PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Energy-Efficient Medium Access Control Protocols and Network Coding in Green Wireless Networks

Raul Palacios-Trujillo

Advisor:

Prof. Fabrizio Granelli

University of Trento

December 2014

Abstract

Wireless networks are a popular means of communications in daily social and business activities of many users nowadays. However, current estimates indicate that wireless networks are expected to significantly contribute to the rapidly increasing energy consumption and carbon emissions of the Information and Communication Technologies (ICT) sector. Crucial factors leading to this trend are the continuous growth of wireless network infrastructure coupled with the increased number of user wireless devices equipped with various radio interfaces and batteries of extremely limited capacity (e.g., smartphones). The serious problem of energy consumption in wireless networks is mainly related to the current standard designs of wireless technologies. These approaches are based on a stack of protocol layers aiming to maximize performance-related metrics, such as throughput or Quality of Service (QoS), while paying less attention to energy efficiency. Although the focus has shifted to energy efficiency recently, most of the existing wireless solutions achieve energy savings at the cost of some performance degradation.

This thesis aims at contributing to the evolution of green wireless networks by exploring new approaches for energy saving at the Medium Access Control (MAC) protocol layer and the combination of these with the integration of the Network Coding (NC) paradigm into the wireless network protocol stack for further energy savings. The main contributions of the thesis are divided into two main parts. The first part of the thesis is focused on the *design and performance analysis and evaluation of novel energy-efficient distributed and centralized MAC protocols for Wireless Local Area Networks (WLANs). The second part of the thesis turns the focus to the design and performance analysis and evaluation of new NC-aware energy-efficient MAC protocols for wireless ad hoc networks. The key idea of the proposed mechanisms is to enable multiple data exchanges (with or without NC data) among wireless devices and allow them to dynamically turn on and off their radio transceivers (i.e., duty cycling) during periods of no transmission and reception (i.e., when they are listening or overhearing).*

Validation through analysis, computer-based simulation, and experimentation in real hardware shows that the proposed MAC solutions can significantly improve both the throughput and energy efficiency of wireless networks, compared to the existing mechanisms of the IEEE 802.11 Standard when alone or combined with the NC approach. Furthermore, the results presented in this dissertation help understand the impact of the on/off transitions of radio transceivers on the energy efficiency of MAC protocols based on duty cycling. These radio transitions are shown to be critical when the available time for sleeping is comparable to the duration of the on/off radio transitions.

Keywords

[Wireless network, energy efficiency, medium access control protocol, network coding, cross-layering]

A mi familia.
Para Susana.

Acknowledgments

This PhD dissertation would not have been possible without the help and support of many people. Herein, I would like to express my gratitude to all of them.

First of all, I would like to thank Prof. Claudio Sacchi, Prof. Tommaso Pecorella, and Prof. Igor Bisio for accepting the invitation to be part of my PhD defense committee.

I would also like to thank my advisor, Prof. Fabrizio Granelli, for his guidance during the last three years of the PhD degree and for giving me the opportunity to work and grow as a researcher at the University of Trento within the framework of the GREENET European research project. You have always been very kind with me and have always tried to provide me with all the materials and tools I needed to properly carry out my research. I feel that you have been far more than my mentor and, besides a great professor and researcher, I believe that you are a very nice person. I wish you all the best for the future and hope we can continue working together for many years.

I would like to extend my gratitude to the GREENET project for several reasons. One is the financial support from the project through the University of Trento, which allowed me to purchase equipment to conduct top-notch research and attend the project meetings and many international conferences and workshops. Another reason is the top-level training I received from the project through very interesting and useful seminars

organized by the partners of the project, exchange activities such as visits and secondments to other partners of the project, and the organization and participation to workshops and industrial dissemination days. Finally, the third reason is the multi-cultural and multi-disciplinary network of talented young researchers and top-level research scientists from whom I have learned to be a better researcher and person and with whom I have spent many hours of work but also some of fun during the social dinners of the project meetings.

More specifically related to the previous point, I would like to thank the GREENET coordinators, Prof. Luis Alonso Zarate, Dr. Christos Verikoukis, and Melani Gurdiel Garcia, for managing such a fantastic project and making it by all of us together a success story. Also, I wish to acknowledge all the GREENET supervisors, Harald Haass, Marco di Renzo, Fabrizio Graziosi, Andreas Foglar, Hugo Marques, Jonathan Rodriguez, Gerrit Shulte, Kimon Kontovasilis, and Cristophe Gruet, for all their valuable comments and suggestions regarding my research activities. A special thanks goes to Marco di Renzo, who was the president of my qualifying exam committee at the end of the first year of the PhD degree and gave me very interesting comments and ideas to complete this thesis.

In addition, I would like to thank Andreas Foglar for organizing my secondment in his company, InnoRoute Gmbh, located in Munich (Germany) and for believing in me and my ideas to prepare and submit a patent. I would like to extend my gratitude to the rest of the InnoRoute team, Klaus Starnberger, Christian Liss, Tamas Fendler, Daniel Scharfen, Lorenzo di Gregorio, and Danica Gajic, with whom I felt very comfortable during the months I spent there. Related to my secondment, I also spent few weeks at CTTC to work with Francisco Vazquez Gallego for the implementation of my ideas on programmable wireless platforms. I would like to thank him for all his help and the lessons learned and I want to take this opportunity to express my special thanks to Dr. Verikoukis for authorizing me to have such a great experience at CTTC.

I will never forget all the former and current GREENET researchers, Konstantinos Ntontin, Athanasios Stavridis, Xiping Wu, Danica Gajic, Saud Althunibat, Alexandra Bousia, Agapi Mesodiakaki, Vasileios Miliotis, Riccardo Bassoli, Victor Sucasas, Eleni-Constantina Davri, Dereje Assefa, Xeonofon Foucas, Gergo Ertli, Pol Torres Compta, Marium Jalal Chaudhry, Sandeedp Narayanan, and Xavier Pons Masbernat, for the great moments we had together during these three years.

Likewise, I cannot forget my main research collaborators during these three years: Jesus Alonso Zarate, Luis Alonso, Dzmitry Kliazovich, Janus Heide, and Frank H. P. Fitzek. I would like to thank you all for all your very useful comments and suggestions to improve our papers and my research in general. I look forward to continuing our fruitful collaboration in the future.

The work presented in this thesis could not have been possible without the invaluable collaboration of some of my bachelor and master's students: Moatez Larbaa, Alessandro De Rossi, Habte Haile, Peter Nakeba, Francesco Franch, and Gedlu Mengistie. Thank you all for your valuable contributions to some parts of the thesis. Also, I want to take this opportunity to thank Mbiika Ceriano (as a former master's student), Carlos Astudillo (as an exchange master's student), and my current master's students (Robel Endeshaw, Biniam Hailu, Habte Dechasa, and Mamun Rashid) for all what I have learned from you to become a better supervisor and person for the future. I must say that I have been very lucky to meet you all and wish you all the best for your future careers.

Also, I wish to acknowledge the great work that all the staff of the DISI department at the University of Trento do, who have made possible that the development of this thesis has been a wonderful experience. My special

gratitude goes to Mirella Collini, Andrea Stenico, and Manuel Zucchelini, who have always been available to help me whenever I have had a problem.

My stay at the University of Trento has also left me very good friends, such as, Qi Wang, Le Dieu Thu, and Mai Anh Nguyen, among others. I thank you all for the time being together where we enjoyed multi-cultural dinners and had very interesting discussions about research, the PhD, and life in general. Also, I cannot forget to thank all my colleagues of "open space" from the DISI department with whom I have shared PhD courses, seminars, and talks.

I would also like to thank my close friends, who, since the time they knew I was going to Trento to do research and pursue the PhD degree at the University of Trento, they have always been supporting me.

As in every step forward in my life, I cannot forget that this achievement could not have been possible without the support of my family: my parents and my sister. You have always been there with me throughout all my life and thanks to your teachings and advice I have become the man who I am today. Without you, I would not be where I am now. In particular, I would like to dedicate this work to my dear father, who few time ago was able to recover from a serious illness. You are a very nice man and are a very important person in my life. I wish you from the deepest of my heart a long, healthy, and happy life from now on.

Finally, I would like to thank Susana for her admirable patience during the last three years, for being always there for me, and for helping me in every sense to finish this thesis. You are the most important person in my life and I hope you will be with me for the rest of my life to help me achieve new goals in the future.

I would like to finish these acknowledgments by extending my gratitude to everyone else who, directly or not, consciously or not, has contributed to this thesis.

A mi familia. Para Susana.

Como en cada paso hacia adelante en mi vida, no puedo olvidar que este logro no hubiera sido posible sin el apoyo incondicional de mi familia: mis padres y mi hermana. Vosotros habéis estado siempre a mi lado a lo largo de toda mi vida y gracias a vuestras enseñanzas y consejos me he convertido en el hombre que soy hoy en día. Sin vosotros, no hubiera llegado a donde estoy ahora. En particular, me gustaría dedicar este trabajo a mi querido padre, el cual ha sido capaz de recuperarse de una grave enfermedad hace poco tiempo. Tú eres un buen hombre y eres una persona muy importante en mi vida. Te deseo desde lo más profundo de mi corazón una vida feliz, saludable, y duradera de ahora en adelante.

Finalmente, quiero agradecer a Susana por su admirable paciencia durante estos tres últimos años, por estar siempre a mi lado, y por ayudarme en todos los sentidos a acabar esta tesis. Tú eres la persona más importante en mi vida y espero que sigas a mi lado el resto de mi vida para ayudarme a conquistar nuevas metas en el futuro.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter introduces the problem and provides its positioning in the framework of the main research areas of ICT. The structure of the chapter is described as follows. Section 1.1 describes the context within which the thesis is positioned. The research problem addressed in the thesis is detailed in Section 1.2. Section 1.3 presents the main contributions of the thesis. Then, Section 1.4 discusses how the thesis improves the state of the art. Finally, a description of the structure of the thesis is presented in Section 1.6.

1.1 Context

Climate change and energy consumption are widely recognized by society as the most important issues for sustainable economic growth in both developed and developing countries. To enable the low-carbon economy in Europe, the European Commission set ambitious targets in 2008 to reduce greenhouse gas emissions by 20% and to improve energy efficiency by 20%, both by 2020 [10]. Indeed, the ICT sector can play an important role to meet these challenges by improving energy efficiency in all economic sectors. However, the energy consumption and carbon emissions of the sector itself are rapidly growing and must be minimized. Therefore, the energy

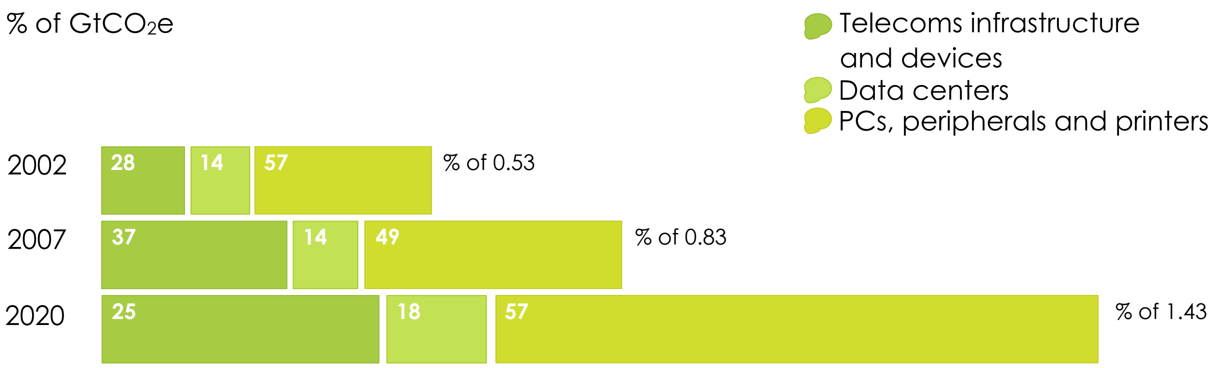


Figure 1.1: ICT carbon footprint by sector [1]

efficiency of ICT has recently gained significant attention from both the European Union and the ICT industry to achieve green ICT.

Currently, the global ICT carbon footprint represents roughly 3% of the worldwide energy consumption and 2% of the total carbon emissions [1,11]. This rather small amount of ICT carbon emissions is, however, comparable to the total carbon emissions caused by international air traffic, or one quarter of global carbon emissions by cars [11]. Analyzing the ICT carbon emissions by sector in Fig. 1.1, it can be seen that the carbon footprint of both telecommunications infrastructure and devices (32% of total ICT carbon emissions in 2002) has been significantly increasing since 2002, in comparison with those of other ICT sub-sectors like Personal Computers (PCs) and data centers.

Also, as it is reported in Fig. 1.2, while the share of wired networks have remained almost constant year after year, the increasing contribution of wireless networks is expected to dominate the total telecommunications carbon footprint by 2020. This prediction is mainly based on the rapid growth of wireless network infrastructure, the increased number of wireless devices, and the last projections of wireless data traffic growth. Thus, developing energy-efficient wireless networking solutions becomes urgent to reduce the global ICT carbon footprint by 2020.

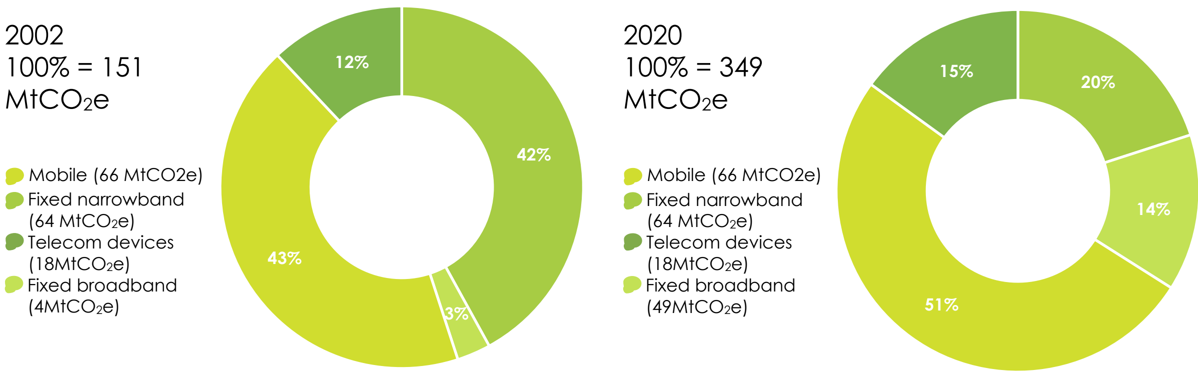


Figure 1.2: Global telecommunications carbon footprint by sub-sector [1]

Wireless networks have become an essential means of communications in our daily social and business activities, since they allow mobility while supporting a wide range of services, such as voice, video, web access, and peer-to-peer content sharing. Commonly used wireless networks among users are infrastructure wireless networks like cellular networks and WLANs and wireless ad hoc (infrastructure-less) networks.

Infrastructure wireless networks are managed by central nodes, referred to as Base Stations (BSs) in cellular networks or Access Points (APs) in WLANs, that are responsible for coordinating access to one or several transmission channels among user wireless devices located in their coverage areas. These nodes usually provide access to the Internet for the connected users through a wired network infrastructure. In infrastructure wireless networks, wireless communications occur in the last hop between the central nodes (a BS or an AP) and the user devices (i.e., single-hop communications).

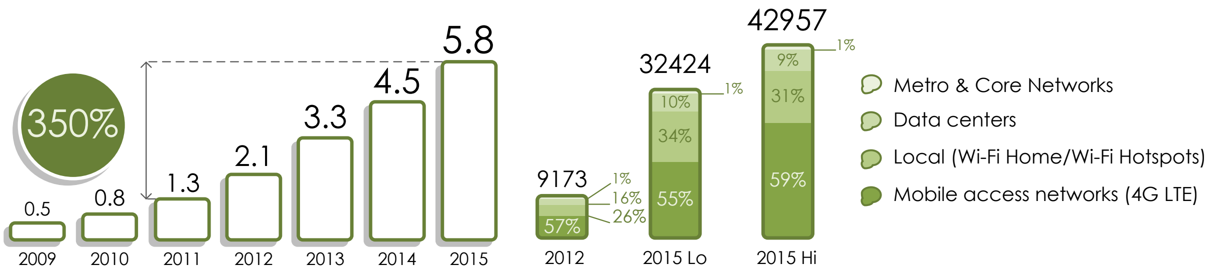
On the other hand, wireless ad hoc networks are coordinated in a distributed manner by wireless devices that are located within the transmission range of each other and dynamically maintain network connectivity among them to exchange data through a shared wireless channel. One of these devices may be connected to a BS or an AP to provide a gateway to the Internet for the rest of network devices. In wireless ad hoc networks, all devices participate in routing by forwarding data for other devices (i.e., multi-hop communications).

There exist various wireless technologies defined by several standards that specify the interconnection and interoperability of wireless devices in different wireless networks. For example, standardization bodies like the 3rd Generation Partnership Project (3GPP) and the Institute of Electrical and Electronics Engineers (IEEE) provide, respectively, the standards of the Universal Mobile Telecommunications System (UMTS) and Long Term Evolution (LTE) technologies for cellular networks and the standard of the 802.11 technology for WLANs, also known as Wireless Fidelity (Wi-Fi). The specifications of these wireless technologies by the standards are based on the traditional Open Systems Interconnection (OSI) protocol stack of the International Organization for Standardization (ISO). As it is defined in the ISO/OSI model, the main functions of a wireless network are split into protocol layers that are independently designed, implemented, and optimized and only then interconnected to work as a whole.

With currently employed wireless technologies, the main problems of energy consumption can be associated with the two main actors operating wireless networks: central coordinators (wireless network infrastructure) and wireless devices.

Wireless network infrastructure

One of the largest mobile telecommunications operators maintains 238.000 BS sites worldwide. Each site contains multiple BSs continuously operating to serve different radio access technologies. Overall, they account for 60% of the total energy consumption and carbon emissions caused by the company as a whole [12]. In addition, Wi-Fi APs have been extensively deployed in public and private areas, such as, university campuses, business parks, and user homes during the last decade. Typically, they are



(a) Number of Wi-Fi public hotspots in the world (in million), 2009-2015 [13]

(b) Total annual wireless cloud energy consumption (GWh), 2012-2015 [14]

Figure 1.3: Growth and energy consumption of Wi-Fi APs

constantly on to provide users with a continuous seamless connection to the Internet, hence consuming significant amounts of energy. More than 3 million Wi-Fi public APs were expected to be deployed worldwide in 2013 and this figure is expected to increase by 75% in 2015, as shown in Fig. 1.3a. As a consequence, Wi-Fi homes and hotspots will contribute by 31-34% to the overall yearly cloud energy consumption by 2015, being the second main contributor after mobile networks, as it can be seen in Fig. 1.3b. With growing energy prices, energy saving methods for BSs and APs are thus of paramount importance for telecommunications operators and business and home users to limit the annual electricity bill.

