in any given DCR, those data packets are discarded and they are not transmitted in subsequent DCRs. This scenario represents very well applications where a number of sensors is transmitting data and some packets can be lost or not transmitted without seriously affecting the application. Just as an example, a meter reader in a smart grid may fail to transmit one reading of the energy consumption, even though the next reading will implicitly include this information. Therefore, no retransmission of a failed data packet would be needed.

In order to focus the study on the performance of the MAC layer, we assume that all packets are always transmitted without errors induced by the wireless channel. We assume that there is no capture effect, i.e., when two or more data packets collide, none of the packets involved in the collision can be decoded by the coordinator. The inclusion of transmission errors and capture effect constitutes part of our future work.

3.3.2 Energy Storage Model

The amount of energy stored in the ESD of an end-device can be modeled as a random variable which depends on the harvested energy and the energy consumed by the end-device throughout the DCRs. The energy stored in the ESD of the i-th end-device is denoted by $E_{ESD,i} \in \{0, 1\delta, 2\delta, ..., N\delta\}$, where δ [Joule] is referred to as *energy unit*, and N is the normalized capacity of the ESD. The end-device enters into active mode if the energy in its ESD at the beginning of the k-th DCR, denoted by $E_{ESD,i}(k)$, is above a certain energy threshold $E_{thr} = \varepsilon_{thr}\delta$, with $\varepsilon_{thr} \in \{0, 1, 2, ..., N-1\}$. The probability that an end-device can take part in the k-th DCR is called activation probability, denoted by $p_{active}(k)$, which can be expressed as

$$p_{active}(k) = \Pr\left\{E_{ESD,i}(k) > E_{thr}\right\}. \tag{3.1}$$

The energy threshold ε_{th} must be selected so as to maximize performance.

3.3.3 Energy Harvesting Model

The energy harvester of the i-th end-device captures an amount of energy, denoted by $E_{H,i}(k)$, for the time interval T_R between any two consecutive DCRs $(k-1)$-th and k-th, which can be expressed as

$$E_{H,i}(k) = \int_{T_R} P_{H,i}(t)dt, \tag{3.2}$$

where $P_{H,i}(t)$ is the instantaneous electrical power delivered by the energy harvester.

The harvested energy $E_{H,i}(k)$ can be modeled as a discrete random variable with a probability mass function $q_j = \Pr\left\{E_{H,i}(k) = j\delta\right\}$ with $j \in \{0, 1, 2, ...\}$ energy units, which depends on the characteristics of the energy source. $E_{H,i}(k)$ is considered to be i.i.d. with regard to other end-devices and DCRs.

The *energy harvesting rate*, denoted by $\overline{E}_H$, is defined as the average harvested energy of an end-device during the time T_R between the beginning of two consecutive DCRs, which can be expressed as

$$\overline{E}_H = \mathbb{E}\left[E_{H,i}(k)\right]. \tag{3.3}$$

We assume that the dynamics of the energy harvesting process is slower than the contention process in a DCR. Therefore, we consider that the amount of energy that is harvested within the duration $T_c(k)$ of the k-th contention process is negligible with respect to $E_{H,i}(k)$, and it is not immediately available to be used during the contention process. Consequently, all the harvested energy $E_{H,i}(k)$ is ready to be used by an end-device at the beginning of the k-th DCR.

3.3.4 Energy Consumption Model

Regarding energy consumption, the end-devices can be in four different modes of operation: (i) transmitting a packet; (ii) receiving; (iii) idle listening; or (iv) sleeping. The associated power consumptions are ρ_{tx}, ρ_{rx}, ρ_{σ}, or ρ_{sleep}, respectively. The model is general for any power-saving mechanism and any value of the power consumptions in different modes of operation. We assume that the energy required to switch between sleep and active modes (*i.e.*, transmitting, receiving, idle listening) is negligible. In sleep mode, the radio interface is fully disabled, and thus, the end-devices consume the lowest power consumption.

3.4 Performance Metrics

The following sections describe the criteria that are considered in this chapter to measure the performance of the energy harvesting-aware MAC protocols: the probability of delivery, the data delivery ratio and the time efficiency.

3.4.1 Probability of Delivery

The *probability of delivery* is considered in the scenario in which each end-device has exactly 1 new data packet ready to be transmitted to the coordinator in every DCR. The probability of delivery is defined as the probability that an end-device becomes active in the k-th DCR and succeeds in transmitting its data packet to the coordinator. An end-device fails to transmit data in a DCR if it enters in energy shortage before its data packet is successfully transmitted. The probability of delivery measures the ability of the MAC protocol to successfully deliver data from an end-device to the coordinator in a DCR without depleting its ESD.

3.4.2 Data Delivery Ratio

The *data delivery ratio* (DDR) is considered in the scenario in which each end-device has one or more new data packets ready to be transmitted to the coordinator in every DCR. The DDR is defined as the ratio between the number of data packets that are successfully transmitted to the coordinator in the k-th DCR, and the number of data packets ready to be transmitted at the beginning of the DCR. The DDR measures the ability of the MAC protocol to successfully deliver long data messages fragmented in small packets from the end-devices to the coordinator in every DCR without depleting their ESD.

3.4.3 Time Efficiency

The *time efficiency* is defined as the ratio between the duration of all the data packets successfully transmitted to the coordinator in the k-th DCR, and the time $T_C(k)$ required to complete the DCR. This value measures the probability that one slot allocated by the MAC layer during a DCR is successfully used. Therefore, the time efficiency is an indicator of the *data collection rate*, which can be obtained by dividing the time efficiency by the duration of a slot.

Due to the fluctuations of the harvested energy, the limited capacity of the ESDs, and collisions, the probability of delivery, the DDR and the time efficiency may be lower than 1. Since the use of energy harvesters potentially allows for perpetual operation, it is interesting to analyze the performance metrics when the system is in *steady-state*, i.e., for a DCR with large index k.

3.5 Scenario I: Single Data-Packet per Data Collection Round

This section focuses on EH-aware MAC protocols for data collection applications where each end-device is equipped with an energy harvester and has exactly one new data packet to transmit in every DCR, i.e., $l(k) = 1$. The structure of this section is organized as follows. Section 3.5.1 focuses on the analysis and performance evaluation of EH-DFSA. Section 3.5.2 focuses on the design, analysis and performance evaluation of EH-CTA.

3.5.1 Energy Harvesting-Aware Dynamic Frame Slotted-ALOHA (EH-DFSA)

This section focuses on the analysis and performance evaluation of the EH-DFSA protocol. The operation of EH-DFSA is decribed in Section 3.5.1.1. The energy consumption model of the end-devices using EH-DFSA is introduced in Section 3.5.1.2. Section 3.5.1.3 presents the theoretical analysis of EH-DFSA and formulates the expressions to calculate the probability of delivery and the time efficiency for EH-DFSA. Section 3.5.1.4 is devoted to validate the analysis and to evaluate the performance of EH-DFSA through comprehensive computer-based simulations.

3.5.1.1 MAC Protocol Description

In EH-DFSA, the contention process in each DCR is composed of a sequence of frames further divided into a variable number of slots. The number of slots is dynamically adjusted to be proportional to the estimated number of end-devices that contend in each frame, i.e., $m_i = \lceil \rho \cdot \hat{n}_i \rceil$, where m_i is the number of slots of frame F_i, $\hat{n}_i$ is the estimated number of end-devices that contend in frame F_i and ρ is a positive real number. An end-device that becomes active in a DCR randomly selects one of the slots in every frame to transmit its data packet. When an end-device either succeeds in transmitting its data packet or falls in energy shortage in a given frame, it enters into sleep mode and stops contending again in subsequent frames of the same DCR. Otherwise, it reattempts transmission of its data packet as long as it has enough energy to do so. At the end of each frame, the coordinator estimates the number of end-devices that will contend in the

next frame and broadcasts the number of slots in the FBP. An estimation algorithm for EH-DFSA is proposed in [28]. The algorithm takes into account the energy harvesting process. For the first frame, the estimation is based on the calculation of the average energy available in the ESDs at the beginning of a DCR in steady-state. For the next frames, the coordinator estimates: *(i)* the number of end-devices that collided in the previous frame by using the number of slots with success and collision [78], and *(ii)* the number of end-devices that will not have enough energy to contend in the next frame [28].

3.5.1.2 Energy Consumption Model

We assume that each time that an end-device transmits its data packet in a certain frame F_i of a DCR it consumes a constant amount of energy, denoted by E_{tx} [Joules], which accounts for all the energy consumption in the communication phases with the coordinator: *(i)* the end-device transmits in 1 slot and listens to the channel to receive an ACK, *(ii)* it remains in sleep mode in the other $m_i - 1$ slots, and *(iii)* it listens to the channel to receive the FBP.

We consider that the energy consumed by an end-device in sleep mode is negligible. For mathematical tractability, we normalize the energy consumption E_{tx} to one energy unit, i.e., $E_{tx} = 1\delta$.

3.5.1.3 Analysis of Performance Metrics

In order to derive an analytical model to compute the probability of delivery and the time efficiency in steady-state for EH-DFSA, we need to evaluate the steady-state probability distribution of the energy available in the ESDs at the beginning of a DCR, which depends on the energy harvesting process and the random slot selection in every frame. Given that the number of end-devices that contend in every frame depends on the energy available in the ESD at each end-device, deriving the exact steady-state probability distribution is not an easy task.

However, if we consider that the number of end-devices that contend in every frame is large, then the probability that an end-device succeeds in transmitting its data packet in a given frame can be approximated by the Poisson distribution, which only depends on the value of ρ. Consequently, we can evaluate the steady-state probability distribution of

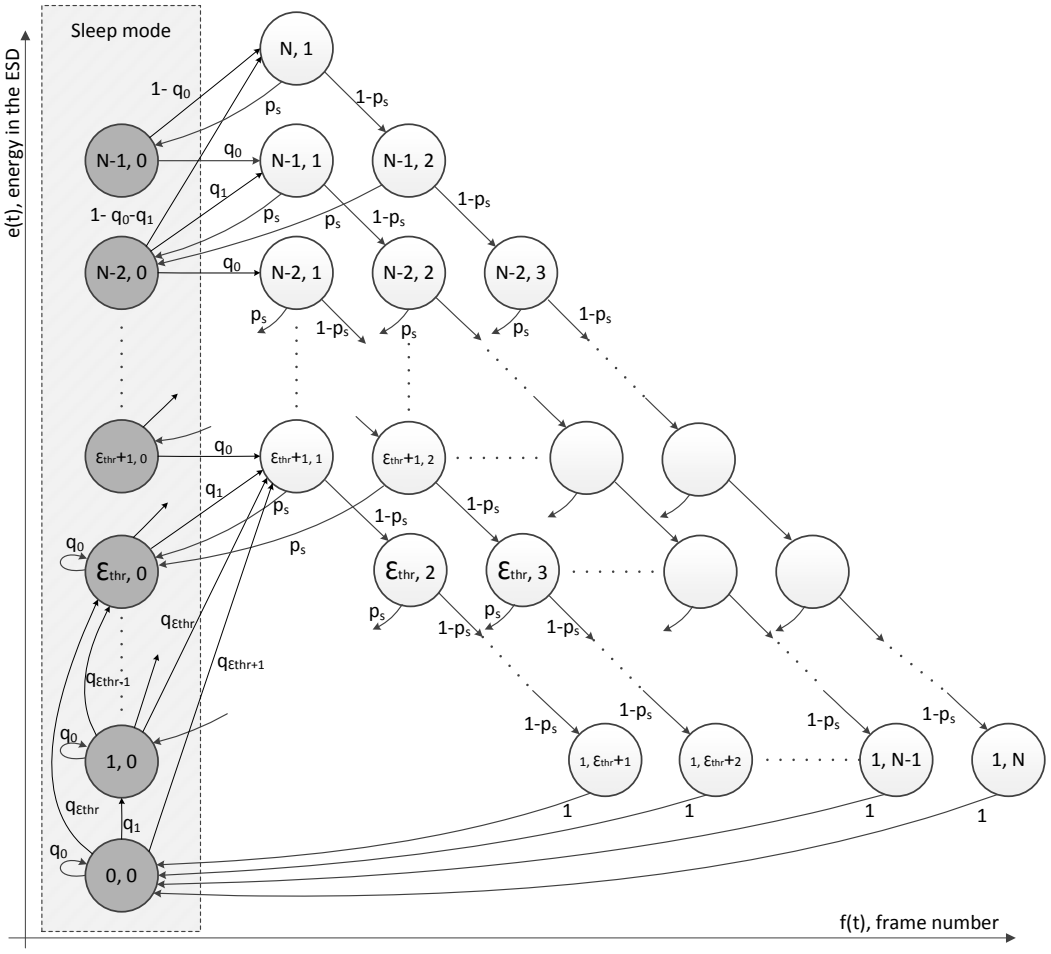


FIGURE 3.3: State transition diagram of the Markov chain that models the evolution of the energy available in an ESD using EH-DFSA. Some transitions have been intentionally omitted for ease of understanding.

the energy available in the ESDs by analyzing the evolution of the energy of a single ESD, which is an approximation that neglects the interactions among the ESDs of different end-devices.

In this Section, we present the Markov chain that models the evolution of the energy available in an ESD, we formulate the probability of success in a frame and the steady-state probability distributions, and we derive analytical expressions to calculate the probability of delivery and the time efficiency for EH-DFSA.

3.5.1.3.1 Markov Chain Model

We use a two dimensional Markov chain to model the energy available in the ESD of an end-device as shown in Figure 3.3. This Markov chain is based on the model developed in [27–29] to analyze the performance of DFSA in networks with energy-harvesting. In this thesis, that model has been adapted to EH-DFSA by including a minimum energy

(or threshold) that an end-device needs to start transmitting data to the gateway in a DCR. Each state of the chain is defined by $\{e(t), f(t)\}$, where $e(t) \in \{0, ...N\}$ is a stochastic process which represents the number of energy units stored in the ESD at time t, and $f(t) \in \{0, ...N\}$ is a stochastic process which represents the number of frame in which an end-device contends. An end-device is in sleep mode when $f(t) = 0$.

The Markov chain is characterized by a transition matrix $\mathrm{P} = [p_{ij}]$, where the one-step transition probabilities are defined as

$$p_{ij} = \Pr\{e(t+1) = e_j, f(t+1) = f_j | e(t) = e_i, f(t) = f_i\}. \tag{3.4}$$

The operations of an end-device across DCRs are as follows. When an end-device is not active in a DCR, it remains in sleep mode waiting for the next DCR. When a DCR starts, the amount of energy harvested during the previous T_R is added. Then, if the energy available in the ESD is above the energy threshold ε_{th}, the state of the end-device changes from sleep to active mode. See Figure 3.3 from $f(t) = 0$ to $f(t+1) = 1$. Otherwise, if the energy available in the end-device is still below or equal to the energy threshold when a DCR starts, the end-device remains in sleep mode.

The transition probability from the states in sleep mode is defined as

$$p_{ij} = \begin{cases} q_\varepsilon, & \text{if}(e_i \leq e_j)\text{and}(e_j \leq \varepsilon_{th})\text{and}(f_j = 0) \\ q_\varepsilon, & \text{if}(e_i \leq e_j)\text{and}(\varepsilon_{th} < e_j < N)\text{and}(f_j = 1) \\ 1 - \sum_{k=0}^{N-1-e_i} q_k, & \text{if}(e_i < e_j)\text{and}(e_j = N)\text{and}(f_j = 1) \\ 0, & \text{otherwise}, \end{cases}$$

where q_ε is the probability that an end-device captures $\varepsilon = (e_j - e_i)$ energy units. There are three different conditions in the transition from any state in sleep mode: (i) the end-device remains in sleep mode because it has not harvested energy, (ii) the end-device harvests ε (with $e_j < N$) energy units, and (iii) the end-device harvests an amount of energy equal to or above the capacity of the ESD.

Once an end-device becomes active at the beginning of a DCR, it will transmit its data packet in successive frames until it either succeeds in transmitting its data packet, enters in energy shortage, or both. The transition probability from any state in active

mode is defined as

$$p_{ij} = \begin{cases} p_s, & \text{if}(e_i > 1)\text{and}(e_j = e_i - 1)\text{and}(f_j = 0) \\ 1 - p_s, & \text{if}(e_i > 1)\text{and}(e_j = e_i - 1)\text{and}(f_j = f_i + 1) \\ 1, & \text{if}(e_i = 1)\text{and}(e_j = 0)\text{and}(f_j = 0) \\ 0, & \text{otherwise}, \end{cases}$$

where p_s is the probability that an end-device succeeds in transmitting its data packet in a given frame, and $1 - p_s$ is the probability that an end-device fails. There are three different conditions in the transition from any state in active mode: (i) if an end-device transmits its data packet successfully, it will change from state (e_i, f_i) to $(e_i - 1, 0)$, (ii) if the end-device does not succeed, it will change from state (e_i, f_i) to $(e_i - 1, f_i + 1)$, and (iii) if the energy level of the end-device is 1, it will change to (0,0), no matter whether the transmission is successful or not.

3.5.1.3.2 Probability of Success in one Frame

The probability that an end-device succeeds in transmitting its data packet in a given frame F_i, assuming that there are neither transmission errors nor capture effect, can be formulated as

$$p_s = m_i \left(\frac{1}{m_i}\right)\left(1 - \frac{1}{m_i}\right)^{n_i - 1}, \tag{3.5}$$

where m_i is the number of slots in the frame, and n_i is the number of end-devices which contend in that frame. Since we consider that $m_i \simeq \rho \cdot n_i$, the expression of p_s can be formulated as

$$p_s = \rho \cdot \binom{n_i}{k}\left(\frac{1}{m_i}\right)^k \left(1 - \frac{1}{m_i}\right)^{n_i - k} \text{ with } k = 1. \tag{3.6}$$

As it can be observed, p_s is expressed as a binomial distribution with $k = 1$. Then, if n_i is large, the expression of p_s can be approximated by the Poisson distribution:

$$p_s \simeq \rho \cdot \frac{e^{-1/\rho}}{\rho^k k!} \text{ with } k = 1. \tag{3.7}$$

Consequently, the probability that an end-device succeeds in a frame of EH-DFSA is constant, $p_s \simeq e^{-1/\rho}$, for all the frames of the contention process.

3.5.1.3.3 Energy Availability in Steady-state

As it can be observed in Figure 3.3, when $0 < p_s < 1$, $q_0 > 0$ and $q_1 > 0$, the Markov

chain is aperiodic and any state of the Markov chain can be reached from any other state with non-zero probability, and therefore the Markov chain is irreducible.

Since the Marchov chain is irreducible and aperiodic, and thus ergodic, it admits a unique steady-state probability distribution $\pi = \pi_{e,f}$ defined as

$$\pi_{e,f} = \lim_{t\to\infty} \Pr\{e(t) = e, f(t) = f\}, \tag{3.8}$$

which satisfies that $(\mathrm{P}' - \mathrm{I})\pi' = 0$, where P is the transition matrix and I is the identity matrix. By calculating the eigenvector of P' that corresponds to an eigenvalue equal to 1, we can solve the equation $(\mathrm{P}' - \mathrm{I})\pi' = 0$ and get π. Then, the probability of activation p^{SS}_{active} can be formulated as

$$p^{SS}_{active} = \pi^{B}_{\varepsilon_{th}+1,1} + \pi^{B}_{\varepsilon_{th}+2,1} + \cdots + \pi^{B}_{N,1} = \sum_{e=\varepsilon_{th}+1}^{N} \pi^{B}_{e,1}, \tag{3.9}$$

where $\pi^{B}_{e,f}$ is the steady-state probability distribution conditioned on being active at the beginning of a DCR, which can be formulated as

$$\pi^{B} = \pi^{S}\mathrm{P}, \tag{3.10}$$

where π^{s} is the steady-state probability distribution conditioned on being in sleep mode, which is calculated from the steady-state probability distribution as

$$\pi^{S}_{e,f} = \begin{cases} \frac{\pi_{e,0}}{\sum_{i=0}^{N-1} \pi_{i,0}}, & \text{if}\,(f = 0) \\ 0, & \text{if}\,(1 \le f \le N)\,. \end{cases} \tag{3.11}$$

3.5.1.3.4 Probability of Delivery

The *probability of delivery*, denoted by $p^{SS}_{delivery}$, is defined as the probability that an end-device becomes active and succeeds in transmitting its data packet in a DCR in steady-state conditions. Once the steady-state probability distribution conditioned on being active at the beginning of a DCR (3.10) is computed, we can formulate the expression to calculate $p^{SS}_{delivery}$. Recall that an end-device that enters in active mode at the begining of a DCR starts from one of the states $(e, 1)$, with $e \in \{\varepsilon_{th}+1, ..., N\}$, and then retransmits its data packet in the next frames as long as the available energy stored in ESD is enough. According to the steady-state probability distribution at the beginning of a DCR, the

expression of the probability of delivery can be formulated as

$$p_{delivery}^{SS} = \sum_{f=1}^{\varepsilon_{thr}} \sum_{e=\varepsilon_{thr}+1}^{N} \pi_{e,1}^{B} (1-p_s)^{f-1} p_s + \sum_{f=\varepsilon_{thr}+1}^{N} \sum_{e=f}^{N} \pi_{e,1}^{B} (1-p_s)^{f-1} p_s. \tag{3.12}$$

3.5.1.3.5 Time Efficiency

The *time efficiency*, denoted by η_t, can be formulated as the average number of slots with successful data packets in a DCR divided by the total number of slots in a DCR:

$$\eta_t = \frac{\sum_{i=1}^{N} \mathbb{E}[m_{S,i}]}{\sum_{i=1}^{N} \mathbb{E}[m_i]} = \frac{\sum_{i=1}^{N} p_t(1) \rho \mathbb{E}[n_i]}{\sum_{i=1}^{N} \rho \mathbb{E}[n_i]} = p_t(1), \tag{3.13}$$

where $\mathbb{E}[m_i]$ is the average number of contention slots in frame $i \in \{1, 2, ..., N\}$, $\mathbb{E}[m_{S,i}]$ is the average number successful slots in frame i, $\mathbb{E}[n_i]$ is the average number of contending end-devices in frame i, and $p_t(1)$ is the probability that one slot contains a successful data packet.

The probability that k of n_i contending end-devices transmit their data packet in the same slot of a frame with m_i slots can be formulated as

$$p_t(k) = \binom{n_i}{k} \left(\frac{1}{m_i}\right) \left(1 - \frac{1}{m_i}\right)^{n_i - k}. \tag{3.14}$$

If we consider a large number of contenders in every frame, we can resort to the standard Poisson approximation and obtain

$$p_t(k) \simeq \frac{e^{-\frac{1}{\rho}}}{\rho^k k!}, \tag{3.15}$$

and the probability that a given slot contains one successful data packet can be calculated as $p_t(1)$. Thus, the time efficiency can be expressed as

$$\eta_t \simeq \frac{e^{-\frac{1}{\rho}}}{\rho}. \tag{3.16}$$

Therefore, the time efficiency of EH-DFSA is constant, regardless of the number of end-devices and the probability distribution of the energy available in the ESDs, and it only depends on the value of ρ.

3.5.1.4 Model Validation and Performance Evaluation

In this section, we evaluate how the probability of delivery for EH-DFSA is influenced by the following parameters: *(i)* the energy harvesting rate $\overline{E}_H$, *(ii)* the total number n of end-devices, *(iii)* the energy threshold ε_{th} that determines whether an end-device becomes active in a DCR, and *(iv)* the capacity N of the ESDs. In addition, we have compared the analytical results with computer-based simulations using MATLAB. While in the theoretical model of EH-DFSA the steady-state probability distribution of the energy in the ESDs is calculated by analyzing the evolution of the energy of a single ESD, which is an approximation of the actual model, the simulation does not neglect the interactions among the ESDs of different end-devices. The simulation results of 500 samples have been averaged for every test case. Results show that the analytical results are tightly matched with the simulation results, thus validating the correctness of the analytical model proposed in Section 3.5.1.3.

3.5.1.4.1 Scenario

We consider a wireless network formed by 1 coordinator surrounded by a number n of end-devices in the transmission range of the coordinator. Each end-device includes an energy harvester and an ESD with $N = 10$ energy units of capacity. We assume that the energy harvested by an end-device during the time between two consecutive DCRs follows a binomial distribution with probability mass function

$$q_j = \begin{pmatrix} N_H \\ j \end{pmatrix} \left(\frac{\overline{E}_H}{N_H}\right)^j \left(1 - \frac{\overline{E}_H}{N_H}\right)^{N_H - j} \tag{3.17}$$

for $j \in \{0, 1, 2, ..., N_H\}$, where $N_H = 10$ is the maximum number of energy units that can be harvested, and $\overline{E}_H \in [0, 1, 2, ...N_H]$ is the energy harvesting rate. We consider that the coordinator adjusts the number of slots in every frame to be equal to the number of contenders, i.e., $\rho = 1$.

3.5.1.4.2 Energy Harvesting Rate

Figure 3.4 shows the probability of delivery for EH-DFSA as a function of the energy harvesting rate $\overline{E_H}$ (from 0.25 to 3 energy units). It has been evaluated by considering: $n \in \{100, 500, 1000\}$, $\varepsilon_{th} = 1$, and $N = 10$.

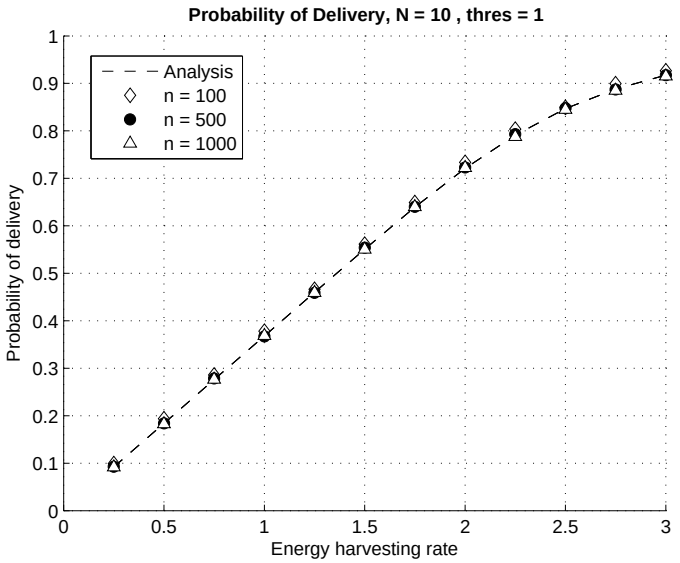


FIGURE 3.4: Probability of delivery over the energy harvesting rate using EH-DFSA.

As it could be expected, results show that the higher the energy harvesting rate, the higher the amount of energy that can be used to transmit data in a DCR. The higher the number of energy units stored in the ESD between two consecutive DCRs, the higher the number of possible retransmissions from each end-device, and thus the greater the probability of delivery.

It is worth noting that according to the analytical model presented in Section 3.5.1.3, the probability of delivery is independent of the number of end-devices in the network. This can be corroborated with the simulation results shown in Figure 3.4. This is due to the fact that the average number of retransmissions of an end-device does not change with the number of end-devices. Note that the probability of success in a given frame is constant for all the end-devices thanks to the dynamic reconfiguration of the number of slots per frame of EH-DFSA.

3.5.1.4.3 Number of End-Devices

Figure 3.5 shows the probability of delivery for EH-DFSA as a function of the number n of end-devices (from 100 to 1000). It has been evaluated by considering: $\overline{E_H} \in \{0.5, 1, 1.5, 2, 2.5, 3\}$, $\varepsilon_{th} = 1$, and $N = 10$.

Results show that the probability of delivery for EH-DFSA is insensitive to the number of end-devices. Since the number of slots per frame is adjusted dynamically by the coordinator based on the number of end-devices that contend in every frame, the

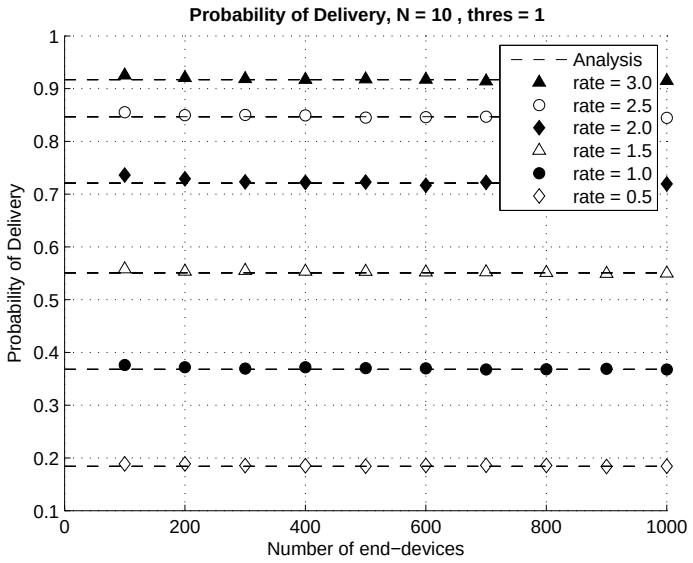


FIGURE 3.5: Probability of delivery over the number of end-devices using EH-DFSA.

probability of collision among end-devices remains approximately constant regardless of the number of end-devices.

In addition, it can be observed that the higher the energy harvesting rate, the higher the probability of delivery. It is worth noting that when the energy harvesting rate is lower than 1 energy unit, the probability of delivery is below 0.4.

3.5.1.4.4 Energy Threshold

Figure 3.6 shows the probability of delivery as a function of the energy threshold ε_{th} (from 1 to 9 energy units). It has been evaluated by considering: $n \in \{100, 1000\}$, $\overline{E_H} \in \{0.5, 1, 2, 3, 4, 5\}$, and $N = 10$.

Results show that the probability of delivery using EH-DFSA does not increase with the energy threshold. This is due to the fact that the probability that an end-device succeeds in a given frame (3.6) is approximately constant for all the frames of the contention process. However, when the energy threshold increases above a certain value, the average number of active end-devices per DCR is lower, i.e., the activation probability decreases, and thus the probability of delivery decreases.

3.5.1.4.5 Capacity of the Energy Storage Device

Figure 3.7 shows the probability of delivery as a function of the capacity N of the ESDs (from 3 to 10 energy units). It has been evaluated by considering: $n \in \{100, 500\}$, $\overline{E_H} \in \{0.5, 1, 2, 3\}$, and $\varepsilon_{th} = 1$.

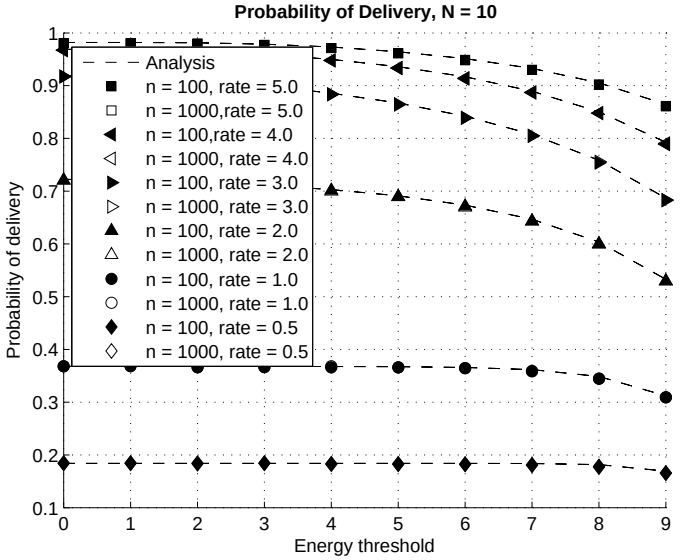


FIGURE 3.6: Probability of delivery over the energy threshold to enter in active mode at the beginning of each DCR using EH-DFSA.