Wireless devices

Currently, a wide variety of portable devices, e.g. laptops, tablets, and smartphones, are equipped with multi-standard radio interfaces, such as UMTS/LTE, Wi-Fi, and Bluetooth, to provide users with a flexible and powerful wireless connection. Unfortunately, multi-standard devices, especially those of limited size (like smartphones), require a significant amount of their energy resources for maintaining two or more radio interfaces, hence quickly depleting their batteries. For instance, downloading data using UMTS or Wi-Fi consumes more energy than what is consumed by the Central Processing Unit (CPU) or the display in some smartphones,

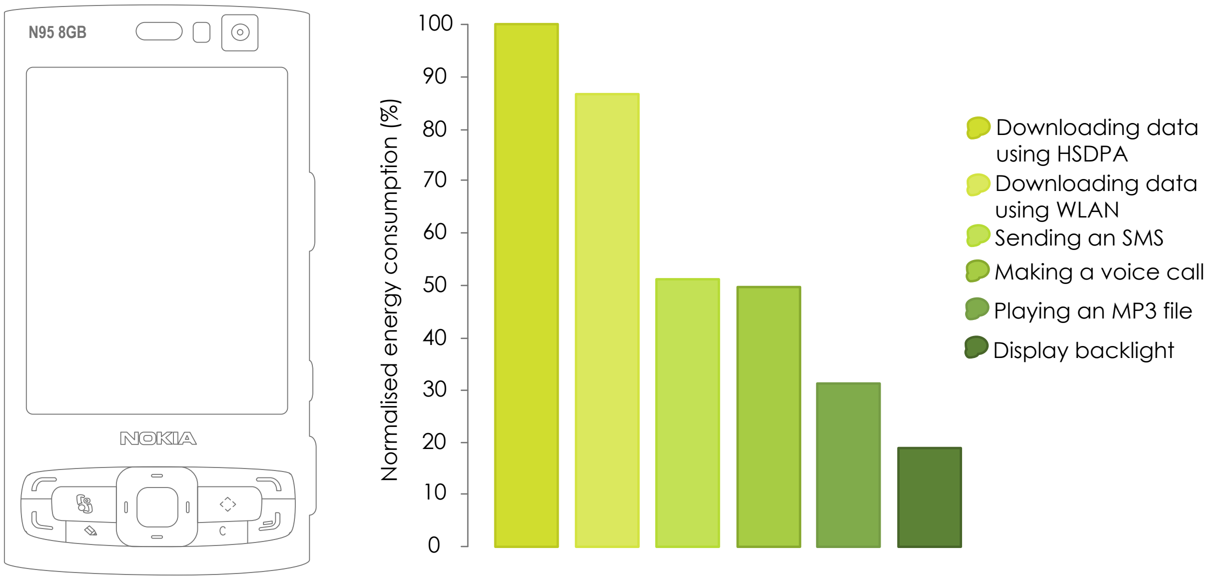
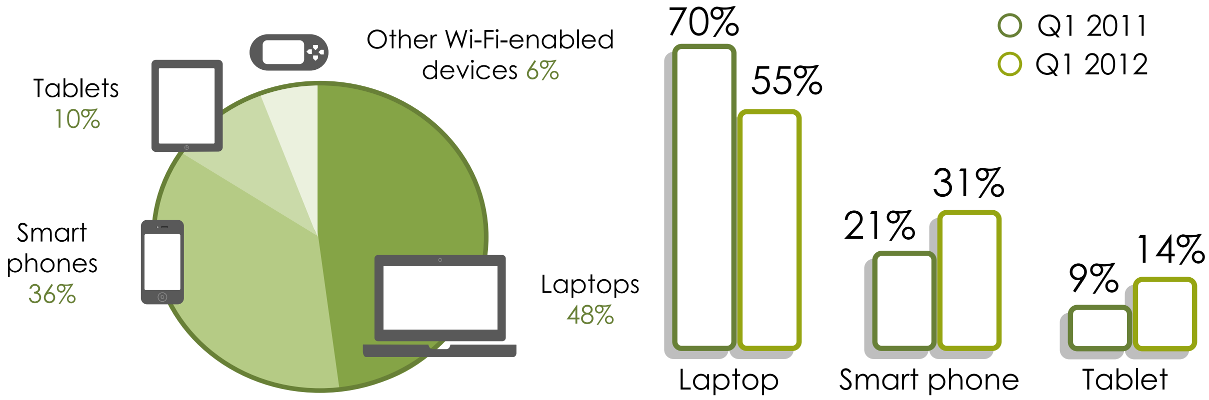


Figure 1.4: Normalized energy consumption of several entities in a Nokia N95 smartphone [2]

as it is shown in Fig. 1.4. In addition, laptops and smartphones are the most commonly used Wi-Fi-enabled devices in Wi-Fi hotspots, as shown in Fig. 1.5a. But more importantly, smartphones are increasingly used year after year (see Fig. 1.5b). Despite the evolution on battery technology, the progress on scaling and circuit design, and the development of novel thermal and cooling techniques, new approaches for energy saving (not only hardware but also software) are needed in order to prolong the operational time of battery-powered devices.

1.2 Problem Statement

The fact that the current standard designs of wireless networks are based on the ISO/OSI protocol stack makes them be unsuitable to face new challenges in wireless networks, such as minimizing energy consumption while guaranteeing the highest possible performance. Currently operating wireless networks have been mainly designed to maximize performance-



(a) Devices used in Wi-Fi hotspots [15] (b) Public Wi-Fi connections by device, 2012 [14]

Figure 1.5: Use and type of Wi-Fi-enabled devices in Wi-Fi networks

related metrics, such as, throughput, QoS, and reliability, while usually paying less attention to energy efficiency. The future designs of wireless networks need to consider energy efficiency across all layers of the protocol stack in a cross-layer approach, where the protocol layers are aware of the requirements of lower and upper layers to maximize energy efficiency.

Many solutions have been proposed to improve the energy efficiency of wireless networks by introducing changes in all the layers of the protocol stack. In this thesis, the focus has been put on the cross-layer interactions between Physical (PHY) and data link layers and data link and network layers of the protocol stack. More specifically, the MAC sublayer of the data link layer directly interfaces to the PHY layer and is responsible for managing access to one or several wireless channels shared among multiple wireless devices. The channel access control is performed by the MAC protocol, which defines the rules that wireless devices need to obey to communicate within a multiple access network. The MAC protocol takes decisions that determine how the wireless interfaces of network devices are used to perform channel access control. Since the wireless interface has shown to be a major source of energy consumption for wireless devices, the MAC sublayer is thus considered as a central point of the protocol stack for energy consumption control and energy saving through cross-layer methods.

The IEEE 802.11 Standard [16] for WLANs (Wi-Fi) specifies a set of MAC protocols that have been widely investigated over the last years. Basically, two main mechanisms are defined for sharing access to the wireless channel: a mandatory contention-based distributed channel access mechanism called Distributed Coordination Function (DCF) and an optional polling-based centralized channel access mechanism called Point Coordination Function (PCF). Unfortunately, these MAC protocols have not been optimized for energy efficiency. As a result, wireless devices implementing them, referred to as Wireless Stations (STAs) in the terminology of the Standard, consume a significant amount of energy for keeping their radio transceivers always on (i.e., constant channel listening) and receiving data addressed to other destinations (i.e., overhearing).

To reduce the energy consumption of wireless devices, the Standard also defines an optional power saving mechanism called Power Save Mode (PSM) that exploits the capability of some wireless interfaces to enable a low-power sleep state where the radio transceiver is turned off. This yields energy savings at the cost of not being able to either transmit or receive data when in this state. Typically, the STAs executing the PSM periodically alternate between awake (i.e., the radio transceiver is turned on) and sleep states to listen to selected beacons periodically transmitted by the AP that contain information about data buffered for them in the AP. Also, they may wake up to transmit data at any time. Unfortunately, this MAC protocol may produce high control packet overhead for retrieving data from the AP and may also cause some performance degradation due to the dependency on the beacon and selected listen intervals. In addition, the STAs may experience high energy consumption during awake periods, where they may execute either the DCF or the PCF.

Therefore, new energy-efficient distributed and centralized MAC protocols need to be investigated aiming to boost both the throughput and energy efficiency of WLANs when either the standard DCF, PCF, or PSM are executed.

In addition, the NC paradigm [17] has emerged as a ground-breaking technology for the efficient operation of wireless networks. In multi-hop (relay-aided) wireless networks, information is delivered from a source node to a destination node by routing through intermediate (or relay) nodes of the network. Each intermediate node is simply required to store and forward the received information to the next intermediate node until reaching the final destination node. In contrast with simple store-and-forward schemes, the basic principle of NC is to allow intermediate nodes to take several received packets and combine them into a single coded packet for transmission by exploiting the broadcast channel. Packets are coded by applying linear coding operations (e.g, XOR) and using an encoding vector added to the header of the transmitted coded packet to allow potential receiver nodes to perform successful decoding, thus introducing additional overheard. Despite the coding overhead, the NC operation allows increasing the information content of each transmission and reducing the total number of channel accesses, hence improving throughput and energy efficiency.

NC has been extensively studied in the literature. The first work dealing with the theory of NC was presented in [17], which showed that combining multiple information flows in wireless network nodes can provide multicast capacity. Since then, NC has gained increasing attention and has been applied to multiple wireless network scenarios, showing improvements in terms of throughput, energy efficiency, robustness, and security. So far, many existing works have been mainly theoretical and have been based on important assumptions related to the structure of the network or the channel access scheme. As an important step forward to bridge the gap between theory and practice, the inspiring work in [18] introduces COPE as the first implementation of a practical NC protocol in Wi-Fi networks (i.e., based on the IEEE 802.11 Standard). COPE seamlessly integrates an NC layer between the data link and network layers of the protocol stack that identifies coding opportunities to forward multiple packets from different sources in a single transmission.

In [18] the authors show that there exist important practical considerations that should be taken into account for the proper implementation of NC in currently operating wireless networks. More specifically, NC awareness of the MAC protocol is essential for the proper NC operation. Unfortunately, the widely used IEEE 802.11 MAC protocol (DCF) presents some limitations to efficiently work with NC: *(i)* lack of per-node and per-packet channel access priority for NC, *(ii)* lack of reliable and collision avoidance mechanisms to broadcast coded packets, *(iii)* retransmission schemes unaware of NC, and *(iv)* need for continuous channel sensing for coding and decoding opportunities (channel listening and overhearing).

COPE addresses some of these issues by introducing various mechanisms that do not require any modifications of the IEEE 802.11 MAC protocol. Unfortunately, COPE still shares most of the limitations of the IEEE 802.11 DCF MAC protocol, which are mainly control packet overhead, collisions, contentions, and continuous channel sensing. The results presented in [18] indicate that the interactions between opportunistic NC and the MAC protocol have to be carefully studied and new enhancements at the MAC layer need to be proposed to achieve high cooperation with the NC protocol layer. Furthermore, new approaches for minimizing the time that nodes spend in channel listening and overhearing while ensuring the proper NC operation are needed in order to achieve further energy savings.

Therefore, new NC-aware energy-efficient MAC protocols need to be investigated aiming to boost the throughput and energy efficiency of wireless networks when the NC operation is executed.

1.3 Contributions

This thesis aims at contributing to the field of green wireless networks by investigating new mechanisms at the MAC layer for energy saving and the combination of these with the NC approach for higher energy savings. The main contributions of the thesis can be divided into two parts. The first part of the thesis is focused on the design and performance analysis and evaluation of novel energy-efficient distributed and centralized MAC protocols for WLANs. The second part of the thesis shifts the focus to the design and performance analysis and evaluation of new NC-aware energy-efficient MAC protocols for wireless ad hoc networks.

In the first part, the new energy-efficient distributed MAC protocols are the Bidirectional MAC protocol (BidMAC) and the Green Bidirectional MAC protocol (GreenBid). The basic idea behind BidMAC is to enable receiver-initiated bidirectional transmissions between the AP and the STAs in a contention-free manner once one of them has gained access to the wireless channel. This approach is very suitable for scenarios with bidirectional traffic as it reduces contention in the wireless channel when compared to the case when the standard DCF is executed. Furthermore, GreenBid extends the BidMAC operation by exploiting the longer duration of bidirectional transmissions to allow those STAs not involved in the communication to go to sleep, in a way similar to the standard PSM. This approach is able to significantly improve both the throughput and energy efficiency of STAs in highly dense networks and with heavy traffic conditions. At the same time, GreenBid can also be used in conjunction with the PSM or other power saving mechanisms to achieve energy saving when the number of STAs and the traffic load in the network are both low. Note that the fact that both MAC protocols are based on the DCF makes them also suitable for wireless ad hoc networks, although the extensions of the MAC protocols for this scenario has been left as a possible future line of research.

In addition, in this first part of the thesis the Bidirectional Polling MAC protocol (BidPoll) and the Green Polling MAC protocol (GreenPoll) are proposed as the new energy-efficient centralized MAC protocols. The basic idea behind BidPoll is to enable contention free periods, based on polling with beacons, during which the AP and the STAs can be reserved slots for sequential downlink and uplink transmissions with a very low overhead of control packets, when compared to the case when the PCF is executed. Moreover, GreenPoll is an extension of BidPoll that allows the STAs involved in a contention free period to save energy by turning off their radio transceiver after exchanging data with the AP, in a way similar to the PSM. In addition, those STAs not involved in data transfer can also enter the sleep state until the contention free period completes. Like GreenBid, GreenPoll can increase throughput and significantly improve the energy efficiency of the STAs during periods of high network activity, while being able to work in combination with the PSM or other power saving mechanisms during periods of low network activity.

The performances of the new BidMAC, GreenBid, BidPoll, and GreenPoll MAC protocols are evaluated by means of theoretical analyses and computer-based simulations in terms of throughput and energy efficiency in a WLAN consisting of an AP and a finite number of STAs. Relevant system parameters, such as, the traffic load, data packet length, data transmission rate, and number of STAs in the network, are used for the evaluation and comparison of the new MAC protocols with the standard

DCF and PCF. Furthermore, an experimental performance evaluation of BidMAC is carried out in a proof-of-concept network formed by an AP and two STAs in order to validate the high performance of the new MAC protocol in real environments when compared to standard DCF.

Regarding the second part of the thesis, the new NC-aware energy-efficient MAC protocols are the Bidirectional NC-aware MAC protocol (BidCode) and the Green NC-aware MAC protocol (GreenCode). The basic idea behind BidCode is to allow intermediate nodes to combine several received packets into coded packets and immediately forward them upon successful reception of data (i.e., receiver-initiated bidirectional transmissions involving coded data). This approach is very suitable for congested relay nodes that have coded data ready to be transmitted because they do not need to contend for channel access with other nodes in their coverage areas, as it would be the case with standard DCF and COPE. Furthermore, GreenCode extends the BidCode operation to allow potential overhearing nodes of a bidirectional coded data transmission to go to sleep when they recognize that the transmitted coded data do not provide any new information. This approach eliminates unnecessary overhearing and increases energy efficiency with no performance degradation of the NC operation. BidCode and GreenCode can be considered as extensions of BidMAC and GreenBid with NC awareness for wireless ad hoc networks.

The performances of the new BidCode and GreenCode NC-aware MAC protocols are evaluated by means of theoretical analyses and computer-based simulations in terms of throughput and energy efficiency in various wireless ad hoc network scenarios. Important system parameters, e.g., the traffic load, data packet length, and data transmission rate, are considered for the evaluation and comparison of the new NC-aware MAC protocols with the standard DCF and COPE and new BidMAC and GreenBid, respectively. Furthermore, an experimental performance evaluation of Bid-

Code is carried out in a proof-of-concept network formed by two sources nodes and a relay node in order to validate the high performance of the new NC-aware MAC protocol in real environments when compared to standard DCF and COPE.

1.4 Innovative Aspects

In the last years, strong efforts from both research and standardization communities have been devoted to the development of new MAC-layer enhancements to achieve high throughput and energy efficiency in WLANs. There are some survey papers that attempt to summarize the major contributions in [19, 20]. Existing energy-efficient MAC solutions for WLANs address the problems of energy consumption of both the DCF and PCF during active periods (or in active mode) and the PSM during low-power periods (or in PS mode). In active mode, the proposed MAC schemes aim at reducing the overhead of control packets and silent periods, reducing the number of contentions and collisions, and minimizing the time for channel monitoring. In PS mode, the proposed solutions minimize the control packet overhead and awake time of STAs to retrieve downlink data from the AP and maximize the sleep period based on prediction of packet arrivals from upper layers.

Despite the huge amount of work on energy efficiency at the MAC layer in WLANs, none of the existing MAC solutions jointly address all the problems of energy consumption during both active and low-power periods and, at the same time, are able to improve the overall WLAN performance. For example, most of the proposed solutions in active mode do not solve the problems of channel listening and overhearing, hence suffering from yet low energy efficiency. In contrast, those that solve these by enabling low-power active state periods may introduce some performance degradation or suffer from scalability limitations in densely populated WLANs. Similarly, most of the proposed solutions in PS mode do not solve the problems of energy consumption when the STAs in PS mode need to be awake to transmit or receive data. Also, the dependency on the beacon and listen intervals may introduce performance degradation and scalability limitations when the number of STAs and the traffic load in the network are high.

In contrast with the state-of-the-art solutions, the energy-efficient distributed and centralized MAC protocols proposed in this thesis are able to achieve high throughput and high energy efficiency in WLANs during both active and low-power periods. The key features of the novel MAC protocols are:

- Avoid the use of listen intervals and beacons attaching information about buffered downlink data to overcome performance and scalability limitations when the number of STAs and the traffic load are high. This feature can be enabled as long as all the STAs operate in active mode.

- Exploit the time that the channel will remain busy to allow the STAs not involved in data transmission or reception (i.e., overhearing) to opportunistically enter the sleep state to save energy. The sleep operation is feasible provided that the amount of time for sleeping (i.e., the duration of a data transmission) is longer than the time required by the radio transceivers of STAs to switch between on and off states. This feature can be applied to STAs in either active or PS mode. The data transmission time depends on the amount of data to be transmitted and the data transmission rate used whereas the duration of the on/off radio transitions, which is typically in the order of hundreds of microseconds [2, 21, 22], depends on the radio hardware design.

- Increase the duration of data transmissions to facilitate the sleep operation in overhearing STAs by enabling bidirectional transmissions or in general any MAC techniques that allow increasing the amount of transmitted data (e.g., batch transmissions and frame aggregation). This also improves the overall network throughput for reducing the overhead of control packets and silent periods and the number of contentions in the network.

In addition, various MAC-layer enhancements being aware of the NC approach have been proposed over the last years in order to achieve higher energy savings by reducing the negative effects of the IEEE 802.11 MAC protocol on the performance of NC. Survey papers that cover this topic at a high level are [23, 24]. Existing NC-aware MAC-layer solutions can be classified into three categories. In the first category, the proposed approaches manage the transmission queues to give a higher transmission priority to coded packets (i.e., queue-level priority access). The second category deals with solutions that provide a higher channel access priority for relay nodes that have coded packets ready to send by adjusting channel access parameters based on different network indicators (i.e., channel-level priority access). Finally, approaches that combine power saving strategies and NC are included in the third category (i.e., low-power overhearing).

It should be note that, despite the strong research efforts, none of the existing NC-aware MAC solutions jointly address all the cross-layer issues of NC with IEEE 802.11 MAC in order to improve both the throughput and energy efficiency of wireless networks. For example, the solutions based on queue-level prioritization are limited by the fair channel access distribution of the IEEE 802.11 DCF MAC protocol among all competing nodes. Also, those based on channel-level prioritization can only provide a higher channel access priority on average, i.e., probabilistic or relative (not absolute) and, as a result, they cannot guarantee immediate channel accesses for congested relay nodes that have coded packets ready to be transmitted.

Furthermore, none of these approaches minimize the time that the nodes spend in channel listening and overhearing in order to achieve further energy savings. On the other hand, those that solve this problem by enabling sleeping periods in conjunction with NC may introduce performance degradation and additional delays.