Results show that the higher the capacity of the ESD, the higher the probability of delivery. For the higher energy harvesting rate (when $\overline{E_H} > 2$), the more sensitive the probability of delivery is with regard to the capacity of the ESD. On the contrary, the lower energy harvesting rate, the less sensitive the probability of delivery is with the increasing capacity of the ESD. When the capacity of the ESD is large, the amount of possible retransmissions per DCR is greater, and then the probability of transmitting a data packet successfully increases. In addition, the probability of delivery approximates

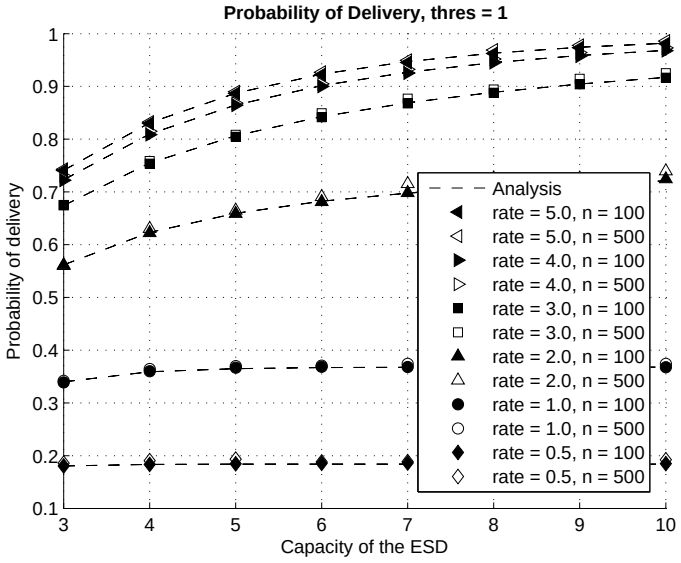


FIGURE 3.7: Probability of delivery over the capacity of the energy storage device using EH-DFSA.

to its maximum value when the capacity is above 6-7 energy units regardless of the energy harvesting rate. This is a very practical design guideline for the scenario considered.

3.5.1.5 Conclusions

In previous research works, the EH-DFSA protocol was proposed for data collection scenarios where the end-devices are equipped with energy harvesters. They propose a Markov chain model to analyze the evolution of the energy stored in an end-device and to compute the probability of delivery. Results show that the probability of delivery increases with the energy harvesting rate.

In this section, the performance of the EH-DFSA protocol has been analyzed and evaluated to demonstrate how the size of the batteries, the number of end-devices and the energy harvesting rate impact on the performance of EH-DFSA. It has been possible to identify optimal configuration regions to extend the lifetime of data collection networks with energy harvesting capabilities. A discrete-time Markov chain model has been used to analyze the performance of EH-DFSA.

3.5.2 Energy Harvesting-Aware Low Power Contention Tree-based Access (EH-CTA)

This section focuses on the design, analysis and performance evaluation of the EH-CTA protocol proposed in this thesis. The operation of EH-CTA is decribed in Section 3.5.2.1. The energy consumption model of the end-devices using EH-CTA is introduced in Section 3.5.2.2. Section 3.5.2.3 proposes a Markov chain model to analyze the evolution of the energy available in an end-device using EH-CTA, and formulates the expressions to calculate the probability of delivery and the time efficiency for EH-CTA. Section 3.5.2.4 is devoted to validate the analysis and to evaluate the performance of EH-CTA through comprehensive computer-based simulations. The comparison with the performance of EH-DFSA is also presented in this section.

3.5.2.1 MAC Protocol Description

The EH-CTA protocol implements a tree-splitting algorithm and takes into account the energy available in the ESDs. In EH-CTA, the end-devices which become active in a DCR are organized into sub-groups using a tree-splitting algorithm, which reduces the probability of collision per transmission attempt. The algorithm can be represented with a *contention tree* like the one depicted in Figure 3.8.a. Each node of the tree represents a time frame with a fixed number m of slots. The number in every slot of a frame denotes the number of end-devices that transmit in that slot. The tree-splitting algorithm works as follows. In the first frame, all the active end-devices randomly select one slot to transmit their data packet. If two or more end-devices collide in a specific slot, a new frame is assigned only to the sub-group of end-devices that caused the collision in order to retransmit, and they are queued into a logical queue referred to as the Collision Resolution Queue (CRQ). Therefore, if there are k slots with collision in one frame of level d of the tree, with $d \in \{1, 2, ..., N\}$, then k new frames are scheduled in level $d + 1$, and k sub-groups of end-devices are queued into the CRQ. The process leads to the formation of a tree whose expansion stops at frames with only successful and/or empty slots, meaning that all collisions have been resolved, or at level N, which is the normalized capacity of the ESD and thus limits the number of levels where an end-device can retransmit its data packet. An end-device transmits its data-packet in one frame of every level d of the tree until it succeeds in one level or enters in energy shortage.

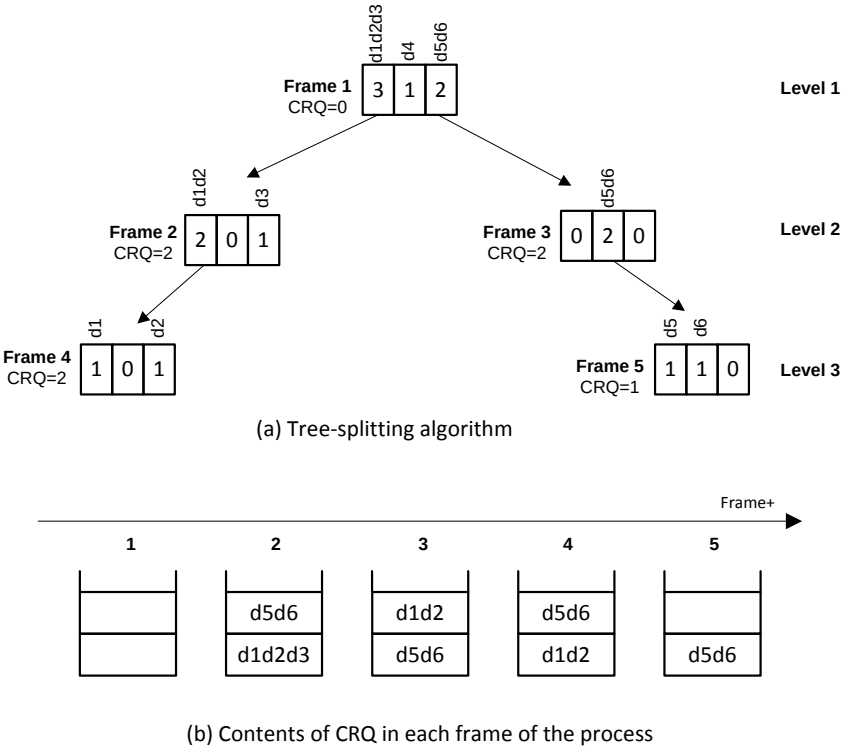


FIGURE 3.8: Example of a data collection round using EH-CTA with n=6 end-devices (d1 to d6) that become active at the beginning of the DCR, m=3 slots per frame, and $N \geq 3$: (a) diagram of the tree splitting algorithm that illustrates the length of CRQ and the end-devices that transmit in each frame; and (b) time diagram with the contents of CRQ in each frame.

The CRQ is represented at each end-device by two integer numbers: *(i)* the position of the end-device in the CRQ, and *(ii)* the total length of the CRQ, i.e., the number of sub-groups of end-devices that are waiting for retransmission. The coordinator broadcasts the status of the m slots and the length of the CRQ in the FBP at the end of each frame. Using this information, an end-device that has collided in a given frame can compute its position in the CRQ. Once an end-device has entered in the CRQ, it will retransmit its data packet in the next frame if it occupies the first position in the CRQ; otherwise, it just waits its turn without contending during successive frames. The position of an end-device in the CRQ is always decremented by one at the end of each frame. Consequently, an end-device only needs to receive the FBP of those frames where it contends in order to know whether it has succeeded (and then leaves the CRQ) or has collided (and then enters into the CRQ again).

An example of the evolution of the contents of the CRQ is shown in Figure 3.8.b. At frame 1, all the end-devices that become active (referred to as d1, d2, ... d6) transmit their data packet: d1, d2 and d3 collide in slot 1 and enter in the first position in the

CRQ; d5 and d6 collide in slot 3 and enter in the second position; and d4 succeeds in slot 2. At frame 2, d1, d2 and d3 contend because they occupy the first position in the CRQ: d1 and d2 collide and enter again in the CRQ; d3 succeeds and leaves the CRQ; and d5 and d6 move to the first position in the CRQ, thus, d5 and d6 do not contend in frame 2. At frame 3, d5 and d6 contend because they occupy the first position in the CRQ, they collide and enter again in the CRQ; and d1 and d2 do not contend and move to the first position in the CRQ. At frame 4, d1 and d2 contend and succeed; and d5 and d6 do not contend and move to the first position in the CRQ. Finally, d5 and d6 contend and succeed at frame 5.

3.5.2.2 Energy Consumption Model

We assume that each time that an end-device transmits its data packet in a certain frame of a DCR it consumes a constant amount of energy, denoted by E_{tx} [Joules], which accounts for all the energy consumption in the communication phases with the coordinator: *(i)* the end-device transmits in 1 slot, *(ii)* it remains in sleep mode in the other $m-1$ slots, and *(iii)* it listens to the channel to receive the FBP.

We consider that the energy consumed by an end-device in sleep mode is negligible. For mathematical tractability, we normalize the energy consumption E_{tx} to one energy unit, i.e., $E_{tx} = 1\delta$.

3.5.2.3 Analysis of Performance Metrics

In order to derive an analytical model to compute the performance metrics in steady-state for EH-CTA, we need to evaluate the steady-state probability distribution of the energy available in the ESDs at the beginning of a DCR, which depends on the energy harvesting process, the random slot selection in every frame and the tree splitting process. Given that the number of end-devices that contend in every level of the contention tree depends on the energy available in the ESD at each end-device, deriving the exact steady-state probability distribution is not an easy task. As an example, an exact stochastic model to analyze the energy evolution at the ESDs would require a set of interconnected Markov chains (one for each end-device).

However, if we adjust the value of the energy threshold ε_{th} (3.1) to guarantee that all the end-devices that become active (n_1) in a DCR will have enough energy to contend in

a certain number of levels, assuming that the number of end-devices that fall in energy shortage in a DCR is negligible, we can consider that the probability that an end-device succeeds in transmitting its data packet in one frame of any level of the contention tree basically depends on the value of n_1, the number of slots per frame, and the number of level. Consequently, we can evaluate the steady-state probability distribution of the energy available in the ESDs by analyzing the evolution of the energy of a single ESD, which is an approximation that neglects the interactions among the ESDs of different end-devices.

3.5.2.3.1 Markov Chain Model

The evolution of the energy available in the ESD of an end-device can be modeled with the discrete-time Markov chain shown in Figure 3.9. Each state in the chain is defined by $\{e(t), d(t)\}$, where $e(t) \in \{0, 1, ..., N\}$ is a stochastic process which represents the number of energy units stored in the ESD at time t; and $d(t) \in \{0, 1, ..., N\}$ is a stochastic process which represents that either an end-device is in sleep mode when $d(t) = 0$, or the level number in the contention tree in which an end-device contends when $d(t) \in \{1, ..., N\}$. Recall that in every level d of the tree, an end-device transmits its data packet in only one frame. Note that state transitions in the discrete-time Markov chain do not occur at fixed time intervals.

The Markov chain is characterized by a transition matrix $\mathrm{P} = [p_{ij}]$, where each element p_{ij} is the one-step transition probability defined as

$$p_{ij} = \Pr\{e(t+1) = e_j, d(t+1) = d_j | e(t) = e_i, d(t) = d_i\} \ . \tag{3.18}$$

An end-device that has either succeeded or entered in energy shortage in a DCR remains in sleep mode (i.e., in one of the states with $d_i = 0$) until the next DCR starts. At the beginning of a DCR, the energy harvested ε_H in the last T_R interval is added to the energy available in the ESD, i.e., $e_j = e_i + \varepsilon_H$. The probability that an end-device harvests $\varepsilon_H \in \{0, 1, 2, ...\}$ energy units is denoted by q_{ε_H}. Then, if the energy available in the ESD is below or equal a threshold ε_{th}, i.e., $e_j \in \{0, 1, ..., \varepsilon_{th}\}$, the end-device makes a transition from state $(e_i, 0)$ to state $(e_j, 0)$ and continues in sleep mode. On the contrary, an end-device becomes active if $e_j \in \{\varepsilon_{th} + 1, ..., N\}$, making a transition from state $(e_i, 0)$ to state $(e_j, 1)$. Note that e_j is limited to the capacity N of the ESD.

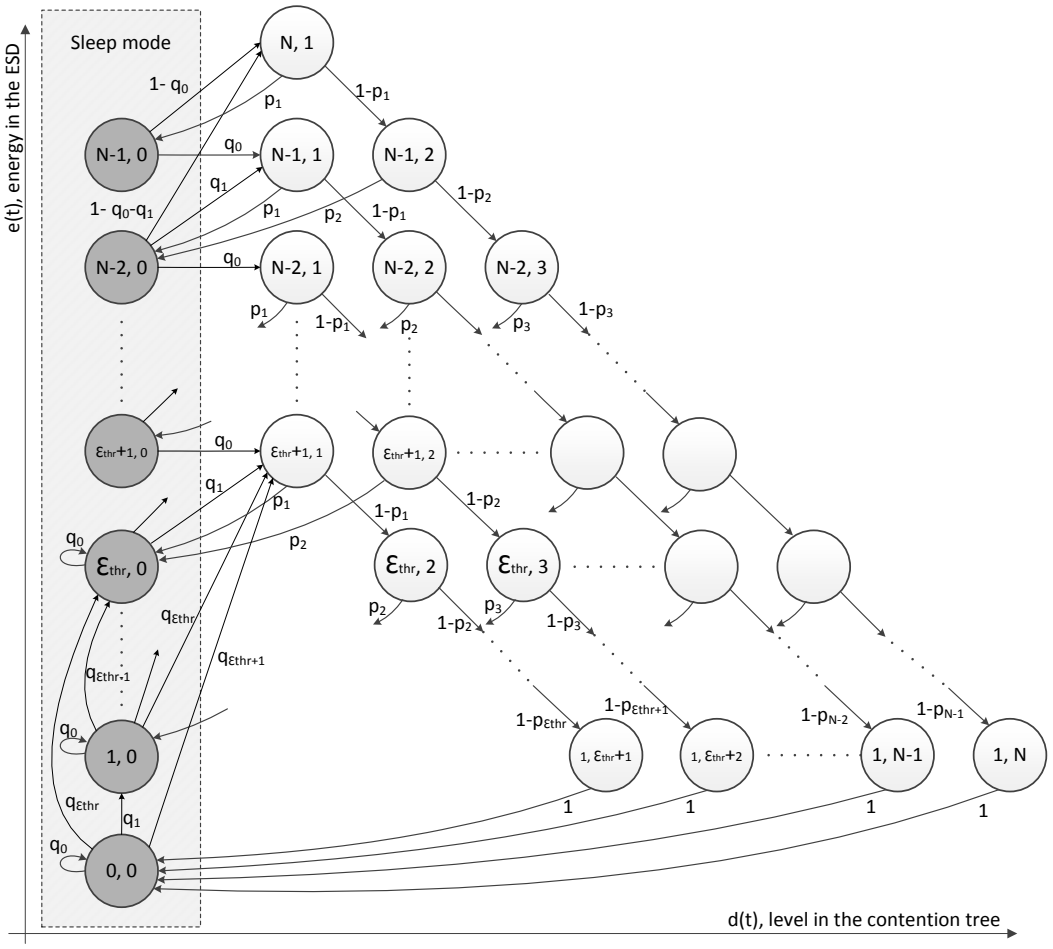


FIGURE 3.9: Generalized state transition diagram of the Markov chain that models the evolution of the energy available in an ESD using EH-CTA. Some transitions have been intentionally omitted for ease of understanding.

The transition probability from state $(e_i, 0)$ to any state (e_j, d_j) at the beginning of a DCR can be expressed as

$$p_{ij} = \begin{cases} q_{\varepsilon_H}, & if\,(e_i \leq e_j)\,and\,(e_j \leq \varepsilon_{th})\,and\,(d_j = 0) \\ q_{\varepsilon_H}, & if\,(e_i \leq e_j)\,and\,(\varepsilon_{th} < e_j < N)\,and\,(d_j = 1) \\ 1 - \sum_{k=0}^{N-1-e_i} q_k, & if\,(e_i < e_j)\,and\,(e_j = N)\,and\,(d_j = 1) \\ 0, & otherwise \end{cases}. \tag{3.19}$$

When an end-device becomes active, it retransmits in one frame of every successive level of the contention tree until either it succeeds or its ESD falls in energy shortage. If the end-device succeeds in one frame of level $d_i \in \{1, ..., N\}$, it makes a transition from state (e_i, d_i) to state $(e_i - 1, 0)$ for $e_i \in \{1, 2, ..., N\}$. However, if the end-device does not succeed, it can make two possible transitions: *(i)* to state $(e_i - 1, d_i + 1)$ when

the end-device has enough energy to retransmit in one frame of the next level, i.e., $e_j \in \{1, 2, ..., N-1\}$; or *(ii)* to state $(0, 0)$ when the end-device falls in energy shortage, i.e., $e_j = 0$.

The transition probability from state (e_i, d_i) to state (e_j, d_j) with $d_i \in \{1, 2, ..., N\}$ can be expressed as

$$p_{ij} = \begin{cases} p_d, & if\,(e_i > 1)\,and\,(e_j = e_i - 1)\,and\,(d_j = 0) \\ 1 - p_d, & if\,(e_i > 1)\,and\,(e_j = e_i - 1)\,and\,(d_j = d_i + 1) \\ 1, & if\,(e_i = 1)\,and\,(e_j = 0)\,and\,(d_j = 0) \\ 0, & otherwise \end{cases}, \tag{3.20}$$

where p_d is the probability that an end-device succeeds in transmitting its data packet in one frame of level $d = d_i$. This probability is derived in the next subsection.

3.5.2.3.2 Probability of Success in one Frame

The probability that an end-device succeeds in transmitting its data packet in one frame of level d can be expressed as

$$p_d = \left(1 - \frac{1}{m}\right)^{n_d - 1} \text{ for } d \geq 1, \tag{3.21}$$

where m is the number slots per frame and n_d is the number of end-devices which contend in one frame of level d.

In the first frame of a steady-state DCR, i.e., in level $d = 1$, the number n_1 of end-devices which contend is equal to the average number of end-devices that become active, which can be expressed as

$$n_1 = n \cdot p_{active}^{SS}. \tag{3.22}$$

where n is the total number of end-devices in the network and p_{active}^{SS} is the activation probability (3.1) in steady-state, i.e., for large index k of DCR, defined as

$$p_{active}^{SS} = \lim_{k \to \infty} p_{active}(k). \tag{3.23}$$

The value of the energy threshold ε_{th} must be set to guarantee that an end-device that becomes active in a DCR has enough energy to contend until success. In this

paper, we assume that the value of this threshold is set to ensure that the number of active end-devices that fall in energy shortage during a DCR is negligible. Under this assumption, the value of n_d for $d > 1$ can be derived as follows. First, the probability that k of n_d end-devices transmit in the same slot of a frame, denoted by $p_s(k)$, can be calculated as

$$p_s(k) = \binom{n_d}{k} \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{n_d - k}, \tag{3.24}$$

and the average number of empty, success, and collision slots in that frame, denoted by S_d^E, S_d^S, and S_d^C, respectively, can be calculated as

$$S_d^E = m \cdot p_s(0) = m\left(1 - \frac{1}{m}\right)^{n_d}, \tag{3.25}$$

$$S_d^S = m \cdot p_s(1) = n_d\left(1 - \frac{1}{m}\right)^{n_d - 1}, \tag{3.26}$$

$$S_d^C = m - S_d^E - S_d^S. \tag{3.27}$$

As described in Section 3.5.2.1, if there are S_d^C (3.27) slots with collision in one frame of level d, then $F_{d+1} = S_d^C$ new frames are scheduled in level $d+1$. Each new frame in level $d+1$ is assigned only to the sub-group of end-devices that caused a collision in the same specific slot of level d. The average number of end-devices that succeed in one frame of level d, denoted by n_d^S, is equal to the average number S_d^S (3.26) of slots with success. Therefore, the average number of end-devices that collide in one frame of level d, denoted by n_d^C, can be calculated as $n_d^C = n_d - S_d^S$. Then, since we assume that the number of end-devices that fall in energy shortage in a DCR is negligible, the n_d^C end-devices will have enough energy to contend again in F_{d+1} new frames of level $d+1$. Consequently, the average number of end-devices that contend in one frame of level $d+1$, denoted by n_{d+1}, can be approximated by the average number of end-devices n_d^C that collide in one frame of level d divided by the average number of frames F_{d+1} scheduled in level $d+1$ from the frame in level d where the n_d^C end-devices collided. Finally, the average number of end-devices that contend in one frame of level $d+1$ can

be expressed as

$$n_{d+1} \simeq \frac{n_d^C}{F_{d+1}} = \frac{n_d^C}{S_d^C} = \frac{n_d - n_d\left(1-\frac{1}{m}\right)^{n_d-1}}{m - m\left(1-\frac{1}{m}\right)^{n_d} - n_d\left(1-\frac{1}{m}\right)^{n_d-1}}. \tag{3.28}$$

The probability of success in one frame of every level of the contention tree (3.21) is represented in Figure 3.10. It has been evaluated with $m \in \{3, 10\}$, $n \in \{10 \cdot m, 100 \cdot m\}$, $p_{active}^{SS} = 1$, and considering that all the end-devices that become active in a DCR have enough energy to contend until they succeed in transmitting their data packet in the DCR regardless of the energy harvesting rate and the capacity of the ESDs. Results show a tight match between analytical and simulated results. As it could be expected, the value of p_d is close to 0 for low values of d, especially when the number m of slots is very low and the number n of end-devices is high.

In order to set the appropriate value for the energy threshold ε_{th} in EH-CTA, the one that guarantees that an end-device does not enter in energy shortage before it succeeds, we need to calculate the average number of frames where an end-device has to contend in a DCR until it succeeds, denoted by $\mathbb{E}[d]$, which can be expressed as

$$\mathbb{E}[d] = \sum_{d=1}^{\infty} d \cdot p_d \cdot \prod_{i=1}^{d-1} (1 - p_i). \tag{3.29}$$

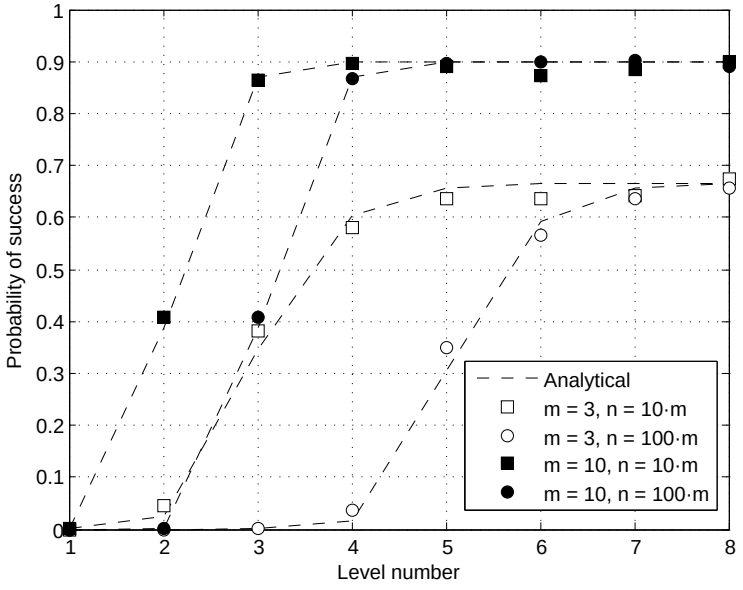


FIGURE 3.10: Probability that an end-device succeeds in transmitting in one frame of every level of the tree.

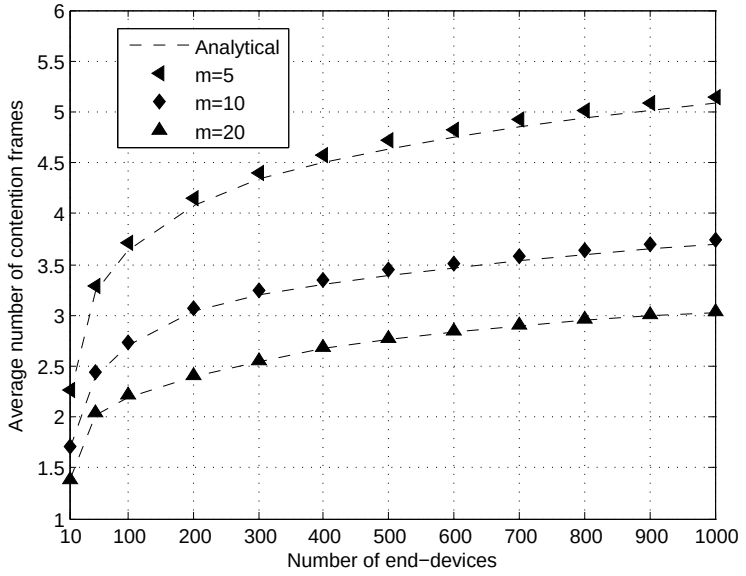


FIGURE 3.11: Average number of levels where an end-device contends until it succeeds in a DCR.

The value of $\mathbb{E}[d]$ is represented in Figure 3.11 as a function of the number of end-devices. It has been evaluated by considering $m \in \{5, 10, 20\}$. As it could be expected, the value of $\mathbb{E}[d]$ increases with n for a given value of m. The energy threshold needs to be adjusted to the value of $\mathbb{E}[d]$, i.e., $\varepsilon_{th} \approx \mathbb{E}[d]$, depending on the number of end-devices and slots. For example, for a network of 1000 end-devices, the energy threshold ε_{th} must be close to 5, 4, or 3 energy units when m is 5, 10, or 20 slots, respectively.

3.5.2.3.3 Energy Availability in Steady-state

As it can be observed in Figure 3.9, when $p_d > 0$ for $d \in \{1, ..., N\}$, $q_0 > 0$ and $q_1 > 0$, the Markov chain is aperiodic and any state of the Markov chain can be reached from any other state with non-zero probability, and therefore the Markov chain is irreducible.

Since the Marchov chain is irreducible and aperiodic, and thus ergodic, it admits a unique steady-state probability distribution, denoted by $\pi = [\pi_{e,d}]$, which can be expressed as

$$\pi_{e,d} = \lim_{t \to \infty} \Pr\{e(t) = e, d(t) = d\}, \tag{3.30}$$

and satisfies that

$$(\mathrm{P}' - \mathrm{I})\pi' = 0, \tag{3.31}$$

where P is the transition matrix and I is the identity matrix. Equation (3.31) can be solved for π by calculating the eigenvector of P$'$ that corresponds to an eigenvalue equal to 1. The steady-state probability distribution π is equal to the eigenvector with its elements normalized to sum one.

Recall that the transition matrix P depends on p_d (3.21), which also depends on the steady-state activation probability p_{active}^{SS} (3.67). On the other hand, p_{active}^{SS} can be expressed from the steady-state probability distribution of the energy available in the ESD at the beginning of a DCR, denoted by $\pi^B = \left[\pi_{e,d}^B\right]$, as follows

$$p_{active}^{SS} = \pi_{\varepsilon_{thr}+1,1}^B + ... + \pi_{N,1}^B = \sum_{e=\varepsilon_{thr}+1}^{N} \pi_{e,1}^B. \tag{3.32}$$

Note that all the values of π^B are zero for $d \in \{2, ..., N\}$. This is due to the fact that at the beginning of a DCR an end-device can only reach either states $(e, 0)$ with $e \in \{0, 1, ..., \varepsilon_{th}\}$ or $(e, 1)$ with $e \in \{\varepsilon_{th} + 1, ..., N\}$. Since the end-devices are in sleep mode before a DCR starts, π^B can be expressed as

$$\pi^B = \pi^S \mathrm{P}, \tag{3.33}$$

where $\pi^S = \left[\pi_{e,d}^S\right]$ is the steady-state probability distribution conditioned on being in sleep mode, which is calculated from the Markov chain steady-state probability distribution as

$$\pi_{e,d}^S = \begin{cases} \frac{\pi_{e,0}}{\sum_{i=0}^{N-1} \pi_{i,0}}, & if\ (d = 0) \\ 0, & if\ (1 \leq d \leq N) \end{cases}. \tag{3.34}$$

Finally, we use Algorithm 1 in order to calculate the steady-state probability distributions. The algorithm consists of the following steps. Firstly, we build the transition matrix P by setting the steady-state activation probability to a test value of 0, i.e., $p_{active-test}^{SS} = 0$. Secondly, we solve equations (3.31), (3.34), and (3.33) to calculate π, π^S, and π^B, respectively. Thirdly, we compute the analytical value of p_{active}^{SS} (3.32) by using π^B. And finally, we check the relative error between the test and analytical values of the activation probability. These steps are repeated iteratively by increasing

Algorithm 1 Calculation of the steady-state probability distributions π, π^S, and π^B

$p^{SS}_{active-test} \leftarrow 0$
repeat
  build P using $p^{SS}_{active-test}$
  compute π (3.31)
  compute π^S (3.34)
  compute π^B (3.33)
  compute p^{SS}_{active} (3.32)
  $error \leftarrow (p^{SS}_{active} - p^{SS}_{active-test})/p^{SS}_{active} \cdot 100$
  if $error \geq 0.1$ **then**
    $p^{SS}_{active-test} \leftarrow p^{SS}_{active-test} + 0.01$
  end if
until $error < 0.1$

$p^{SS}_{active-test}$ until the error is below 0.1%, which indicates that $p^{SS}_{active-test}$ satisfies (3.31), (3.34) and (3.33), and the results obtained for π, π^S, and π^B are correct.

3.5.2.3.4 Probability of Delivery

Once the steady-state probability distribution π^B of the energy available in the ESD at the beginning of a DCR is computed, we can calculate the steady-state *probability of delivery* for EH-CTA, denoted by $p^{SS}_{delivery}$. Recall that an end-device which becomes active in a steady-state DCR starts in one of the states $(e, 1)$ with probability $\pi^B_{e,1}$ for $e \in \{\varepsilon_{th}+1, ..., N\}$. Then, the end-device retransmits its data packet in one frame of every level d until the packet is successfully decoded by the coordinator with probability p_d. The end-device fails if it enters in energy shortage before the packet is successfully decoded. Consequently, the expression of $p^{SS}_{delivery}$ can be formulated as

$$p^{SS}_{delivery} = \sum_{d=1}^{\varepsilon_{thr}} \sum_{e=\varepsilon_{thr}+1}^{N} \pi^B_{e,1} \prod_{k=1}^{d-1} (1-p_k)\, p_d + \sum_{d=\varepsilon_{thr}+1}^{N} \sum_{e=d}^{N} \pi^B_{e,1} \prod_{k=1}^{d-1} (1-p_k)\, p_d, \quad (3.35)$$

where the first term of the summation corresponds to the probability that an end-device succeeds in levels $d \in \{1, ..., \varepsilon_{thr}\}$ and the second term is the probability that an end-device succeeds in levels $d \in \{\varepsilon_{thr}+1, ..., N\}$. As it can be observed in Figure 3.9, an end-device can succeed in levels $d \in \{1, ..., \varepsilon_{thr}\}$ when it becomes active at the beginning of a DCR with an energy level $e \in \{\varepsilon_{thr}+1, ..., N\}$, and it can succeed in levels $d \in \{\varepsilon_{thr}+1, ..., N\}$ when its energy level is $e \in \{d, ..., N\}$ at the beginning of the DCR.

3.5.2.3.5 Time Efficiency

The *time efficiency*, denoted by η_t, can be formulated as the average number of slots with successful data packets in a DCR divided by the total number of slots in a DCR, which can be expressed as

$$\eta_t = \frac{\sum_{d=1}^{N} \mathbb{E}\left[m_{S,d}\right]}{\sum_{d=1}^{N} \mathbb{E}\left[m_d\right]}, \tag{3.36}$$

where $\mathbb{E}\left[m_d\right]$ is the average number of contention slots in level $d \in \{1, ..., N\}$ and $\mathbb{E}\left[m_{S,d}\right]$ is the average number successful slots in level d. Recall that in EH-CTA if there are S_d^C (3.27) slots with collision in one frame of level d, then S_d^C new frames are scheduled in level $d+1$, and thus the average number of frames in level d can be calculated as $\prod_{i=2}^{d} S_{i-1}^C$. Therefore, the time efficiency of EH-CTA can be formulated as

$$\eta_t = \frac{S_1^S + \sum_{d=2}^{N} S_d^S \prod_{i=2}^{d} S_{i-1}^C}{m + \sum_{d=2}^{N} m \prod_{i=2}^{d} S_{i-1}^C}, \tag{3.37}$$

where m is the number of slots per frame, S_d^S (3.26) is the average number of successful slots in a frame of level d, and S_d^C (3.27) is the average number of collided slots in a frame of level d.

In the next section, we validate the analytical model proposed to calculate the steady-state probability of delivery and time efficiency for EH-CTA.

3.5.2.4 Model Validation and Performance Evaluation

In this section, we evaluate the performance of EH-CTA, in terms of probability of delivery and time efficiency, and compare with the performance of EH-DFSA. In addition, we validate the analytical model proposed in Section 3.5.2.3 by comparing the analytical results with computer-based simulations that execute the rules of the EH-CTA protocol without any simplification. While in the theoretical model of EH-CTA the steady-state probability distribution of the energy in the ESDs is calculated by analyzing the evolution of the energy of a single ESD, which is an approximation of the actual model, the simulation does not neglect the interactions among the ESDs of different end-devices.

In the next paragraphs, we first describe the considered scenario, and then discuss the numerical results in order to study how the performance of the protocols is influenced by the following parameters: the energy threshold ε_{thr}; the number m of slots per frame in EH-CTA; the energy harvesting rate $\overline{E}_H$; the number n of end-devices; and the capacity N of an ESD.

3.5.2.4.1 Scenario

We consider a wireless network formed by 1 coordinator surrounded by a number n of end-devices in the transmission range of the coordinator. Each end-device includes an energy harvester and an ESD with $N = 10$ energy units of capacity. We assume that the energy harvested by an end-device during the time between two consecutive DCRs follows a binomial distribution with probability mass function

$$q_j = \begin{pmatrix} N_H \\ j \end{pmatrix} \left(\frac{\overline{E}_H}{N_H}\right)^j \left(1 - \frac{\overline{E}_H}{N_H}\right)^{N_H - j} \tag{3.38}$$

for $j \in \{0, 1, 2, ..., N_H\}$, where $N_H = 10$ is the maximum number of energy units that can be captured by an energy harvester, and $\overline{E}_H \in [0...N_H]$ is the energy harvesting rate (3.3).

Results for EH-DFSA have been obtained through computer-based simulations using MATLAB. We consider an ideal EH-DFSA protocol in which the number of contenders per frame is perfectly estimated. The results of 1000 simulation samples have been averaged for each test case.

3.5.2.4.2 Energy Threshold

The probability of delivery using EH-CTA and EH-DFSA is represented in Figure 3.12 as a function of the energy threshold ε_{thr} (from 0 to 9 energy units). It has been evaluated by considering: $\overline{E}_H = 3$, $n \in \{100, 1000\}$, $m \in \{5, 10, 20\}$ in EH-CTA, and $N = 10$. Recall that an end-device becomes active at the beginning of a DCR if the energy available in its ESD is above the energy threshold, which guarantees a minimum number $\varepsilon_{thr} + 1$ of packet retransmissions by an end-device in a DCR.

As it could be expected, results show that there exists an optimum energy threshold that maximizes the probability of delivery for EH-CTA. This optimum value is close to the average number of frames $\mathbb{E}[d]$ (3.70) in which an end-device has to contend in a

DCR until it succeeds, i.e., $\varepsilon_{thr} \approx \mathbb{E}[d]$. The value of $\mathbb{E}[d]$ is represented in Figure 3.30 as a function of the number of end-devices and considering $m \in \{5, 10, 20\}$. As it can be observed in Figure 3.12, the optimum energy threshold is close to 5, 4, or 3 energy units when m is 5, 10, or 20 slots, respectively.

Figure 3.12 shows a slight deviation between analytical and simulation results when the energy threshold is lower than the optimum value (e.g., for $n = 1000$, $m = 5$ and $\varepsilon_{thr} < 3$). This is due to the fact that our Markov chain is an approximation of the actual model which assumes that the number of end-devices that fall in energy shortage in a DCR is negligible, and in order to fulfill this assumption, the energy threshold must be high enough to guarantee that the end-devices that become active in a DCR will have enough energy to contend in a certain number of levels.

While the probability of delivery using EH-CTA increases with the energy threshold until it reaches its maximum value, results show that the probability of delivery using EH-DFSA does not increase with the energy threshold and it is insensitive to the number of end-devices. This is due to the fact that in EH-DFSA, since we consider that the number of slots per frame is adjusted to be equal to the number of end-devices that contend in every frame, and assuming that there are neither transmission errors nor capture effect, the probability that an end-device succeeds in a given frame of EH-DFSA is approximately constant ≈ 0.36 (3.7) for all the frames [39, 83]. However, as it

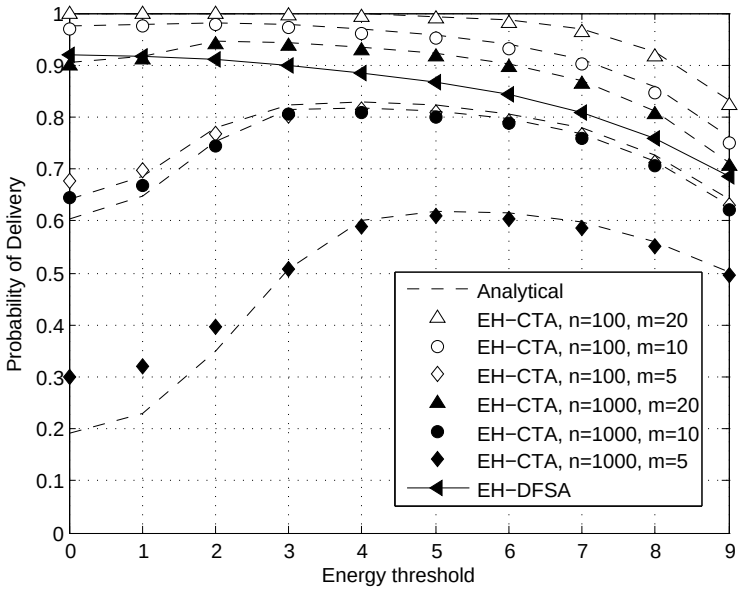


FIGURE 3.12: Probability of delivery over the energy threshold using EH-CTA and EH-DFSA.

can be observed in Figure 3.29, the probability that an end-device succeeds in a given frame of level d in EH-CTA is much higher than 0.36 when the level number in the contention tree increases. For example, when $m = 10$ slots per frame and $n = 10 \cdot m$ end-devices, the probability that and end-device succeeds in one frame of level 2 and 3 is 0.4 and 0.9, respectively.

Finally, as it can be observed in Figure 3.12, the probability of delivery using EH-CTA and EH-DFSA decays dramatically when the energy threshold increases above a certain value. Indeed, when the energy threshold is too high, the activation probability decreases, thus reducing the probability of delivery.

3.5.2.4.3 Number of Slots

The probability of delivery and the time efficiency, for EH-CTA and EH-DFSA, are represented in Figure 3.13 and Figure 3.14, respectively, as a function of the number m of slots per frame (from 2 to 40 slots). They have been evaluated by considering: $n \in \{100, 1000\}$, $\overline{E}_H \in \{0.25, 0.5, 2, 4, 8\}$, and $N = 10$. In EH-CTA, the value of the energy threshold has been set to $\varepsilon_{thr} = 6$ for $m < 5$, $\varepsilon_{thr} = 5$ for $m = 5$, $\varepsilon_{thr} = 4$ for $5 < m \leq 10$, and $\varepsilon_{thr} = 3$ for $10 < m \leq 40$. In EH-DFSA, $\varepsilon_{thr} = 0$. Recall that in EH-DFSA the number of slots per frame is adjusted to be equal to the number of contenders in every frame.

Results show that the probability of delivery for EH-CTA increases when the number of slots per frame increases. This is due to the fact that the probability p_d that and

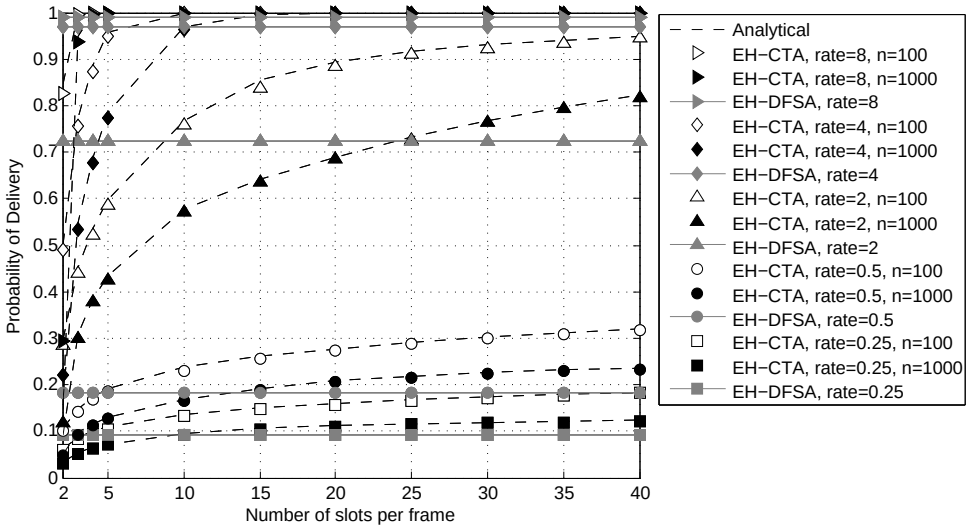


FIGURE 3.13: Probability of delivery for EH-CTA over the number of slots per frame.

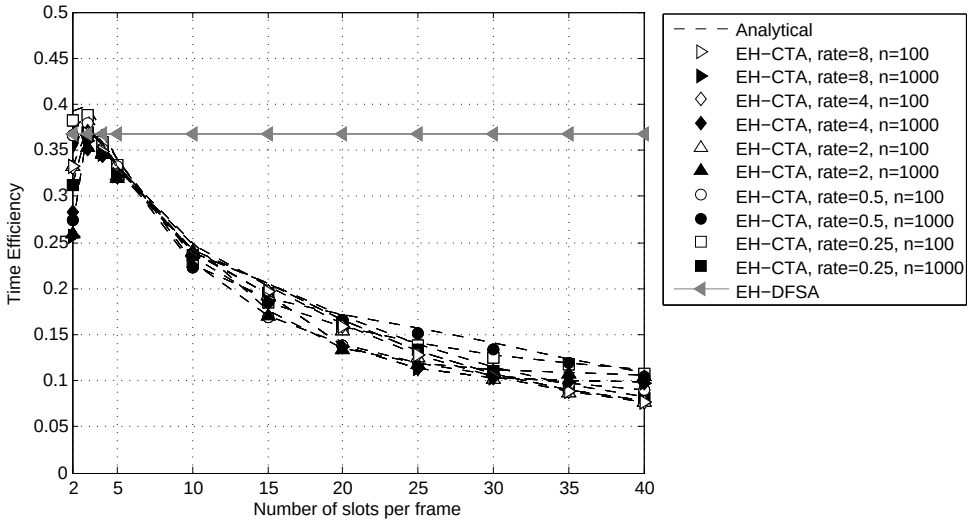


FIGURE 3.14: Time efficiency for EH-CTA over the number of slots per frame.

end-device succeeds in one frame of level d of the contention tree increases when m increases, thus leading the average number of retransmissions and the energy consumed per end-device to decrease, and the probability of delivery to increase.

The probability of delivery for EH-CTA and EH-DFSA increases with the energy harvesting rate. Indeed, the higher the number of energy units available in the ESDs, the higher the number of end-devices that become active in a DCR and the higher the number of possible packet retransmissions. In addition, if the total number of end-devices increases, the probability of delivery for EH-CTA is degraded because the probability of collision is higher. However, the probability of delivery for EH-DFSA is insensitive to the number of end-devices because the number of slots is equal to the number of contenders in every frame.

As it can be observed in Figure 3.13, EH-CTA can outperform EH-DFSA in terms of probability of delivery, for any energy harvesting rate and number of end-devices, if the number of slots per frame in EH-CTA is properly adjusted. For example, if $n = 100$ and $\overline{E}_H \in \{0.25, 0.5, 2, 4, 8\}$, then the value of m in EH-CTA must be equal or greater than 4, 5, 8, 7 and 3 slots, respectively; and if $n = 1000$ and $\overline{E}_H \in \{0.25, 0.5, 2, 4, 8\}$, then the value of m in EH-CTA must be equal or greater than 10, 15, 25, 10 and 4 slots, respectively.

As it can be observed in Figure 3.14, the time efficiency for EH-CTA is maximized for $m = 3$ slots, $\eta_t \approx 0.38$, and it is degraded as the number of slots per frame increases. Indeed, while the number of frames required to resolve the contention in EH-CTA tends

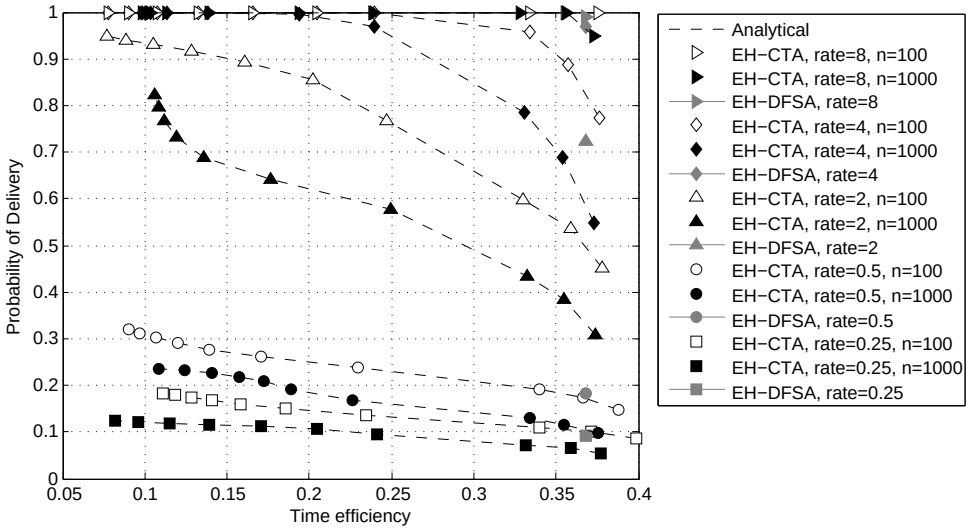


FIGURE 3.15: Trade-off between probability of delivery and time efficiency.

to its minimum value as the number of slots increases, the delay (in slots) is minimized for $m = 3$ slots regardless of the number of end-devices that contend in the DCR [82]. In addition, EH-CTA slightly outperforms EH-DFSA in terms of time efficiency at $m = 3$ and it is very similar for different energy harvesting rates and numbers of end-devices.

The trade-off between probability of delivery and time efficiency is shown in Figure 3.15 considering a number of slots per frame $m \in \{3, ..., 40\}$ in EH-CTA. When the number of slots per frame increases in EH-CTA, more end-devices can eventually succeed in transmitting data to the coordinator in a DCR, thus increasing the probability of delivery, at the cost of reducing the time efficiency and the data collection rate. In addition, as it can be observed in Figure 3.13 and Figure 3.15, with low (e.g., 0.25) and high (e.g., 8) energy harvesting rates, EH-CTA can be configured with a very low number of slots per frame, at almost no cost in the probability of delivery, and increase the time efficiency to a certain value close to the maximum. However, with intermediate energy harvesting rates (e.g., between 0.5 and 4 energy units), EH-CTA must be configured with a number of slots per frame which depends on the total number of end-devices in the network and the harvesting rate. This can be easily implemented in EH-CTA by including in the RFD and FBP packets one field to inform about the frame length to be used in every DCR.

In addition, while the number of slots per frame in EH-DFSA must be adapted to the number of contenders in every frame, thus requiring very long frames in highly dense networks, results show that EH-CTA can outperform EH-DFSA even using a short and

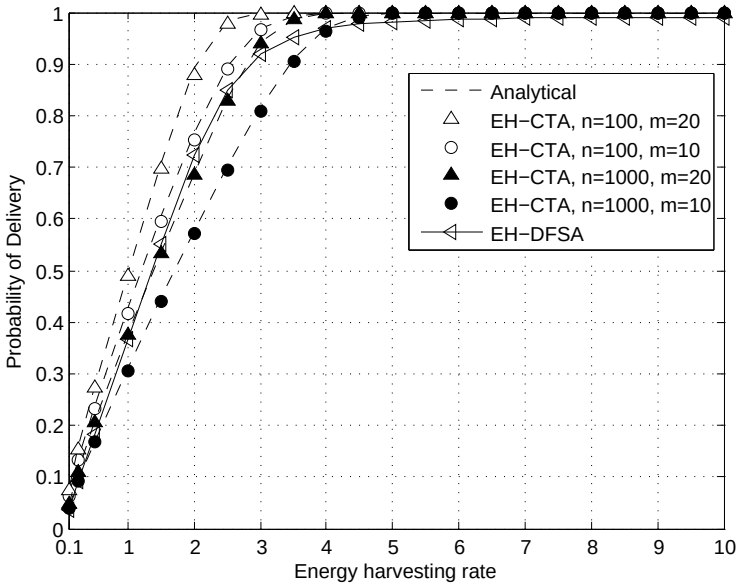


FIGURE 3.16: Probability of delivery over the energy harvesting rate using EH-CTA and EH-DFSA.

fixed frame length in EH-CTA, i.e., $n/m \gg 1$. Therefore, EH-CTA facilitates scalability and synchronization as the number of end-devices in the network increases.

3.5.2.4.4 Energy Harvesting Rate

The probability of delivery using EH-CTA and EH-DFSA is represented in Figure 3.16 as a function of the energy harvesting rate $\overline{E}_H$ (from 1 to 10 energy units). It has been evaluated by considering: $n \in \{100, 1000\}$, $m \in \{10, 20\}$ in EH-CTA, $N = 10$, and the optimum energy thresholds.

Results show that the probability of delivery using EH-CTA and EH-DFSA increases almost linearly with the energy harvesting rate. Indeed, the higher the number of energy units available in an ESD, the higher the number of possible retransmissions in a DCR. In addition, it can be observed that the slope of the probability of delivery using EH-CTA is higher when the number of end-devices is low and the numbers of slots increases. Indeed, as the probability of collision is lower, the energy consumption due to retransmissions is reduced, thus increasing the probability of delivery.

As it could be expected, EH-CTA provides a probability of delivery equal to 1 when the energy harvesting rate is above the optimum energy threshold, i.e., 4 or 3 energy units when m is 10 or 20 slots, respectively, and it outperforms the probability of delivery provided by EH-DFSA for $n \in \{100, 1000\}$ end-devices when the number of slots in EH-CTA is $m = 20$. Indeed, since the tree splitting algorithm organizes the end-devices into

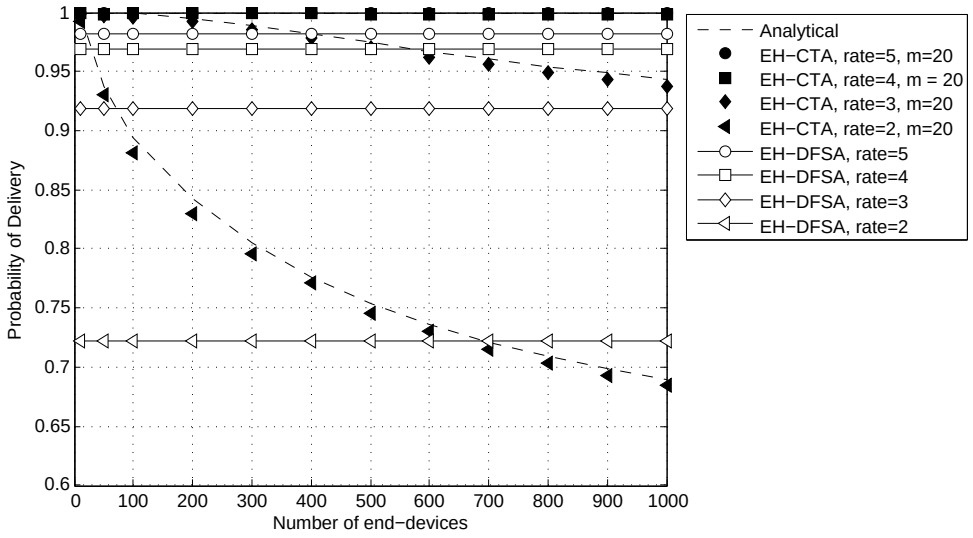


FIGURE 3.17: Probability of delivery over the number of end-devices using EH-CTA and EH-DFSA.

sub-groups to reduce the probability of collision in every level of the contention tree, the average number of retransmissions is lower in EH-CTA than in EH-DFSA, and EH-CTA allows that the data packets of the active end-devices can be eventually transmitted in a DCR with a finite number of retransmissions and energy consumption.

Results show that EH-CTA with $m = 20$ requires lower energy harvesting rate than EH-DFSA to get the same probability of delivery. For example, while EH-DFSA requires $\overline{E}_H \approx 5$ to obtain $p^{SS}_{delivery} \approx 0.98$, EH-CTA requires $\overline{E}_H \approx 2.5$ and $\overline{E}_H \approx 3.5$ if $n = 100$ and $n = 1000$, respectively, which means a reduction of 50% and 30% in energy harvesting rate. Consequently, EH-CTA allows reducing the total time between consecutive DCRs and thus increases the network throughput.

3.5.2.4.5 Number of End-Devices

The probability of delivery using EH-CTA and EH-DFSA is represented in Figure 3.17 as a function of the number n of end-devices (from 10 to 1000). It has been evaluated by considering: $\overline{E}_H \in \{2, 3, 4, 5\}$, $m = 20$ in EH-CTA, and $N = 10$.

As it could be expected, results show that the probability of delivery using EH-CTA decays exponentially with the number n of end-devices for any energy harvesting rate and number of slots. Indeed, the higher the number of end-devices, the higher the probability of collision, and the higher the probability of energy shortage. It can be observed that if the energy harvesting rate is $\overline{E}_H = 2$, the probability of delivery using EH-CTA decays rapidly when the value of n increases up to 700. However, if $\overline{E}_H \geq 3$,

the probability of delivery using EH-CTA decreases much more slowly with n. Indeed, a higher energy harvesting rate compensates the higher energy consumption due to a large value of n. Results show that when the number of slots per frame in EH-CTA is $m = 20$, EH-CTA outperforms the probability of delivery provided by EH-DFSA for $n \leq 1000$ end-devices. For example, if $\overline{E}_H = 2$, EH-CTA provides a gain of 24% in probability of delivery when $n = 100$, and a gain of 4% when $n = 500$. If $\overline{E}_H = 3$, EH-CTA provides a gain of 10%, 8% and 3% when $n = 100$, $n = 500$, and $n = 1000$, respectively. If $\overline{E}_H = 4$ or $\overline{E}_H = 5$, EH-CTA provides a gain of 4% or 2%, respectively, for $10 \leq n \leq 1000$.

Furthermore, while EH-DFSA requires to estimate the number of contending end-devices in every frame in order to adjust dynamically the frame length to the number of contenders, EH-CTA does not require any complex estimation algorithm. On the contrary, EH-CTA uses a fixed frame length during all the contention process. In addition, while EH-DFSA requires very long frame lengths as the number of end-devices in the network increases, the frame length of EH-CTA can be very short, e.g., 20 slots. Consequently, EH-CTA provides the system with higher probability of delivery than EH-DFSA using short and fixed frame lengths for a wide range in the number of end-devices, which enables scalability and facilitates synchronization in wireless M2M networks.

3.5.2.4.6 Capacity of the Energy Storage Device

The probability of delivery using EH-CTA and EH-DFSA is represented in Figure 3.18 as a function of the capacity N of the ESD (from 3 to 10 energy units). It has been evaluated by considering: $\overline{E}_H \in \{0.5, 1, 2\}$, $n \in \{100, 500\}$, and $m = 20$ in EH-CTA.

As it could be expected, results show that the probability of delivery using EH-CTA and EH-DFSA increases with the capacity of the ESD. The variation in the probability of delivery is higher when the energy harvesting rate is above 0.5. On the contrary, it is less sensitive to the capacity of the ESD when the energy harvesting rate is low. Indeed, a larger capacity allows storing larger amounts of harvested energy, and consequently, increases the probability of delivery. As it can be observed, the probability of delivery tends to its maximum value when the capacity is above 5-6 energy units regardless of the energy harvesting rate.

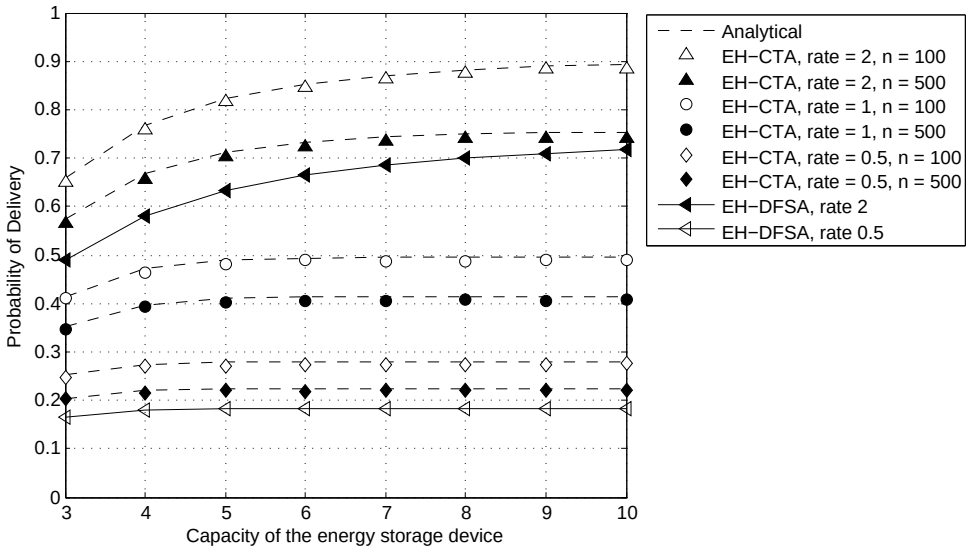


FIGURE 3.18: Probability of delivery over the capacity of the ESD using EH-CTA and EH-DFSA.

3.5.2.5 Conclusions

A new MAC protocol, named Energy Harvesting-aware Low Power Contention Tree-based Access protocol (EH-CTA), has been proposed in this section. EH-CTA is suitable for M2M data collection networks where each end-device is equipped with an energy harvester and an energy-storage device (ESD). EH-CTA uses an m-ary tree-splitting algorithm to resolve collisions and takes the energy availability into account. An energy efficient implementation of the m-ary tree splitting algorithm based on a Collision Resolution Queue has been proposed.

A Markov chain model has been presented to analyze the evolution of the energy available in an end-device using EH-CTA, and to compute analytically two performance metrics: the probability of delivery, which measures the ability of the MAC protocol to successfully transmit data to the gateway from the end-devices without depleting their energy reserves; and the time efficiency (or data collection rate), which measures the average number of data packets received by the coordinator per time slot.

The performance of EH-CTA has been evaluated and compared with that of EH-DFSA. Results show the trade-off between both performance metrics and how they are influenced by the number of slots per frame, the energy harvesting rate and the number of end-devices. The time efficiency for EH-CTA is maximized for $m = 3$ slots, and it is degraded as the number of slots increases. However, if the number of slots is high, more end-devices can eventually succeed in transmitting data to the gateway, thus increasing

the probability of delivery. In addition, results show that with very low and very high energy harvesting rates, EH-CTA can be configured with a very low number of slots, at almost no cost in the probability of delivery, and increase the time efficiency to a certain value close to the maximum. However, with intermediate energy harvesting rates, EH-CTA must be configured with a number of slots per frame which depends on the total number of end-devices and the harvesting rate.

Results show that EH-CTA can outperform EH-DFSA in terms of probability of delivery, for any energy harvesting rate and number of end-devices, if the number of slots in EH-CTA is properly adjusted. For example, with 1000 end-devices and energy harvesting rates of 0.25, 0.5, 2, 4 or 8 energy units, the number of slots per frame in EH-CTA must be equal or greater than 10, 15, 25, 10 or 4 slots, respectively, to improve the probability of delivery with respect to EH-DFSA. Results show that EH-CTA with 20 slots per frame requires lower energy harvesting rate to get the same probability of delivery provided by EH-DFSA. Therefore, EH-CTA allows reducing the time between data collection rounds and increases the throughput with respect to EH-DFSA.

Furthermore, while EH-DFSA requires to estimate the number of contending end-devices in every frame to adapt the frame length, EH-CTA uses a fixed frame length. While the frame length of EH-CTA can be very short, the frame length of EH-DFSA must be as high as the number of contenders, thus causing scalability problems as the number of end-devices increases. Taking that into account, EH-CTA is an interesting alternative for data collection scenarios with energy harvesting. The analytical model proposed for EH-CTA is useful to define specific design requirements, such as the capacity of the ESDs, the configuration of the MAC layer, and the energy harvesting rate, according to the expected density of the network and the required performance.