In contrast with the state-of-the-art solutions, the NC-aware energy-efficient MAC protocols presented in this thesis are able to boost the throughput and energy efficiency of wireless networks when the NC operation is executed. The key features of the novel NC-aware MAC protocols are:

- Allow congested relay nodes to transmit combined data without competing for channel access by exploiting bidirectional channel accesses upon successful reception of data. This approach ensures an immediate channel access for a relay node when it has a coded packet ready to send as soon as it receives a data packet from any other node located in its coverage range. It is also possible to combine the approach with batch coded data transmissions and coded data aggregation for a more efficient operation.

- Allow overhearing nodes to decide when they can go to sleep during a coded data transmissions based on whether they will benefit or not from overhearing the transmission. This approach saves energy while maintaining high cooperation with the NC operation. As explained earlier for the new energy-efficient MAC protocols, the sleep operation can only be realized if the (either coded or not) data transmission time is longer than the duration of the on/off radio transitions.

Also, it is worth mentioning that the analysis of the on/off radio transitions is an important aspect that has been neglected in many works available in the literature. As shown in [3–5], these transitions between on and off states require a certain time and need extra power consumption that should not be neglected, especially, the off-on or wakeup transition whose power consumption is significantly high. In this thesis, they are carefully analyzed to understand their impact on the potential energy saving of the novel MAC protocols based on low-power states.

Finally, another important aspect to be taken into account is that most of the existing works related to this thesis have been based on theoretical studies supported by computer-based simulations. Whereas theoretical models typically adopt simplified assumptions for mathematical tractability, computer-based simulations usually lack PHY-layer modeling accuracy, thus possibly leading to inaccurate results and conclusions. However, a new trend is arising recently focused on experimentally evaluating and measuring the benefits of novel strategies for green wireless networks, such as, [25, 26]. Real-world implementation can help reveal unexpected challenges to the development of new energy-efficient MAC protocols alone or combined with the NC approach and also provide new insights in the operation of communication protocols. This is one of the main motivations for the work presented in this thesis where several proposed solutions (i.e., BidMAC and BidCode) are implemented in real hardware to demonstrate that their superior performances compared to the reference mechanisms (i.e., standard DCF and COPE) can also be attained in real environments.

Specifically, among the various available wireless platforms for prototyping at the MAC layer [27], the Wireless Open-Access Research Platform (WARP) [28] has been selected because it provides a reference design that can interact with commercial Wi-Fi devices, acting as either AP or STA. The DCF MAC source code of the reference design has been modified to implement the proposed BidMAC and BidCode protocols. The focus has been put on the evaluation of energy efficiency, which has been measured in each node using Energino [8].

1.5 Dissemination of Results

All the contributions of this thesis have been published in top-level international conferences. The list of publications is presented as follows:

- R. Palacios, F. Franch, F. Vazquez-Gallego, J. Alonso-Zarate, and F. Granelli, "Experimental Evaluation of Reverse Direction Transmissions in WLAN Using the WARP Platform," in IEEE International Conference on Communications (ICC'15), June 2015, submitted - under review.

- R. Palacios, G. M. Mekonnen, J. Alonso-Zarate, D. Kliazovich, and F. Granelli, "Analysis of an Energy-Efficient MAC Protocol Based on Polling for IEEE 802.11 WLANs," in IEEE International Conference on Communications (ICC'15), June 2015, submitted - under review.

- R. Palacios, H. Haile, J. Alonso-Zarate, and F. Granelli, "Analysis of a Network Coding-Aware MAC Protocol for IEEE 802.11 Wireless Networks with Reverse Direction Transmissions," in IEEE Global Communications Conference (GLOBECOM'14), Dec. 2014, pp. 1230–1236.

- R. Palacios, E. M. B. Larbaa, J. Alonso-Zarate, and F. Granelli, "Performance Analysis of Energy-Efficient MAC Protocols using Bidirectional Transmissions and Sleep Periods in IEEE 802.11 WLANs," in IEEE Global Communications Conference (GLOBECOM'14), Dec. 2014, pp. 1269–1275.

- R. Palacios, D. Kliazovich, and F. Granelli, "Reverse Direction Transmissions and Network Coding for Energy-Efficient Wi-Fi Networks," in IEEE International Workshop on Computer-Aided Modeling Analysis of Design of Communication Links and Networks (CAMAD'14), Dec. 2014, pp. 1–5.

- R. Palacios, F. Granelli, A. Paramanathan, J. Heide, and F. H. P. Fitzek, "Coding-aware MAC: Providing Channel Access Priority for Network Coding with Reverse Direction DCF for IEEE 802.11-based Wireless Networks," in IEEE International Conference on Communications (ICC'14), June 2014, pp. 1272-1277.

- R. Palacios, F. Granelli, D. Kliazovich, L. Alonso, and J. Alonso-Zarate, "An Energy Efficient Distributed Coordination Function Using Bidirectional Transmissions and Sleep Periods for IEEE 802.11 WLANs," in IEEE Global Communications Conference (GLOBECOM'13), Dec. 2013, pp. 1641-1647.

- R. Palacios, F. Granelli, D. Kliazovich, L. Alonso, and J. Alonso-Zarate, "Energy Efficiency of an Enhanced DCF Access Method Using Bidirectional Communications for Infrastructure-based IEEE 802.11 WLANs," in IEEE International Workshop on Computer-Aided Modeling Analysis of Design of Communication Links and Networks (CAMAD'13), Sept. 2013, pp. 38-42.

- R. Palacios, F. Granelli, D. Gajic, C. Li, and D. Kliazovich, "An Energy-Efficient Point Coordination Function Using Bidirectional Transmissions of Fixed Duration for Infrastructure IEEE 802.11 WLANs," in IEEE International Conference on Communications (ICC'13), June 2013, pp. 1036-1041.

- R. Palacios, F. Granelli, D. Gajic, and A. Foglar, "An Energy- Efficient MAC Protocol for Infrastructure WLAN Based on Modified PCF/DCF Access Schemes Using a Bidirectional Data Packet Exchange," in IEEE International Workshop on Computer-Aided Modeling Analysis of Design of Communication Links and Networks (CAMAD'12), Sept. 2012, pp. 216-220.

Although not directly related to the contents of the thesis, it is also worth mentioning another contribution published in a top-level international conference:

- R. Palacios, J. Heide, F. H. P. Fitzek, and F. Granelli, "Design and Performance Evaluation of Underwater Data Dissemination Strategies using Interference Avoidance and Network Coding," in IEEE International Conference on Communications (ICC'12), June 2012, pp. 1410-1415.

Besides the main contributions of this thesis, a number of other research works have been carried out in collaboration with other researchers while this thesis was being written. These works have been focused on energy efficiency in cognitive radio, anonymous, smart grid, and cellular networks. They have been published in top-level international conferences and journals:

- Q. Wang, R. Palacios, F. Granelli, "Power Saving in Smart Meters Data Transmission via Scheduling Strategy in Future Smart Grid," in IEEE International Conference on Communications (ICC'15), June 2015, pp. 1–6, submitted - under review.
- A. Ortega, R. Palacios, F. Granelli, A. S. Akira, and C. M. Schweitzer, "Performance Evaluation of the DNP3 Protocol for Smart Grid Applications over IEEE 802.3/802.11 Networks and Heterogeneous Traffic," in IEEE International Conference on Communications (ICC'15), June 2015, pp. 1–6, submitted - under review.
- X. Pons i Masbernat, R. Palacios, C. Gruet, L. Naviner, H. Marques, J. Rodriguez, and F. Granelli, "Uplink Energy Efficiency in LTE Systems," IEEE International Workshop on Computer-Aided Modeling Analysis of Design of Communication Links and Networks (CAMAD'13), Sept. 2013, pp. 114–118.

- Q. Wang, R. Palacios, and F. Granelli, "A Novel Architecture for the Distribution Section of Smart Grid with Renewable Sources and Power Storage," IEEE International Conference on Computer Communications and Networks (ICCCN'13), July 2013, pp. 1–5.

- V. Sucasas, R. Palacios, F. Granelli, H. Marques, J. Rodriguez, and R. Tafazolli, "An Energy Efficient Cooperative Approach for Anonymous Wireless Communications," European Wireless (EW), Apr. 2013, pp. 1-6.

- S. Althunibat, R. Palacios, and F. Granelli, "Performance Optimisation of Soft and Hard Spectrum Sensing Schemes in Cognitive Radio," IEEE Communications Letters, vol. 16, no. 7, pp. 998-1001, 2012.

- S. Althunibat, R. Palacios, and F. Granelli, "Energy-Efficient Spectrum Sensing in Cognitive Radio Networks by Coordinated Reduction of the Sensing Users," in IEEE International Conference on Communications (ICC'12), June 2012, pp. 1399-1404.

1.6 Structure of the Thesis

The thesis is divided into the following chapters:

- Chapter 2 presents the background of wireless networks considered throughout the thesis, describes how to achieve energy saving through the MAC layer and how to integrate NC into the wireless network operation for high energy efficiency, and comprehensively reviews the state of the art of energy-efficient MAC protocols and NC protocols.

- Chapter 3 presents the new energy-efficient distributed MAC protocols (i.e., BidMAC and GreenBid) and provides a detailed description of each novel MAC protocol together with the theoretical analysis of the throughput and energy efficiency and a comprehensive performance evaluation by means of analysis, computer-based simulation, and real-life experimentation.

- Chapter 4 describes the new energy-efficient centralized MAC protocols (i.e., BidPoll and GreenPoll) and theoretically analyzes and comprehensibly evaluates the throughputs and energy efficiencies of the novel MAC protocols via analysis and computer-based simulation.
- Chapter 5 introduces the new NC-aware energy-efficient MAC protocols (i.e., BidCode and GreenCode) and presents a detailed description of each new approach along with the mathematical model of throughput and energy efficiency and a comprehensive performance evaluation through analysis, computer-based simulations, and experiments performed on real hardware.
- Chapter 6 concludes the thesis, summarizes the main findings, and outlines future lines of research.

Chapter 2

State of the Art

2.1 Introduction

The rapid growth of wireless networks, such as cellular networks and WLANs, in the last few decades has been possible thanks to the strong research and standardization efforts mainly driven by academia and industry to design and optimize new wireless networking solutions that support more and more sophisticated wireless services. For many years, major research efforts related to wireless networks have been focused on improving throughput, delay, and fairness or achieving some degree of QoS. However, recently energy efficiency has become a major design objective, being a hot research topic nowadays, due to the wide spread of portable wireless devices equipped with extremely limited battery capacities (e.g., smartphones). Therefore, this chapter is aimed at providing a comprehensive review of existing energy-efficient network protocols for wireless networks. The main focus has been put on the MAC layer and the NC paradigm as key elements that properly combined can significantly improve the energy efficiency of wireless networks.

The chapter is structured as follows:

- Section 2.2 describes two commonly-used wireless network architectures: infrastructure and ad hoc wireless networks. Also, a discussion of the protocol stack of a generic wireless network is included together with a brief description of each individual protocol layer.

- Section 2.3 discusses the main research challenges to achieve energy saving through the MAC layer and to efficiently integrate the NC operation into the wireless network protocol stack for further energy saving.

- Section 2.4 details the most relevant energy-efficient MAC protocols and NC protocols available in the literature and discusses their advantages and disadvantages.

- Section 2.5 concludes the chapter by summarizing the main research directions of the current scientific literature and highlighting topics that have not been properly tackled and require further research to be undertaken.

2.2 Wireless Networks: Background

This section describes the wireless network architectures considered in this thesis. Also, a discussion of the wireless protocol stack is included together with a brief description of each individual protocol layer.

2.2.1 Architectures

The reference scenario in Fig. 2.1 shows an heterogeneous wireless network deployment consisting of a UMTS/LTE cellular network with relays for coverage extension, an infrastructure Wi-Fi network (i.e., a WLAN) for indoor users and a Wi-Fi ad hoc network for opportunistic information exchange among indoor/outdoor users in the short range. Two basic wireless network architectures can be identified in Fig. 2.1: infrastructure

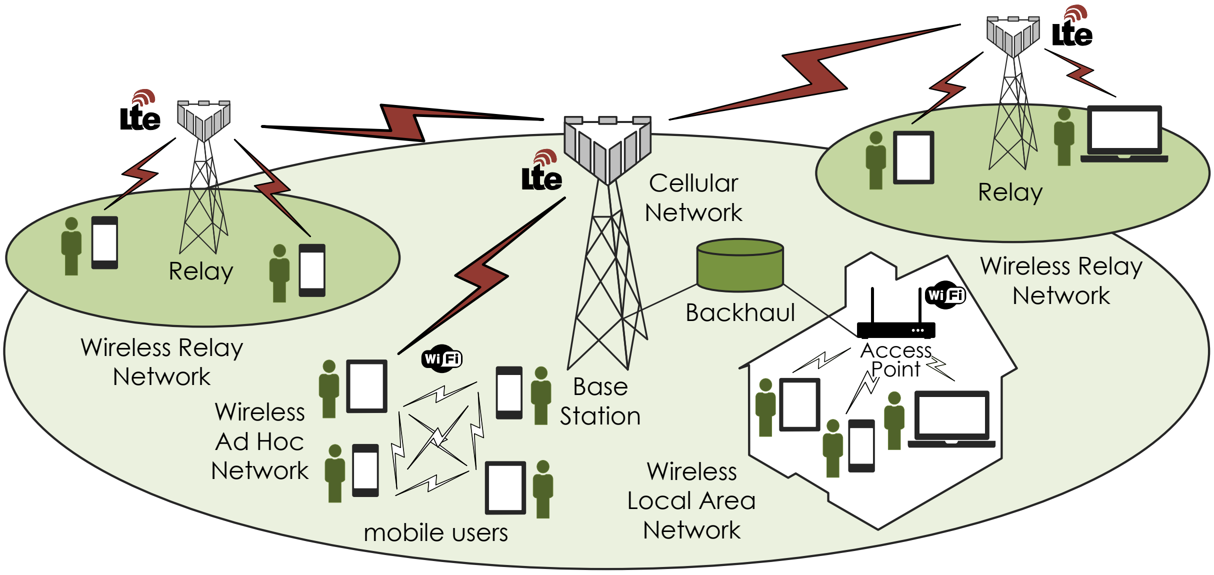


Figure 2.1: Reference scenarios: a relay-aided cellular network, an infrastructure wireless local area network, and a wireless ad hoc network

and ad hoc wireless networks. A description of each system architecture is presented below.

Infrastructure: Infrastructure wireless networks are used to extend, rather than replace, wired networks. Central nodes (i.e., BSs in LTE systems or APs in WLANs) are connected to a hierarchy of wide area and local area wired networks, which is used to provide backhaul connectivity (e.g. the backbone network or the Internet), and coordinate access to the shared wireless channels among mobile users located in their coverage area. Wireless channels may be individual frequencies in Frequency Division Multiple Access (FDMA), time slots in Time Division Multiple Access (TDMA), orthogonal codes or hopping patterns in the case of Code Division Multiple Access (CDMA), or subset of sub-carriers in Orthogonal Frequency Division Multiplexing Access (OFDMA). In general, wireless communications from and to the wired network within infrastructure networks occur in the last hop between the central node and the mobile users. However, one or several relay nodes can be placed between the central node and the mobile users to extend coverage and improve capacity, thus forming multi-hop wireless networks. In infrastructure networks, the main causes of energy consumption are associated with the operation of the central node and its capability to efficiently allocate wireless channels among mobile users.

Ad hoc: A wireless ad hoc network is a multi-hop wireless network in which a set of mobile devices exchange information by cooperatively maintaining network connectivity. This on-demand architecture does not require the help of a central coordinator and is typically characterized by no infrastructure support. Since the ad hoc environment is constantly varying, the network topology may change frequently (e.g, chain, cross, or wheel topology). Thus, monitoring network topology is a fundamental process to properly route information from source to destination across intermediate nodes. In ad hoc networks, the main causes of energy consumption are attributed to network maintenance and multi-hop communication. Due to energy constraints of network nodes, energy consumption is a critical issue to prolong the network lifetime.

2.2.2 Protocol Layers

The internal functions of wireless communication systems are implemented in software running in each wireless device. Application programs using the wireless network do not directly interact with the wireless hardware (or interface) of a device. Instead, a set of protocols organized by layers (i.e. a stack of protocol layers) interact with the wireless interface and cooperate to fulfill the requirements of application programs. Using the lower protocol layer Service Data Unit (SDU), each protocol layer performs a specific task with its Protocol Data Unit (PDU) and passes it to the upper protocol layer by removing the protocol-layer header (i.e., protocol-layer SDU). These protocol layers are independently designed, implemented, and optimized and only then interconnected to work as a

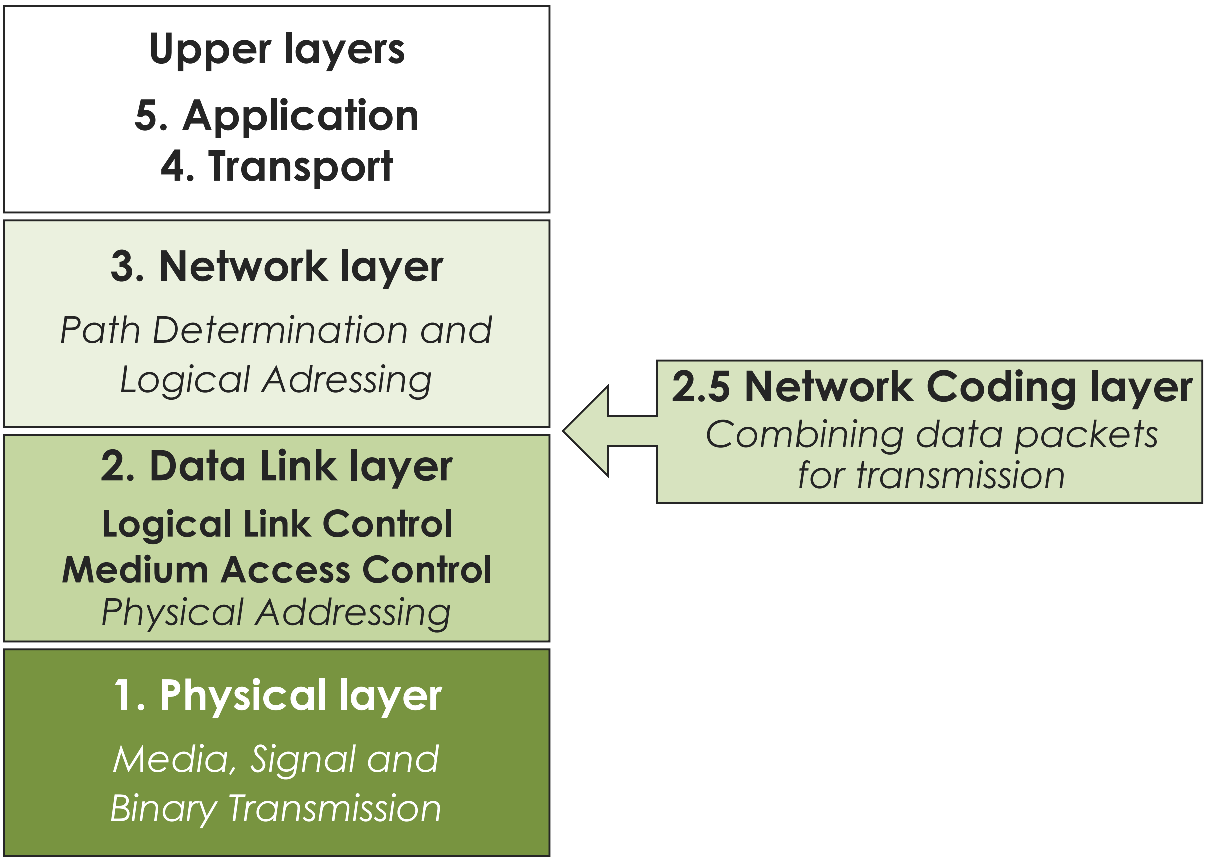


Figure 2.2: Protocol stack of a wireless network and integration of network coding

whole. This aspect allows flexibility for system designers to modify a specific layer without significantly influencing the overall performance of the protocol stack. However, introducing changes into isolated layers may no be sufficient to face new challenges in wireless networks, such as minimizing energy consumption while maintaining performance above a desired bound. New approaches need to consider energy efficiency across all layers of the protocol stack (i.e., cross-layering). Therefore, understanding the interactions across protocol layers is important to carry out cross-layer designs.