3.6 Scenario II: Multiple Data-Packets per Data Collection Round

This section focuses on EH-aware MAC protocols for data collection applications where each end-device is equipped with an energy harvester and has a number $l(k) \geq 1$ of data packets ready to transmit to the coordinator in every DCR k, where $l(k)$ is a discrete random variable. The value of $l(k)$ is independent of other end-devices and has a new realization in every DCR.

The structure of this section is organized as follows. Section 3.6.1 focuses on the design, analysis and performance evaluation of EH-RDFSA. Section 3.6.2 focuses on the design, analysis and performance evaluation of EH-DQ.

3.6.1 Energy Harvesting-Aware Reservation Dynamic Frame Slotted-ALOHA (EH-RDFSA)

This section focuses on the design and analysis of the EH-RDFSA protocol proposed in this thesis and the performance evaluation of EH-RDFSA, EH-DFSA and TDMA. The operation of EH-RDFSA and TDMA is decribed in Section 3.6.1.1. The energy consumption model of the end-devices is introduced in Section 3.6.1.2. Section 3.6.1.3 proposes a discrete time Markov chain model to analyze the evolution of the energy available in an end-device using EH-RDFSA, and formulates the expressions to calculate the data delivery ratio and the time efficiency for EH-RDFSA. An algorithm to estimate at the coordinator the number contenders in every frame of EH-RDFSA is proposed in Section 3.6.1.4. Section 3.6.1.5 is devoted to validate the analysis and to evaluate the performance of EH-RDFSA through comprehensive computer-based simulations. The comparison with the performance of EH-DFSA and TDMA is also presented in this section.

3.6.1.1 MAC Protocols Description

3.6.1.1.1 EH-RDFSA

In EH-RDFSA, each frame is divided in three parts as shown in Figure 3.19: *(i)* a variable number m_R of reserved slots, *(ii)* a variable number m_C of contention slots, and *(iii)* a feedback packet (FBP). A guard time called Inter Frame Space (IFS) is left

between reception and transmission modes to compensate propagation and processing delays and the time required to switch the radio transceivers between reception and transmission. The number of reserved slots in the first frame is always 0.

An end-device that becomes active in the k-th DCR randomly selects one of the contention slots in every frame to transmit the first of its data packets. From the point of view of the coordinator, a contention slot can be in one of three states: empty, if no packet has been transmitted in that slot; success, if only one packet has been transmitted; or collision, if more than one end-device has transmitted in that slot.

When an end-device succeeds in transmitting the first packet in a given frame, one reserved slot will be allocated to the end-device for subsequent frames. The coordinator informs the end-device about the specific reserved slot with an ACK transmitted within the same slot where the data packet was transmitted. Then, the end-device transmits its other $l(k) - 1$ data packets in the reserved slot (one packet per frame) as long as it has enough energy. The coordinator responds with an ACK to each data packet decoded successfully in each slot. The duration of a slot is adjusted to fit the transmission time of a data packet from an end-device and of an ACK from the coordinator, with the necessary guard times and considering both packet length and data transmission rates.

A reserved slot is released either once an end-device has transmitted all its data packets or enters in energy shortage. The header of every data packet includes a flag that indicates whether it is the last packet of the sequence of $l(k)$ packets to be transmitted in this DCR, and one field which informs about the energy available in the end-device. This information is used by the coordinator to calculate the number m_R of reserved slots in the next frame.

In every frame i of a DCR, the coordinator estimates the number n_i of active end-devices that contend to transmit their first data packet, i.e., the number of end-devices that have failed in previous frames and still have enough energy to contend in frame i.

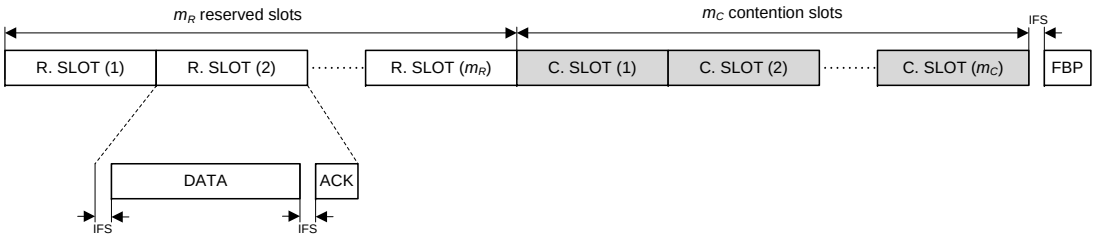


FIGURE 3.19: Frame structure of EH-RDFSA.

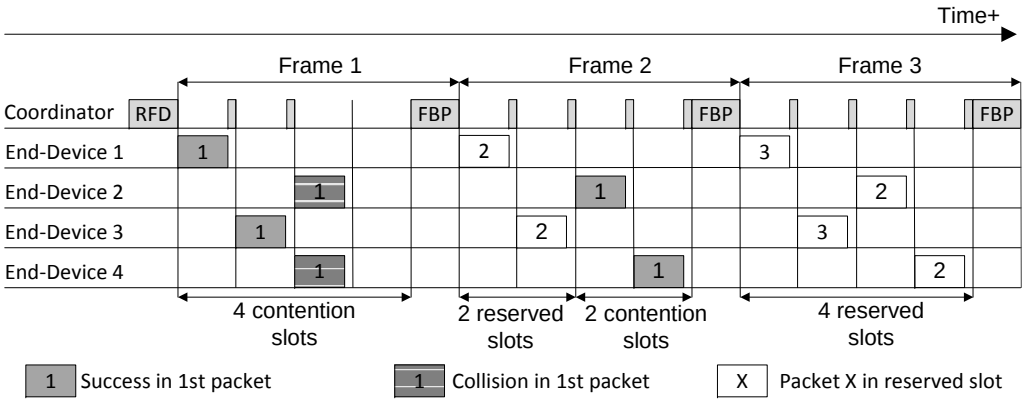


FIGURE 3.20: Example of a DCR using EH-RDFSA considering that 4 end-devices enter in active mode when the DCR starts. For the sake of simplicity, the example only shows the 3 first frames of one possible realization.

The coordinator adjusts the number m_C of contention slots as $m_C = \lceil \rho \cdot \hat{n}_i \rceil$, where ρ is a positive real number and $\hat{n}_i$ is the estimated number of contenders in frame i. The coordinator broadcasts a FBP at the end of each frame to inform about the values of m_R and m_C for the next frame.

The example of Figure 3.20 shows the operation of EH-RDFSA with 4 end-devices in active mode. In frame 1, end-devices 1 and 3 succeed in transmitting their first packet. Thus, 2 new reserved slots are allocated in frame 2 for both end-devices. Since end-devices 2 and 4 collide in frame 1, they contend again to transmit their first packet in frame 2, where they succeed. Thus, 2 new reserved slots are allocated in frame 3 for end-devices 2 and 4. While the number of reserved slots increases in every frame (from 0 to 4), the number of contention slots decreases (from 4 to 0). The communication process continues after frame 3 until every end-device has either transmitted all its packets or entered in energy shortage.

3.6.1.1.2 TDMA

In TDMA, each frame is composed of a fixed number m_R of reserved slots that is equal to the number n of end-devices in the network. Each slot is allocated to one end-device. An end-device that becomes active in the k-th DCR will transmit its $l(k)$ data packets in its reserved slot in $l(k)$ successive frames (one packet per frame) as long as the end-device has enough energy. The coordinator responds with an ACK to each data packet decoded successfully within the same slot where the data packet has been transmitted. The coordinator broadcasts a feedback packet (FBP) or beacon at the end of each frame to enable the synchronization of the active end-devices.

3.6.1.2 Energy Consumption Model

We assume that each time that an end-device transmits its data packet in a certain frame of a DCR it consumes a constant amount of energy, denoted by E_{tx} [Joule], which accounts for all the energy consumption in the communication phases with the coordinator: *(i)* the end-device transmits in 1 slot and listens to the channel to receive an ACK, *(ii)* it remains in sleep mode in the other slots, and *(iii)* it listens to the channel to receive the FBP.

For mathematical tractability, we consider that the energy consumption E_{tx} is equal to K energy units δ [Joule], i.e., $E_{tx} = K\delta$, where K is a positive integer number. We consider that the energy consumed by an end-device in sleep mode is negligible.

3.6.1.3 Analysis of Performance Metrics

In order to derive an analytical model to compute the DDR and the time efficiency in steady-state for EH-RDFSA, we need to evaluate the steady-state probability distribution of the energy available in the ESDs at the beginning of a DCR, which depends on the energy harvesting process, the random slot selection in every frame, and the number of data packets transmitted by each end-device in previous DCRs. Given that the number of end-devices that contend in every frame depends on the energy available in the ESD at each end-device, deriving the exact steady-state probability distribution is not an easy task. However, if we consider that the number of end-devices that contend in every frame is large, then the probability that an end-device succeeds in transmitting its first data packet in a given frame can be approximated by the Poisson distribution, which only depends on the value of ρ. Consequently, we can evaluate the steady-state probability distribution of the energy available in the ESDs by analyzing the evolution of the energy of a single ESD, which is an approximation that neglects the interactions among the ESDs of different end-devices.

In this section, we describe a theoretical model to compute the performance metrics. To this end, in Section 3.6.1.3.1 we propose a discrete-time Markov chain model to analyze the energy evolution of an ESD across DCRs using EH-RDFSA. In Section 3.6.1.3.2, we derive the probability that an end-device succeeds in transmitting its first data-packet in a given frame. Then, in Section 3.6.1.3.3 we derive the steady-state probability distribution of the energy available in an ESD at the beginning of a DCR. In

Section 3.6.1.3.4, we formulate the data delivery ratio. In Section 3.6.1.3.5, we derive the probability that one contention slot of a frame is empty, successful or collided. Finally, we formulate the time efficiency in Section 3.6.1.3.6.

3.6.1.3.1 Markov Chain Model

The evolution of the energy available in the ESD of an end-device can be modeled with a discrete-time Markov chain as shown in Figure 3.21. Each state in the chain is defined by $\{e(t), f(t)\}$, where $e(t) \in \{0, 1, ..., N\}$ is a stochastic process which represents the number of energy units stored in the ESD at time t, and $f(t) \in \left\{0, 1, ..., \left\lfloor \frac{N}{K} \right\rfloor\right\}$ is a stochastic process which represents that either an end-device is in sleep mode, when $f(t) = 0$, or the number of frame in which an end-device contends to transmit the first of its data packets. Recall that K is the number of energy units δ consumed by the end-device when it transmits a data packet, and $N\delta$ is the capacity of the ESD.

The Markov chain is characterized by a transition matrix $\mathrm{P} = [p_{ij}]$, where p_{ij} is the one-step transition probability defined as

$$\Pr\{e(t+1) = e_j, f(t+1) = f_j | e(t) = e_i, f(t) = f_i\} \ . \tag{3.39}$$

The operations of an end-device across DCRs using EH-RDFSA are as follows. When an end-device has either succeeded or entered in energy shortage, it remains in sleep mode (with $f_i = 0$) until the next DCR starts. At the beginning of a DCR, the amount ε_H of energy harvested in the previous DCR is added to the energy in the ESD, i.e., $e_j = e_i + \varepsilon_H$. Then, if the energy available in the ESD is above the energy threshold ε_{th}, i.e., $e_j \in \{\varepsilon_{th} + 1, ..., N\}$, the state of the end-device changes from sleep $(e_i, 0)$ to active mode $(e_j, 1)$. Otherwise, if the energy in the ESD is below or equal to the energy threshold when a DCR starts, the end-device remains in sleep mode, i.e., it makes a transition to state $(e_j, 0)$. Consequently, the transition probability from a state $(e_i, 0)$ in sleep mode to any state (e_j, f_j) can be expressed as

$$p_{ij} = \begin{cases} q_{\varepsilon_H}, & if\,(e_i \leq e_j)\,and\,(e_j \leq \varepsilon_{th})\,and\,(f_j = 0) \\ q_{\varepsilon_H}, & if\,(e_i \leq e_j)\,and\,(\varepsilon_{th} < e_j < N)\,and\,(f_j = 1) \\ 1 - \sum_{k=0}^{N-1-e_i} q_k, & if\,(e_i < e_j)\,and\,(e_j = N)\,and\,(f_j = 1) \\ 0, & otherwise \end{cases} , \tag{3.40}$$

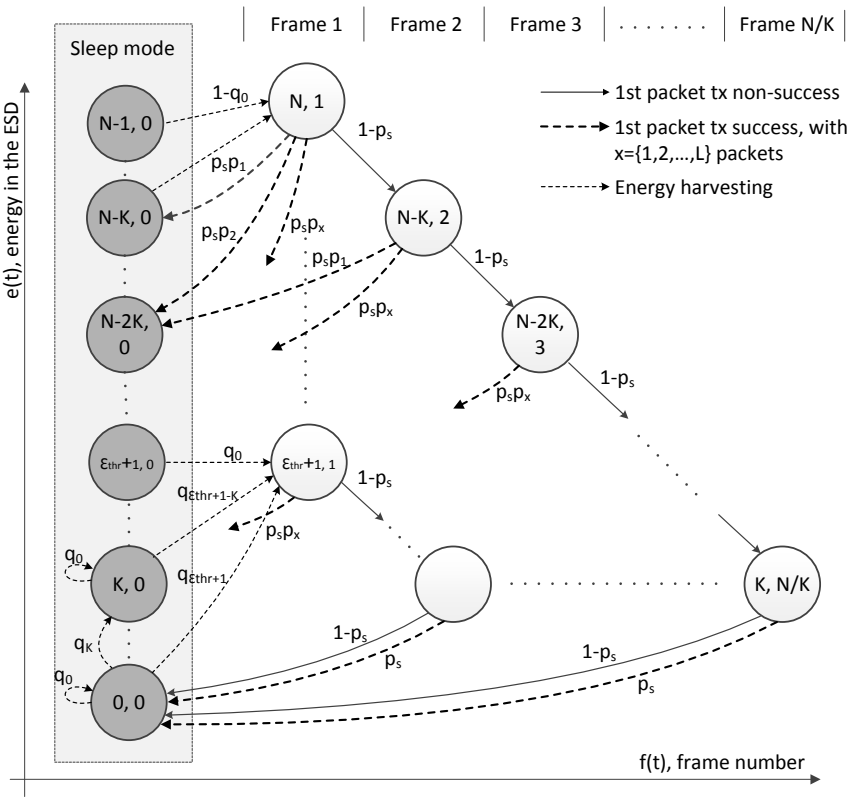


FIGURE 3.21: Generalized state transition diagram of the Markov chain that models the energy evolution of an ESD using EH-RDFSA. Some transitions have been intentionally ommited for ease of understanding.

where q_{ε_H} is the probability that an end-device harvests ε_H energy units, being $\varepsilon_H = e_j - e_i$ with $e_i \leq e_j$.

Once an end-device becomes active at the beginning of a DCR, it will transmit its first data packet in successive frames until either it is successfully received by the coordinator, or the end-device enters in energy shortage, or both. If the end-device does not succeed in transmitting its first data packet in frame f_i, which happens with probability $1 - p_s$, it can make two possible transitions: *(i)* to state $(e_i - K, f_i + 1)$ if it has enough energy to retransmit in the next frame, i.e., $e_j \geq K$; or *(ii)* it enters in sleep mode when $e_j < K$.

Once an end-device succeeds in transmitting its first data packet in frame f_i, the end-device will transmit a number $l \in \{1, 2, ..., l_{max}\}$ of data packets in a reserved slot of successive frames. The value of l_{max} is limited by the energy available in the ESD, i.e., $l_{max} = \left\lfloor \frac{e_i}{K} \right\rfloor$. Thus, the end-device makes the following transitions from state (e_i, d_i): *(i)* to states $(e_i - lK, 0)$ with probability $p_s \cdot p_l$ for $l \in \{1, 2, ..., l_{max} - 1\}$, or *(ii)* to state $(e_i - l_{max}K, 0)$ with probability $p_s \cdot \left(1 - \sum_{l=1}^{l_{max}-1} p_l\right)$ for $l \geq l_{max}$. Recall that p_l is the

probability that an end-device has a number $l \in \{1, 2, ..., L\}$ of data packets ready to transmit at the beginning of a DCR. Consequently, the transition probability from state (e_i, f_i) to state (e_j, f_j) with $f_i \in \left\{1, 2, ..., \left\lfloor \frac{N}{K} \right\rfloor\right\}$ can be formulated as in Equation (3.41).

$$p_{ij} = \begin{cases} (1 - p_s), & if\,(e_i \geq 2K)\,and\,(e_j = e_i - K)\,and\,(f_j = f_i + 1) \\ (1 - p_s), & if\,(K < e_i < 2K)\,and\,(e_j = e_i - K)\,and\,(f_j = 0) \\ 1, & if\,(e_i = K)\,and\,(e_j = 0)\,and\,(f_j = 0) \\ p_s \cdot p_l, & if\,(e_i > K)\,and\,(1 \leq l < l_{max})\,and\,(e_j = e_i - Kl)\,and\,(f_j = 0) \\ p_s \cdot \left(1 - \sum_{l=1}^{l_{max}-1} p_l\right), & if\,(e_i > K)\,and\,(l \geq l_{max})\,and\,(e_j = e_i - Kl_{max})\,and\,(f_j = 0) \\ 0, & otherwise \end{cases} \tag{3.41}$$

The probability p_s that an end-device succeeds in transmitting its first data packet in a given frame is derived in Section 3.6.1.3.2.

3.6.1.3.2 Probability of Success in one Frame

The probability that an end-device succeeds in transmitting its first data packet in a given frame, assuming that there are neither transmission errors nor capture effect, can be formulated as

$$p_s = \left(1 - \frac{1}{m_C}\right)^{n_C - 1}, \tag{3.42}$$

where m_C is the number of contention slots and n_C is the number of end-devices which contend in that frame. Since we consider that $m_C \simeq \rho \cdot n_C$, the expression of p_s can be formulated as

$$p_s = \rho \cdot \binom{n_C}{k} \left(\frac{1}{m_C}\right)^k \left(1 - \frac{1}{m_C}\right)^{n_C - k} \text{ with } k = 1. \tag{3.43}$$

As it can be observed, p_s is expressed as a binomial distribution with $k = 1$. Then, if n_C is large, the expression of p_s can be approximated by the Poisson distribution:

$$p_s \simeq \rho \cdot \frac{e^{-1/\rho}}{\rho^k k!} \text{ with } k = 1. \tag{3.44}$$

Consequently, the probability that an end-device succeeds in transmitting its first data packet in a given frame is constant, $p_s \simeq e^{-1/\rho}$, for all the frames of the contention process.

3.6.1.3.3 Energy Availability in Steady-state

As it can be observed in Figure 3.21, when $p_s > 0$, $q_0 > 0$ and $q_1 > 0$, the Markov chain is aperiodic and any state of the Markov chain can be reached from any other state with non-zero probability, and therefore the Markov chain is irreducible.

Since the Marchov chain is irreducible and aperiodic, and thus ergodic, it admits a unique steady-state probability distribution, denoted by $\pi = [\pi_{e,f}]$, which can be expressed as

$$\pi_{e,f} = \lim_{t\to\infty} \Pr\{e(t) = e, f(t) = f\}, \tag{3.45}$$

and satisfies that $(\mathrm{P}' - \mathrm{I})\pi' = 0$, where P is the transition matrix and I is the identity matrix. The steady-state probability distribution conditioned on being in sleep mode, denoted by $\pi^S = \left[\pi^S_{e,f}\right]$, can be obtained from π as

$$\pi^S_{e,f} = \begin{cases} \frac{\pi_{e,0}}{\sum_{i=0}^{N-1} \pi_{i,0}}, & if\,(f = 0) \\ 0, & if\left(1 \le f \le \left\lfloor \frac{N}{K} \right\rfloor\right) \end{cases}. \tag{3.46}$$

Since an end-device is always in sleep mode before a DCR starts, the steady-state probability distribution conditioned on being at the beginning of a DCR, denoted by $\pi^B = \left[\pi^B_{e,f}\right]$, can be expressed as $\pi^B = \pi^S \cdot \mathrm{P}$, where all the elements are zero for $f \in \left\{2, ..., \left\lfloor \frac{N}{K} \right\rfloor\right\}$ because when a DCR starts an end-device can only reach states $(e, 0)$ with $e \in \{0, 1, ..., \varepsilon_{th}\}$, or $(e, 1)$ with $e \in \{\varepsilon_{th} + 1, ..., N\}$.

3.6.1.3.4 Data Delivery Ratio

The *data delivery ratio* (DDR) can be formulated as

$$\mathrm{DDR} = \frac{\sum_{l=1}^{L} \mathbb{E}\left[N_d(l)\right] \cdot p_l}{\sum_{l=1}^{L} l \cdot p_l}, \tag{3.47}$$

where $\mathbb{E}\left[N_d(l)\right]$ is the average number of data packets successfully transmitted by an end-device in a DCR when it has l data packets ready to transmit at the beginning of

the DCR, which can be expressed as

$$\mathbb{E}\left[N_d(l)\right] = \sum_{f=1}^{\varepsilon_{thr}} \sum_{e=\varepsilon_{thr}+1}^{N} \pi_{e,1}^{B} \left(1-p_s\right)^{f-1} p_s \cdot \min\left(l, \left\lfloor \frac{e-(f-1)K}{K} \right\rfloor\right) + \sum_{f=\varepsilon_{thr}+1}^{\lfloor \frac{N}{K} \rfloor} \sum_{e=f}^{N} \pi_{e,1}^{B} \left(1-p_s\right)^{f-1} p_s \cdot \min\left(l, \left\lfloor \frac{e-(f-1)K}{K} \right\rfloor\right) . \tag{3.48}$$

Recall that an end-device which enters in active mode retransmits its first data packet in subsequent frames until it is successfully decoded by the coordinator. Then, the end-device transmits successfully a number l_s of data packets which depends on the number l of packets ready, the amount e of energy available at the beginning of the DCR, and the frame number f where the first packet succeeds, i.e., $l_s = \min\left(l, \left\lfloor \frac{e-(f-1)K}{K} \right\rfloor\right)$.

3.6.1.3.5 Probability of Empty, Successful and Collided Slot

The probability that k of n_C contending end-devices transmit their first data packet in the same slot of a frame with m_C contention slots can be formulated as

$$p_t(k) = \binom{n_C}{k} \left(\frac{1}{m_C}\right) \left(1 - \frac{1}{m_C}\right)^{n_C - k}. \tag{3.49}$$

If we consider a large number of contenders in every frame, we can resort to the standard Poisson approximation and obtain

$$p_t(k) \simeq \frac{e^{-\frac{1}{\rho}}}{\rho^k k!}, \tag{3.50}$$

and the probability that a given contention slot is empty, successful or collided can be calculated as $p_t(0) \simeq e^{-\frac{1}{\rho}}$, $p_t(1) \simeq \frac{e^{-\frac{1}{\rho}}}{\rho}$, and $1 - p_t(0) - p_t(1)$, respectively.

3.6.1.3.6 Time Efficiency

The *time efficiency*, denoted by η_t, can be formulated as the average number of slots with successful data packets in a DCR divided by the total number of slots in a DCR:

$$\eta_t = \frac{\sum_{f=1}^{\lfloor \frac{N}{K} \rfloor} \left(\mathbb{E}\left[m_{S,f}\right] + \mathbb{E}\left[m_{R,f}\right]\right)}{\sum_{f=1}^{\lfloor \frac{N}{K} \rfloor} \left(\mathbb{E}\left[m_{C,f}\right] + \mathbb{E}\left[m_{R,f}\right]\right)}, \tag{3.51}$$

where $\mathbb{E}[m_{C,f}]$ is the average number of contention slots in frame $f \in \{1, 2, ..., \lfloor \frac{N}{K} \rfloor\}$, $\mathbb{E}[m_{R,f}]$ is the average number of reserved slots in frame f, and $\mathbb{E}[m_{S,f}] = \mathbb{E}[m_{C,f}] \cdot p_t(1)$ is the average number successful slots in frame f, being $p_t(1)$ the probability that a contention slot of frame f contains 1 successful data packet. Recall that the capacity of the ESD limits the maximum number of frames in which an end-device can transmit in a DCR. If we consider that each end-device has a constant number $L_c > 1$ of packets in every DCR, the average number of contention and reserved slots in frame f can be formulated as

$$\mathbb{E}[m_{C,f}] = \sum_{e=\max(\varepsilon_{thr}+1,f)}^{N} n \cdot \pi_{e,1}^{B} \cdot (1-p_s)^{f-1} \tag{3.52}$$

$$\mathbb{E}[m_{R,f}] = \sum_{j=\max(1,f-L_c)}^{f-1} \sum_{e=\max(\varepsilon_{thr}+1,f)}^{N} n \cdot \pi_{e,1}^{B} \cdot (1-p_s)^{j-1} p_s \ . \tag{3.53}$$

3.6.1.4 Number of Contenders' Estimation Algorithm

In EH-RDFSA, the number of contention slots in frame i, denoted by $m_{C,i}$, is adjusted by the coordinator according to $m_{C,i} = \lceil \rho \cdot \hat{n}_i \rceil$, where ρ is a positive real number and $\hat{n}_i$ is the estimation of the number of end-devices that contend in frame i. In this section, we propose an algorithm to estimate the number of end-devices that contend in every frame of a DCR using EH-RDFSA.

The estimation algorithm works as follows. In the first frame of a DCR, the number $\hat{n}_1$ of end-devices that contend is equal to the average number of end-devices that become active, i.e., their energy level is above a predefined threshold, which can be expressed as

$$\hat{n}_1 = n \cdot p_{active}^{SS}, \tag{3.54}$$

where n is the total number of end-devices and p_{active}^{SS} is the activation probability in steady-state, which can be expressed as

$$p_{active}^{SS} = \sum_{e=\varepsilon_{thr}+1}^{N} \pi_{e,1}^{B}. \tag{3.55}$$

For the next frames, the estimation of the number of contenders in every frame $i+1$ is obtained as follows. First, the coordinator counts the number of slots with collision in

frame i, denoted by S_i^C, and estimates the number of end-devices that collide in frame i as $\hat{n}_{c,i} = \bar{n}_c \cdot S_i^C$, where $\bar{n}_c$ is the average number of end-devices that collide in one slot. The value of $\bar{n}_c$ can be calculated as

$$\bar{n}_c = \sum_{k=2}^{\infty} k \frac{p_t(k)}{1 - p_t(0) - p_t(1)}, \tag{3.56}$$

where $p_t(k)$ (3.49) is the probability that k end-devices transmit in the same slot of the frame, and $1-p_t(0)-p_t(1)$ is the probability that a slot is not empty and does not contain a successful data packet, i.e., it is a slot with collision. If we consider a large number of contenders in every frame, we can resort to the standard Poisson approximation of $p_t(k)$ (3.50) and obtain

$$\bar{n}_c \simeq \frac{1}{e^{\frac{1}{\rho}} - 1 - \frac{1}{\rho}} \cdot \sum_{k=2}^{\infty} \frac{1}{\rho^k (k-1)!}. \tag{3.57}$$

Finally, the coordinator estimates the number $\hat{n}_{i+1}$ of end-devices that collided in frame i and will have enough energy to contend in frame $i+1$ as $\hat{n}_{i+1} = \hat{n}_{c,i} \cdot p_a(i+1)$, where $p_a(i+1) = \Pr[e \geq (i+1)K | e \geq iK]$ is the probability that an end-device that contended in frame i will have enough energy to contend again in frame $i+1$, which can be expressed as

$$p_a(i+1) = \frac{\sum_{e=(i+1)K}^{N} \pi_{e,1}^B}{\sum_{e=iK}^{N} \pi_{e,1}^B}, \tag{3.58}$$

where e is the energy level in the ESD at the beginning of the DCR, N is the capacity of the ESD, and K is the number of energy units consumed by the end-device when it transmits in one frame.

To summarize, the estimation algorithm can be formulated as

$$\hat{n}_i = \begin{cases} n \cdot p_{active}^{SS}, & if \quad i = 1 \\ \bar{n}_c \cdot S_{i-1}^C \cdot p_a(i), & if \quad i > 1 \end{cases}. \tag{3.59}$$

3.6.1.5 Model Validation and Performance Evaluation

In this section, we evaluate the performance of EH-RDFSA, in terms of DDR and time efficiency, and compare with the performance of TDMA and EH-DFSA. In addition,

we validate the analytical model proposed in Section 3.6.1.3 by comparing the analytical results with computer-based simulations that execute the rules of the EH-RDFSA protocol without any simplification. While in the theoretical model of EH-RDFSA the steady-state probability distribution of the energy in the ESDs is calculated by analyzing the evolution of the energy of a single ESD, which is an approximation of the actual model, the simulation does not neglect the interactions among the ESDs of different end-devices.

In the following sections, we first describe the considered scenario, and then discuss the numerical results in order to study how the performance of the protocols is influenced by the energy harvesting rate $\overline{E}_H$ and the number data packets per end-device in a DCR. Results for TDMA and EH-DFSA are obtained through computer-based simulations. The results of 1000 simulation samples have been averaged for each test case.

3.6.1.5.1 Scenario

We consider a wireless network formed by 1 coordinator and $n = 1000$ end-devices equipped with an energy harvester and an ESD with $N = 10$ energy units of capacity. We consider that $\varepsilon_{thr} = 0$, $\rho = 1$, $K = 1$, and the energy harvested by an end-device follows a binomial distribution with probability mass function

$$q_j = \begin{pmatrix} N_H \\ j \end{pmatrix} \left(\frac{\overline{E}_H}{N_H}\right)^j \left(1 - \frac{\overline{E}_H}{N_H}\right)^{N_H - j}, \tag{3.60}$$

for $j \in \{0, 1, 2, ..., N_H\}$, where $N_H = 10$ is the maximum number of energy units that can be harvested, and $\overline{E}_H \in [0...N_H]$ is the energy harvesting rate.

3.6.1.5.2 Data Delivery Ratio

The DDR is represented in Figure 3.22 as a function of the energy harvesting rate $\overline{E}_H$ considering that each end-device has a uniformly distributed random number $l \in \{1, 2, ..., L\}$ of data packets ready to be transmitted to the coordinator in every DCR, with $L \in \{5, 10\}$.

As it could be expected, the DDR increases with $\overline{E}_H$ for EH-RDFSA, EH-DFSA, and TDMA. Indeed, the higher the energy available in an ESD at the beginning of a DCR, the higher the number of possible packet transmissions in the DCR and the DDR. TDMA shows a DDR equal to 1 when $\overline{E}_H$ (or average number of harvested energy units)

is close to or above L. Indeed, since there are no collisions in TDMA, its performance is only limited by the amount of harvested energy and the capacity of the ESD, and TDMA is the upper bound for random access protocols. In EH-RDFSA, the DDR is greater than 0.9 when $\overline{E}_H > L$. Therefore, the performance of EH-RDFSA in terms of DDR can be considered acceptable if $\overline{E}_H > L$. In addition, it can be observed that the slope and the maximum value of the DDR using EH-RDFSA are always higher than with EH-DFSA. Indeed, as in EH-RDFSA the end-devices only contend to transmit their first data packet, the collisions and the energy consumption due to retransmissions are reduced with respect to EH-DFSA, thus increasing the DDR. As it can be observed, EH-DFSA and EH-RDFSA perform similarly for low $\overline{E}_H$ because the end-devices have very low energy available, thus transmitting only in a very low number of frames.

The DDR is represented in Figure 3.23 as a function of the energy harvesting rate $\overline{E}_H$ considering that each end-device has a constant number $L_c \in \{5, 10\}$ of data packets ready to be transmitted to the coordinator in every DCR. Results show that the DDR decreases in EH-RDFSA, EH-DFSA and TDMA with respect to the case of uniformly distributed random number of data packets. This is due to the fact that each end device has to transmit more data packets in every DCR, which means more energy consumption and thus lower probability of activation.

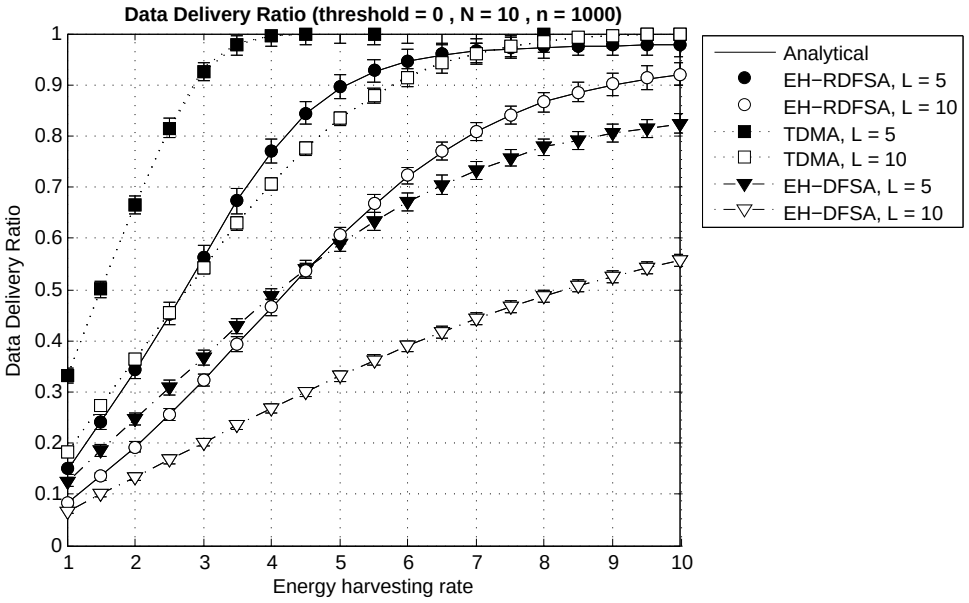


FIGURE 3.22: Data delivery ratio over the energy harvesting rate for EH-RDFSA, EH-DFSA and TDMA with a uniformly distributed random number of data packets.

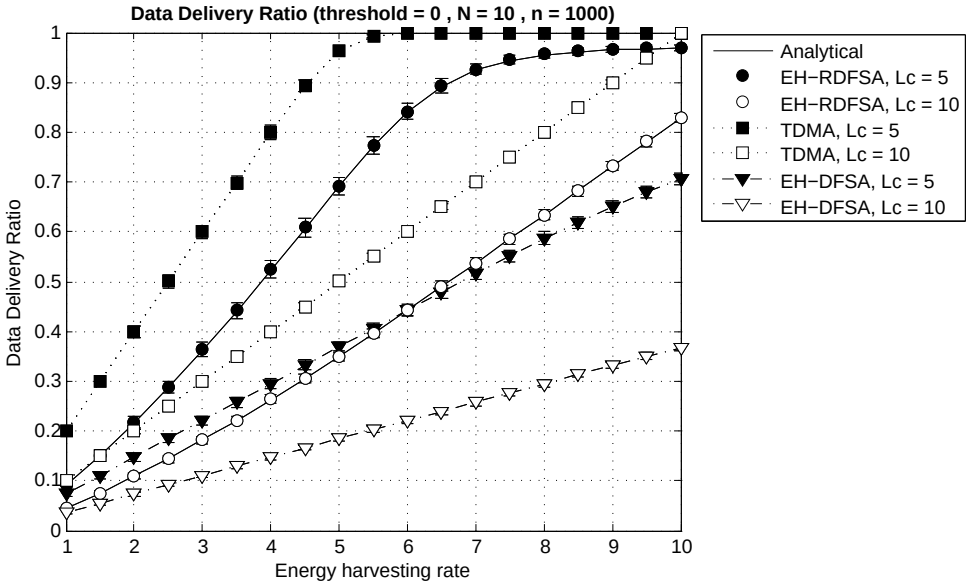


FIGURE 3.23: Data delivery ratio over the energy harvesting rate for EH-RDFSA, EH-DFSA and TDMA with a constant number of data packets.

3.6.1.5.3 Time Efficiency

The time efficiency is represented in Figure 3.24 as a function of the energy harvesting rate $\overline{E}_H$ considering that each end-device has a constant number $L_c \in \{4, 6, 8, 10\}$ of data packets ready to be transmitted to the coordinator in every DCR.

As it could be expected, the time efficiency using TDMA increases with $\overline{E}_H$. Indeed, the higher the energy harvesting rate, the higher the energy available in the ESDs at the beginning of a DCR, and the higher the number of possible packet transmissions per end-device. Consequently, the probability that one slot in every frame of TDMA contains one successful data packet increases with $\overline{E}_H$. Similarly, the time efficiency using EH-RDFSA also increases with $\overline{E}_H$. Indeed, the probability that one reserved slot in every frame of EH-RDFSA contains one successful data packet increases with the energy available in the ESDs at the beginning of a DCR.

As it can be observed, while the time efficiency using TDMA increases linearly up to 1, which indicates that every slot contains one successful data packet, the maximum time efficiency using EH-RDFSA is 0.8. Indeed, while in TDMA each end-device transmits in its reserved slot, in EH-RDFSA the end-devices have to contend until they succeed in transmitting their first data packet, with the consequent waste in contention slots due to collisions. In addition, the time efficiencies using TDMA and EH-RDFSA tend to their maximum when $\overline{E}_H > L_c$.

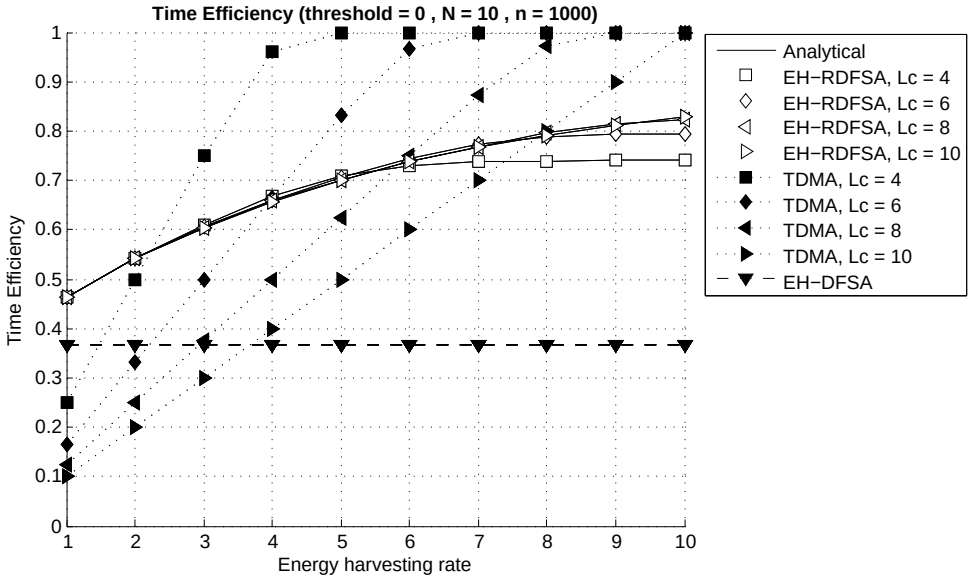


FIGURE 3.24: Time efficiency over the energy harvesting rate for EH-RDFSA, EH-DFSA and TDMA with a constant number of data packets.

As it could be expected, when the energy harvesting rate is below a certain threshold, the time efficiency using EH-RDFSA is greater than using TDMA. As it can be observed in Figure 3.24, EH-RDFSA outperforms TDMA in terms of time efficiency when $\overline{E}_H < L_c - 2$. Indeed, while the number of slots per frame in TDMA is constant, equal to the total number of end-devices in the network regardless of the number of active end-devices, the number of reserved slots per frame in EH-RDFSA is adjusted dynamically according to the number of end-devices that have not entered into sleep mode due to energy shortage, thus leading to higher time efficiency.

Finally, EH-DFSA shows the worst time efficiency with respect to EH-RDFSA and TDMA, and it is insensitive to $\overline{E}_H$. This is due to the fact that in EH-DFSA the end-devices have to contend for the transmission of every data packet, and the number of contention slots is adjusted to the number of end-devices which contend in each frame. However, when $\overline{E}_H$ is very low, even EH-DFSA outperforms TDMA in terms of time efficiency.

3.6.1.5.4 Frame Length Adaptation

In this section, we provide the analytical and simulation results that demonstrate the correctness of the algorithm proposed in Section 3.6.1.4 for the estimation of the number of contenders in every frame of EH-RDFSA.

We consider a wireless network formed by 1 coordinator and $n = 1000$ end-devices equipped with an energy harvester and an ESD with $N = 40$ energy units of capacity and the energy harvested by an end-device follows the binomial distribution with probability mass function of Equation 3.60 with $N_H = 40$ and $\overline{E}_H \in [5, 20]$. We consider that $\varepsilon_{thr} = 20$, $K = 4$ and $\rho \in \{0.75, 1, 1.25\}$. We consider that each end-device has a uniformly distributed random number $l \in [1, 2, \ldots, 10]$ of data packets ready to be transmitted to the coordinator in every DCR. The results of 1000 simulation samples have been averaged for each test case.

Figure 3.25 shows the evolution of the average number of contention slots and reserved slots in every frame of a DCR using EH-RDFSA. Recall that in EH-RDFSA the number of contention slots $m_{C,i}$ in frame i is adjusted according to $m_{C,i} = \lceil \rho \cdot \hat{n}_i \rceil$, where $\hat{n}_i$ is the estimation of the number of end-devices that contend in frame i. Results show that the analytical results are tightly matched with the simulation results, thus validating the correctness to the estimation algorithm.

As it can be observed, the higher the energy harvesting rate, the higher the average number of contention slots in the first frame. This is due to the fact that the probability of activation increases with higher energy harvesting rates, and thus more end-devices become active at the beginning of a DCR and contend to transmit their first data packet in the first frame.

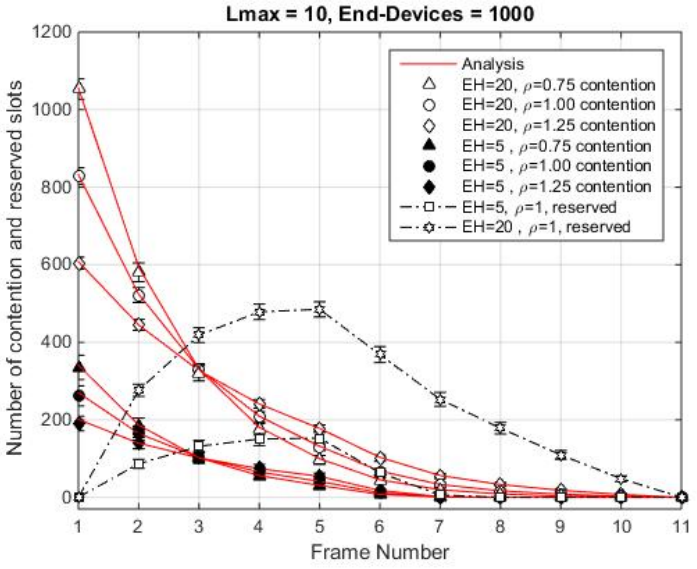


FIGURE 3.25: Evolution of the average number of contention slots and reserved slots in every frame of a DCR using EH-RDFSA.

As it could be expected, the average number of contention slots decreases almost exponentially as the number of frame increases, and it tends to 0 slots at frame 10. Indeed, a certain number of end-devices succeed in transmitting their first data packet in every frame and thus a new reserved slot is allocated to each end-device. Therefore, the number of contention slots decreases and the number of reserved slots increases until all the end-devices have succeeded. Finally, the number of reserved slots decreases down to 0 at the last frame because all the end-devices have either transmitted all their data packets or entered in energy shortage.

It is worth noting that with high ρ (e.g., 1.25) the average number of contention slots in the first frames (1 to 3) is larger than with low ρ, but then it becomes lower from frame 4. Indeed, the higher the value of ρ, the higher the number of contention slots and the lower the probability of collision, and thus the higher the number of end-devices that succeed in the first frames.

3.6.1.6 Conclusions

A new MAC protocol, named Energy Harvesting-aware Reservation Dynamic Frame Slotted-ALOHA (EH-RDFSA), has been proposed in this section. EH-RDFSA is suitable for wireless M2M networks where a gateway periodically collects long data messages from a group of end-devices equipped with energy harvesters.

The data delivery ratio (DDR) and the time efficiency have been considered as performance metrics. A discrete-time Markov chain model has been proposed to analyze the evolution of the energy available in an end-device and to compute analytically the DDR and the time efficiency. The performance of EH-RDFSA has been evaluated and compared with that of EH-DFSA and TDMA in order to demonstrate how they are affected by the amount of harvested energy.

Results show that the DDR and the time efficiency using TDMA and EH-RDFSA increase with the energy harvesting rate. EH-RDFSA outperforms DFSA in terms of DDR and time efficiency. TDMA outperforms EH-RDFSA and DFSA in terms of DDR. However, EH-RDFSA outperforms TDMA in terms of time efficiency when the energy harvesting rate decreases below a certain threshold which depends on the amount of data to transmit from each end-device to the gateway.

3.6.2 Energy Harvesting-Aware Low Power Distributed Queuing with Reservation (EH-DQ)

This section focuses on the design and analysis of the EH-DQ protocol proposed in this thesis and the performance evaluation of EH-DQ, EH-RDFSA and TDMA. The operation of EH-DQ is decribed in Section 3.6.2.1. The energy consumption model of the end-devices is introduced in Section 3.6.2.2. Section 3.6.2.3 proposes a discrete time Markov chain model to analyze the evolution of the energy available in an end-device using EH-DQ, and formulates the expressions to calculate the data delivery ratio and the time efficiency for EH-DQ. Section 3.6.2.4 is devoted to validate the analysis and to evaluate the performance of EH-DQ through comprehensive computer-based simulations. The comparison with the performance of EH-RDFSA and TDMA is also presented in this section.

3.6.2.1 MAC Protocol Description

In EH-DQ, the active end-devices request access to the channel in a short contention window at the beginning of each frame, thus confining collisions to a specific part of the frame. Collisions are resolved by using a tree-splitting algorithm [20] that organizes the end-devices into sub-groups to reduce the probability of collision per transmission attempt. When an end-device succeeds in transmitting its access request, it waits for its turn to transmit data in collision-free slots. Each frame of EH-DQ is divided in three parts as shown in Figure 3.26: *(i)* m contention slots devoted to the transmission of access request (ARS) packets, *(ii)* one collision-free slot for the transmission of a data packet, and *(iii)* a feedback packet (FBP). A guard time called Inter Frame Space (IFS) is left between reception and transmission modes to compensate propagation and processing delays and the time required to switch the radio transceivers between reception and transmission.

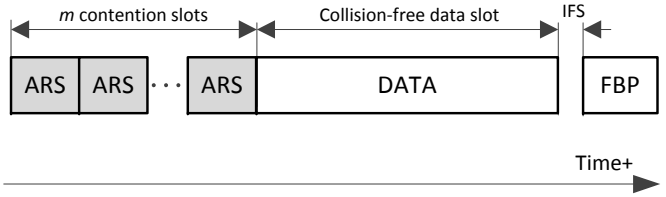


FIGURE 3.26: Frame structure of EH-DQ.

Every active end-device randomly selects one of the contention slots in every frame to transmit an ARS. Each ARS only contains one field (e.g., 1 byte) that indicates the number of data packets that must be transmitted by an end-device, i.e., the number l_R of collision-free slots to be reserved, which depends on the energy available in the end-device. Note that the ARS does not need to identify the end-device. Depending on whether the ARS collides or is successfully decoded by the coordinator, an end-device is queued into one of two logical and distributed queues:

1) The end-devices that have collided in a given contention slot when transmitting their ARS are queued into the *Collision Resolution Queue* (CRQ), sharing the same position in the queue. Note that after every frame, at most m new entries enter into the CRQ, being each one associated to each of the collisions that occurred in the last contention window. The length of the CRQ and the position of the end-devices in the CRQ is updated by executing the tree-splitting algorithm represented in Figure 3.27.a. Each node of the tree represents a frame of m contention slots (m=3 in the example), and the number in each contention slot denotes the number of end-devices that transmit an ARS in that slot. In every level of the tree, an end-device transmits its ARS in only one frame until it succeeds in one level or enters in energy shortage. The algorithm works as follows. At frame 1, all the active end-devices contend. If two or more end-devices collide in a slot, a new frame is assigned only to the end-devices that caused the collision in order to reattempt access, and they are queued into the CRQ. Therefore, if there are k slots with collision in one frame of level d, then k new frames are scheduled in level $d+1$, and k sub-groups of end-devices are queued into the CRQ. Once an end-device has entered in the CRQ, it will retransmit its ARS in a given frame only if it has enough energy in its ESD and it occupies the first position in the CRQ; otherwise, the end-device enters in sleep mode and waits until it reaches the first position of the CRQ.

2) The end-devices that succeed in transmitting their ARS are queued into the *Data Transmission Queue* (DTQ). In principle, any queue discipline could be used to this end. For example, devices could enter into the DTQ following the same chronological order of the contention slots. Contrarily to the CRQ, in this case, every position of the DTQ is occupied by just one end-device. Indeed, an end-device occupies a number of positions in the DTQ that is equal to the number l_R of collision-free data slots reserved by the end-device for this particular DCR. When an end-device reaches the first position of the DTQ, it transmits its data packets in the collision-free slot of successive frames.

The CRQ and DTQ are represented at each end-device by 2 integer numbers per queue representing: *1)* the position of the end-device in the queue, and *2)* the total length of the queue. The length of the CRQ represents the number of sub-groups of end-devices waiting to retransmit an ARS. The length of the DTQ represents the total number of collision-free slots reserved by the end-devices that have succeeded in transmitting their ARS and wait for their first collision-free slot.

The coordinator updates the length of the CRQ and DTQ at the end of each frame according to the following rules: *1)* the length of the CRQ is incremented by the number of contention slots with collision; *2)* if the CRQ was not empty in the previous frame, then its length is decremented by one; *3)* the length of the DTQ is incremented by the total number of collision-free slots reserved in each frame; and *4)* if the DTQ was not empty in the previous frame, then its length is decremented by one.

The coordinator broadcasts in every FBP: *(i)* the length of the CRQ (2 bytes); *(ii)* the length of the DTQ (2 bytes); and *(iii)* the state of the m contention slots (empty, success, or collision) and the number of collision-free slots reserved in every slot with one successful ARS (1 byte per contention slot). Using the information of the FBP, an end-device that transmitted an ARS can compute its position in the CRQ if it collides, or its position in the DTQ if it succeeds. The positions in the CRQ and DTQ are always decremented by one at the end of each frame. Therefore, the end-devices only receive the FBP in those frames where they transmit either an ARS or a data packet, and they enter into sleep mode in those frames where they do not transmit either ARS or data, in order to save energy.

Figure 3.27 shows an example of the operation of EH-DQ. The contents of the slots and the lengths of the CRQ and DTQ in every frame are shown in Figure 3.27.a. The contents of both queues are shown in Figure 3.27.b. At frame 1, all the end-devices (d1 to d6) transmit an ARS: d1, d2 and d3 collide in slot 1; d4 succeeds in slot 2; d5 and d6 collide in slot 3. Thus, d1, d2 and d3 enter in the first position of the CRQ; d4 enters in the first position of the DTQ reserving 1 collision-free slot; d5 and d6 enter in the second position of the CRQ. At frame 2, d4 transmits its data packet (because it occupies the first position of the DTQ), and d1, d2 and d3 transmit an ARS (because they occupy the first position of the CRQ): d1 and d2 collide and enter in the CRQ again; d3 succeeds and enters in the DTQ reserving 2 collision-free slots; d5 and d6 move to the first position of the CRQ. At frame 3, d5 and d6 transmit an ARS, collide,

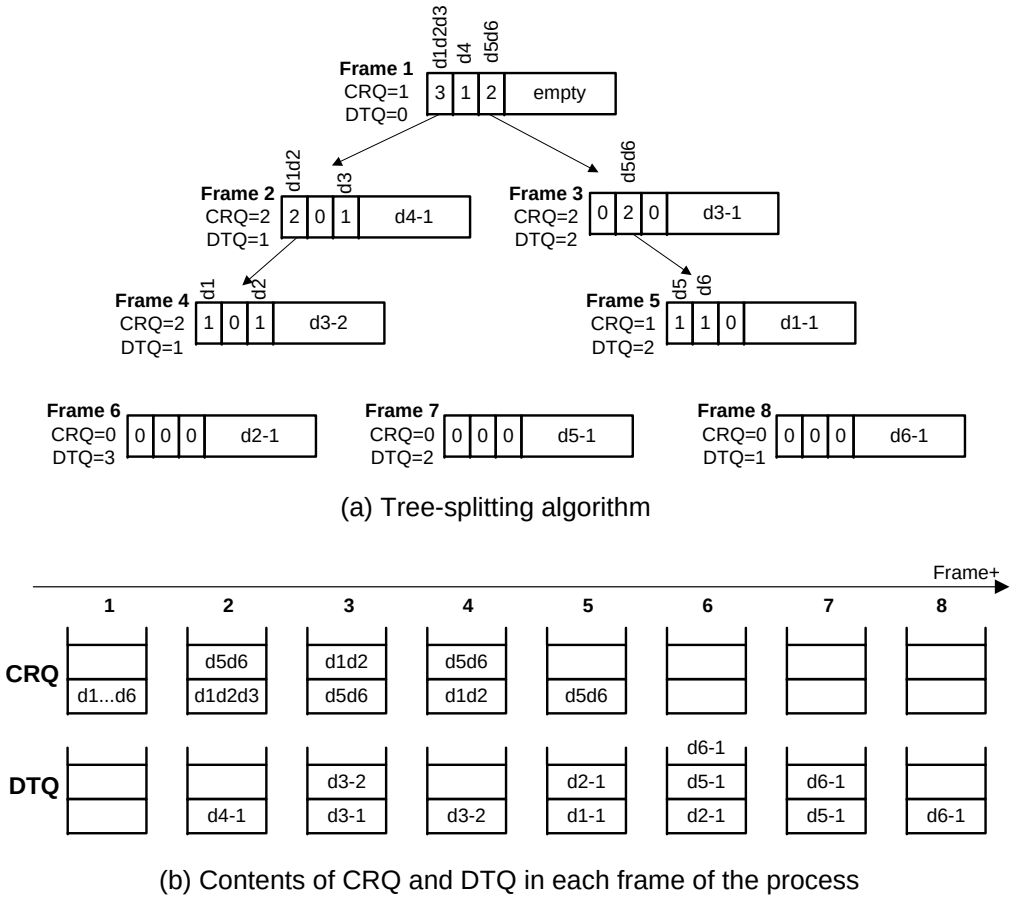


FIGURE 3.27: Example of EH-DQ with 6 end-devices (d1 to d6) and 3 contention slots: (a) tree-splitting algorithm, and (b) contents of the CRQ and DTQ in each frame.

and enter in the second position of the CRQ again; d1 and d2 move to the first position of the CRQ; and d3 transmits its first data packet. At frame 4, d3 transmits its second data packet; d1 and d2 transmit an ARS, succeed, and enter in the DTQ reserving 1 collision-free slot each; and d5 and d6 move to the first position of the CRQ. The process continues until the end of the DCR.

3.6.2.2 Energy Consumption Model

Every time that an end-device transmits an ARS in a certain frame of a DCR, it consumes a constant amount of energy, denoted by E_{ARS} [Joule], which accounts for the energy used in the following communication phases: *(i)* the end-device transmits the ARS in 1 contention slot, *(ii)* it remains in sleep mode in the other $m-1$ contention slots and in the collision-free slot, and *(iii)* it receives the FBP. Then, E_{ARS} can be

formulated as

$$E_{ARS} = \rho_{tx} T_{ARS} + (m-1)\,\rho_{sleep} T_{ARS} + \rho_{sleep} T_{data} + \rho_{rx} T_{FBP}, \tag{3.61}$$

where T_{ARS}, T_{data} and T_{FBP} are the duration of a contention slot, a collision-free slot, and the time of transmission of a FBP, respectively; and ρ_{tx}, ρ_{rx} and ρ_{sleep} are the power consumption in transmission, reception and sleep mode, respectively.

Every time that an end-device transmits one data packet in a certain frame of a DCR, it consumes a constant amount of energy, denoted by E_{data} [Joule], which accounts for the energy used in the following communication phases: *(i)* the end-device remains in sleep mode in m contention slots, *(ii)* it transmits data in the collision-free slot, and *(iii)* it receives the FBP. Then, E_{data} can be formulated as

$$E_{data} = m\rho_{sleep} T_{ARS} + \rho_{tx} T_{data} + \rho_{rx} T_{FBP}. \tag{3.62}$$

We assume that the energy consumed by an end-device in those frames where it is in sleep mode is negligible.

For convenience, we normalize these energy consumptions to $E_{ARS} = 1\delta$ and $E_{data} \approx K\delta$, where K is a positive integer number. Therefore, an end-device consumes 1 and K energy units δ when it transmits an ARS or a data packet in a frame, respectively.

3.6.2.3 Analysis of Performance Metrics

In order to derive an analytical model to compute the DDR and the time efficiency in steady-state for EH-DQ, we need to evaluate the steady-state probability distribution of the energy available in the ESDs at the beginning of a DCR, which depends on the energy harvesting process, the random slot selection in every frame, the tree splitting process, and the number of data packets transmitted by each end-device in previous DCRs. Given that the number of end-devices that contend in every frame depends on the energy available in the ESD at each end-device, deriving the exact steady-state probability distribution is not an easy task. However, if we adjust the value of the energy threshold ε_{th} (3.1) to guarantee that all the end-devices that become active (n_1) in a DCR will have enough energy to contend in a certain number of levels, assuming that the number of end-devices that fall in energy shortage in a DCR is negligible, we can

consider that the probability that an end-device succeeds in transmitting an ARS packet in one frame of any level of the contention tree basically depends on the value of n_1, the number of slots per frame, and the number of level. Consequently, we can evaluate the steady-state probability distribution of the energy available in the ESDs by analyzing the evolution of the energy of a single ESD, which is an approximation that neglects the interactions among the ESDs of different end-devices.

In this section, we derive an analytical model for EH-DQ to compute the DDR and the time efficiency in steady-state. To this end, in Section 3.6.2.3.1 we first propose a discrete-time Markov chain model to analyze the evolution of the energy available in the ESD of a given end-device. In Section 3.6.2.3.2, we derive the probability that an end-device succeeds in transmitting an ARS packet in one frame of a DCR. Finally, we derive in Section 3.6.2.3.3 the steady-state probability distribution of the energy available in the ESDs at the beginning of a DCR, we formulate the DDR in Section 3.6.2.3.4 and the time efficiency in Section 3.6.2.3.5.

3.6.2.3.1 Markov Chain Model

The evolution of the energy available in the ESD of a an end-device can be modeled with the discrete-time Markov chain shown in Figure 3.28. Each state in the chain is defined by $\{e(t), d(t)\}$, where $e(t) \in \{0, 1, ..., N\}$ is a stochastic process that represents the number of energy units δ available in the ESD at time t; and $d(t) \in \{0, 1, ..., N-K\}$ is a stochastic process that represents that either an end-device is in sleep mode when $d(t) = 0$, or the level number in the contention tree where an end-device transmits an ARS when $d(t) \in \{1, ..., N-K\}$. Recall that in every level of the tree, an end-device transmits an ARS in only one frame. Note that the state transitions in the Markov chain do not occur at fixed time intervals.

The Markov chain is characterized by a transition matrix $\mathrm{P} = [p_{ij}]$, where each element p_{ij} is the one-step transition probability defined as

$$p_{ij} = \Pr\{e(t+1) = e_j, d(t+1) = d_j | e(t) = e_i, d(t) = d_i\} \ . \tag{3.63}$$

An end-device that has successfully transmitted all its data packets or entered in energy shortage in a DCR, remains in sleep mode (i.e., in a state with $d_i = 0$) until the next DCR starts. At the beginning of a DCR, the number ε_H of energy units

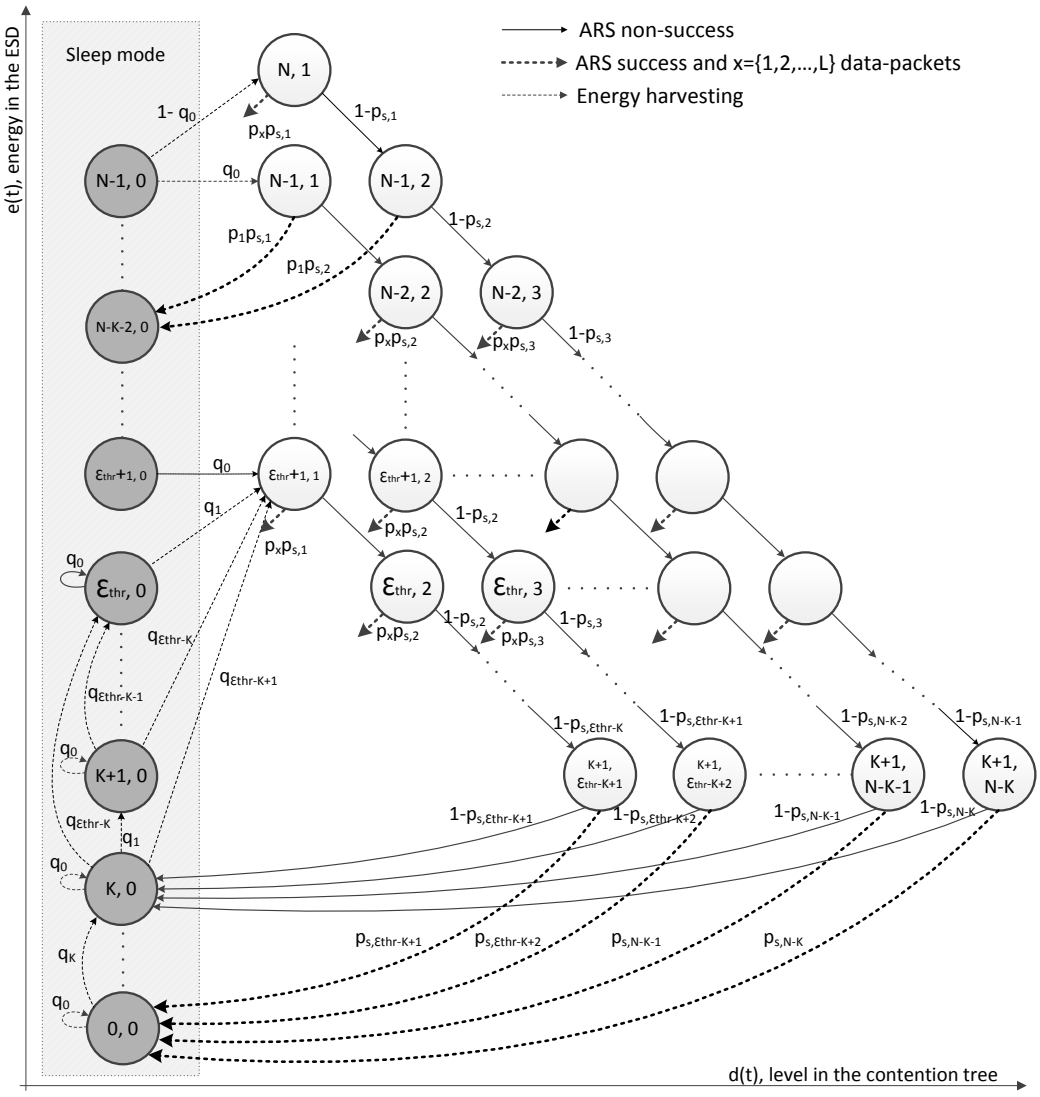


FIGURE 3.28: Generalized state transition diagram of the Markov chain that models the evolution of the energy available in an ESD using EH-DQ. Some state transitions have been omitted for ease of understanding.

harvested in the last T_R interval is added to the energy in the ESD, i.e., $e_j = e_i + \varepsilon_H$. Then, if the number of energy units available in the ESD is above the threshold ε_{th}, i.e., $e_j \in \{\varepsilon_{th}+1, ..., N\}$, the state of the end-device changes from sleep $(e_i, 0)$ to active mode $(e_j, 1)$. Otherwise, if the number e_j of energy units in the ESD is below or equal to ε_{th}, the end-device makes a transition from state $(e_i, 0)$ to state $(e_j, 0)$ and remains in sleep mode. The transition probability from state $(e_i, 0)$ to any state (e_j, d_j) at the

beginning of a DCR can be expressed as

$$p_{ij} = \begin{cases} q_{\varepsilon_H}, & if\,(e_i \leq e_j)\,and\,(e_j \leq \varepsilon_{th})\,and\,(d_j = 0) \\ q_{\varepsilon_H}, & if\,(e_i \leq e_j)\,and\,(\varepsilon_{th} < e_j < N)\,and\,(d_j = 1) \\ 1 - \sum_{k=0}^{N-1-e_i} q_k, & if\,(e_i < e_j)\,and\,(e_j = N)\,and\,(d_j = 1) \\ 0, & otherwise \end{cases}, \tag{3.64}$$

where q_{ε_H} is the probability that an end-device harvests a number ε_H of energy units, being $\varepsilon_H = e_j - e_i$ with $e_i \leq e_j$.