Fig 2.2 (left side) illustrates the typical protocol stack adopted in wireless networks in accordance with the ISO/OSI model specifications. The following protocol layers can be found:

- **Physical:** The PHY layer specifies the relationship between a device and the PHY transmission medium (i.e., wireless radio link) and deals with Radio Frequency (RF) signals, modulation, and channel coding.

- **Data link:** The data link layer provides reliable (and possibly encrypted for security reasons) point-to-point communications over unreliable wireless channels. The data link layer includes a MAC sublayer and a Logical Link Control (LLC) sublayer.

  - **MAC:** The MAC sublayer is responsible for managing access to the shared wireless channels among connected nodes. The IEEE 802.11 Standard (Wi-Fi) for WLANs and the IEEE 802.3 Standard for Ethernet networks are examples of MAC/PHY protocol stack architectures.

  - **LLC:** The LLC sublayer is in charge of wireless link error control and packet synchronization. For example, the IEEE 802.2 Standard defines the specifications of the LLC-layer protocol for IEEE 802.x networks.

- **Network:** The network layer is responsible for addressing, routing, and (not necessarily reliably) delivering variable length data sequences (i.e., datagrams) from a source node (possibly across intermediates nodes) to a destination node. An example of network-layer protocol in the standard Internet stack (or Internet protocol suite) is the Internet Protocol (IP).

- **Transport:** The transport layer manages end-to-end communications to provide efficient and reliable data transport between network endpoints via one or more networks. An example of a transport-layer protocol in the Internet protocol suite is the Transmission Control Protocol (TCP), built on top of IP. Also, the User Datagram Protocol (UDP) of the Internet protocol suite, built on top of TCP/IP, is commonly considered as a transport-layer protocol within OSI.

- **Application:** The application, presentation, and session layers involve a wide variety of functions that are mainly application specific (e.g., managing a session between end-user application processes, data representation, and network process to application).

2.3 Research Challenges

This section discusses the main research challenges to achieve energy saving through the MAC sublayer of the data link layer and to efficiently integrate an NC layer into the wireless network protocol stack for further energy saving.

2.3.1 Energy Saving Through the MAC Layer

The MAC sublayer of the data link layer provides addressing and channel access control mechanisms that determine the procedures to be executed by several terminals or network nodes in order to communicate within a multiple access network that incorporates one or several shared channels. The hardware that implements the MAC is referred to as a medium access controller. The primary functions performed by the MAC layer are:

- Frame delimiting and recognition.
- Addressing of destination nodes (unicast, multicast, or broadcast).
- Conveyance of source-node addressing information.
- Transparent data transfer of LLC PDUs, or of equivalent information in the Ethernet sublayer.

- Error protection by means of generating and checking frame check sequences.
- Control of access to the PHY transmission channels.
- QoS control.
- Store-and-forward switching or cut-through switching.
- Data packet queuing or scheduling.
- Acknowledgment (ACK) and retransmission procedures.

The MAC sublayer directly interfaces to the PHY layer and takes decisions on the use of the wireless interface to regulate access to the communication channel. Since the wireless interface is regarded as a major source of energy consumption for mobile devices [2, 21, 22], the MAC sublayer represents a strategic point of the protocol stack for energy consumption control and energy saving through cross-layer methods. In this sense, understanding the power characteristics of the wireless interface is important for the energy-efficient design of MAC protocols.

A wireless interface can be in one of the following five modes, as shown in Fig. 2.3:

- **Transmit:** A wireless interface in this mode acts as a transmitter to transmit packets.
- **Receive:** A wireless interface in this mode acts as a receiver to receive packets destined to itself or to other destinations (i.e., overhearing).
- **Idle:** A wireless interface in this mode is inactive or in standby (just listening) but ready to transmit and receive.
- **Sleep (or Doze):** Most of the radio hardware components of the wireless interface are turned off in this mode and it is not able to either transmit or receive any information.

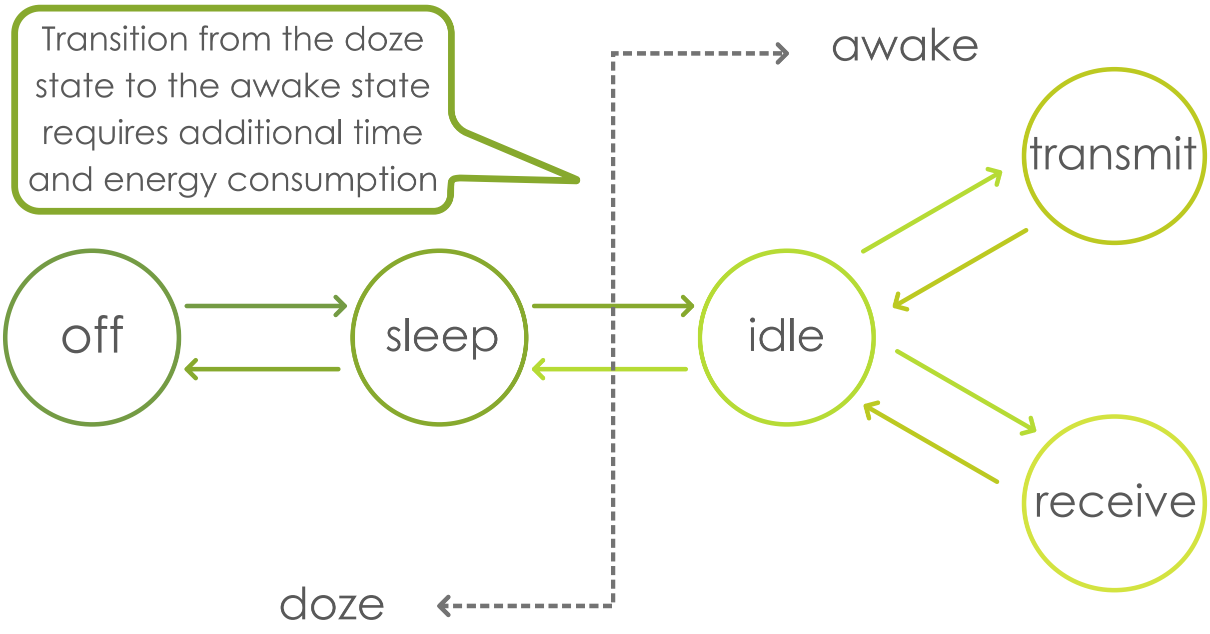


Figure 2.3: Operational states of a wireless interface

- **Off:** The wireless interface is switched off in this mode and so no power is consumed.

Fig. 2.4 reports the power consumptions of a Lucent IEEE 802.11 Wave-LAN card in all the previous modes [3–5]. As it can been seen, maximum power is consumed for transmitting (1.65 W) whereas the power consumption of the receive mode is only 15 % lower than that of the transmit mode (1.4 W). The idle mode consumes less power (1.15 W), although the power consumption is still relevant with respect to the power consumed for transmitting (only 30% lower than that in transmission). The sleep state represents the lowest power consumption mode (only 45 mW) whose power consumption is 95%, 85%, and 70% lower than transmit, receive, and idle power consumptions, respectively. Finally, it is evident that the power consumption of the off mode is zero.

Recently, some studies [2, 21, 22] have shown that the wireless interface not only consumes a significant amount of energy from mobile devices during active modes, but also in idle mode. The energy consumption (in

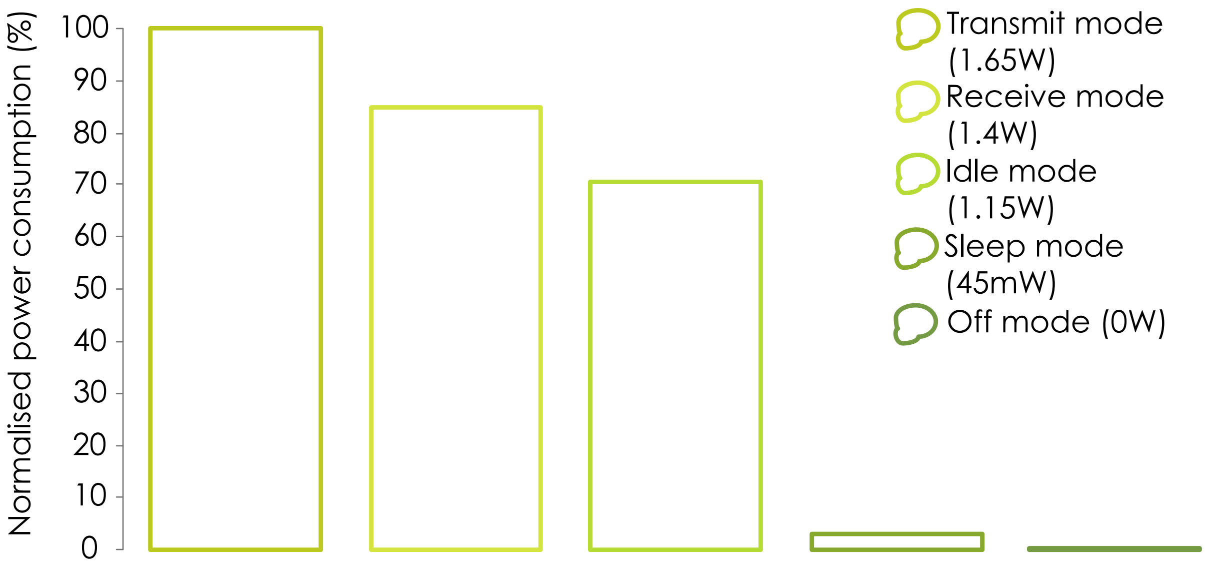
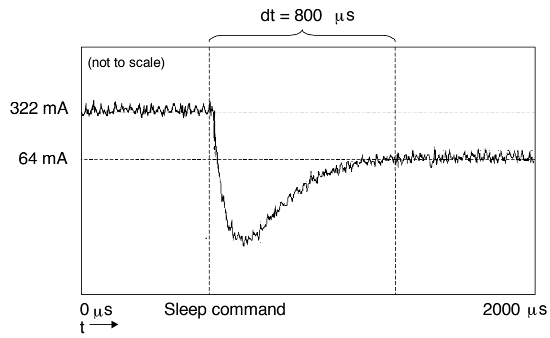


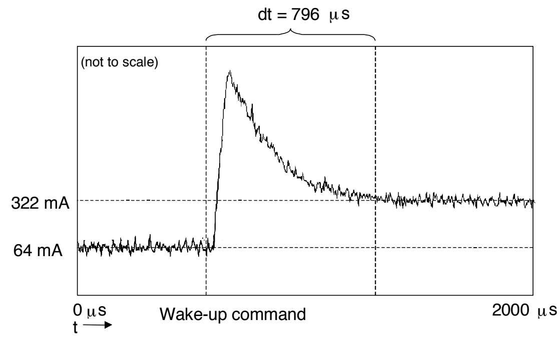
Figure 2.4: Power characteristics of a Lucent IEEE 802.11 WaveLAN card [3–5]

Joules) of a wireless interface is determined by the power consumption (in Watts) of the wireless interface during transmit, receive, idle, and sleep modes, and how long (in hours) the wireless interface remains in such operating modes. This indicates that the time, and not only the power consumption, can significantly influence the energy consumption of a lower power state like the idle mode. Therefore, the MAC protocol should be designed to minimize the time that the wireless interface operates in power consuming states, such as transmitting and receiving, and also in idle state.

Furthermore, MAC designs exploiting the sleep mode can substantially reduce the time that the wireless interface stays idle. In this case, it is important to consider the impact of the radio transitions of a wireless interface between modes. A wireless interface is unable to transmit and receive at the same time and requires additional time and consumes extra power to switch between transmit and receive states. Similarly, the transitions between idle and sleep modes, also known as awake/sleep or on/off radio transitions, have specific timing and power consumption requirements that should not be neglected, as it is highlighted in Fig. 2.3.



(a) Idle to sleep transition



(b) Sleep to idle transition

Figure 2.5: Time and power consumption during the transitions between idle and sleep modes for a WaveLAN interface [4]

Experimental measurements performed in the WaveLAN card are shown in Fig. 2.5 from [4]. As it can be seen, the power consumed during an idle to sleep transition is significantly lower than the power consumed in the sleep mode for approximately 800 μs. In contrast, a sleep to idle transition consumes significantly more energy than in the idle steady mode for approximately 796 μs. Note that the durations of both radio transitions are similar.

Therefore, the MAC protocol should also be designed to maximize the sleep period being aware of the on/off radio transitions while maintaining the desired network performance, since these radio transition introduce certain delay that cannot be neglected.

The main causes of energy consumption at the MAC layer were discussed in [29,30] and can be summarized into: collisions, control packet overhead, idle-listening and overhearing. Since the wireless channel is shared among multiple nodes, collisions may take place when two or more nodes attempt access to the wireless channel at the same time. A collision occurs due to the inability of a receiver node to receive multiple packets simultaneously. When a packet transmission fails, retransmission is required, hence increasing latency and energy consumption. Control packets are necessary to ensure the proper operation of MAC functions. However, in multi-user environments the control packet overhead, i.e. increased number of control packets, represents an important source of bandwidth and energy waste for continuous MAC operations. Monitoring channel activity to transmit or receive packets is another important stage in which idle-listening (i.e., listening to an idle channel) and overhearing (i.e., receiving packets addressed to other nodes) lead to significant energy consumption. Therefore, removing these sources of energy consumption is a primary goal to achieve energy efficiency.

Depending on the characteristics of MAC protocols, some of the energy consumption issues mentioned above may be eliminated by default whereas some others may remain inherent due to the nature of the MAC protocols. MAC protocols can be classified in different ways, depending on which of their characteristics is the focus of attention. From the point of view of where the channel access control is exercised, a possible classification is the following:

- **Centralized:** This sort of MAC protocols are based on deterministic channel accesses controlled by a master node that decides how to grant access to the wireless channel to other nodes. The main advantages of centralized MAC protocols are mentioned below:
  - Greater control to provide features like priority, overrides, and guaranteed bandwidth.
  - Simpler logic at each node.
  - Easy coordination.
  - Collisions can be completely avoided.

  Although centralized approaches may be easier to implement, they may be vulnerable to failure of the master node and reduce efficiency.

- **Distributed:** This sort of MAC protocols are based on random channel accesses coordinated in a distributed manner by all nodes, which dynamically decide which node to be granted access to the wireless channel at a given time. Distributed MAC protocols are more reliable than centralized ones. However, they are limited by collisions due to random access, hidden nodes, exposed nodes, captured nodes, and lack of channel access priority. Note that the hidden node problem occurs when a node transmits to a receiver node that is receiving data from another node out of range of the transmitter node. In addition, the exposed node problem occurs when a node is prevented from transmitting to an idle receiver node because of an ongoing transmission of a neighboring node to a receiver node that is out of range of such node and its intended receiver node.

Another possible taxonomy of MAC protocols from the point of view of how the channel access control is exercised can be the one that splits them into four categories: contention-based, round-robin, channelization-based, and reservation-based. In this thesis, the focus has been put on the MAC protocols of the first two categories, which have been used in IEEE 802.11/Wi-Fi WLANs [16]. Channelization-based MAC protocols are used in cellular networks (e.g., FDMA, TDMA, CDMA, and OFDMA). Finally, reservation-based MAC protocols are used in satellite networks, which can be centralized or distributed.

In Round-robin MAC protocols, each node of a network is given the chance to transmit by rotation. When a node gets its turn to send, it may either decline to send, if it has no data ready to be transmitted, or may send if it has got data to send. After getting a transmission opportunity, it must wait for a maximum period of time to get its turn to transmit again. The right to transmit is predetermined by a logical sequence and can be controlled in a centralized or distributed manner. Polling is an example of centralized control whereas token passing is an example of distributed control. The mechanism of polling has been used in the centralized channel access mechanism of the IEEE 802.11 Standard for WLANs, named PCF, which is described as follows.

In PCF, a central controller (or point coordinator) polls each node, referred to as an STA in the terminology of the standard, to be granted channel access by sending a poll message to its address. Although all nodes receive the message, only the addressed node responds and then it sends data or a null data message if it has no data. After sending an ACK message to the node for notification of successful data reception, the point coordinator addresses the next node to be polled in a round-robin fashion, one after the other. When the last node is polled, the point coordinator begins again the round-robin polling scheme from the first node. In a WLAN the point coordinator is usually executed in the AP, which maintains a polling list that contains the polling order and the association identifiers of the STAs of the WLAN to be polled during the round-robin polling activity. The polling order can be used to give higher priority of access to some STAs, hence ensuring some degree of QoS.

Round-robin MAC protocols work efficiently when majority of the stations have data to send most of the time. However, in situations where only a few nodes have data to send for short periods of time, round-robin MAC protocols are unsuitable. Thus contention-based MAC protocols can be used, which are suitable for dynamic traffic patterns (i.e. bursty traffic). In this case, there is no centralized control and when a node has data to send, it contends for gaining control of the wireless channel. The main advantages of this sort of MAC protocols are their simplicity and easy implementation in each node. Contention-based MAC protocols work efficiently under light to moderate load, although performance rapidly falls under heavy load. ALOHA and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) are examples of contention-based distributed MAC protocols. The CSMA/CA mechanism has been used in the distributed channel access mechanism of the IEEE 802.11 Standard for WLANs, named DCF, which is described as follows.

In DCF, the AP and the STAs of a WLAN execute a basic access mechanism based on the CSMA/CA protocol (i.e., listen before talk) in combination with a Binary Exponential Backoff (BEB) algorithm. In this mechanism, an STA having a data packet to transmit first senses the wireless channel (i.e., performs PHY-layer carrier sensing) to check whether another transmission is in progress or not. If the channel is sensed idle, the STA starts sending. If the channel is sensed busy, the STA continues to monitor the channel activity until the channel is sensed idle. During a busy channel, the STA performs virtual carrier sensing by which it updates its Network Allocation Vector (NAV) timer with the time that the channel will remain busy through control information carried in overheard packets.

After NAV expiration, the STA monitors the channel and if the channel is sensed idle it waits for a random amount of time by executing the BEB procedure. In the BEB stage, the STA selects a random value (i.e., a backoff counter) uniformly distributed within a Contention Window (CW), starting with a PHY-defined minimum. The backoff counter decrements down to zero when the channel is sensed idle. When the backoff counter reaches zero, the STA sends data and waits for an ACK from the receiver. If no response is received, the STA understands that a collision occurred and reschedules a retransmission by executing the BEB procedure. In this case, the CW doubles after each failed retransmission attempt up to a PHY-defined maximum, and is reset to a PHY-defined minimum after successful transmission. Also, there is a retransmission limit for each data packet, delimited by a retry limit and a retry counter that increments after transmission failure. When the retry counter exceeds the threshold limit, the data packet is discarded and upper layer is notified via MAC interface.

An optional collision avoidance mechanism consisting in a handshake of Request-To-Send (RTS) and Clear-To-Send (CTS) control packets can also be implemented in conjunction with the basic mechanism described above when data packets are longer than a threshold. The RTS/CTS exchange method is performed between source and destination before the transmission of data and is aimed at reducing the impact of collisions of data packets and at combating the presence of hidden nodes.

The IEEE 802.11 MAC layer specifications define a set of timing intervals for channel access control and that provide channel access priorities through different Interframe Spaces (IFSs):

- **Slot Time:** It is PHY medium dependent, derived from propagation delay, transmitter/receiver turnaround time, etc. It is the basic unit of time for MAC, e.g., the backoff time is a multiple of slot time.

- **Short Interframe Space (SIFS):** It is used for highest priority channel access, e.g., ACK and CTS, and allow Data-ACK and RTS-CTS to be automatic transactions.

- **PCF Interframe Space (PIFS):** It is used for channel access through the PCF and allows medium channel access priority, after ACKs but before contention-based access.

- **DCF Interframe Space (DIFS):** It is used for channel access through the DCF and results in lower channel access priority than using SIFS or PIFS.

- **Extended Interframe Space (EIFS):** It is used in the event that the MAC receives a packet with an error and provides an opportunity for a fast retransmission of the error packet.

A hybrid channel access mechanism is also defined in the IEEE 802.11 Standard for the coexistence of both centralized polling-based and distributed contention-based MAC protocols (PCF/DCF). The AP announces through beacons the beginning of a Contention Free Period (CFP) repetition interval whose maximum CFP duration and periodicity are specified in the beacons. The beacons are transmitted after a PIFS to allow the AP to gain control of channel access. After receiving a beacon, all the STAs update their NAVs and cannot transmit any data unless they are granted channel access or their NAVs expire and the channel is sensed idle for a DIFS. A CFP repetition interval is composed of a CFP and a Contention Period (CP). During a CFP, the AP sequentially sends poll packets (possibly combined with data) to the STAs of the polling list to grant them transmission opportunities by using the PCF. The end of a CFP is indicated by a CFP End (CE) packet transmitted from the AP. After that, the AP and the STAs enter a CP wherein they contend for channel access after a DIFS by using the DCF. Note that the duration of a CFP repetition interval must be computed to allow at least the transmission of data packet during a CP, as required for the coexistence of both time-bounded (PCF) and bes-effort (DCF) traffic.

The fact that the DCF is the mandatory channel access method of the IEEE 802.11 Standard coupled with its limitations to provide an optimum performance under heavy load conditions in densely populated WLANs have generated a lot of interest over the last years. Well-known problems related to the DCF due to CSMA/CA and BEB as the contention resolution mechanism are:

- Large overhead per MAC data packet.
- Lack of QoS guarantess (best effort).
- Hidden and exposed node problems.

- Capture effect.
- Congestion under heavy traffic loads (collisions).
- Anomaly problem due to adaptive rate (slowest stations occupy the channel for longer periods).

Thus many research works have been undertaken to deeply analyzed and optimize the performance of the DCF, focusing on improving throughput, delay, and fairness or achieving QoS.

From the point of view of energy consumption, the main limitations of DCF were summarized in [19] and are listed below:

- **Control packet overhead and IFS:** The control packets like RTS, CTS and ACK as well as silent periods such as DIFS and SIFS ensure the proper operation of the DCF. However, in multi-user environments the overheads of both control packets and IFS represent important sources of bandwidth wastage and energy consumption for continuous operation.
- **Collisions:** Since the wireless channel is shared among multiple STAs competing for access to the wireless channel, collisions may happen. They occur due to the inability of the receiver to receive multiple packets simultaneously. When packet collisions occur, packet retransmissions are required. Therefore, collisions increase latency and energy consumption.
- **Monitoring channel activity:** When the wireless channel is sensed busy, an STA with data to transmit enters the backoff stage before transmitting to avoid collisions with other transmitting STAs. The backoff time is unpredictable for each STA, since it depends on the channel activity. An STA performs continuous channel listening to decrement the backoff counter. Regardless of whether the channel

is sensed idle or busy, the STA's wireless interface is constantly on. Therefore, monitoring channel activity is another important stage in which idle listening and overhearing lead to significant energy consumption.

On the other hand, due to the fact that it is an optional channel access method of the IEEE 802.11 Standard, the PCF has received much less attention than the DCF in the literature, despite its superior performance and capability to provide QoS. Nevertheless, several research works on PCF have been undertaken over the last years to improve QoS and energy efficiency. In this sense, the main limitations of PCF were summarized in [19, 31] and are listed below:

- **Control packet overhead and IFS:** Control packets associated with the polling process such as poll and null packets and silent periods like PIFS and SIFS are required to guarantee the proper operation of the PCF, as well as the ACK packets provide reliability in data transmission. However, in multi-user environments the overhead generated by the exchange of control packets in PCF leads to bandwidth inefficiency and energy consumption for continuous operation.

- **Monitoring channel activity:** During a CFP, all the STAs need to perform constant channel listening to wait for a transmission opportunity from the AP, thus consuming significant amounts of energy to monitor incoming packets addressed to other STAs. Furthermore, when the number of active STAs is large, the last STAs need to overhear all the previous transmissions between the AP and the rest of STAs. Therefore, monitoring channel activity to transmit or receive data during the polling activity represents an important source of energy consumption.

- **Packet transmission duration:** The transmission time of an STA that has been polled by the AP is unpredictable and unconstrained. Any polled STA can transmit a data packet of any length up to a maximum length. This aspect may severely compromise the performance of other STAs of the polling list, since the AP may not be able to serve all of them in a given CFP. Therefore, additional time and energy are required for the transmission/delivery of data packets.

- **Polling list management:** Any associated STAs intending to register to or unregister from the polling list need to first send a reassociation frame to the AP during a CP. Since in CPs the STAs compete for an access to the same shared wireless channel using the DCF, additional time to perform a reassociation may be required when the channel contention increases. For an STA intending to register to the polling list, this may result in unlimited reassociation delays to obtain the contention-free service. On the other hand, for an admitted STA with no more data to transmit but having no chances to unregister from the polling list, this may degrade the bandwidth usage due to the transmission of a null data packet whenever it is polled by the AP. Furthermore, all these inefficiencies may cause unnecessary energy consumption at both the AP and the STAs of the network.

The IEEE 802.11 Standard tackles the problem of energy consumption by specifying two modes of power management for the STAs of a WLAN operating under either the DCF or the PCF: active mode and Power Save (PS) mode. In active mode, STAs maintain fully powered radio interfaces (i.e. awake state) and can transmit or receive data at any time, thus consuming significant amounts of energy. In PS mode, STAs enter a low-power doze (or sleep) state wherein their radio transceivers are switched off and they are not able to transmit or receive when in this state.

Two independent power saving mechanisms are defined in the Standard depending on whether the WLAN deployment is the infrastructure or ad hoc mode. Since most of WLANs are deployed in the infrastructure mode, the infrastructure power saving mechanism, named PSM, is described as follows.

STAs that wish to enable the PS mode should inform the AP during the association process or at any time through the power management bits in the control field of transmitted packets. In this procedure, the STAs should specify their preferred listen intervals. This interval determines the next time instant at which the STAs will awake to listen to a beacon periodically transmitted by the AP. Once the STAs receive approval from the AP through an ACK packet, they can activate the PS mode and then enter the sleep state.

During the time that the STAs remain in sleep state, the AP stores in its buffer all the data packets destined to them. Periodically, the AP broadcasts a beacon that indicates if STAs in PS mode have buffered data. This period of time is known as the beacon interval (usually, 100 ms). Periodically, STAs enter the awake state based on their listen intervals selected at the time of activation of the PS mode. The listen interval is a multiple of the beacon interval. When STAs wake up at its listen interval, they wait to receive the selected beacons. Then they read the Traffic Indication Map (TIM) field of the received beacon to determine whether the AP have buffered data packets destined to their addresses. The TIM element contains the identifiers of those STAs in PS mode with data packets buffered in the AP. If there are no data packets to retrieve from the AP, STAs can return to sleep until their next listen intervals. However, if an STA recognizes its identifier in the TIM element, it should remain awake and send PS-Poll frames to retrieve all its data packets from the AP using the standard DCF procedure, or otherwise using the PCF during a CFP without PS-Poll frames (i.e., waiting to be polled).

When operating under the DCF, STAs in PS mode transmit PS-Poll frames to the AP. Then the AP can immediately respond with data or just send an ACK packet. The transmission of an ACK packet is more feasible from a practical perspective since the AP may not be able to find the requested data packets in its queue within a SIFS. In this case, the AP will deliver the data packets as soon as possible. The AP indicates whether there are more data stored in its buffer through the More Data (MD) bit in the control field of transmitted packets. If the MD bit of the received packet is one, an STA should stay awake and send a new PS-Poll frame to retrieve the remaining data. Only when the MD bit is zero, an STA can return to sleep. In case of errors or delays during the delivery of packets, the AP may implement an aging function by which packets stored for an excessive time are deleted from the buffer.

STAs in PS mode can awake at any time to transmit data packets to the AP. They should also wake up to receive broadcast and multicast packets from the AP. The AP notifies the STAs of buffer status for these packets through the Delivery Traffic Indication Map (DTIM) subfield contained in the TIM element of specific beacons. An STA in PS mode should wake up at each DTIM interval to listen to those beacons when ReceiveDTIM is true.

Fig 2.6 illustrates an example of operation of the legacy PSM in the infrastructure mode when no PCF is operating. STA 1 has a listen interval of two beacon intervals and STA 2 has a listen interval of three beacon intervals. The DTIM interval corresponds to five beacon intervals.

In this figure, STA 1 wakes up at the DTIM interval because its ReceiveDTIM is true. STA 1 receives the beacon, reads the DTIM field, and stays awake to receive Broadcast (BC) and Multicast (MC) packets from the AP. After that, STA 1 returns to sleep until its next listen interval.

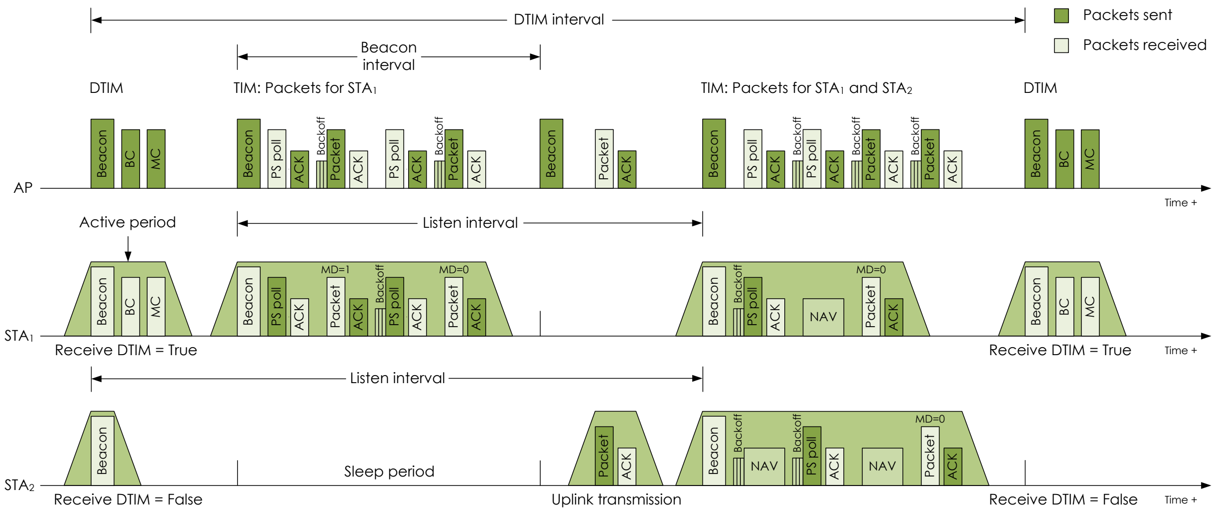


Figure 2.6: Example of operation of the PSM of the IEEE 802.11 in the infrastructure mode when no PCF is operating

STA 2 also wakes up to monitor buffer status in the AP even though its ReceiveDTIM is false. STA 2 receives the beacon, reads the TIM field, and returns to the doze state as no unicast packets are buffered in the AP for itself.

At its listen interval, STA 1 enters the awake state to receive the beacon. STA 1 identifies its identifier in the TIM element and remains awake to retrieve all its data packets from the AP. STA 1 sends a PS-Poll frame and the AP replies with an ACK packet after a SIFS. The AP performs the backoff procedure and then delivers a data packet to STA 1 with the MD bit in the control field set to one. STA 1 responds with an ACK packet after a SIFS and sends a new PS-Poll frame. After a SIFS, the AP replies with an ACK packet and after a while delivers a data packet to STA 1 with the MD bit equals zero. After a SIFS, STA 1 responds with an ACK packet and returns to sleep.

STA 2 wakes up to send a data packet to the AP and, after a SIFS, the AP replies with an ACK packet. STA 2 then returns to the doze state. Both STA 1 and STA 2 enter the awake state based on their listen intervals to receive the beacon packet from the AP. The TIM element indicates that both STAs have packets buffered in the AP. The STAs compete for access to the channel. Due to a lower backoff counter, STA 1 gains access earlier and sends a PS-Poll packet to the AP. STA 2 overhears the PS-Poll frame, reads the duration field, and updates its NAV. The AP replies with an ACK packet after a SIFS. After a DIFS, the AP and STA 2 perform contention. STA 2 is first to seize the channel and transmits a PS-Poll frame to the AP. The AP freezes its backoff counter and responds with an ACK packet. After a DIFS, the AP resumes decrementing its backoff counter down to zero. The AP sends a data packet to STA 1 with the MD field set to zero. STA 1 acknowledges it with the transmission of an ACK packet and returns to the doze state. The AP performs the backoff procedure and then sends a data packet to STA 2 with the MD bit equals zero. After a SIFS, STA 2 replies with an ACK packet and returns to sleep.

At the DTIM interval, STA 1 enters the awake state to receive the beacon. Based on the DTIM information, STA 1 stays awake to receive the buffered packets from the AP. After that, STA 1 returns to sleep. STA 2 does not need to wake up because its ReceiveDTIM is false and remains in the doze state until its next listen interval.

The fact that the PSM of the IEEE 802.11 Standard is based on periodic beacon and listen intervals and on the DCF or the PCF as the standard delivery mechanisms of downlink data leads to some inefficiencies that have been deeply analyzed and improved in the literature over the last years. The main limitations of the legacy PSM are described as follows.

- **Overhead of PS-Poll frames:** When operating under the DCF, an STA in PS mode needs to contend for channel access to transmit a PS-Poll frame that only allows the delivery of a single data packet from the AP. Therefore, when the number of STAs in the network and the traffic load are both high, the increased number of PS-Poll frames results in a waste of bandwidth and energy resources for the STAs in PS mode.

- **Beacon/listen interval dependency:** The beacon interval is fixed by the AP for all the STAs in PS mode and the listen interval is a multiple of the beacon interval. Depending on the downlink traffic characteristics of each STA in PS mode, data packets for a STA in PS mode may arrive at the AP while the STA is in the sleep state. Therefore, the dependency on the selected listen interval may lead to increased packet delivery delays or even frame dropping by the AP.

- **Energy consumption during awake periods:** STAs may awake periodically to listen to selected beacons and remain awake to retrieve buffered data from the AP or may wake up at any time to transmit uplink data. In both of these cases, the STAs cannot go back to sleep until they complete their procedures. During awake periods, the STAs in PS mode experience the same problems of energy consumption as those STAs in active mode when operating either in the DCF or the PCF (i.e., collisions, control packet overhead, idle-listening, and overhearing). Therefore, in situations of densely populated networks and high traffic loads long awake periods for delivery of downlink data and transmission of uplink data (i.e., bidirectional traffic flows) will significantly reduce the energy savings that can be achieved by using the PSM.

Later in this chapter, Section 2.4 - Subsection 2.4.1 will present a comprehensive review of existing energy-efficient MAC-layer enhancements for both active and PS modes based on the legacy DCF, PCF, and PSM channel access mechanisms and variants or derivative MAC protocols for WLANs.

2.3.2 Network Coding Integration for Energy Efficiency

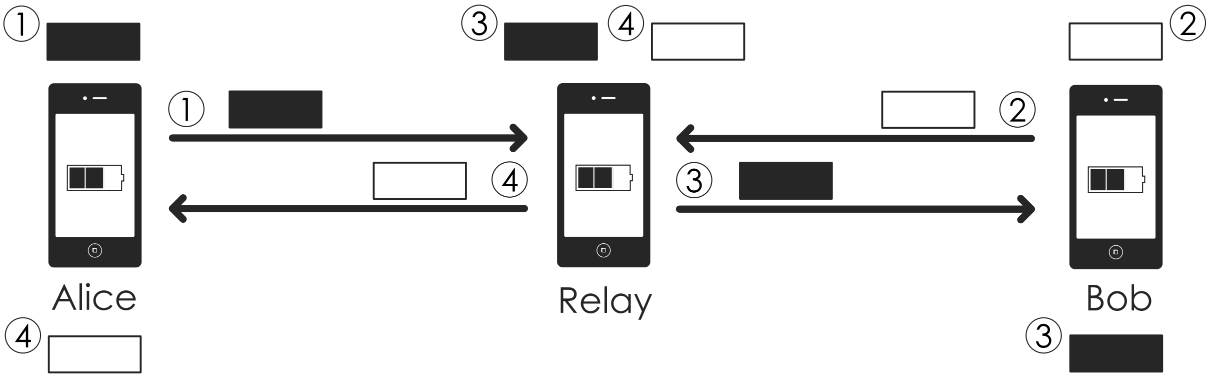
NC [17] has emerged as a new concept that breaks with the traditional operation of wireless networks. In multi-hop (or relay-aided) wireless networks, information is delivered from a source node to a destination node by routing through intermediate (or relay) nodes of the network. At the network layer of each node, the routing protocol determines and maintains the path through which information need to be routed to reach the final destination node. In simple routing schemes, each intermediate node along the computed path is simply required to store and forward the received information to the next intermediate node until reaching the end node of the path. Due to the broadcast nature of the wireless medium and the overhearing capabilities of wireless nodes, multi-hop communication and data redundancy are the main causes of energy consumption in multi-hop wireless networks.

In contrast with traditional store-and-forward routing protocols, NC exploits the broadcast nature of the wireless channel and the overhearing capabilities of wireless nodes to transmit combined information to multiple receivers simultaneously. More specifically, instead of relaying the data packets they receive, the nodes of a network take several received packets and combine them into a single coded packet for transmission. Packets are coded by applying linear coding operations (e.g, XOR) and using an encoding vector added to the header of the coded packet to perform decoding at receiver nodes, thus introducing additional overheard. Despite the coding overhead introduced, the NC operation allows increasing the information content of each transmission and reducing the total number of transmissions, hence improving throughput and energy efficiency.