Once an end-device becomes active at the beginning of a DCR, it will transmit an ARS packet in one frame of every successive level of the contention tree until either it succeeds or its ESD falls below $(1+K)$ energy units. Recall that an end-device consumes 1 energy unit when transmits an ARS, and K energy units when transmits a data packet.

If the end-device does not succeed in transmitting an ARS in one frame of level $d_i \in \{1, ..., N-K\}$, it can make two possible transitions: *(i)* to state $(e_i - 1, d_i + 1)$, if the end-device has enough energy to retransmit an ARS in the next level and to transmit one or more data packets, i.e., $e_i \in \{2 + K, ..., N\}$; or *(ii)* to state $(K, 0)$, if the end-device has not enough energy to retransmit an ARS and a data packet, i.e., $e_i = 1 + K$.

Once an end-device succeeds in transmitting an ARS in one frame of level d_i, which happens with probability $p_{s,d}$ with $d = d_i$ (derived in Section 3.6.2.3.2), the end-device will transmit a number $l \in \{1, 2, ..., l_{max}\}$ of data packets in the collision-free slot of subsequent frames. The maximum value l_{max} is limited by the energy available in the ESD, i.e., $l_{max} = \left\lfloor \frac{e_i - 1}{K} \right\rfloor$. Thus, the end-device makes the following transitions from state (e_i, d_i): *(i)* to states $(e_i - 1 - lK, 0)$ with probability $p_{s,d} \cdot p_l$ for $l \in \{1, 2, ..., l_{max} - 1\}$, or *(ii)* to state $(e_i - 1 - l_{max}K, 0)$ with probability $p_{s,d} \cdot \left(1 - \sum_{l=1}^{l_{max}-1} p_l\right)$ for $l \in \{l_{max}, ..., L\}$, where p_l is the probability that an end-device has a number $l \in \{1, 2, ..., L\}$ of data packets ready to transmit at the beginning of a DCR. Consequently, the transition probability from state (e_i, d_i) to state (e_j, d_j) with $d_i \in \{1, 2, ..., N-K\}$ can be formulated as in

Equation (3.65).

$$p_{ij} = \begin{cases} (1-p_{s,d}), & if\,(e_i \geq 2+K)\,and\,(e_j = e_i - 1)\,and\,(d_j = d_i + 1) \\ (1-p_{s,d}), & if\,(e_i = 1+K)\,and\,(e_j = K)\,and\,(d_j = 0) \\ p_{s,d} \cdot p_l, & if\,(1 \leq l < l_{max})\,and\,(e_j = e_i - 1 - Kl)\,and\,(d_j = 0) \\ p_{s,d} \cdot \left(1 - \sum_{l=1}^{l_{max}-1} p_l\right), & if\,(l \geq l_{max})\,and\,(e_j = e_i - 1 - Kl_{max})\,and\,(d_j = 0) \\ 0, & otherwise \end{cases} \quad (3.65)$$

3.6.2.3.2 Probability of Success in one Frame

The probability that an end-device succeeds in transmitting an ARS packet in one frame of level $d \in \{1, 2, ..., N-K\}$, denoted by $p_{s,d}$, can be expressed as

$$p_{s,d} = \left(1 - \frac{1}{m}\right)^{n_d - 1}, \quad (3.66)$$

where m is the number slots per frame and n_d is the number of end-devices which contend in one frame of level d.

In the first frame of a steady-state DCR, i.e., in level $d = 1$, the number n_1 of end-devices that contend is equal to the average number of end-devices that become active, which can be expressed as $n_1 = n \cdot p_{active}^{SS}$, where n is the total number of end-devices and p_{active}^{SS} is the activation probability in steady-state, i.e., for large index k of DCR, defined as

$$p_{active}^{SS} = \lim_{k \to \infty} p_{active}(k). \quad (3.67)$$

We can assume that all the end-devices that become active in a steady-state DCR will have enough energy to contend until they succeed in transmitting an ARS. Note that this can be guaranteed by properly adjusting the value of the threshold ε_{th}. Under this assumption, the value of n_d for $d > 1$ can be derived as follows. First, the probability that k of n_d end-devices transmit in the same slot of a frame, denoted by $p_s(k)$, can be calculated as

$$p_s(k) = \binom{n_d}{k} \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{n_d - k}, \quad (3.68)$$

and the average number of empty, success, and collision slots in that frame can be calculated as $S_d^E = m \cdot p_s(0)$, $S_d^S = m \cdot p_s(1)$, and $S_d^C = m - S_d^E - S_d^S$, respectively.

As described in Section **??**, if there are S_d^C slots with collision in one frame of level d, then $F_{d+1} = S_d^C$ new frames are scheduled in level $d+1$, where each new frame in level $d+1$ is assigned only to the sub-group of end-devices that caused a collision in the same specific slot of level d. The average number of end-devices that succeed in one frame of level d, denoted by n_d^S, is equal to the average number S_d^S of slots with success. Therefore, the average number of end-devices that collide in one frame of level d, denoted by n_d^C, can be calculated as $n_d^C = n_d - S_d^S$. Since we assume that the n_d^C end-devices have enough energy, they will contend again in F_{d+1} new frames of level $d+1$. Then, the average number of end-devices that contend in one frame of level $d+1$ can be calculated as

$$n_{d+1} = \frac{n_d - S_d^S}{F_{d+1}} = \frac{n_d - n_d\left(1 - \frac{1}{m}\right)^{n_d - 1}}{m - m\left(1 - \frac{1}{m}\right)^{n_d} - n_d\left(1 - \frac{1}{m}\right)^{n_d - 1}}. \tag{3.69}$$

The probability that an end-device succeeds in transmitting an ARS in one frame of every level of the contention tree (3.66) is represented in Figure 3.29. It has been evaluated with $m \in \{3, 10\}$, $n \in \{10 \cdot m, 100 \cdot m\}$, $p_{active}^{SS} = 1$, and considering that all the end-devices that become active in a DCR have enough energy to contend until they succeed in transmitting their ARS packet in the DCR regardless of the energy harvesting rate and the capacity of the ESDs. Results show a tight match between analytical and simulated results. As it could be expected, the value of $p_{s,d}$ is close to 0 for low values of d, especially when the number m of slots is low and the number n of end-devices is high.

In order to set an appropriate value for the threshold ε_{th} which minimizes the probability that an end-device enters in energy shortage before it succeeds in transmitting an ARS, it is necessary to calculate the average number of frames where an end-device has to contend until it succeeds, denoted by $\mathbb{E}[d]$, which can be expressed as

$$\mathbb{E}[d] = \sum_{d=1}^{\infty} d \cdot p_{s,d} \cdot \prod_{i=1}^{d-1} (1 - p_{s,i}). \tag{3.70}$$

The value of $\mathbb{E}[d]$ is represented in Figure 3.30 as a function of the number of end-devices. It has been evaluated by considering $m \in \{5, 10, 20\}$. As it could be expected,

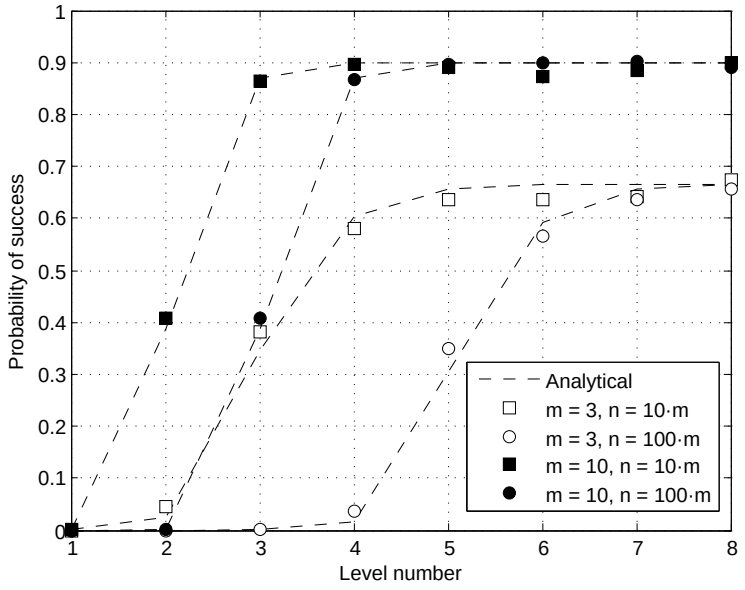


FIGURE 3.29: Probability that an end-device succeeds in transmitting an ARS in one frame of every level of the tree.

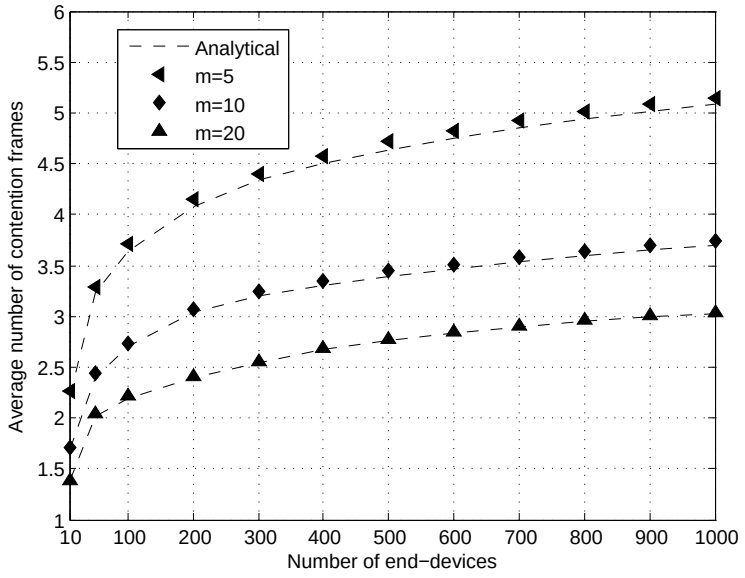


FIGURE 3.30: Average number of levels where an end-device contends until it succeeds in transmitting an ARS in a DCR.

the value of $\mathbb{E}[d]$ increases with n for a given value of m. The energy threshold needs to be adjusted as $\varepsilon_{th} \geq \mathbb{E}[d]$ depending on the values of m and n. For example, for a network of 1000 end-devices, the minimum energy threshold must be close to 5, 4, or 3 energy units when m is 5, 10, or 20 slots, respectively.

3.6.2.3.3 Energy Availability in Steady-state

As it can be observed in Figure 3.28, when $p_d > 0$ for $d \in \{1, 2, ..., N-K\}$, $q_0 > 0$

and $q_1 > 0$, the Markov chain is aperiodic and any state of the Markov chain can be reached from any other state with non-zero probability, and therefore the Markov chain is irreducible.

Since the Marchov chain is irreducible and aperiodic, and thus ergodic, it admits a unique steady-state probability distribution, denoted by $\pi = [\pi_{e,d}]$, which can be expressed as

$$\pi_{e,d} = \lim_{t\to\infty} \Pr\{e(t) = e, d(t) = d\}, \tag{3.71}$$

and satisfies that

$$(\mathrm{P}' - \mathrm{I})\pi' = 0, \tag{3.72}$$

where P is the transition matrix and I is the identity matrix. Equation (3.72) can be solved for π by calculating the eigenvector of P' that corresponds to an eigenvalue equal to 1. The steady-state probability distribution π is equal to the eigenvector with its elements normalized to sum one.

Recall that the transition matrix P depends on $p_{s,d}$ (3.66), which also depends on p^{SS}_{active} (3.67). On the other hand, p^{SS}_{active} can be expressed from the steady-state probability distribution of the energy available in the ESD at the beginning of a DCR, denoted by $\pi^B = \left[\pi^B_{e,d}\right]$, as follows

$$p^{SS}_{active} = \pi^B_{\varepsilon_{thr}+1,1} + ... + \pi^B_{N,1} = \sum_{e=\varepsilon_{thr}+1}^{N} \pi^B_{e,1}. \tag{3.73}$$

Note that all the values of π^B are zero for $d \in \{2, ..., N-K\}$. This is due to the fact that at the beginning of a DCR an end-device can only reach either states $(e, 0)$ with $e \in \{0, 1, ..., \varepsilon_{th}\}$ or $(e, 1)$ with $e \in \{\varepsilon_{th}+1, ..., N\}$. Since an end-device is in sleep mode before a DCR starts, π^B can be expressed as

$$\pi^B = \pi^S \mathrm{P}, \tag{3.74}$$

where $\pi^S = \left[\pi^S_{e,d}\right]$ is the steady-state probability distribution conditioned on being in sleep mode, which is calculated as

$$\pi^S_{e,d} = \begin{cases} \frac{\pi_{e,0}}{\sum_{i=0}^{N-1} \pi_{i,0}}, & if\,(d=0) \\ 0, & if\,(1 \leq d \leq N-K) \end{cases} . \tag{3.75}$$

Finally, we compute the steady-state probability distributions as follows. Firstly, we build the transition matrix P by setting the steady-state activation probability to a test value of 0, i.e., $p^{SS}_{active-test} = 0$. Secondly, we solve equations (3.72), (3.75), and (3.74) to calculate π, π^S, and π^B, respectively. Thirdly, we compute the analytical value of p^{SS}_{active} (3.73) by using π^B. And finally, we check the relative error between the test and analytical values of the activation probability. These steps are repeated iteratively by increasing $p^{SS}_{active-test}$ until the error is below 0.1%, which indicates that it satisfies (3.72), (3.75) and (3.74), and the results obtained for π, π^S, and π^B are correct.

3.6.2.3.4 Data Delivery Ratio

Once the steady-state probability distribution π^B of the energy available in the ESD at the beginning of a DCR is computed, we can formulate the expression to calculate the *data delivery ratio* in steady-state for EH-DQ as follows

$$\text{DDR} = \frac{\mathbb{E}\left[N_S\right]}{\mathbb{E}\left[N_R\right]} = \frac{\sum_{l=1}^{L} \mathbb{E}\left[N_d(l)\right] \cdot p_l}{\sum_{l=1}^{L} l \cdot p_l}, \tag{3.76}$$

where $\mathbb{E}\left[N_S\right]$ is the average number of data packets that are successfully transmitted to the coordinator in a DCR, $\mathbb{E}\left[N_R\right]$ is the average number of packets ready to be transmitted at the beginning of the DCR, and $\mathbb{E}\left[N_d(l)\right]$ is the average number of packets successfully transmitted by an end-device when it has $l \in \{1, 2, ..., L\}$ packets ready at the beginning of the DCR, which can be expressed as

$$\mathbb{E}\left[N_d(l)\right] = \sum_{d=1}^{N-K} \sum_{e=d+K}^{N} \pi^B_{e,1} \prod_{i=1}^{d-1} \left(1 - p_{s,i}\right) p_{s,d} \cdot \min\left(l, \left\lfloor \tfrac{e-d}{K} \right\rfloor\right) . \tag{3.77}$$

Recall that an end-device which enters in active mode retransmits an ARS packet in subsequent frames until it is successfully decoded by the coordinator. Then, the end-device transmits a number l_R of data packets which depends on the number l of

packets ready, the amount of energy e available at the beginning of the DCR, and the level number d where the ARS succeeds, i.e., $l_R = \min\left(l, \left\lfloor \frac{e-d}{K} \right\rfloor\right)$.

3.6.2.3.5 Time Efficiency

The *time efficiency* for EH-DQ, denoted by η_t, can be formulated as

$$\eta_t = \frac{\mathbb{E}\left[N_S\right] T_{data}}{\left(\mathbb{E}\left[N_E\right] + \mathbb{E}\left[N_S\right]\right)\left(mT_{ARS} + T_{data} + T_{FBP}\right)}, \tag{3.78}$$

where $\mathbb{E}\left[N_E\right]$ is the average number of frames with the collision-free slot empty, i.e., frames which do not contain data. Since every active end-device transmits a number $l_R \geq 1$ of data packets, we can assume that once an end-device has first succeeded in transmitting an ARS in a given frame, every frame until the end of the DCR contains data. Therefore, $\mathbb{E}\left[N_E\right]$ can be approximated as the average number of frames where an end-device contends until it succeeds, i.e., $\mathbb{E}\left[N_E\right] \approx \mathbb{E}\left[d\right]$ (3.70). Since $\mathbb{E}\left[d\right] \ll \mathbb{E}\left[N_S\right]$, the expression of η_t can be approximated as

$$\eta_t \simeq \frac{T_{data}}{mT_{ARS} + T_{data} + T_{FBP}}, \tag{3.79}$$

where recall that T_{ARS}, T_{data} and T_{FBP} are the duration of a contention slot, a collision-free slot, and the time of transmission of a FBP, respectively.

3.6.2.4 Model Validation and Performance Evaluation

In this section, we evaluate the performance of EH-DQ, in terms of the DDR and the time efficiency, and compare it with the performance of TDMA and EH-RDFSA. While in the theoretical model of EH-DQ the steady-state probability distribution of the energy in the ESDs is calculated by analyzing the evolution of the energy of a single ESD, which is an approximation of the actual model, the simulation does not neglect the interactions among the ESDs of different end-devices.

In the following sections, we first describe the considered scenario, and then discuss the numerical results to show how the performance is influenced by the energy threshold, the number of contention slots, and the energy harvesting rate.

3.6.2.4.1 Scenario

We consider a wireless network formed by 1 coordinator and $n = 1000$ end-devices.

Each end-device has a number of $L = 5$ data packets ready to be transmitted to the coordinator in every DCR. Each data packet has a payload of 114 bytes. At the end of each frame of EH-DQ, the coordinator broadcasts a FBP with a payload of 24 bytes that informs about the length of the CRQ and the DTQ, the status of the contention slots, and the number of collision-free slots reserved in every contention slot. All the packets are composed of a physical layer preamble, a MAC header, a payload and a cyclic redundancy code (CRC) of 2 bytes.

The system parameters used to evaluate the performance are summarized in Table 3.1. They have been selected according to the IEEE 802.15.4 standard [79] and the specifications of the CC2520 radio transceiver [80]. Being $\delta = 143$ μJoule (energy unit), an end-device consumes 1 energy unit δ to transmit an ARS packet in one frame of EH-DQ and $K = 4$ energy units to transmit a data packet in one frame of EH-DQ.

Each end-device includes an energy harvester and an ESD with $N = 40$ energy units δ of capacity. We assume that the energy harvested by an end-device in a DCR follows a binomial distribution with probability mass function

$$q_j = \begin{pmatrix} N_H \\ j \end{pmatrix} \left(\frac{\overline{E_H}}{N_H}\right)^j \left(1 - \frac{\overline{E_H}}{N_H}\right)^{N_H - j} \tag{3.80}$$

for $j \in \{0, 1, 2, ..., N_H\}$, where $N_H = 40$ is the maximum number of energy units that can be captured and $\overline{E_H} \in [0, ..., N_H]$ is the average number of energy units harvested per DCR, i.e., the energy harvesting rate.

Results for EH-DQ have been obtained analytically and through computer-based simulations using MATLAB. The results of 1000 simulation samples have been averaged for each test case. The tight match between analytical and simulation results validate the accuracy of the analytical model proposed in Section 3.6.2.3.

TABLE 3.1: System parameters for EH-DQ.

Parameter	**Value**	**Parameter**	**Value**
MAC header	8 bytes	Data-rate	250 kbps
Data payload	114 bytes	T_{data}	4.1 ms
ARS payload	1 byte	T_{ARS}	512 μs
FBP payload	24 bytes	T_{FBP}	1.2 ms
ρ_{tx}	100.8 mW	ρ_{stby}	525 μW
ρ_{rx}	66.9 mW	ρ_{sleep}	90 nW
E_{ARS}	1δ	E_{data}	4δ

In TDMA and EH-RDFSA, we consider that an end-device consumes $K = 4$ energy units δ when transmits a data packet in one frame. We consider an ideal EH-RDFSA where the number of contenders per frame is perfectly estimated and the number of contention slots is adjusted in every frame to be equal to the number of end-devices that contend in order to transmit their first data packet, i.e., $\rho = 1$. Results for TDMA and EH-RDFSA have been obtained through computer-based simulations.

3.6.2.4.2 Energy Threshold

The DDR and the time efficiency for EH-DQ, EH-RDFSA and TDMA are represented in Figure 3.31 and Figure 3.32, respectively, as a function of the energy threshold ε_{thr} by considering $\overline{E_H} \in \{10, 20\}$ and $m \in \{3, 10\}$, where m is the number of contention slots in one frame of EH-DQ. As it can be observed, the value of the DDR for EH-DQ, EH-RDFSA and TDMA increases with the energy harvesting rate. Indeed, the higher the number of energy units available in the ESDs, the higher the number of end-devices that become active in a DCR and the higher the number of possible packet transmissions.

Recall that an end-device becomes active at the beginning of a DCR if the energy available in its ESD is above ε_{thr} energy units, and in EH-DQ an end-device consumes 1 energy unit when it transmits an ARS, and $K = 4$ energy units when it transmits a data packet. As shown in Figure 3.30, the average number of frames in which an end-device has to contend until succeeds in transmitting an ARS, $\mathbb{E}[d]$ (3.70), is close to 5, 4, or 3 frames when $n = 1000$ and m is 5, 10, or 20 slots, respectively. Therefore, an end-device consumes an average of $\mathbb{E}[d]$ energy units in the transmission of ARS packets. Consequently, an end-device will need at least an energy level of $\varepsilon_{thr} \simeq \mathbb{E}[d] + KL$ in its ESD at the beginning of a DCR in order to maximize the DDR. As it can be observed in Figure 3.31, the optimum value of ε_{thr} that maximizes the DDR for EH-DQ is within 20-25 energy units when m is 3 or 10 slots, $n = 1000$ end-devices, and $L = 5$ data packets.

The DDR for EH-RDFSA increases with the energy threshold, but much more slightly than in EH-DQ. This is due to the fact that in EH-RDFSA, since we consider that the number of contention slots per frame is adjusted to be equal to the number of end-devices that contend in every frame, the probability that an end-device succeeds in a given frame of EH-RDFSA is approximately constant (≈ 0.36) for all the frames.

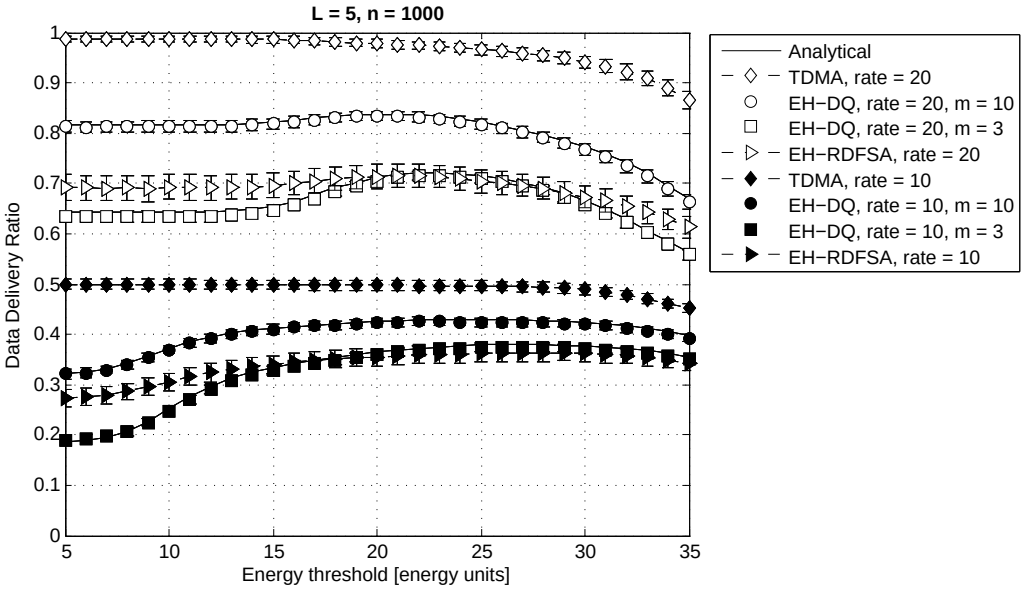


FIGURE 3.31: Data Delivery Ratio over the energy threshold using EH-DQ, EH-RDFSA and TDMA.

However, as it can be observed in Figure 3.29, in EH-DQ the probability that an end-device succeeds in transmitting an ARS is very low in the first levels of the contention tree and then it increases above 0.36 when the level number increases. For example, when $m = 10$ slots and $n = 10 \cdot m$ end-devices, the probability that and end-device succeeds in one frame of level 2 and 3 is 0.4 and 0.9, respectively.

While the DDR for EH-DQ and EH-RDFSA increases with the energy threshold until it reaches its maximum value, results show that the DDR for TDMA does not increase with the energy threshold. This is due to the fact that in TDMA there is no energy waste in collisions and thus the DDR for TDMA only depends on the energy harvesting rate.

Finally, as it can be observed in Figure 3.31, the DDR for EH-DQ, EH-RDFSA and TDMA decays dramatically when the energy threshold increases above a certain value. Indeed, when the energy threshold is too high, the activation probability decreases, thus reducing the DDR.

As it can be observed in Figure 3.32, the time efficiency for EH-DQ decreases as the number m of contention slots increases. This is due to the fact that once all the end-devices have succeeded in transmitting their ARS, the higher the number of contention slots per frame, the higher the overhead and the time wasted in the DCR. In addition,

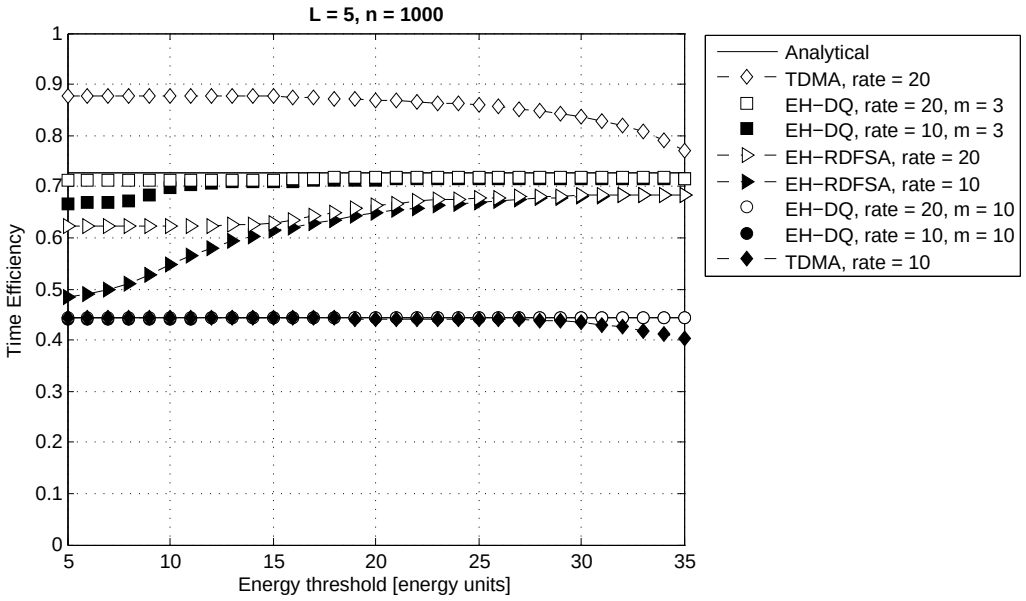


FIGURE 3.32: Time efficiency over the energy threshold using EH-DQ, EH-RDFSA and TDMA.

as it could be expected according to Equation (3.79), the time efficiency for EH-DQ is insensitive to the energy harvesting rate and the energy threshold.

The time efficiency for TDMA decreases as the energy threshold increases. This is due to the fact that the higher the energy threshold, the lower the number of end-devices that become active in a DCR, the higher the number of empty slots, and thus the lower the time efficiency. The time efficiency for TDMA increases as the energy harvesting rate increases. Indeed, the higher the energy harvesting rate, the higher the energy available in the ESDs at the beginning of a DCR, and the higher the number of possible packet transmissions per end-device. Consequently, the probability that one slot in every frame of TDMA contains one successful data packet increases with the energy harvesting rate. Similarly, the time efficiency using EH-RDFSA increases with the energy harvesting rate and the energy threshold. Indeed, the probability that one reserved slot in every frame of EH-RDFSA contains one successful data packet increases with the energy available in the ESDs at the beginning of a DCR.