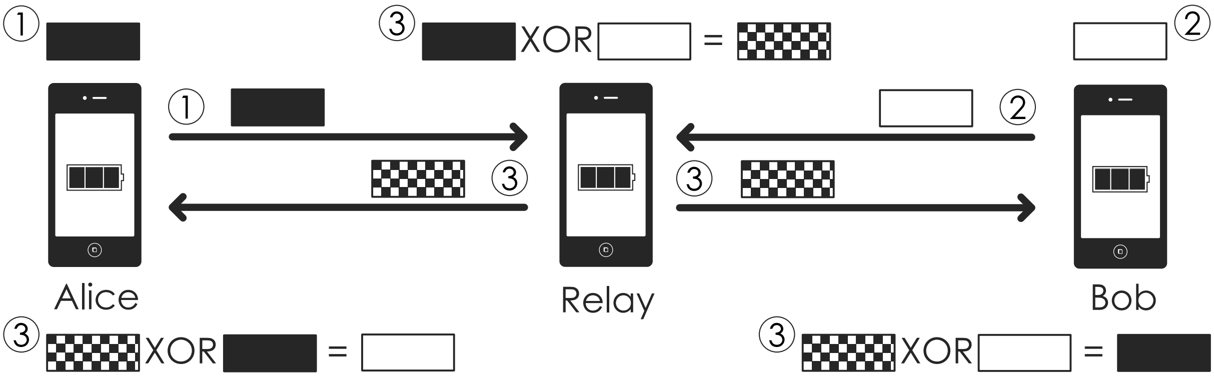
In order to show the potential advantages of NC, Fig. 2.7 describes the NC principle in a simple topology, the so-called Alice and Bob network. In this example, Alice and Bob want to exchange a pair of packets but do not hear each other. So, they need the help of a relay node to forward their packets. As shown in Fig. 2.7a for the case when NC is not used, Alice sends her packet to the relay and the relay sends the packet to Bob, who sends his packet through the relay to Alice. In total, 4 transmissions are required in order to exchange two packets between Alice and Bob. Now, consider the case with NC in Fig. 2.7b. Both Alice and Bob send their packets to the relay, which encodes the packets and broadcasts the coded version. Then, Alice and Bob can decode the packet from each other by using the received coded packet and their own packets. In this case, 3 transmissions, instead of 4, are required. Therefore, the NC approach improves the wireless throughput, since 1 transmission out of 4 can be used to send new data. In addition, NC reduces the number of collisions in contention-based MAC protocols and redundant transmissions, hence improving energy efficiency [32, 33].

NC has been extensively studied in the literature. The first work dealing with the theory of NC was presented in [17], which showed that combining multiple information flows in wireless network nodes can provide multicast capacity. Since then, NC has gained increasing attention and has been applied to multiple wireless network scenarios, showing improvements in terms of throughput, energy efficiency, robustness, and security. Although NC was originally proposed to be used at the network layer (see Fig. 2.2), in wireless networks, NC has been widely used in either the MAC layer or PHY layer. It has been shown that in both cases NC can increase the end-to-end throughput and overall network energy efficiency [23, 24].

The mechanisms of NC can be classified from the point of view of how codes used to combine packets are generated or from the point of view of which packets can be coded. In the first case, the coding operations can be linear, distributed randomized, or random, among others. The focus of this thesis is on linear NC (i.e., XOR). In the second case, encoding packets



(a) Without network coding the relay node forwards the two packets from sources nodes Alice and Bob to their respective destinations. In total, 3 transmissions are required to exchange data from end to end



(b) With network coding the relay node encodes the packets of source nodes Alice and Bob and broadcasts a single coded packet. In total, 4 transmissions are required to exchange data from end to end

Figure 2.7: An example of network coding in the Alice and Bob network

from the same flow is referred to as intra-flow (or intra-session) NC whereas encoding packets from different flows is referred to as inter-flow (or inter-session) NC. Multi-path Opportunistic Routing Engine (MORE) [34] and COPE [18] are well-known examples of intra-session and inter-session NC protocols, respectively. These inspiring works are considered as the most important implementations of NC in wireless networks. In this thesis, the focus is on inter-session NC and therefore on COPE.

COPE is the first implementation of a practical NC protocol for Wi-Fi networks (i.e., based on the IEEE 802.11 Standard). COPE defines an NC layer between the data link and network layers (see Fig. 2.2), which identifies coding opportunities to forward multiple packets in a single transmission. The authors of [18] showed that there exist important practical considerations that should be taken into account for the proper implementation of NC in currently operating wireless networks. More specifically, one of the main contributions of [18] was to show the impact of the IEEE 802.11 MAC protocol (i.e., DCF - CSMA/CA and BEB) on the performance of NC. As described earlier, this MAC protocol has been widely investigated in the literature due to several inefficiencies. But, more importantly, it presents additional limitations to efficiently work with NC, as described as follows.

Randomized channel access. The IEEE 802.11 MAC protocol (DCF) is a random channel access scheme that equally distributes channel access opportunities among all competing nodes of the network. This leads to very low end-to-end performance since congested relay nodes can only use a nearly equal share of the channel as any other node in its coverage area. Since relay nodes enabling NC provide more information in a single coded packet transmission than those that forward single packets, it would be desirable to provide a higher channel access priority for these relay nodes when they have coded packets ready to be transmitted.

Unicast reliability and collision detection. Since NC exploits the broadcast channel to forward several packets in a single transmission for multiple nodes simultaneously, multicast reliability is required to ensure successful reception and decoding of a coded packet at the intended receiver. The IEEE 802.11 MAC protocol supports both unicast and broadcast modes. In the unicast mode, data packets are immediately acknowledged with an ACK packet after successful reception. If a data packet is lost due to channel errors or packet collisions, retransmission is required. The data packet is retransmitted for a fixed number of times by following the backoff rules until a synchronous ACK is received. On the other hand, the broadcast mechanism specified in the IEEE 802.11 Standard does not provide backoff rules and reliable transmission. A broadcast packet is destined to several receivers and so it is not well specified who should reply with an ACK packet. In the absence of the ACK packets, it is not possible to detect collisions. As a result, there are no retransmissions following the backoff rules, hence leading to very low network throughput and extra energy consumption.

Simple retransmission and queue management schemes. When a collision occurs and retransmission is required, the IEEE 802.11 MAC layer is not aware whether a retransmitted packet is coded or not. Before retransmitting a packet, it is important to know if a packet can be encoded or not with other packets to increase coding opportunities while ensuring high decoding probability at the intended receivers. Therefore, efficient NC-aware retransmission schemes are very important to reduce the energy consumed for packet retransmissions. In addition, NC-aware queue management is required to give transmission priority to coded packets, since coded packets provide more information for the network than non-coded packets.

Continuous channel sensing. To increase coding opportunities and the decoding probability at the receiver nodes, NC requires that all nodes overhear all packet transmissions. This significantly increases energy consumption for the nodes since they need to consume energy to monitor channel activity and receive all the transmitted packets. Thus, it would be desirable to put some nodes to sleep for a given time while ensuring the proper operation of NC through the wireless network. This can be achieved by any of the power saving mechanisms of the IEEE 802.11 Standard (e.g., PSM). Therefore, new studies are needed to investigate the feasibility of combining the PS mode and NC for more aggressive energy savings by optimizing coding opportunities and energy consumption due to overhearing.

Indeed, NC awareness of the MAC protocol is essential for the proper NC operation. In this sense, COPE proposes a number of techniques that allow seamlessly integrating NC into the IEEE 802.11 protocol stack with no modifications of the MAC protocol. However, the results presented in [18] show that the interactions between opportunistic NC and the MAC protocol have to be carefully studied and new enhancements at the MAC layer need to be proposed to achieve high cooperation with the NC protocol layer. The COPE protocol along with MAC-layer enhancements being aware of the NC approach will be described in the next section. More specifically, they will be presented in Subsection 2.3.2.

2.4 State-of-the-art Solutions

This section describes the most relevant energy-efficient MAC protocols and NC protocols available in the literature.

2.4.1 Review of Energy-Efficient MAC Protocols

Existing MAC solutions for energy efficiency in WLANs can be classified into two categories: active mode or PS mode. In active mode, MAC solutions that reduce channel contentions, avoid IFSs and retransmission overheads, and optimize the speeds for packet transmission are proposed to minimize energy consumption. In PS mode, new approaches that enhance the PSM are designed by minimizing the contention time for an STA to retrieve packets from the AP, using scheduling of packets at the AP, or dynamically optimizing the length of each listening interval to maximize the sleep period without increasing packet delivery delays. A survey in this area can be found in [19].

Active mode

Several aspects could have substantial impact on the energy consumption of STAs actively participating in data transmission over WLANs. Energy-efficient solutions can be classified into the following three main categories, namely reducing the overhead of control packets and IFSs, reducing the number of transmissions and retransmissions, and minimizing the time for channel monitoring.

1) Reducing the Overhead of Control Packets and IFSs:

To reduce the overhead associated to the polling process in the PCF, the work in [35] introduces the SuperPoll protocol where the AP, rather than individually polling each STA, broadcasts a superpoll frame with the polling order of the STAs admitted to the polling list. This approach implies that the length of data packets of polled STAs has to be fixed for the duration of the CFP. The work in [36] tackles the limitations of [35] by providing a robust and reliable mechanism, with no additional overhead, where any polled STA includes the MAC address of the next STA to be polled into the header of its uplink packet. A modified operation of the PCF is proposed in [37], where a CFP is divided into the distributed polling protocol period for uplink transmissions, without any polling overhead, and the real-time traffic downlink period. The Distributed Point Coordination Function (DPCF) protocol was first proposed in [38] and deeply analyzed later in [39] as a novel MAC protocol combining the advantages of both the DCF and the PCF. The D-PCF aims to reduce collisions by using the polling-based access method in a distributed manner when the traffic load is high. In the D-PCF system, a reduction in the number of control packets is achieved by detecting periods of inactivity of polled STAs.

In DCF a source STA can initiate the RTS/CTS handshake before transmitting data to an intended destination STA. However, RTS/CTS packets increase the control overhead. Thus, a polling CTS [40], where the receiver STA, rather than the transmitter STA, initiates the connection, can be used to remove the RTS and increase bandwidth efficiency. The CTS polling may be inefficient in some scenarios, since polled STAs may have no data to send, which is a waste of bandwidth. To address this issue, hybrid schemes that alternate between the RTS/CTS exchange and the CTS polling are proposed in [41, 42], together with the negative CTS for dense traffic situations at the receiver.

In addition, the last amendments of the IEEE 802.11 Standard introduce new MAC techniques for specific purposes that indirectly contribute to a reduction of control packet overhead and IFSs. More specifically, the IEEE 802.11e amendment of the Standard introduces MAC enhancements for QoS guarantees through the definition of a new Hybrid Coordination Function (HCF). Two new channel access methods are defined in HCF: Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA).

The EDCA mechanism is an extension of the DCF to provide traffic prioritization for the STAs by adjusting the values of different parameters involved in the contention process according to the QoS requirements of conveyed traffic. Four Access Categories (ACs) are defined depending on the target application, namely, AC_VO for voice, AC_VI for video, AC_BE for best effort, and AC_BK for background. Based on each of these ACs, STAs contend for the channel with different access priorities. An Arbitration Interframe Space (AIFS) determines the amount of time that an STA senses the channel to be idle before backing off or transmitting. A variable CW size is used to randomly select a backoff counter during the backoff process. The transmission time of an STA when it seizes the channel is given by the duration of a Transmission Opportunity (TXOP), also known as EDCA TXOP, for each AC. In TXOP, an STA may initiate a burst transmission in which several data packets are transmitted (i.e., batch transmission). Each data transmission is separated by a SIFS and immediately acknowledged, after a SIFS, with the transmission of an ACK packet from the destination STA.

The HCCA mechanism is an extension of the PCF to provide parameterized QoS per STA by defining different Traffic Categories (TCs) and for STAs having multiple Traffic Streams (TSs) with different Traffic Specifications (TSPECs). To guarantee both per-user and per-flow QoS, the AP can initiate reserved time intervals, known as Controlled Access Phases (CAPs), in either CFPs or CPs after the channel is sensed idle for a PIFS. A CAP may include consecutive TXOPs in which the AP delivers data sequences to the STAs, also known as HCCA TXOPs, and polled TXOPs in which STAs are polled to transmit bursts of data.

Therefore, AIFSs, TXOPs, variable CWs of the IEEE 802.11e HCF channel access schemes help reduce the overheard of control packets and IFSs.

In addition, the IEEE 802.11n amendment of the Standard defines block and compressed ACKs, packet aggregation, Reduced Interframe Space (RIFS), and Reverse Direction Protocol (RDP) for high throughput, which can also reduce the overhead required for data transmission. Block ACK specifies that an STA can send one ACK to acknowledge multiple data packets, hence reducing the energy required to transmit multiple ACK frames. Compressed ACKs have a shorter lenght than normal ACKs. In packet aggregation, the basic approach is to combine several small packets into a MAC frame. Thus, only one contention and one ACK are required to convey multiple packets. A RIFS is shorter than a SIFS. Finally, in RDP the holder of a TXOP can grant part of its TXOP to the receiver for reverse data transfer (i.e., transmitted-initiated), hence providing similar advantages to batch transmission or packet aggregation. Reverse transmissions (both transmitter-initiated and receiver-initiated) have also been proposed for different purposes in [43–47].

It should also be noted that the enhanced packet aggregation scheme defined in the IEEE 802.11ac allows aggregating significantly more data than that defined in the IEEE 802.11n. In addition, the multi-channel capability of the IEEE 802.11ac through Multiple-Input Multiple-Output (MIMO) support allows Multi-User Transmission Opportunities (MU-TXOPs), where the AP can use a TXOP to deliver data to multiple STAs simultaneously. These MAC techniques also contribute to reducing the overhead of control packets and IFSs.

2) Reducing the Number of Transmissions and Retransmissions:

These solutions aim to minimize transmissions and retransmissions due to collisions or errors by using Transmission Power Control (TPC), optimizing the speeds for packet transmission, and adjusting MAC-layer parameters, such as the fragmentation threshold and the RTS/CTS threshold. In [48] the TPC approach combined with PHY-layer rate adaptation are applied to the PCF in order to determine the most energy-efficient strategy to transmit a packet. An adaptive mechanism for dynamic adjustment of the RTS/CTS threshold in the DCF is proposed in [49] to minimize average energy consumption. Link adaptation can minimize packet losses and the transmission time to save energy during packet transmissions. To identify the most energy-efficient configuration, the work in [50] introduces a cross-layer methodology that optimizes the transmission time and the transmission energy for any given signal-to-noise ratio. In [51] a game-theoretic approach is proposed to set the optimal transmission rate that maximizes reliability with minimum energy consumption. Note that some of the standard MAC-layer improvements discussed above, such as TXOPs and RDP, can also be used to reduce the number of collisions under high traffic loads.

3) Minimizing the Time for Channel Monitoring:

These approaches focus on minimizing contentions, i.e. the time that an STA needs to wait before transmitting a packet, overhearing, i.e. the time that an STA needs to monitor packet transmissions from other STAs, and conserving energy during contention and contention-free periods.

To conserve energy during the polling activity, a group-polling frame that contains the polling order of the STAs of the polling list and their assigned transmission times is proposed in [36]. It allows the STAs of the network to switch off their radio transceiver to conserve energy during most of the CFP, except for when they intend to transmit data. In [31] the Unified Point Coordination Function (UPCF) is specified, which defines a vector-list poll frame and a power-conserving scheduling that allows the STAs in PS mode to spend as less energy as possible during the polling process. UPCF is designed to addresses most of the issues of energy consumption in the PCF. However, in UPCF the last STAs of the polling list tend to overhear more time, hence consuming more energy. The Energy-Efficient Multi-Polling (EE-MultiPoll) mechanism is presented in [52]. EE-MultiPoll determines optimal wake-up intervals to fulfill a desired bandwidth utilization, hence reducing the energy consumption of STAs of later polling orders in comparison with UPCF. All these approaches refer to multi-polling frames to poll several STAs at once. Unfortunately, they may suffer from scalability limitations when the number of STAs to be polled is very large, due to the need to attach identifiers of the STAs and scheduling information to the multi-polling frames.

To save energy during channel contention, the Energy-efficient Distributed Access (EDA) mechanism is proposed in [53]. This MAC protocol is based on the DCF and allows contending STAs to enter a low-power idle mode while a packet is being transmitted (i.e., during NAV periods) and then remain in this state during subsequent backoff periods. Thus, the STAs do not perform carrier sensing to decrement their backoff counters but only wait for the backoff timers to expire and then awake to sense the wireless channel for a PIFS and transmit if the wireless channel is sensed idle. Otherwise, if the wireless channel is sensed busy, the STAs double their CWs and draw new backoff counters that exponentially increase until they seize the wireless channel. Unfortunately, the EDA scheme requires a WLAN interface implementing a low-power mode with a negligible radio transition time into the transmitting and receiving modes with respect to a packet transmission. In addition, the exponentially increased backoff mechanism without carrier sensing may cause some throughput degradation and increase delays. Different from the work in [53], an analytical framework is presented in [54] to optimize the CW size in DCF, which reduces the backoff periods in order to balance throughput and energy consumption. Similarly, the work in [55] derives the CW sizes that maximize throughput under both saturated and non-saturated conditions.

PS mode:

The STAs in PS mode need to awake at their listen intervals and contend for the channel in order to receive buffered packets from the AP. When many downlink packets must be sent to more than one STA in PS mode, increasing waiting times consume extra energy. Thus, the energy consumption of an STA in PS mode involves all issues of an active STA using DCF. In this procedure, two aspects influencing the energy consumption of a STA can be identified. The first refers to the amount of time required for an STA to successfully get access to the channel and received downlink data. The second is related to the optimization of the listening interval and the sleep period.

1) Minimizing contentions to retrieve downlink frames:

In this category, solutions minimize the time that STAs have to wait to receive downlink packets by packet scheduling at the AP based on a packet service sequence [56] and differentiation of packet transmissions for STAs in PS mode and STAs in active mode [57]. STAs in active mode may not suffer from energy constraints but are competing for the channel concurrently with the STAs in PS mode, hence forcing them to spend more time and energy during contentions.

2) Maximizing the sleep period without increasing delays:

Indeed, an STA with a longer listen interval can sleep more time and save more energy. However, a longer listen interval introduces packet delays. Previous studies in this area proposed to dynamically adjust the listen intervals to reduce energy consumption without increasing packet delays. A theoretical model, where the probabilities of an STA being in active, idle or sleeping, number of packets buffered, and average packet delay are obtained, is presented in [58]. Based on this model, a mechanism for efficient power management is designed to optimize the idle time and the sleep duration. Since packet delays depend on packet arrivals, solutions usually have to consider cross-layer effects and the characteristics of packet arrivals, such as TCP and web accesses. This includes cross-layer approaches that account for the behavior of upper layers to improve energy efficiency.

In addition, the last amendments of the IEEE 802.11 Standard introduce new power-saving QoS-constrained MAC techniques to reduce the overhead of PS-Poll frames and optimize the amount of time the STAs in PS mode spend in awake state for transmitting and receiving data. More specifically, the IEEE 802.11e amendment of the Standard introduces the Automatic Power Save Delivery (APSD) as an extension of the PSM. In APSD, the STAs in PS mode are awake during Service Periods (SPs) in which they may receive several data packets from the AP. SPs can be unscheduled or scheduled depending on whether they are initiated by the STAs in PS mode or the AP. The end of SP occurs when the AP sends a data packet whose End of Service Period (EOSP) subfield of the QoS control field is set to one. Alternatively, the AP may send a null data packet in case of having no buffered data for an STA that started an SP.

The APSD mechanism defines a distributed power saving mechanism called Unscheduled Automatic Power Save Delivery (U-APSD) for unscheduled SPs wherein EDCA can only be executed. A centralized power saving mechanism called Scheduled Automatic Power Save Delivery (S-APSD) is also defined for scheduled SPs wherein both HCCA and EDCA can be executed. The main novelty of U-APSD is to exploit the time intervals at which STAs wake up for the transmission of data packets to deliver data packets buffered in the AP. This is particularly convenient for bidirectional traffic, although alternative methods are provided for other scenarios. For example, an STA in PS mode may decide when to awake to send a null data packet that triggers the beginning of an SP for the delivery of downlink data. In contrast, the key idea of S-APSD is to schedule the time intervals at which STAs should wake up to receive data packets from the AP.