3.6.2.4.3 Number of Contention Slots

The DDR and the time efficiency of EH-DQ are represented in Figure 3.33 and Figure 3.34, respectively, as a function of the number m of contention slots (from 2 to 20 slots) by considering $\overline{E_H} \in \{5, 10, 20, 30\}$ and $\varepsilon_{thr} = 20$, which is a value of the energy threshold close to the one that maximizes the DDR and the time efficiency for EH-DQ,

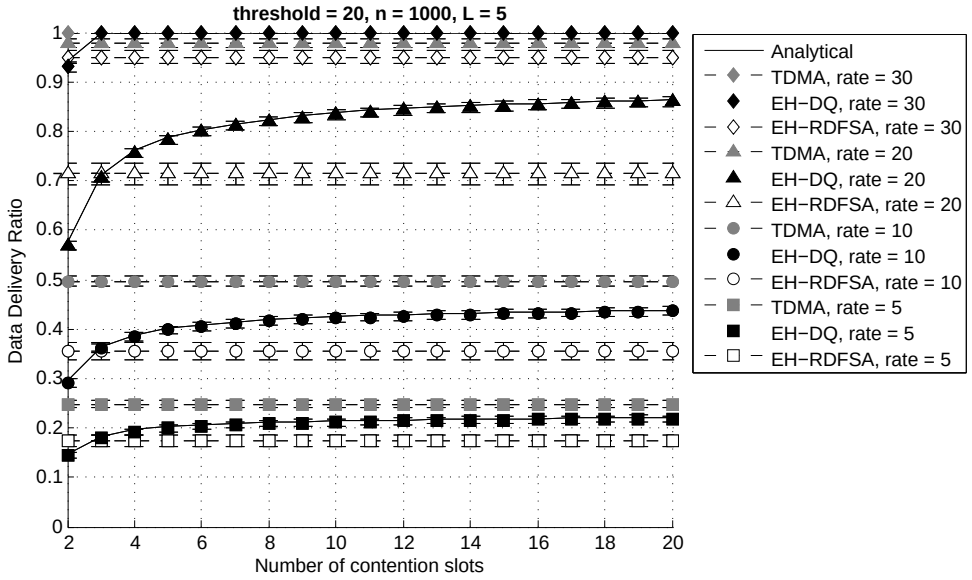


FIGURE 3.33: Data Delivery Ratio over the number of contention slots per frame.

EH-RDFSA and TDMA, as it can be observed in Figure 3.31 and Figure 3.32. Recall that in EH-RDFSA the number of contention slots per frame is adjusted to be equal to the number of contenders in every frame, and in TDMA the number of slots is equal to the total number of end-devices in the network.

Results show that the DDR for EH-DQ increases when the number of contention slots per frame increases. Indeed, the higher the number of contention slots, the lower the probability that an ARS collides in a given frame, and the lower the energy wasted in retransmissions, thus increasing the DDR.

The DDR for EH-DQ, EH-RDFSA and EH-DFSA increases with the energy harvesting rate. Indeed, the higher the number of energy units available in the ESDs, the higher the number of end-devices that become active in a DCR and the higher the number of possible packet retransmissions.

As it can be observed in Figure 3.33, EH-DQ can outperform EH-RDFSA in terms of DDR, for any energy harvesting rate, if the number of slots per frame in EH-DQ is properly adjusted. For example, if $n = 1000$ and $\overline{E_H} \in \{5, 10, 20, 30\}$, then the value of m in EH-DQ must be equal or greater than 3 slots.

As it can be observed in Figure 3.34, the time efficiency for EH-DQ is maximized for 2-3 contention slots, $\eta_t \approx 0.80$, and it is degraded as the number of contention slots per frame increases. Indeed, the higher the number of contention slots, the higher the time wasted in every frame once all the end-devices have succeeded in transmitting their

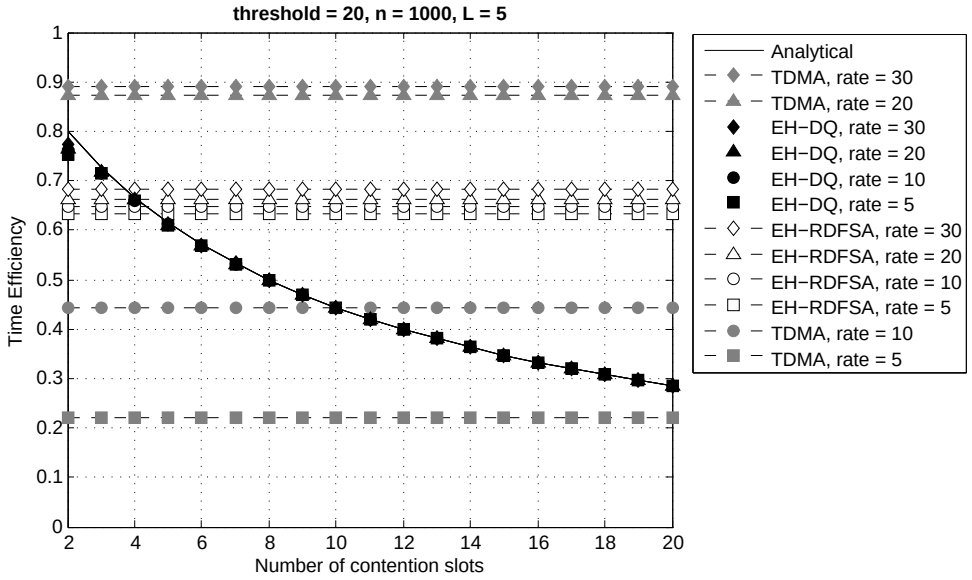


FIGURE 3.34: Time efficiency over the number of contention slots per frame.

ARS, thus reducing the time efficiency. In addition, EH-DQ can outperform EH-RDFSA and TDMA in terms of time efficiency if the number of contention slots is low, and the time efficiency for EH-DQ is very similar for different energy harvesting rates.

There is a trade-off between DDR and time efficiency for EH-DQ. When the number of contention slots per frame increases, more end-devices can eventually succeed in transmitting ARS packets to the coordinator in a DCR, thus increasing the DDR, at the cost of reducing the time efficiency and the data collection rate. However, as it can be observed in Figure 3.33, with low (e.g., 5) and high (e.g., 30) energy harvesting rates, EH-DQ can be configured with a very low number of contention slots (e.g., $m = 3$), at almost no cost in the DDR, and increase the time efficiency to a certain value close to the maximum. However, with intermediate energy harvesting rates (e.g., between 10 and 20 energy units), EH-DQ must be configured with a number of contention slots per frame which depends on the harvesting rate.

3.6.2.4.4 Energy Harvesting Rate

The DDR and the time efficiency for EH-DQ, EH-RDFSA and TDMA are represented in Figure 3.35 and Figure 3.36, respectively, as a function of the energy harvesting rate $\overline{E_H}$ by considering $m \in \{3, 10\}$ and $\varepsilon_{thr} = 20$, which is a value of the energy threshold close to the one that maximizes the DDR and the time efficiency for EH-DQ, EH-RDFSA and TDMA, as it can be observed in Figure 3.31 and Figure 3.32.

The value of the DDR increases almost linearly with $\overline{E_H}$ for EH-DQ, EH-RDFSA, and TDMA. Indeed, the higher the energy available in an ESD at the beginning of a DCR, the higher the number of possible packet transmissions and thus the value of the DDR. As it could be expected, TDMA yields a value of DDR equal to 1 when $\overline{E_H} \geq 4L = 20$ energy units. Indeed, since there are no collisions in TDMA, its performance is only limited by the amount of harvested energy and the capacity of the ESD. Note that ideal TDMA could be considered as the upper bound for random access protocols in terms of absence of collisions. In EH-RDFSA, however, the value of the DDR is close to 1 when $\overline{E_H} > 5L$. In its turn, EH-DQ attains a value of the DDR equal to 1 when $\overline{E_H} \geq 5L = 25$ energy units for $m = 10$, and when $\overline{E_H} \geq 6L = 30$ for $m = 3$. Indeed, as the probability of collision is lower when the number of contention slots increases, the energy consumption due to retransmissions of ARS packets is reduced, thus increasing the DDR. In addition, it can be observed that EH-DQ outperforms the DDR provided by EH-RDFSA. Indeed, as in EH-DQ the end-devices only contend to transmit short ARS packets, the collisions and the energy consumption due to retransmissions are reduced with respect to EH-RDFSA, thus increasing the DDR. Furthermore, the tree splitting algorithm of EH-DQ allows that the ARS packets can be eventually transmitted with a finite number of retransmissions. Results show that EH-DQ with $m = 10$ requires lower energy harvesting rate than EH-RDFSA to get the same DDR. For example, while EH-RDFSA requires $\overline{E_H} = 30$ energy units to obtain DDR $= 0.95$ with $L = 5$, EH-DQ requires $\overline{E_H} = 23$, which means a reduction of 23% in energy harvesting rate. Consequently, EH-DQ allows reducing the total time between consecutive DCRs and thus increases the network throughput with respect to EH-RDFSA.

As it can be observed in Figure 3.36, the time efficiency for TDMA increases with $\overline{E_H}$ and tends to its maximum value when $\overline{E_H} > 4L$. Indeed, the higher the energy harvesting rate, the higher the energy available in the ESDs at the beginning of a DCR, and the higher the number of possible packet transmissions per end-device. Consequently, the probability that one slot in every frame of TDMA contains one successful data packet increases with $\overline{E_H}$. In EH-RDFSA, the time efficiency increases slightly with $\overline{E_H}$. Contrarily, the time efficiency of EH-DQ is insensitive to the energy harvesting rate.

While the time efficiency in TDMA increases linearly up to 0.9, which indicates that every slot contains one successful data packet, the maximum time efficiency in

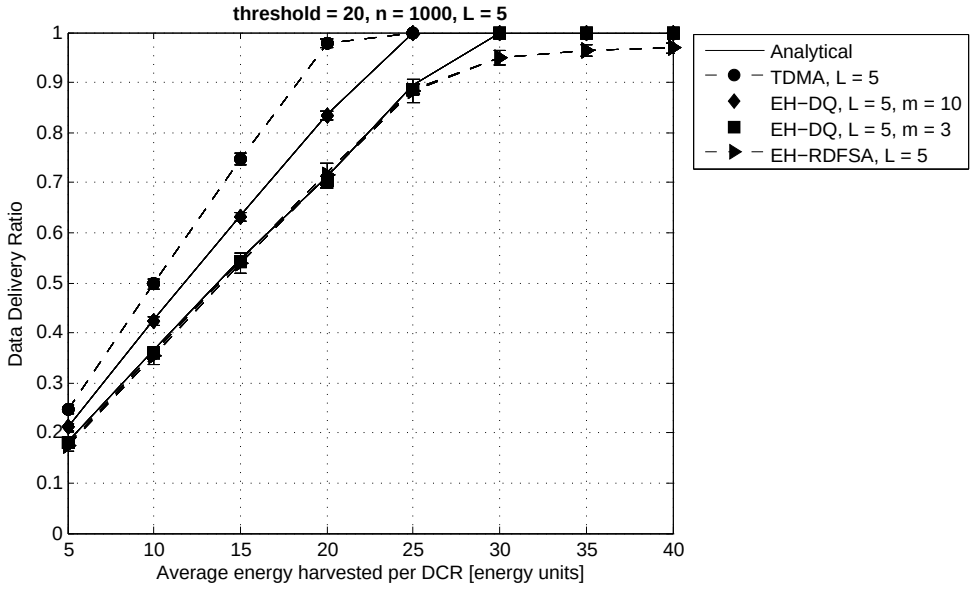


FIGURE 3.35: Data Delivery Ratio over the energy harvesting rate using EH-DQ, EH-RDFSA and TDMA.

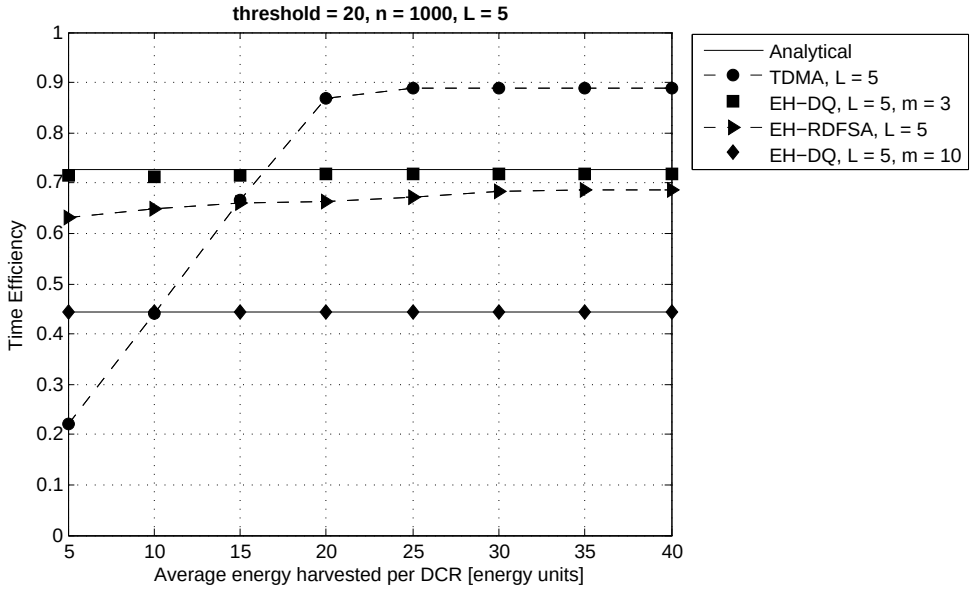


FIGURE 3.36: Time Efficiency over the energy harvesting rate using EH-DQ, EH-RDFSA and TDMA.

EH-RDFSA is 0.7, and it is 0.72 and 0.45 in EH-DQ with $m = 3$ and $m = 10$ number of contention slots, respectively. Indeed, while in TDMA each end-device transmits in its reserved slot, in EH-RDFSA and EH-DQ the end-devices have to contend until they succeed in transmitting their first data packet and the ARS, respectively, with the consequent waste of time in contention slots.

When the energy harvesting rate is below a certain threshold, the time efficiency in EH-DQ is greater than in TDMA. As it can be observed in Figure 3.36, EH-DQ

outperforms TDMA when $\overline{E_H} < 3L$. Indeed, while the number of slots per frame in TDMA is constant, equal to the total number of end-devices in the network regardless of the number of active end-devices, every frame in EH-DQ contains a very short contention window and 1 collision-free slot reserved for a specific end-device, thus leading to higher time efficiency.

3.6.2.5 Conclusions

A new MAC protocol, named Energy Harvesting-aware low power Distributed Queuing with reservation, has been proposed in this section. EH-DQ is suitable for data collection networks where each end-device is equipped with an energy harvester and generates messages which are fragmented into small data packets to be transmitted to a coordinator. In EH-DQ, the end-devices request access in a short contention window at the beginning of each frame and data transmissions are kept collision-free. EH-DQ uses a tree-splitting algorithm to resolve collisions and takes the energy availability into account.

A discrete-time Markov chain model has been presented to analyze the evolution of the energy available in an end-device using EH-DQ and to compute analytically two performance metrics: the DDR and the time efficiency. The performance of EH-DQ has been evaluated and compared with that of EH-RDFSA and the upper-bound of TDMA.

Results show that there is a trade-off between the DDR and the time efficiency for EH-DQ and how they are influenced by the number of contention slots per frame and the energy harvesting rate. The time efficiency for EH-DQ is maximized for a very low number of contention slots (e.g., $m = 3$). When the number of contention slots per frame increases, more end-devices can eventually succeed in transmitting an ARS to the coordinator in a DCR, thus increasing the DDR, at the cost of reducing the time efficiency and the data collection rate. However, with very low and very high energy harvesting rates, EH-DQ can be configured with a very low number of contention slots, at almost no cost in the DDR, and increase the time efficiency to a certain value close to the maximum. However, with intermediate energy harvesting rates, EH-DQ must be configured with a number of contention slots per frame which depends on the harvesting rate.

Results show that the DDR increases with the energy harvesting rate for EH-DQ, EH-RDFSA and TDMA. EH-DQ and TDMA provide the maximum DDR = 1, and both

outperform EH-RDFSA in terms of DDR and time efficiency. While the time efficiency for TDMA increases with the energy harvesting rate, the time efficiency for EH-DQ is insensitive to the harvesting rate. EH-DQ outperforms TDMA in terms of the time efficiency in a certain range of the energy harvesting rate which depends on the number of data packets to be transmitted by each end-device. Furthermore, while EH-RDFSA requires to estimate the number of contenders in each frame to adapt the frame length dynamically, EH-DQ uses a short and fixed number of contention slots per frame. In addition, while TDMA requires to update the knowledge of the network topology to maintain a collision-free schedule, EH-DQ does not require topology knowledge, thus reducing overhead and energy consumption. Taking that into account, EH-DQ is an interesting alternative for data collection scenarios with energy harvesting.

3.7 Chapter Conclusions

This chapter focuses on the integration of the MAC layer with the use of energy harvesters in data collection networks. While battery-powered networks can guarantee fully continuous operation of the end-devices within a limited lifetime, the use of energy harvesters can theoretically ensure unlimited lifetime, but with discontinuous operation. Due to the variability and random nature of the energy sources, the energy available may not be enough for the operation of the end-devices during certain periods of time. This fact is taken into account in this chapter for the design, analysis and performance evaluation of energy harvesting-aware MAC protocols by considering three key performance metrics: the probability of delivery, the data delivery ratio (DDR) and the time efficiency (or data collection rate).

Previous research works in networks with energy harvesting capabilities mainly focus on slotted-ALOHA and CSMA on homogeneous networks where each end-device generates packets following a random distribution. They evaluate the throughput and packet drop rate over the energy harvesting rate by considering that the harvested energy arrives in fixed amounts at random time intervals. The first works related to the MAC layer in data collection networks with energy harvesting focus on the analysis and performance evaluation of DFSA (or EH-DFSA). Results show how the probability of delivery increases with the energy harvesting rate. In this chapter, a comprehensive analysis and performance evaluation of EH-DFSA is presented in Section 3.5.1, which shows how the performance of EH-DFSA is affected by the number of end-devices, the capacity of the batteries and the minimum energy (or threshold) required by an end-device to become active and start transmitting data to the gateway in a data collection round. Results show that the probability of delivery decreases when the energy threshold is above a certain value, and the probability of delivery increases with the capacity of the batteries. In addition, since the frame length in EH-DFSA is dynamically adjusted to the number of contenders in every frame, and thus the average number of packet retransmissions of an end-device is almost insensitive to the number of end-devices, the performance of EH-DFSA is almost independent of the number of end-devices in the network. The complexity of the protocol resides on the algorithm to estimate the number of contenders, which must take the energy availability of the end-devices into account. Furhermore, since the frame length must be equal to the number of contenders, the synchronization

of the end-devices becomes difficult when the total number of end-devices in the network is huge.

Since the use of tree splitting-based access protocols in networks with energy harvesting capabilities has never received attention, the EH-CTA protocol is proposed in Section 3.5.2 as an adaptation of LP-CTA that takes the energy harvesting process into account. An analytical model is formulated to calculate the performance of EH-CTA in steady-state conditions. The probability of delivery and time efficiency of EH-CTA are evaluated and compared with that of EH-DFSA over the amount of harvested energy, the number of end-devices, the capacity of the batteries and the energy threshold. Results show that EH-CTA outperforms EH-DFSA if the frame length and the energy threshold are properly configured. There is a trade-off in EH-CTA; the time efficiency is maximized when the frame length is 3 slots, and it is degraded as the frame length increases. However, if the frame length is high, more end-devices can succeed in transmitting data to the gateway, and the probability of delivery increases. At very low and very high energy harvesting rates, a very short frame length can be used, at almost no cost in the probability of delivery, and get a time efficiency close to the maximum. However, at intermediate energy harvesting rates, the frame length must set depending on the maximum number of end-devices in the network. While EH-DFSA requires the estimation of the number of contenders in every frame to adapt the frame length dynamically, EH-CTA uses a fixed frame length. In addition, while the frame length in EH-DFSA must be as high as the number of contenders, the frame length in EH-CTA can be short, which enables scalability and facilitates synchronization in highly dense networks.

In data collection networks with energy harvesting capabilities where each each end-device transmits data fragmented into small packets it is better to adopt a reservation mechanism similar to the one proposed in RFSA or LPR-DQ. In this chapter, the EH-RDFSA and EH-DQ protocols are proposed in Section 3.6.1 and Section 3.6.2, respectively, and their performance is evaluated in terms of DDR and time efficiency. EH-DFSA is a combination of RFSA and DFSA, and EH-DQ is an extension of LPR-DQ. The performance of both protocols is evaluated and compared with that of EH-DFSA and TDMA. Results show that the DDR of EH-DQ, EH-RDFSA and TDMA increases with the energy harvesting rate. EH-DQ and TDMA provide the maximum DDR=1,

and both outperform EH-RDFSA in terms of DDR. In terms of time efficiency, in EH-RDFSA and TDMA it increases with the energy harvesting rate, but in EH-DQ the time efficiency is insensitive to the harvesting rate. While TDMA is the upper bound in terms of DDR, EH-RDFSA and EH-DQ provide better time efficiency, which yields higher data collection rate and higher energy efficiency at the gateway than in TDMA. Furthermore, while TDMA requires to maintain a collision-free schedule, EH-RDFSA and EH-DQ do not require any schedule, thus reducing overhead and energy consumption. And finally, while EH-RDFSA requires to estimate the number of contenders and adjust the frame length dynamically, EH-DQ uses a short and fixed frame length regardless of the number of contenders. Taking that into account, EH-DQ is an promising alternative for highly dense data collection networks with energy harvesting capabilities.

Chapter 4

Conclusions and Future Work

A summary of the main contributions and findings of this thesis is presented in this chapter. The main conclusions are presented in Section 4.1 and the open research lines are outlined in Section 4.2.

4.1 Conclusions

This thesis has focused on wireless M2M networks for data collection scenarios where a gateway gathers data from a huge and dynamic number of end-devices. The end-devices remain in sleep mode for certain periods of time in order to save energy, and wake up to transmit bursts of data to the gateway, periodically or triggered by an event. In these scenarios, the network changes abruptly from idle into saturation (which has also been referred to as delta traffic condition) when hundreds of end-devices wake up to transmit simultaneously. Consequently, two of the main challenges for the deployment of these M2M networks are: (i) to provide high scalability and capacity to handle hundreds or thousands of simultaneous end-devices, and (ii) to ensure very high energy efficiency in order to prolong the network lifetime and reduce the maintenance costs in battery recharge or replacement.

This thesis has contributed to resolve the abovementioned challenges by combining two complementary strategies. Firstly, in Chapter 2, the thesis has focused on the design, analysis and performance evaluation of MAC protocols which can manage abrupt transitions in the traffic load and minimize the energy consumption devoted to communications in star topology wireless networks. And secondly, the thesis has considered

end-devices equipped with energy harvesters to enable perpetual operation. Although energy harvesting can provide infinite lifetime, the high variability and unpredictability of the energy sources may not guarantee fully continuous operation and the end-devices may enter temporarily in energy shortage. This fact has been considered in the second part of the thesis, in Chapter 3, which has focused on the design, analysis and performance evaluation of energy harvesting-aware MAC protocols for data collection scenarios, and has proposed new MAC performance metrics that take the uncertainty on the available energy into account.

While reservation-based MAC protocols (e.g., TDMA) allow every end-device to transmit without collisions, thus theoretically minimizing energy consumption, they are not optimal for M2M networks with a large and dynamic number of end-devices due to the delay and energy required to update the knowledge of the network topology and to create, maintain and distribute the network schedule. Contrarily, the simplicity and distributed operation of random access protocols (e.g., ALOHA and its variants) make them ideal for simple and low-cost end-devices in M2M networks with an unknown and dynamic number of end-devices.

The Frame Slotted-ALOHA (FSA) protocol has been traditionally adopted to handle the delta traffic in data collection scenarios. In fact, it is used in the ISO/IEC 18000-7 standard for RFID. In FSA, time is divided into frames which are further divided into slots where the end-devices contend to transmit data; and in Dynamic FSA (FSA), the frame length is dynamically adjusted according to the number of contenders in every frame. Previous works in FSA and DFSA have mainly focused on the minimization of the delay, however, they did not analyze how the energy consumption is affected by the frame length and the number of end-devices, and did not determine the optimal configuration that minimizes the energy consumed by the gateway and the end-devices during the data collection process. This thesis has filled this gap with the following contribution. The analytical models to calculate the average delay and the average energy consumption under delta traffic have been formulated for two variants of FSA and DFSA, their performance has been evaluated over the frame length and the number of end-devices, and the optimal configuration of the protocols has been determined. Results show that there is a trade-off; if the frame length increases above the number of contenders, each end-device consumes less energy, but the delay and the energy consumption of the gateway increase. In FSA, there is a frame length which minimizes

delay and the energy consumed by the gateway. This optimum frame length is approximately equal to half the number of end-devices. Regarding the energy consumed per end-device using FSA, it tends to a minimum value when the frame length increases above the number of end-devices. In DFSA, the frame length has to be equal to the number of contenders in every frame to minimize delay and the energy consumed by the gateway; and the energy consumed per end-device tends to a minimum value when the frame length is proportional to the number of contenders. While DFSA outperforms FSA in terms of delay and energy consumption of the gateway, FSA outperforms DFSA in terms of energy consumed per end-device if an acknowledgement packet is sent by the gateway to every data packet successfully decoded.

In FSA, since the number of end-devices in the network is dynamic, the gateway must execute an estimation algorithm of the number of end-devices in order to adjust the frame length periodically. Indeed, the estimation of the number of contenders is performed at every frame in DFSA. In addition, since the number of end-devices can be very large (hundreds or thousands), the frame lengths need to be large, and this may cause synchronization problems as the number of slots per frame increases. The end-devices are time synchronized after decoding a feedback packet (or beacon) sent by the gateway at the end of every frame. The higher the precision of the synchronization, the smaller the guard times have to be in every slot, and thus the higher the time and energy efficiency of the MAC layer. However, due to the offset and drifts of cheap crystal oscillators, the time difference between the clocks of the end-devices increases as time advances along a frame. This deserves further research to improve the accuracy of the internal clocks in wireless transceivers so that very long frames can be executed.

An alternative to improve the performance of random access protocols is to use tree splitting algorithms which resolve collisions among end-devices by organizing the retransmission of colliding packets in such a way that they are always transmitted with finite delay. The group of contending end-devices is iteratively split into smaller sub-groups in order to reduce the probability of collision. Previous works in tree splitting-based access protocols for data collection (e.g., Query Tree, Binary Tree, or m-ary Tree in RFID) evaluated their performance in terms of delay and energy consumption. However, the analytical models are incomplete, because they neglected the energy consumed by the end-devices in reception and sleep modes, and did not determine the optimal configuration of the protocols to minimize delay and energy consumption. This thesis has filled

this gap with the following contribution. A new tree splitting-based random access protocol, named LP-CTA, has been proposed for data collection networks. In LP-CTA, time is organized in frames divided into m slots where the end-devices contend, and collisions are resolved using an m-ary tree splitting algorithm. In addition, in order to reduce the energy consumed by the end-devices in reception mode, the tree splitting algorithm has been implemented using a Collision Resolution Queue which allows the end-devices to sleep in complete successive frames until their turn to contend arrives. The analytical models to compute the average delay and the average energy consumption have been formulated for LP-CTA.

While LP-CTA uses data slots for contention, it is possible to separate contention from data transmission through the use of access requests in minislots where collisions are resolved with tree splitting. Since these minislots (or short contention slots) can be much shorter than the duration of a data packet, the performance of the network can be improved. This concept was the foundation of the DQ protocol, which combined an m-ary tree splitting algorithm with the logic of two distributed queues that manage the contention resolution and the collision-free data transmission, respectively. Previous research works in DQ considered homogeneous networks where each end-device generates packets following a random Poisson distribution and evaluated the performance of the protocol in terms of throughput in steady-state. However, none of the existing analyses has evaluated the performance of DQ under delta traffic. This thesis has filled this gap with the following contribution. The LP-DQ protocol has been proposed for data collection networks. An analytical model to compute the average energy consumption of the end-devices has been formulated, and the performance of LP-DQ has been evaluated and compared with respect to LP-CTA, FSA and DFSA. Similarly to FSA and DFSA, results show that in LP-CTA and LP-DQ there is a trade-off; as the number of contention slots increases, the end-devices consume less energy, but the delay and the energy consumption of the gateway increase. In LP-CTA and LP-DQ there exists a number of contention slots that minimizes delay and the energy consumed by the gateway. This optimum value is 3 contention slots regardless of the number of end-devices. Results show that LP-CTA and LP-DQ outperform DFSA and FSA in terms of delay and energy consumed by the gateway. In particular, LP-DQ provides delay reductions of up to a 60% and LP-CTA of up to a 40%; and LP-DQ provides energy savings at the gateway of up to 60% and LP-CTA of up to a 30%. Regarding the energy consumed

per end-device, it tends to a minimum value when 20 or more contention slots are used in LP-CTA, and 10 or more contention slots are used in LP-DQ. Results show that LP-DQ outperforms LP-CTA, DFSA and FSA. In particular, LP-DQ provides energy savings of more than a 27% with respect to LP-CTA, 13% with respect to DFSA and 45% with respect to FSA. However, if the frame length of FSA is over-dimensioned, it could theoretically provide lower energy consumption per end-device than LP-DQ.

The LP-CTA and LP-DQ protocols relax the need to know the size of the network and adapt smoothly to any change in the number of end-devices. In both protocols, the gateway does not need to estimate the number of contenders, and the frame length can be very short and fixed, thus facilitating the synchronization of the end-devices and the scalability of the network.

The approach of the FSA, DFSA, LP-CTA and LP-DQ protocols is convenient when the messages transmitted by each end-device fit in one single data slot. However, if the end-devices generate long data messages that have to be fragmented into a certain number of small packets, it is better to add a reservation mechanism in order to boost the performance of the protocols. In this sense, the Reservation FSA (RFSA) protocol was proposed in previous works as an extension of FSA for scenarios where the end-devices transmit long messages with fragmentation. In RFSA, when an end-device succeeds in transmitting the first packet of a message in a given slot, that slot is reserved in subsequent frames until the last packet of the message is sent. Previous works in RFSA considered networks where each end-device generates messages following a given random Poisson distribution, where each message is divided into a random number of packets, and evaluated the performance of the protocol in terms of throughput. However, none of the existing works has evaluated the performance of RFSA in the case of delta traffic. This thesis has filled this gap with the following contribution. An analytical model to calculate the average delay and the average energy consumption under delta traffic has been formulated for RFSA, considering that the number of data packets per message is a random variable that follows an exponential distribution. The performance of RFSA has been evaluated over the frame length and the number of end-devices, the optimal configuration of the protocol has been determined, and the performance of RFSA has been compared with respect to FSA. In RFSA, there is a frame length that minimizes delay and the energy consumed by the gateway. This optimum frame length is approximately equal to a fifth of the number of end-devices. Regarding the energy

consumed per end-device using RFSA, it tends to a minimum value when the frame length increases above half the number of end-devices. In FSA, the frame length has to be equal to half the number of end-devices to minimize delay and the energy consumed by the gateway; and the energy consumed per end-device tends to a minimum value when the frame length is above the number of end-devices. RFSA outperforms FSA in terms of delay and energy consumption. In particular, RFSA reduces the delay and the coordinator's energy consumption by more than 25% and the energy consumed per end-device by more than 50%. Besides the delay reduction and the energy savings, the frame length needed in RFSA is much shorter than in FSA, thus facilitating the synchronization of the end-devices in highly dense networks. Therefore, the use of reserved slots in successive frames can considerably improve the delay and energy performance of M2M networks for data collection.