Furthermore, the Power Save Multi-Poll (PSMP) defined in the IEEE 802.11n extends the operation of APSD (both unscheduled and scheduled) by allowing the AP to begin an SP that includes an uplink and downlink transmission phase in order to minimize the awake time of the STAs in PS mode. Specifically, the AP transmits a PSMP frame addressed to those STAs in PS mode that are awake and containing a schedule of uplink and downlink transmissions for each of them. During a PSMP period, the STAs in PS mode are only awake at their assigned transmission and reception slots.

On the contrary, the Transmission Opportunity Power Save Mode (TXOP PSM) has recently been defined in the IEEE 802.11ac as a new power saving mechanism that breaks with the basic idea of PSM, APSD, and PSMP of listen intervals and beacons attaching a TIM. STAs in this PS mode may opportunistically go to sleep when the AP transmits to other STAs and when other STAs transmit to the AP (i.e., during TXOPs where they are not involved) by exploiting the virtual carrier sense mechanism (NAV). More specifically, the NAVs of STAs are set to the duration that the wireless channel will remain busy, based on the duration information carried in the duration field of overheard control and data frames. During this period of time, STAs may enter the sleep state and awake before such waiting time expires. Since the duration of data transmissions is increased by TXOPs (both batch transmission and aggregation) and MU-TXOPs, STAs in this PS mode can sleep when they listen to TXOPs where they are not involved.

Power-saving cross-layer approaches:

The characteristics of upper layers can be used to determine the duration of sleep periods. STAs can estimate the arrival of packets based on the nature of the flow to be conveyed, sleep during periods without packets, and only wake up to receive packets when they arrive.

Cross-layer methodologies are then employed to improve WLAN energy efficiency by investigating the characteristics of upper-layer packets and predict packet arrivals. As a brief overview, solutions focus on TCP traffic, web access, and Voice over Internet Protocol (VoIP). TCP connections may cause unnecessary control overhead, resulting in unnecessary transmissions and additional energy consumption. In web access, for example, if the connection speed between a web and an STA is slow, an STA may suffer from longer awake periods to retrieve the packets, thus significantly increasing energy consumption. The energy consumption of an STA with VoIP traffic is a critical issue, since it determines the maximal talking time of a mobile user.

To fix the TCP problem, the work in [59] introduced a TCP ACK at the AP on behalf of the STA in PS mode in order to remove the duplicate ACKs for TCP and MAC frames. Also, the work in [25] proposed and experimentally evaluated a Self-Tuning Power Management (STPM) that allows the STAs to dynamically switch between active and PS modes depending on access patterns and user requirements to maximize performance and/or save energy. For the problem of longer awake periods to retrieve data in web accesses, a power-aware web proxy between an STA and the Internet servers was proposed in [60]. Based on this proxy server, which catches any web page contents an STA in PS mode may request, the STA can retrieve information at higher transmission rates, thus increasing opportunities for sleeping. Finally, an algorithm in [61] is proposed to determine the sleep and wake-up intervals to conserve energy during VoIP sessions, using the end-to-end network delay and the packet loss rate.

Summary:

After discussion of the state-of-the-art MAC solutions for WLANs, a general picture consisting of blocks is drawn in Fig. 2.8. For clarity, the different research areas are classified into the categories used above. For more comprehension, a summary of the reviewed techniques together with the energy saving compared to IEEE 802.11 is also shown in Table 2.1. Finally, all the MAC-layer improvements of the IEEE 802.11 Standard across its amendments considering IEEE 802.11a/b/g/n/ac/ad for both high throughput and energy efficiency are summarized in Fig. 2.9. A complete survey on PHY/MAC enhancements for QoS and throughput is presented in [20]. A survey that describes some of the power saving mechanisms of the IEEE 802.11 Standard is presented in [19]. Unfortunately, there exists no complete survey on the evolution of power saving mechanisms in the IEEE 802.11 Standard up to date.

Energy-efficient derivative MAC protocols from IEEE 802.11

Several MAC protocols derivative from the IEEE 802.11 Standard were also proposed to improve energy efficiency in wireless networks. These are the Energy Conservation MAC protocol (EC-MAC) [62], the dominating-awake-interval protocol [63], the Dynamic Power Saving Mech-

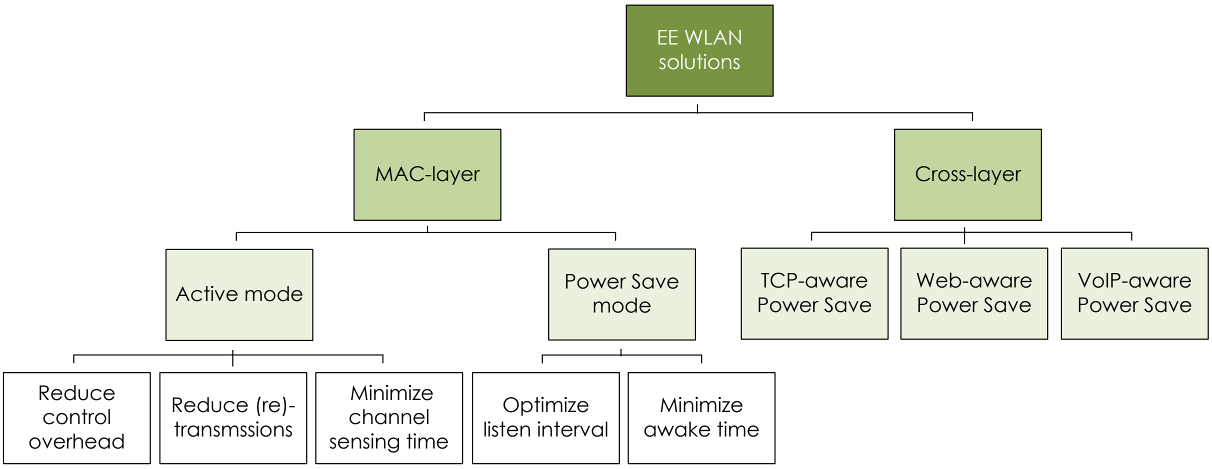


Figure 2.8: General picture of existing energy-efficient MAC designs for WLANs

Table 2.1: Summary of Existing Energy-Efficient MAC designs for WLANs

Category	Description	Energy Saving	Mode
MAC-layer	SuperPoll frame [35, 36]	10-90%	Active
	Negative CTS/CTS polling	-	
	Hybrid channel accesses [37–39]	250%	
	Reverse transmissions [43–47]	100-300%	
	TPC and link/rate adaptation [48, 50, 51]	30-60%	
	Optimize CW size [55] [54]	5-40%	
	Optimize RTS/CTS threshold [49]	-	
	Saving energy during multi-polling [31, 36, 52]	70-90%	
	Saving energy during channel contention [53]	28-80%	
	Downlink packet scheduling [56, 57]	35-50%	Power Save
	Optimize sleep and wakeup intervals [58]	28%	
Cross-layer	TCP ACK at the AP [59]	50%	
	Self-tunning power management (STPM) [25]	21%	
	Power-aware web proxy server [60]	50%	
	VoIP-aware sleep and wakeup intervals [61]	30%	

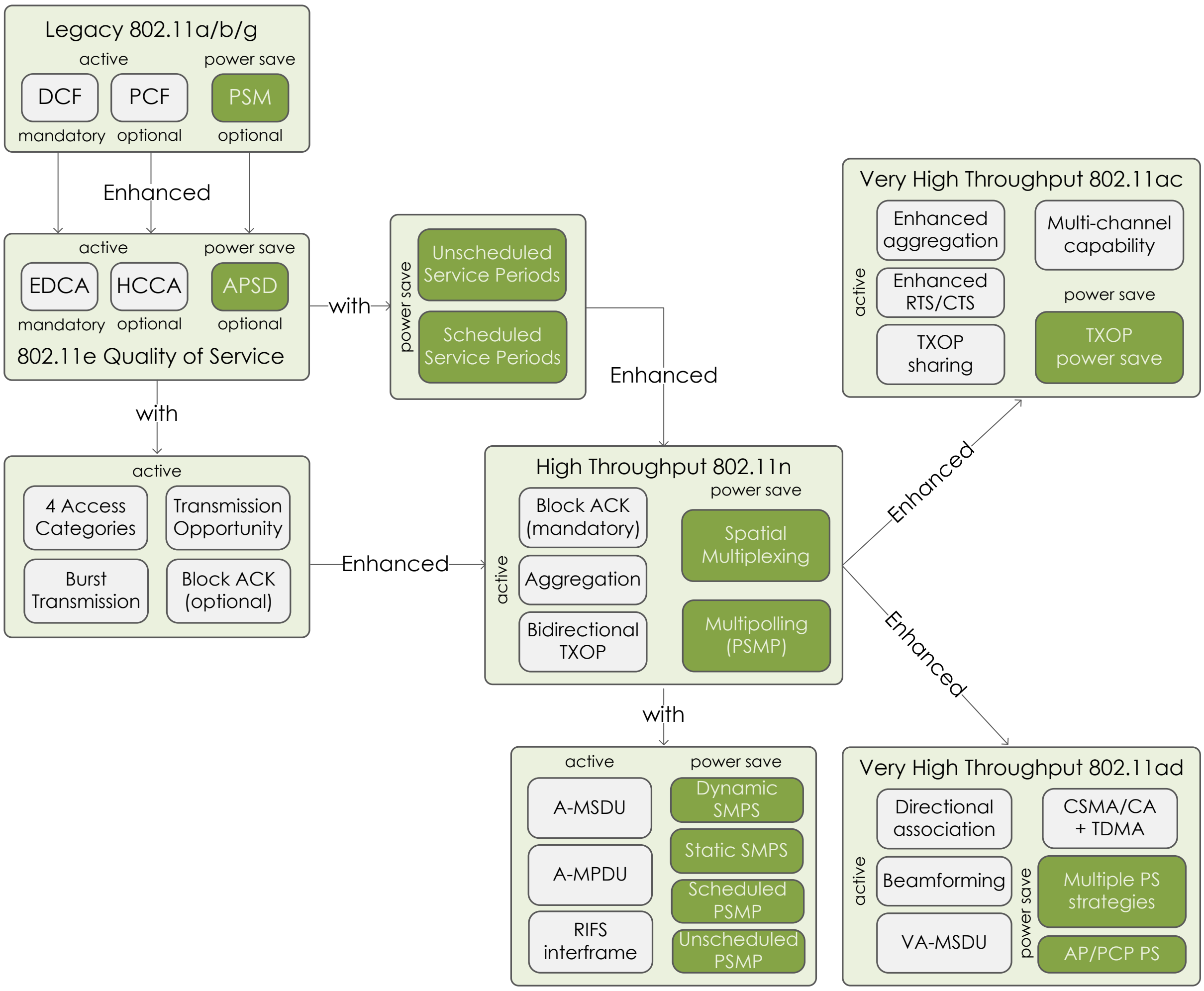


Figure 2.9: MAC-layer evolution of the IEEE 802.11 Standards towards high-throughput and high energy-efficient WLANs

anism (DPSM) [5], the Low Energy Priority Oriented Hybrid Access (LEPOHA) [64], the Neighborhood Aware Power Save Mode (NA-PSM) [65], the multi-level PSM [66], and the Improved Power Save Mode (IPSM) [67]. A brief description of these MAC protocols together with their energy savings when compared to IEEE 802.11 and their application scenarios (infrastructure or ad hoc) are summarized in Table 2.2.

Table 2.2: Variants of the IEEE 802.11 PSM MAC Protocol

MAC Protocols	Description	Energy Saving	Infrastructure	Ad Hoc
EC-MAC [62]	Combination of reservation and scheduling mechanisms	50-70%	✓	✓
Dominating-awake-interval protocol [63]	Multiple beacons and overlapping awake intervals	10-35%	✗	✓
DPSM [5]	Variable Announcement Traffic Indication Message (ATIM) window	60-75%	✗	✓
LEPOHA [64]	Polling-based/TDMA access	5-50%	✓	✗
NA-PSM [65]	Neighborhood based PSM	10-20%	✗	✓
Multi-level PSM [66]	K power levels added to PSM	40%	✗	✓
IPSM [67]	Dynamic ATIM window and piggybacking of pending packets	10-60%	✗	✓

2.4.2 Review of Network Coding Protocols

This section describes the main features of COPE [18] and data link-layer approaches that modify MAC-related functions to improve the performance of NC.

COPE has the following main features:

- **Opportunistic listening:** COPE exploits the broadcast nature of the wireless channel. Since wireless stations are usually equipped with omni-directional antennas, they can overhear packets being transmitted by other wireless stations located in the transmission range. For that, all the wireless stations enable the promiscuous mode to monitor ongoing packet transmissions and store the received packets for a limited time. This mechanism increases coding opportunities at a given relay station and also the decoding probability of coded packets at its potential receiver stations.

- **Opportunistic coding:** COPE defines a coding algorithm that combines (i.e. XOR) as many packets as possible and, at the same time, ensures that all the receivers of a coded packet are able to decode it with a high probability.

- **Neighbor state learning:** In order to provide a high decoding probability, a wireless station needs to know which packets its neighbors have stored to determine the optimal coding decision of available native packets. To provide this information, each wireless station includes in the data packet header a reception report that contains the list of currently buffered packets. In case of having no data to transmit, the reception report is periodically sent in special control packets. Additionally, COPE uses the routing computation to determine the delivery probability between each pair of wireless stations and estimate whether a neighboring station has a particular packet. This information is very useful in the absence of deterministic information. For example, when the channel contention is high packet collision occurs and the reception reports are lost frequently. Similarly, when the channel activity is low the reception reports may arrive too late, after a suboptimal coding decision has been made.

To address the limitations of the IEEE 802.11 DCF MAC protocol for the proper NC operation, COPE introduces the following mechanisms:

- **Pseudo-broadcast:** Coded packets are transmitted using the unicast mode. One of the intended receivers is chosen as the destination of the unicast packet, which generates a synchronous ACK. This provides reliability and allows the transmitter station to detect collisions and perform backoff properly. COPE adds a new header that is placed between the MAC and IP headers of the standard format packet. The

COPE header includes the whole list of receivers of a coded packet. Since all the wireless stations are in promiscuous mode, they can overhear packets not destined to them. When a wireless station receives a packet not destined to its address, it checks in the COPE header whether it is one of the intended receivers. If so, it proceeds to decode the coded packet. Otherwise, it stores the packet in its buffer as an opportunistically received packet.

- **Asynchronous ACK:** Non-coded packets are acknowledged using synchronous ACKs. Applying this mechanism to coded packets would be certainly inefficient since an ACK packet would be required from each intended receiver. As a result, the overhead of ACK packets would be significantly increased. Therefore, COPE adopts an alternative solution by which coded packets are acknowledged asynchronously. When an intended receiver of a coded packet is able to decode the coded packet to obtain its native packet, an ACK event is immediately scheduled. As soon as the wireless station has a new data packet to send, it includes all the pending ACK events in the COPE header of the transmitted data packet. In case of having no data to transmit, the ACKs can also be sent in periodic control packets as those used to send reception reports. In addition, the transmitter station of the coded packet also schedules a retransmission event for each native packet used to obtain the coded packet. If any of these packets is not acknowledged within a given period of time, the packet is retransmitted. Retransmitted packets may also be combined with other packets by following the coding algorithm previously described.

The COPE protocol seamlessly integrates the NC-layer protocol operation on top of the IEEE 802.11 MAC/PHY protocol stack without requiring any modifications of the MAC-layer protocol operation. This means that the main limitations of the IEEE 802.11 DCF MAC protocol remain inherent in the COPE protocol. These are mainly control packet overhead, collisions, contentions, and continuous channel sensing. As a result, the proper operation of the NC layer in COPE can be severely compromised due to a reduction of the coding opportunities, hence limiting the achievable throughput improvement and energy efficiency gain of NC.

A number of MAC-layer enhancements being aware of the NC approach have been proposed over the last years in order to reduce the negative effects of the IEEE 802.11 MAC protocol on the performance of NC. Available NC-aware MAC-layer solutions can be classified into three categories. In the first category, the proposed approaches manage the transmission queues to give a higher transmission priority to coded packets. The second category deals with solutions that provide a higher channel access priority for relay nodes that have coded packets ready to send by adjusting the CW size based on different network indicators. Finally, approaches that combine power saving strategies and NC are included in the third category.

1) Queue-based priority schemes:

In COPE, each station maintains a single output queue. When there is a transmission opportunity, the first packet of the queue is taken to combine it with any other packet in the queue from a different flow. If such packet exists, the two packets are combined together and the new coded packet is transmitted. Otherwise, the packet is transmitted alone. Therefore, coded packets and non-coded packets obtain an equal share of transmission opportunities.

To address this issue, the Coded Packet Priority Access (CPPA) protocol is introduced in [68]. In CPPA, coded packets are assigned higher transmission opportunities than native packets at relay wireless stations. Each wireless station maintains the queue of buffered packets. The basic idea of CPPA is that the native packets of the queue are transmitted with the h_n probability whereas coded packets are transmitted with the h_c probability, where $h_c > h_n$.

Similarly, the work in [69] proposes the Network Coding-Aware MAC level Packet Prioritization (NCAPP) scheme to give higher priority to coded packets at a relay station. In NCAPP, a relay station manages the output queue as a number of virtual queues proportional to the number of ongoing flows. The first packets of these virtual queues are checked for network coding and the new coded packets are reinserted in the output queue. When a transmission opportunity takes place, coded packets are transmitted with a higher probability than non-coded packets based on the number of coded flows of each packet in the queue. This allows increasing coding opportunities, since more packets from different sources are likely to be coded together. A round-robin scheme is then used to schedule packet transmissions from the virtual queues.

In addition, the Network Coding-Aware Queue Management (NCQAM) scheme is proposed in [70], which stores coded packets and drops packets from the flows with more packets in the queue based on both congestion and NC information to increase coding opportunities. In NCAPP, packets are dropped without any differentiation and there may be an unbalanced number of packets from each flow, thus reducing the number of coded packets. To overcome this problem, the work in [71] presents the Network Coding-Aware Priority Queuing (NCAPQ) protocol, which combines both the NCAQM and NCAPP schemes to further improve the COPE performance.

2) Channel-based priority schemes:

In COPE, each station must follow the rules of DCF to access the wireless channel. With DCF, the relay station experiences a significantly lower amount of channel access opportunities than any other station when the traffic load and the number of competing stations increase.

To reduce the influence of MAC contention on the system performance and increase the efficiency of content distribution in wireless ad hoc networks, the Popularity Aware Scheduling (PAS) approach is proposed in [72]. In PAS, the indicator of popularity measures how much data is encoded in a single coded packet. The coded packets that contain more information are more valuable, i.e. popular, for the neighboring wireless stations. Depending on the amount of encoded data in the coded packet, different levels of channel access priorities can be assigned by adjusting the CW sizes to randomly select a backoff counter. Also, in [73] Rainbow is presented as a novel MAC protocol using NC for content distribution in multi-hop wireless ad hoc networks. In Rainbow, a higher transmission priority is given to those wireless stations that are able to deliver more useful information to their neighbors. For that, each wireless station adjusts its transmission rate according to the level of innovation of own coded packets in comparison with those available in other stations. The more information a coded packet contains, the higher the transmission rate is.