The first part of the thesis has been finished with the design, analysis and performance evaluation of the LPR-DQ protocol. LPR-DQ has been proposed as an extension of the LP-DQ protocol for data collection networks where the end-devices transmit long messages with fragmentation. The concept of reservation has been integrated in LP-DQ to allow the end-devices reserve as many collision-free slots as they need in every data collection round. The analytical model to calculate the average energy consumption of the end-devices has been formulated, and the performance of LPR-DQ has been evaluated and compared with respect to RFSA and FSA. LPR-DQ outperforms RFSA and FSA in terms of delay and energy consumption. LPR-DQ provides delay reductions of a 40% with respect to RFSA and up to 75% with respect to FSA; in terms of the energy consumed by the gateway, it provides energy savings of a 40% with respect to RFSA and 70% with respect to FSA; and in terms of the energy consumed per end-device, it provides energy savings of more than a 52% with respect to RFSA and up to a 70% with respect to FSA.

The second part of the thesis has tackled the integration of the MAC layer with the use of energy harvesters in data collection networks. While traditional battery-powered systems guarantee fully continuous operation within a limited lifetime, energy harvesters theoretically provide unlimited lifetime, but with discontinuous operation. Due to the variability and random nature of the energy sources, the energy available may not be enough for the operation of the end-devices during certain periods of time. These fluctuations in the available energy have been taken into account in this thesis

for the assessment of energy harvesting-aware MAC protocols by considering three key performance metrics: the probability of delivery, the data delivery ratio (DDR) and the time efficiency (or data collection rate).

Previous research works related to the MAC layer on networks with energy harvesting capabilities focused on slotted-ALOHA and CSMA on homogeneous networks where each end-device generates packets following a given random distribution. They considered that the harvested energy arrives in fixed amounts at random time instants and evaluated the throughput and packet drop rate over the energy harvesting rate. The first research works on data collection networks focused on the analysis and performance evaluation of DFSA (or EH-DFSA) in terms of the probability of delivery and time efficiency. Results showed how the probability of delivery increases with the energy harvesting rate. However, these works did not analyze how the performance of EH-DFSA is affected by the number of end-devices, the capacity of the batteries and the minimum energy (or threshold) required by an end-device to become active and transmit data to the gateway. This thesis has filled this gap with a comprehensive analysis and performance evaluation of EH-DFSA. Results show that the probability of delivery decreases when the energy threshold is above a certain value, and it increases with the capacity of the batteries. Furthermore, the performance of EH-DFSA is independent of the number of end-devices in the network. This is due to the fact that the frame length is dynamically adjusted to the number of contenders in every frame, and thus the average number of packet retransmissions of an end-device to get access is almost insensitive to the number of end-devices. The complexity of EH-DFSA resides on the algorithm to estimate the number of contenders, which must take the energy availability of the end-devices into account, and also on the synchronization of the network when the number of contenders is huge. The results are useful to define practical design guidelines such as the capacity of the batteries (or super-capacitors) and the energy harvester rate, according to the expected network density and the data collection efficiency needed.

While the performance of random access protocols based on tree splitting was analyzed in previous works, and the LP-CTA protocol has been proposed for data collection networks in the first part of this thesis, the use of tree splitting-based access protocols for networks with energy harvesting capabilities has never received attention. This thesis has filled this gap with the following contribution. The EH-CTA protocol has been proposed as an adaptation of LP-CTA that takes the energy harvesting process into

account. An analytical model to compute the probability of delivery and the time efficiency in steady-state conditions has been formulated for EH-CTA. The performance of EH-CTA has been evaluated, and compared with that of EH-DFSA over the amount of harvested energy, the number of end-devices, the capacity of the batteries and the energy threshold. Results show that there is a trade-off in EH-CTA; the time efficiency is maximized when the frame length is 3 slots, and it is degraded as the frame length increases. However, if the frame length is high, more end-devices can succeed in transmitting data to the gateway, and the probability of delivery increases. At very low and very high energy harvesting rates, a very short frame length can be used, at almost no cost in the probability of delivery, and get a time efficiency close to the maximum. However, at intermediate energy harvesting rates, the frame length must set depending on the maximum number of end-devices in the network. EH-CTA outperforms EH-DFSA if the frame length and the energy threshold for an end-device to become active are properly configured. In addition, while EH-DFSA needs the estimation of the number of contenders in every frame in order to adapt the frame length dynamically, EH-CTA uses a fixed frame length. Furthermore, while the frame length of EH-CTA can be short, the frame length of EH-DFSA must be as high as the number of contenders, thus causing scalability and synchronization problems as the number of end-devices increases. Consequently, the EH-CTA protocol is a promising alternative for data collection networks with energy harvesting capabilities.

The EH-DFSA and EH-CTA protocols have been conceived for data collection networks with energy harvesting capabilities where the data transmitted by each end-device fits in one single data slot. However, they are not the best approach for scenarios where the data messages must be fragmented into smaller packets. In these scenarios, it is better to adopt a reservation mechanism like the one used in RFSA or LPR-DQ. Since there are not EH-aware random access protocols which integrate any reservation mechanism, this thesis has filled this gap with the following contribution. The EH-RDFSA and EH-DQ protocols have been proposed and their performance has been evaluated in terms of DDR and time efficiency. EH-RDFSA is a combination of RFSA and DFSA, and EH-DQ is an extension of LPR-DQ. The performance of both protocols has been evaluated and compared with that of EH-DFSA and TDMA. Results show that the DDR using EH-DQ, EH-RDFSA and TDMA increases with the energy harvesting rate. EH-DQ and TDMA provide the maximum DDR=1, and both outperform EH-RDFSA

in terms of DDR. While the time efficiency using EH-RDFSA and TDMA increases with the energy harvesting rate, the time efficiency for EH-DQ is insensitive to the harvesting rate. EH-RDFSA outperforms TDMA in terms of time efficiency when the energy harvesting rate decreases below a certain threshold which depends on the number of packets transmitted by each end-device to the gateway. EH-DQ outperforms TDMA and EH-RDFSA in terms of time efficiency in a certain range of the energy harvesting rate which depends on the number of packets transmitted by each end-device. While TDMA is the upper bound in terms of DDR, the EH-RDFSA and EH-DQ protocols provide better time efficiency, which yields higher data collection rate and higher energy efficiency at the gateway. In EH-RDFSA, this is due to the fact that the frame length is adapted dynamically depending on the number of active end-devices; and in EH-DQ, it is thanks to the tree splitting algorithm used to resolve the contention of short access requests, which are also separated from data transmission. In addition, while TDMA requires to maintain and distribute a collision-free schedule to all the end-devices in the network, the EH-RDFSA and EH-DQ protocols do not require any topology knowledge to build schedules, thus reducing overhead and energy consumption. Finally, while EH-RDFSA requires to estimate the number of contenders, EH-DQ uses a short and fixed frame length regardless of the number of contenders. Taking that into account, EH-DQ is an interesting alternative for highly dense data collection networks with energy harvesting where each end-device generates bursts of data fragmented in multiple packets.

4.2 Future Work

This thesis has addressed a number of significant issues in the design, analysis and performance evaluation of MAC protocols for wireless M2M networks with energy harvesting capabilities in data collection scenarios. Many other open topics have also been identified through the course of this thesis that have not been covered. Some open research lines are listed below to suggest the future work needed to extend the scope of the research presented in this thesis.

1) The delay and energy models of all the MAC protocols proposed and analyzed in this thesis have been formulated by considering that all packets are always transmitted without transmission errors induced by the wireless channel, and also assuming that if two or more packets collide, none of the packets can be decoded by the gateway, i.e.,

there is no capture effect. Since the number of packet retransmissions can be affected by transmission errors and the capture effect, they may have an impact on the delay and energy consumption of the network. Therefore, the theoretical models to compute delay, energy consumption, probability of delivery, DDR and time efficiency could be extended by including packet error probabilities and integrating an interference model which accounts for communication errors due to interferences arising from collisions among transmitting end-devices. The performance of the protocols could be evaluated and compared to study how they are affected by transmission errors and capture effect.

2) One of the main drawbacks of the FSA, DFSA, RFSA, EH-DFSA and EH-RDFSA protocols is that their frame lengths (i.e., number of slots per frame) must be long enough in order to guarantee good performance in terms of delay, energy consumption, probability of delivery, DDR and time efficiency. The time difference between the clocks of the end-devices increases as the slot number increases along a frame due to the inaccuracies of the clocks, and thus a guard time is needed in every slot in order to ensure that all the end-devices are well synchronized. Therefore, the longer the frame length, the longer the guard times needed. In this thesis, the duration of the time slots has been considered constant regardless of the frame length. Therefore, it could be interesting to evaluate the impact of the guard times on the protocols performance, especially in highly dense data collection networks.

3) One of the assumptions considered in this thesis for the analysis and performance evaluation of EH-aware MAC protocols, is that if an end-device fails to transmit one or more data packets due to energy shortage in a data collection round (DCR), those data packets are discarded and they are not transmitted in subsequent DCRs. This assumption is valid for scenarios where data is outdated in successive DCRs or where end-devices have memory constraints. As future work, the Markov chain models of EH-DFSA, EH-RDFSA, EH-CTA and EH-DQ could be extended by considering that non-transmitted packets are queued in a buffer of the end-devices and wait to be transmitted in future DCRs.

4) In the system model presented in this thesis, the total number of end-devices that have data ready to transmit to the gateway in a DCR has been considered constant and equal to the total number of end-devices in the network. Indeed, although the number of active end-devices in the case of networks with energy harvesting capabilities can

differ depending on the energy availability at the beginning of the DCR, all the end-devices have data ready in every DCR. Therefore, the system model could be extended by considering that the number of end-devices with data ready in a DCR is a random variable that follows a certain probability distribution depending on the statistics of the information.

5) In the analysis and performance evaluation of the EH-aware MAC protocols, the energy harvested by an end-device during the time period between two consecutive DCRs has been modeled with an arbitrary probability distribution. The energy models could be more sophisticated and closer to the features of real energy sources. To this end, a test campaign could be carried out in order to study the statistics of real energy sources and energy harvesters. These energy harvesting models could be plugged into the Markov chains, defining the transition probabilities from every state in sleep mode at the beginning of a DCR.

6) The MAC protocols presented in this thesis have been designed for wireless M2M networks in terrestrial data collection scenarios. All the work presented in this thesis could be extrapolated and applied to M2M satellite networks where the satellite operates as gateway and the end-devices are the earth terminals. One of the main issues to consider in this extension is the long round-trip time (RTT) between the satellite and the end-devices, which compromises the use of immediate feedback, and thus, the frame structure of the MAC protocols proposed in this thesis should be modified taking the long RTT into account. Consequently, the analytical delay and energy models formulated should be updated accordingly.

7) Finally, the actual implementation of the MAC protocols in real hardware is a very interesting work ahead. In fact, the FSA and LP-DQ protocols have been implemented and evaluated on 70 OpenMote devices operating in the 433MHz band. The experimental results have been published in a conference paper [84] that obtained the BEST DEMO RUNNER-UP AWARD at the IEEE INFOCOM 2014. The experimental evaluation of both protocols corroborated the delay reductions and energy savings provided by LP-DQ with respect to FSA in data collection networks in star topology. Future work could focus on the implementation of TDMA, EH-RDFSA and EH-DQ in wireless motes equipped with energy harvesters. The experiments could evaluate and compare the performance of the protocols in terms of DDR and time efficiency.

This thesis has contributed to solve some of the main challenges that arise in the deployment of highly dense M2M networks for data collection. The existing MAC protocols traditionally used in data collection scenarios have been comprehensively analyzed and evaluated, and novel MAC schemes based on tree splitting and Distributed Queuing have been proposed. The LP-DQ and LPR-DQ protocols show the greatest results in terms of delay and energy consumption under delta traffic, providing high scalability and capacity to handle hundreds of simultaneous devices, and they have been successfully extended to the EH-DQ protocol for its application on networks with energy harvesting capabilities. The use of the EH-DQ protocol will facilitate the deployment of future wireless M2M networks that are fully autonomous and it will contribute to reduce drastically the maintenance costs.

Bibliography

[1] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, October 2010. doi: 10.1016/j.comnet.2010.05.010. URL http://dx.doi.org/10.1016/j.comnet.2010.05.010.

[2] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi. From today's INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view. *IEEE Wireless Communications*, 17(6):44–51, December 2010. ISSN 1536-1284. doi: 10.1109/MWC.2010.5675777.

[3] NGMN Alliance, 5G White Paper, February 2015. URL https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf.

[4] *Technical Specification, Machine-to-Machine communications (M2M); Functional Architecture. TS 102 690, v1.1.1.* ETSI, October 2011.

[5] *The need for low cost, high reach, wide area connectivity for the Internet of Things.* Machina Research report, October 2014.

[6] SigFox. URL http://www.sigfox.com/.

[7] Weightless. URL http://www.weightless.org/.

[8] LoRa Alliance. URL https://www.lora-alliance.org/.

[9] *Low Throughput Networks (LTN): Protocols and Interfaces. ETSI GS LTN 003, Group Specification, v1.1.1.* ETSI, September 2014.

[10] L. Kleinrock. *Queueing Systems*, volume I: Theory. Wiley Interscience, 1975.

[11] A. Ephremides. Energy concerns in wireless networks. *IEEE Wireless Communications*, 9(4):48–59, August 2002. ISSN 1536-1284. doi: 10.1109/MWC.2002.1028877.

[12] S. Sudevalayam and P. Kulkarni. Energy harvesting sensor nodes: Survey and implications. *IEEE Communications Surveys Tutorials*, 13(3):443–461, September 2011. ISSN 1553-877X. doi: 10.1109/SURV.2011.060710.00094.

[13] M. Pinuela, P. D. Mitcheson, and S. Lucyszyn. Ambient RF energy harvesting in urban and semi-urban environments. *IEEE Transactions on Microwave Theory and Techniques*, 61(7):2715–2726, July 2013. ISSN 0018-9480. doi: 10.1109/TMTT.2013.2262687.

[14] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002. ISSN 0163-6804. doi: 10.1109/MCOM.2002.1024422.

[15] Lin Gu and J. A. Stankovic. Radio-triggered wake-up for wireless sensor networks. *Real-time Systems*, 29:157–182, 2005.

[16] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, May 2009. ISSN 1570-8705. doi: 10.1016/j.adhoc.2008.06.003. URL http://dx.doi.org/10.1016/j.adhoc.2008.06.003.

[17] Pei Huang, Li Xiao, S. Soltani, M.W. Mutka, and Ning Xi. The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys Tutorials*, 15(1):101–120, 2013. ISSN 1553-877X. doi: 10.1109/SURV.2012.040412.00105.

[18] L. Roberts. Aloha packet system with and without slots and capture. In *ACM SIGCOMM Comput. Commun. Rev.*, volume 5, 1975.

[19] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels:Part 1: CSMA modes and their throughput-delay characteristics. *IEEE Trans. Commun.*, 23:1400–1416, 1975.

[20] J. I. Capetanakis. Tree Algorithms for Packet Broadcast Channels. *IEEE Transactions on Information Theory*, 25(5):505–515, Sept. 1979.

[21] J. L. Massey. *Collision resolution algorithm and random access in Multiuser Communication Systems*. New York: Springer-Verlag, 1981.

[22] Y. Xinqing and L. Xuemei. Evaluating the energy consumption of RFID tag collision resolution protocols. In *International Conference on Mobile Ad-hoc and Sensor Networks*, pages 215–221, February 2010.

[23] V. Namboodiri and Lixin Gao. Energy-aware tag anticollision protocols for RFID systems. *IEEE Transactions on Mobile Computing*, 9(1):44–59, January 2010. ISSN 1536-1233. doi: 10.1109/TMC.2009.96.

[24] EnOcean White Paper, August 2011. URL http://www.enocean.com.

[25] M. Moradian and F. Ashtiani. Throughput analysis of a slotted aloha-based network with energy harvesting nodes. In *IEEE PIMRC*, pages 351–356, September 2012. doi: 10.1109/PIMRC.2012.6362809.

[26] O. Briante, A. M. Mandalari, A. Molinaro, G. Ruggeri, F. Vazquez-Gallego, and J. Alonso-Zarate. Duty-Cycle Optimization for Machine-to-Machine Area Networks based on Frame Slotted-ALOHA with Energy Harvesting Capabilities. In *European Wireless*, May 2014.

[27] F. Iannello, O. Simeone, and Spagnolini. Dynamic Framed-ALOHA for Energy-Constrained Wireless Sensor Networks with Energy Harvesting. In *IEEE GLOBECOM*, pages 1–6, December 2010. doi: 10.1109/GLOCOM.2010.5683738.

[28] F. Iannello, O. Simeone, and U. Spagnolini. Medium Access Control Protocols for Wireless Sensor Networks with Energy Harvesting. *IEEE Transactions on Communications*, 60(5):1381–1389, May 2012. ISSN 0090-6778. doi: 10.1109/TCOMM.2012.030712.110089.

[29] F. Iannello, O. Simeone, P. Popovski, and U. Spagnolini. Energy group-based dynamic framed ALOHA for wireless networks with energy harvesting. In *Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, March 2012. doi: 10.1109/CISS.2012.6310797.

[30] V. Sharma, U. Mukherji, and V. Joseph. Efficient energy management policies for networks with energy harvesting sensor nodes. In *Annual Allerton Conference on Communication, Control, and Computing*, pages 375–383, September 2008. doi: 10.1109/ALLERTON.2008.4797582.

[31] Ger Yang, Guan-Yu Lin, and Hung-Yu Wei. Markov chain performance model for IEEE 802.11 devices with energy harvesting source. In *IEEE GLOBECOM*, pages 5212–5217, December 2012. doi: 10.1109/GLOCOM.2012.6503948.

[32] F. C. Schoute. Control of ALOHA Signalling in a Mobile Radio Trunking System. In *International Conference on Radio Spectrum Conservation Techniques*, pages 38–42, 1980.

[33] Y. Xinqing and Z. Fan. Rapid RFID tag collision resolution with the frame slotted ALOHA protocol. In *International Conference on Computer Application and System Modeling*, volume 1, pages 517–521, November 2010.

[34] E. Egea-Lpez, J. Vales-Alonso, A. S. Martnez-Sala, M. V. Bueno-Delgado, and J. Garca-Haro. Performance evaluation of non-persistent CSMA as anti-collision protocol for active RFID tags. In *International Conference on Wired/Wireless Internet Communications*, June 2007.

[35] H. Vogt. Efficient object identification with passive RFID tags. In *International Conference on Pervasive Computing*, volume 2414, pages 98–113, August 2002.

[36] H. Vogt. Multiple object identification with passive RFID tags. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, page 6, October 2002. doi: 10.1109/ICSMC.2002.1176119.

[37] B. Knerr, M. Holzer, C. Angerer, and M. Rupp. Slot-wise maximum likelihood estimation of the tag population size in FSA protocols. *IEEE Transactions on Communications*, 58(2):578–585, February 2010. ISSN 0090-6778. doi: 10.1109/TCOMM.2010.02.080571.

[38] Z. G. Prodanoff. Optimal frame size analysis for framed slotted ALOHA based RFID networks. *Elsevier Computer Communications*, 33(5):648–653, 2010.

[39] V. Namboodiri, M. DeSilva, K. Deegala, and S. Ramamoorthy. An extensive study of slotted Aloha-based RFID anti-collision protocols. *Elsevier Computer Communications*, 35(16):1955–1966, 2012.

[40] D. K. Klair, K. Chin, and R. Raad. On the suitability of framed slotted Aloha based RFID anti-collision protocols for use in RFID-enhanced WSNs. In *International*

Conference on Computer Communications and Networks, pages 583–590, August 2007.

[41] E. Casini, R. De Gaudenzi, and Od.R. Herrero. Contention resolution diversity slotted aloha (CRDSA): An enhanced random access scheme for satellite access packet networks. *IEEE Transactions on Wireless Communications*, 6(4):1408–1419, April 2007. ISSN 1536-1276. doi: 10.1109/TWC.2007.348337.

[42] G. Liva. Graph-based analysis and optimization of contention resolution diversity slotted ALOHA. *IEEE Transactions on Communications*, 59(2):477–487, February 2011. ISSN 0090-6778. doi: 10.1109/TCOMM.2010.120710.100054.

[43] C. Stefanovic, K.F. Trilingsgaard, N. K. Pratas, and P. Popovski. Joint estimation and contention-resolution protocol for wireless random access. In *IEEE International Conference on Communications (ICC)*, pages 3382–3387, June 2013. doi: 10.1109/ICC.2013.6655070.

[44] F. Ricciato and P. Castiglione. Pseudo-random ALOHA for enhanced collision-recovery in RFID. *IEEE Communications Letters*, 17(3):608–611, March 2013. ISSN 1089-7798. doi: 10.1109/LCOMM.2013.011513.122109.

[45] L. Choudhury and S. Rappaport. Diversity ALOHA - a random access scheme for satellite communications. *IEEE Transactions on Communications*, 31(3):450–457, Mar 1983. ISSN 0090-6778. doi: 10.1109/TCOM.1983.1095828.

[46] W. Crowther, R. Rettberg, D. Walden, S. Ornstein, and F. Heart. A system for broadcast communication: Reservation-ALOHA. In *International Conference on Systems Sciences*, pages 371–374, 1973. doi: 70.

[47] S. S. Lam. Satellite packet communication–multiple access protocols and performance. *IEEE Transactions on Communications*, 27(10):1456–1466, October 1979. ISSN 0090-6778. doi: 10.1109/TCOM.1979.1094313.

[48] S. Tasaka. Multiple-access protocols for satellite packet communication networks: A performance comparison. *Proceedings of the IEEE*, 72(11):1573–1582, November 1984. ISSN 0018-9219. doi: 10.1109/PROC.1984.13054.

[49] S. S. Lam. Packet broadcast networks–a performance analysis of the R-ALOHA protocol. *IEEE Transactions on Computers*, C-29(7):596–603, July 1980. ISSN 0018-9340. doi: 10.1109/TC.1980.1675630.

[50] S. Tasaka. Stability and performance of the R-ALOHA packet broadcast system. *IEEE Transactions on Computers*, C-32(8):717–726, August 1983. ISSN 0018-9340. doi: 10.1109/TC.1983.1676309.

[51] K. Young-An and M. Nakagawa. R-ALOHA protocol for SS inter-vehicle communication network using head spacing information. In *IEEE Intelligent Vehicles Symposium*, pages 278–283, September 1996. doi: 10.1109/IVS.1996.566391.

[52] R. Verdone. Multihop R-ALOHA for intervehicle communications at millimeter waves. *IEEE Transactions on Vehicular Technology*, 46(4):992–1005, November 1997. ISSN 0018-9545. doi: 10.1109/25.653073.

[53] N. Alsbou, D. Henry, and H. Refai. R-ALOHA with priority (PR-ALOHA) in non ideal channel with capture effects. In *IEEE International Conference on Telecommunications (ICT)*, pages 566–570, April 2010. doi: 10.1109/ICTEL.2010.5478849.

[54] L. MinWoo, L. Jong-Kwan, L. Jae-Joon, and L. Jaesung. R-CRDSA: Reservation-Contention Resolution Diversity Slotted ALOHA for satellite networks. *IEEE Communications Letters*, 16(10):1576–1579, October 2012. ISSN 1089-7798. doi: 10.1109/LCOMM.2012.082012.120573.

[55] M. Kaplan and E. Gulko. Analytic properties of multiple-access trees. *IEEE Transactions on Information Theory*, 31(2):255–263, March 1985. ISSN 0018-9448. doi: 10.1109/TIT.1985.1057015.

[56] H. Wu, Y. Zeng, J. Feng, and Y. Gu. Binary tree slotted ALOHA for passive RFID tag anticollision. *IEEE Transactions on Parallel and Distributed Systems*, 24(1): 19–31, January 2013.

[57] X. Su and Y. Xiao. Analysis of Energy Consumption for Multiple Object Identification System with Active RFID Tags. In *IEEE Wireless Communications and Networking Conference*, pages 4019–4023, March 2007.

[58] P. Mathys and P. Flajolet. Q-ary collision resolution algorithms in random-access systems with free or blocked channel access. *IEEE Transactions on Information Theory*, 31(2):217–243, Mar 1985. ISSN 0018-9448. doi: 10.1109/TIT.1985.1057013.

[59] A. Janssen and M. Jong. Analysis of Contention Tree Algorithms. *IEEE Transactions on Information Theory*, 46(6):2163–2172, September 2000.

[60] V. Namboodiri and Lixin Gao. Energy-aware tag anti-collision protocols for RFID systems. In *IEEE International Conference on Pervasive Computing and Communications*, pages 23–36, March 2007. doi: 10.1109/PERCOM.2007.14.

[61] W. Xu and G. Campbell. A Near Perfect Stable Random Access Protocol for a Broadcast Channel. In *IEEE International Conference on Communications (ICC)*, volume 1, pages 370–374, June 1992. doi: 10.1109/ICC.1992.268230.

[62] J. Alonso-Zarate, C. Verikoukis, E. Kartsakli, A. Cateura, and L. Alonso. A Near-Optimum Cross-Layered Distributed Queuing Protocol for Wireless LAN. *IEEE Wireless Communications*, 15(1):48–55, 2008. ISSN 1536-1284. doi: 10.1109/MWC.2008.4454704.

[63] Wu Chien-Ting and G. Campbell. Extended DQRAP (XDQRAP). A cable TV protocol functioning as a distributed switch. In *International Workshop on Community Networking Integrated Multimedia Services to the Home*, pages 191–198, July 1994. doi: 10.1109/CN.1994.337349.

[64] Lin Harn-Jier and G. Campbell. PDQRAP-prioritized distributed queueing random access protocol. In *IEEE Conference on Local Computer Networks*, pages 82–91, 1994. doi: 10.1109/LCN.1994.386612.

[65] Wu Chien-Ting and G. Campbell. *Interleaved DQRAP with Global TQ*. Dept. of Comp. Sci., Illinois Inst. of Technology, 1995.

[66] L. Alonso, R. Agusti, and O. Sallent. A near-optimum MAC protocol based on the distributed queueing random access protocol (DQRAP) for a CDMA mobile communication system. *IEEE Journal on Selected Areas in Communications*, 18 (9):1701–1718, September 2000. ISSN 0733-8716. doi: 10.1109/49.872957.

[67] J. Alonso-Zarate, L. Alonso, C. Skianis, and C. Verikoukis. Analysis of a Distributed Queuing Medium Access Control Protocol for Cooperative ARQ. In *IEEE GLOBECOM*, pages 1–5, December 2010. doi: 10.1109/GLOCOM.2010.5683302.

[68] J. Alonso-Zarate, D. Gregoratti, P. Giotis, C. Verikoukis, and L. Alonso. Medium access control priority mechanism for a DQMAN-based wireless network. *IEEE Communications Letters*, 13(7):495–497, July 2009. ISSN 1089-7798. doi: 10.1109/LCOMM.2009.081502.

[69] B. Otal, L. Alonso, and C. Verikoukis. Energy-Efficiency Analysis of a Distributed Queuing Medium Access Control Protocol for Biomedical Wireless Sensor Networks in Saturation Conditions. *Sensors*, 11(2):1277–1296, 2011. ISSN 1424-8220. doi: 10.3390/s110201277. URL http://www.mdpi.com/1424-8220/11/2/1277.

[70] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang. Energy harvesting wireless communications: A review of recent advances. *Selected Areas in Communications, IEEE Journal on*, 33(3):360–381, March 2015. ISSN 0733-8716. doi: 10.1109/JSAC.2015.2391531.

[71] A. Seyedi and B. Sikdar. Modeling and analysis of energy harvesting nodes in wireless sensor networks. In *Annual Allerton Conference on Communication, Control, and Computing*, pages 67–71, September 2008. doi: 10.1109/ALLERTON.2008.4797537.

[72] Jing Lei, R. Yates, and L. Greenstein. A generic model for optimizing single-hop transmission policy of replenishable sensors. *IEEE Transactions on Wireless Communications*, 8(2):547–551, February 2009. ISSN 1536-1276. doi: 10.1109/TWC.2009.070905.

[73] J. Ventura and K. Chowdhury. Markov modeling of energy harvesting body sensor networks. In *IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 2168–2172, September 2011. doi: 10.1109/PIMRC.2011.6139899.

[74] P. Lee, Zhi Ang Eu, Mingding Han, and H. Tan. Empirical modeling of a solar-powered energy harvesting wireless sensor node for time-slotted operation. In *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pages 179–184, March 2011. doi: 10.1109/WCNC.2011.5779157.

[75] Chin Keong Ho, Pham Dang Khoa, and Pang Chin Ming. Markovian models for harvested energy in wireless communications. In *Communication Systems (ICCS), 2010 IEEE International Conference on*, pages 311–315, Nov 2010. doi: 10.1109/ICCS.2010.5686445.

[76] M. Gorlatova, A. Wallwater, and G. Zussman. Networking low-power energy harvesting devices: Measurements and algorithms. In *INFOCOM, 2011 Proceedings IEEE*, pages 1602–1610, April 2011. doi: 10.1109/INFCOM.2011.5934952.

[77] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver. Low energy operation in WSNs: A survey of preamble sampling MAC protocols. *Computer Networks*, 55 (15):3351–3363, 2011.

[78] F. C. Schoute. Dynamic frame length aloha. *IEEE Transactions on Communications*, 31(4):565–568, Apr 1983. ISSN 0090-6778. doi: 10.1109/TCOM.1983.1095854.

[79] *IEEE Std 802.15.4-2006: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. IEEE Std., September 2006.

[80] CC2520 datasheet. URL http://www.ti.com/lit/ds/symlink/cc22520.pdf.

[81] L. Kleinrock. *Communication Nets; Stochastic Message Flow and Delay*. McGraw-Hill Book Company, New York, 1964. (Out of Print.) Reprinted by Dover Publications, 1972 and in 2007. Published in Russian, 1971, Published in Japanese, 1975.

[82] F. Vazquez-Gallego, L. Alonso, and J. Alonso-Zarate. Energy and Delay Analysis of Contention Resolution Mechanisms for Machine-to-Machine Networks based on Low-Power WiFi. In *IEEE International Conference on Communications (ICC)*, June 2013.

[83] F. Vazquez-Gallego, J. Alonso-Zarate, S. Wu, Y. Chen, and K. Chai. Analysis and performance evaluation of dynamic frame slotted-ALOHA in wireless Machine-to-Machine networks with energy harvesting. In *IEEE GLOBECOM Workshop on Green Broadband Access*, December 2014.

[84] P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso-Zarate, X. Vilajosana, and L. Alonso. "Demonstrating Low-Power Distributed Queuing for Active RFID communications at 433 MHz. In *IEEE INFOCOM*, May 2014.