To give higher transmission priority to the relay station, an autonomous mechanism that optimizes the minimum CW size based on the number of competing stations is defined and evaluated in [74]. This mechanism also helps improve the network throughput and achieve fairness at the relay station. Similarly, the work in [26] analyzes and implements in a testbed a new MAC protocol that dynamically adapts the CW size of the relay station based on the amount of traffic to be conveyed and considering the influence of NC.

In contrast with the previous approaches, a queue management approach is proposed in [75] to increase coding opportunities in multi-rate wireless networks. The key idea is to adaptively prioritize the channel access of the wireless stations located in the transmission range of a relay station based on the information available from the virtual queues of a relay station. The channel access priority is given by adjusting the CW size. A relay station suggests the minimum CW sizes that the wireless stations around it should use to balance the information content of the virtual queues of the relay station. The values of the suggested minimum CW sizes depend on the number of packets that the relay station has stored in its virtual queues and the quality of the links with its neighboring wireless stations. This information is included in the COPE header of transmitted data by the relay station to one of the neighboring wireless stations. Note that this approach can also fit well into the first category, since the channel access priority is assigned based on information related to the virtual queues of a relay station.

3) Power saving NC-aware MAC schemes:

To increase coding opportunities and the decoding probability at the receiver stations, COPE specifies that all the wireless stations should activate the promiscuous mode to overhear all the packet transmissions. This significantly increases energy consumption for the wireless stations since they need to consume energy to monitor channel activity and receive all the overheard packets. Thus, it would be desirable to put some intermediate wireless stations to sleep for a given time while ensuring the proper operation of NC through the wireless network.

The inspiring work in [76] proposes to combine NC and duty cycling for more aggressive energy savings in wireless sensor networks. Duty cycling is a technique that increases energy efficiency by allowing a node to turn off part or all of its systems for some periods of time, thus cutting idle listening and also overhearing. However, NC saves energy by exploiting overhearing. Thus, these techniques achieve energy saving by conflicting means. The focus of this work is on applications such as data dissemination or flooding where, due to the redundancy of coding, there are periods of time when a node does not benefit from overhearing coded data packets being transmitted. The proposed solution, named DutyCode, supports streaming to predict packet arrival and introduces random sleep periods using elastic intervals based on the NC operation. DutyCode is the only existing solution so far that falls into this promising area of research.

2.5 Summary and Conclusions

This chapter has reviewed the most relevant works on energy-efficient MAC protocols and NC protocols for wireless networks. In this section, a summary of existing solutions together with final conclusions on the related topics are presented for each specific area of research in the two following sections, respectively.

2.5.1 Energy-Efficient MAC Protocols

In this chapter it has been shown that the MAC sublayer of the data link layer is a central point of the protocol stack to achieve energy saving in wireless devices. The reason is that this sublayer directly interfaces to the PHY layer and takes decisions that determine how the wireless interface is used to perform channel access control.

The wireless interface not only consumes a significant amount of the limited energy resources of wireless devices for transmitting and receiving data, but also during periods of no activity (i.e., idle listening). Thus, some wireless interfaces provide a low-power sleep state by which the radio transceivers can be turned off, hence saving energy but not being able to either transmit or receive data when in this state. In this case, it is important to consider that the sleep operation requires transitions between on and off states during which the radio transceiver needs a certain switching time and generates extra power consumption. This is particularly critical during the off-on or wakeup transition, whose peak power consumption is significantly high.

Therefore, MAC protocols need to be designed to minimize the transmission and reception periods of a wireless interface and convert the idle periods into low-power (sleep) periods being aware of the on/off radio transitions. And, at the same time, they have to fulfill high-performance requirements for QoS guarantees.

The IEEE 802.11 Standard for WLANs specifies a set of MAC protocols that have been widely analyzed and optimized over the last years. These are a mandatory contention-based distributed channel access mechanism (DCF) and an optional polling-based centralized channel access mechanism (PCF). The main limitations of these MAC protocols in terms of energy consumption during active periods are the control packet overhead, collisions, and continuous channel sensing (i.e., idle-listening, and overhearing).

To address these issues of energy consumption, the Standard also defines an optional power saving mechanism (PSM) that allows the STAs to periodically alternate between awake and sleep states to listen to selected beacons containing information about data buffered in the AP. The main limitations of this MAC protocol are the overhead of PS-Poll frames and the dependency on the beacon and selected listen intervals, which may lead to some performance degradation. In addition, the STAs in PS mode experience high energy consumption during awake periods, where they may execute either the DCF or the PCF.

Existing energy-efficient MAC solutions for WLANs address the problems of energy consumption of both the DCF and PCF during active periods (or in active mode) and the PSM during low-power periods (or in PS mode).

In active mode, the proposed MAC schemes aim at reducing the overhead of control packets and silent periods (IFS), reducing the number of transmissions and retransmissions, and minimizing the time for channel monitoring. Relevant MAC techniques in this area of research are: *(i)* hybrid and reverse channel accesses, which combine both contention (DCF) and contention-free periods (PCF) in a distributed manner; *(ii)* multi-polling frames, which allow the STAs to be polled to only awake at their assigned transmission and reception slots; and *(iii)* low-power contention periods, which allow contending STAs to enter a low-power idle state with a negligible radio transition time into transmitting and receiving state with respect to a packet transmission.

In addition, other interesting MAC techniques defined in the subsequent amendments of the Standard that can reduce the energy consumption of STAs during active periods are: *(i)* IEEE 802.11e EDCA and HCCA batch transmissions, which allow sending a burst of data frames in a single TXOP; *(ii)* IEEE 802.11n frame aggregation, block ACK, and RDP, which allow exchanging multiple bidirectional data frames aggregated in single MAC frames and acknowledged by the receivers once for all aggregated frames within a single TXOP; and *(iii)* IEEE 802.11ac MU-TXOP, where the AP can deliver data to multiple STAs simultaneously within a TXOP.

In PS mode, the proposed solutions minimize the time that STAs have to be awake to retrieve downlink frames and maximize the sleep period based on prediction of packet arrivals from upper layers, such as TCP and web access. Relevant power saving MAC strategies in this area of research are: *(i)* downlink packet scheduling based on packet service sequence; *(ii)* differentiation of packets transmissions between STAs in PS mode and STAs in active mode; and *(iii)* TCP ACK at the AP, which allows removing duplicate ACKs for TCP and MAC frames.

Also, additional interesting power saving MAC strategies defined in the subsequent amendments of the Standard that reduce the overhead of PS-Poll frames and optimize the awake time of STAs in PS mode are: *(i)* IEEE 802.11e APSD, which allows the STAs in PS mode to initiate SPs where they can retrieve burst of buffered data from the AP at any time through a trigger frame or at fixed intervals (S-APSD); *(ii)* IEEE 802.11n PSMP, which allows the STAs in PS mode to only awake to transmit and receive data within an SP initiated by the AP through a PSMP frame; and *(iii)* IEEE 802.11ac TXOP PSM, which allows the STAs in PS mode to opportunistically go to sleep when a packet is being transmitted in the wireless channel. Note that, while PSM, APSD, and PSMP have generated a lot of interest in recent years, TXOP PSM has received very little attention, despite its potential advantages to significantly improve the energy efficiency of STAs during awake periods in densely populated WLANs.

To conclude the assessment of the state of the art on energy-efficient MAC protocols for WLANs, it is worth mentioning that, despite the strong efforts of both research and standardization communities, none of the existing MAC solutions jointly address all the problems of energy consumption during both active and low-power periods and, at the same time, are able to improve the overall WLAN performance. For example, most of the proposed solutions in active mode do not solve the problems of idle listening and overhearing, hence suffering from yet low energy efficiency. In contrast, those that solve these in active mode by enabling low-power state periods may introduce some performance degradation, additional delays, and scalability limitations when the traffic load and the number of STAs in the network are both high. Similarly, most of the proposed solutions in PS mode do not solve the problems of energy consumption when the STAs in PS mode need to be awake to transmit or receive data, and may also introduce performance degradation, additional delays, and scalability limitations.

Therefore, new win-win (i.e., high-throughput energy-efficient) MAC protocols need to be investigated aiming to boost the throughput and energy efficiency of WLANs when either the standard DCF, PCF, or PSM are executed. Furthermore, the design of such MAC protocols need to account for the time and power consumption of the on/off transitions of radio transceivers, as they have shown to be critical when enabling short low-power periods, e.g., based on the IEEE 802.11ac TXOP PSM. Alternative methods that could be used in conjunction with the TXOP PSM are batch transmissions (i.e., TXOPs), reverse transmissions (e.g, RDP), and frame aggregation to increase the duration of data transmissions, hence enabling the TXOP operation and increasing both throughput and energy efficiency.

All these concepts and ideas will be investigated in the next two chapters. More specifically, Chapter 3 will present, analyze, and implement new high-throughput energy-efficient distributed MAC protocols based on the combination of reverse transmissions (receiver-initiated) and power saving through TXOP PSM on top of the standard DCF. Also, Chapter 4 will present and evaluate new high-throughput energy-efficient centralized protocols based on the combination of the TXOP PSM and standard PCF along with additional novel strategies.

2.5.2 Network Coding Protocols

This chapter has also shown that the NC paradigm can help improve the energy efficiency of wireless devices by letting them combine multiple received packets for transmission, with marginal overhead due to coding. The reason is that this operation leads to a reduction of the number of channel accesses, hence reducing the amount of energy consumed per delivered bit of information.

Different approaches of NC have been proposed, which can be classified in different ways. From the point of view of which packets can be used for coding, they may encode packets from different flows/sessions (intersession NC) or from the same flow/session (intra-session NC). COPE is the first packet-oriented forwarding architecture that implements an inter-session NC protocol over Wi-Fi networks (IEEE 802.11). COPE seamlessly integrates an NC layer between the data link (MAC) and network (IP) layers of the protocol stack based on the IEEE 802.11 MAC/PHY layer specifications. This new layer identifies coding opportunities to forward multiple packets from different sources in a single transmission. The main features of COPE are: *(i)* opportunistic listening, which allows the nodes to store all overheard packets for a limited time; *(ii)* opportunistic coding, which allows the nodes to take several received packets and combine them for transmission; and *(iii)* neighbor state learning, which allows the nodes to tell their neighbors what packets they have stored to increase coding and decoding opportunities.

Through COPE it has been shown that NC awareness of the MAC protocol is essential for the proper NC operation. Specifically, the IEEE 802.11 MAC protocol (DCF), which has been widely investigated over the last years for several reasons, presents some limitations to efficiently work with NC. These are: *(i)* lack of per-node and per-packet channel access priority for NC, *(ii)* lack of reliable and collision avoidance mechanisms to broadcast coded packets, *(iii)* lack of retransmission schemes aware of NC, and *(iv)* need for continuous channel sensing for coding and decoding opportunities (idle listening and overhearing).

COPE addresses some of these issues by introducing two mechanisms that do not require any modifications of the IEEE 802.11 MAC protocol, which are: *(i)* pseudo-broadcast, which allows the nodes to send coded packets to one of the intended receiver nodes and specify the whole list of receiver nodes in a new header added to the coded packet; and *(ii)* asynchronous ACK, which allows the nodes to acknowledge successfully decoded packets through a new header attached to transmitted data packets or through periodic control frames. Unfortunately, COPE still shares most of the limitations of the IEEE 802.11 DCF MAC protocol, which are control packet overhead, collisions, contentions, and continuous channel sensing.

Existing MAC-layer solutions being aware of the NC approach cope with the negative effects of the IEEE 802.11 MAC protocol on the performance of NC mainly by three different approaches. These are: *(i)* queue-level priority access, which provides a higher transmission priority for coded packets in the queue; *(ii)* channel-level priority access, which provides a higher channel access priority for nodes that have coded packets ready to send by adjusting the CW sizes based on different network indicators; and *(iii)* low-power overhearing, which allows the nodes to cut overhearing by duty cycling with limited performance degradation of NC.

To conclude the assessment of the state of the art on NC protocols for Wi-Fi networks, it should be noted that, despite the strong research efforts, none of the existing NC-aware MAC solutions jointly address all the cross-layer issues of NC with IEEE 802.11 MAC in order to improve both the throughput and energy efficiency of wireless networks. For example, the solutions based on queue-level prioritization are limited by the fair channel access distribution of the IEEE 802.11 DCF MAC protocol among all competing nodes. Also, those based on channel-level prioritization can only provide higher channel access priority on average, i.e., probabilistic or relative (not absolute) and, as a result, they cannot guarantee immediate channel accesses for congested relay nodes that have coded packets ready to be transmitted. Furthermore, none of these approaches minimize the time that the nodes spend in idle listening and overhearing in order to achieve further energy savings. On the other hand, those that solve this problem by enabling sleeping periods in conjunction with NC may introduce performance degradation and additional delays.

Therefore, new NC-aware energy-efficient MAC protocols need to be investigated aiming to boost the throughput and energy efficiency of Wi-Fi networks when the NC operation is executed. Existing standard techniques that can be used for these purposes and have received little attention so far in combination with NC are batch transmissions (i.e., TXOPs), reverse transmissions (e.g., IEEE 802.11n RDP), frame aggregation, and low-power overhearing periods (e.g., IEEE 802.11ac TXOP PSM) being aware of the on/off radio transitions.

All these concepts and ideas will be investigated in the last main chapter of this thesis, Chapter 5. More specifically, this chapter will present, analyze, and implement new NC-aware energy-efficient MAC protocols based on the combination of reverse transmissions (receiver-initiated) and power saving through TXOP PSM on top of the standard DCF in conjunction with NC (i.e., COPE-based).

Chapter 3

Energy-Efficient Distributed MAC Protocols

3.1 Introduction and Related Work

Currently, most of WLANs are based on the MAC and PHY layer specifications of the IEEE 802.11 Standard [16]. In a typical WLAN deployment (see Fig. 3.1), an AP and several STAs compete for access to the shared wireless channel, i.e., a TXOP, using a mandatory distributed contention-based access method called DCF.

The basic access rules of the DCF for both the AP and the STAs are to sense the wireless channel before transmitting to an intended receiver and back off during a random period of time for collision avoidance, when the wireless channel is sensed busy. The backoff period exponentially increases after transmission failure to resolve collisions. Upon initial transmission or subsequent retransmission, reception of a positive ACK after a short period of no transmission (i.e., silent period) indicates transmission success. Therefore, collisions, backoff periods, and the overheads of ACK frames and silent periods are the main problems of energy efficiency when the DCF is executed, as shown in Fig. 3.1.

The STAs of a WLAN operating in the DCF mode can choose between two modes of power management. In active mode, STAs remain

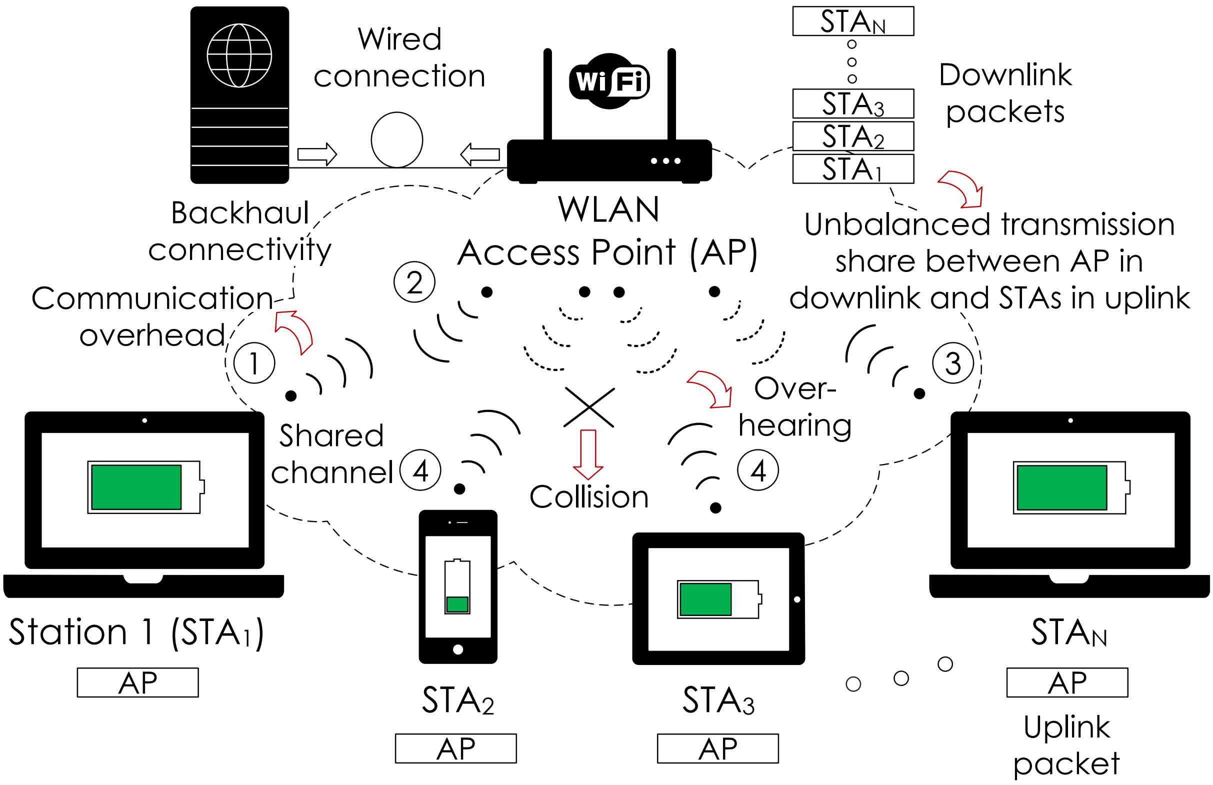


Figure 3.1: Problems of energy efficiency in an infrastructure WLAN operating under the legacy DCF

in an awake state where their radio transceivers are always switched on, thus continuously listening to the wireless channel (being ready to either transmit or receive data) and consuming significant amounts of energy for powering their radio transceivers during idle periods (i.e., idle listening) and when receiving packets addressed to other destinations (i.e., overhearing). In PS mode, instead, STAs enter a low-power doze (or sleep) state wherein their radio transceivers are turned off. This yields energy savings at the cost of not being able to either transmit or receive when in this state.

Typically, the STAs operating in PS mode alternate between awake and sleep states periodically to listen to selected beacons broadcasted periodically by the AP (every listen interval is negotiated with the AP). These beacons inform them about data buffered in the AP through a TIM. This TIM contains the list of identifiers of the STAs that must remain awake until the AP delivers all their buffered data. In the PSM specified in the original version of the IEEE 802.11, STAs retrieve buffered data from the AP by transmitting PS-Poll frames using the DCF (each PS-Poll frame is used to retrieve a single data frame). In addition, STAs may also wake up at any time to transmit data. Therefore, the overhead of PS-Poll frames, the long contention periods to retrieve buffered data from the AP, and packet dropping by the AP under high traffic conditions represent the main causes of throughput degradation, increased packet delivery delays, and extra energy consumption of the STAs in PS mode.

Many research works available in the literature have proposed MAC-layer enhancements addressing the problems of energy efficiency of DCF during both active and low-power periods (i.e., PSM) [19, 29]. Along the various amendments of the Standard, different methods backwards compatible with the PSM have also been specified to reduce the amount of PS-Poll frames and optimize the amount of time that the STAs in PS mode spend in awake state for transmitting and receiving data. For example, the APSD defined in the IEEE 802.11e is a mechanism for the delivery of downlink data buffered in the AP. STAs enabling APSD decide when to awake to transmit a trigger frame, similar to the PS-Poll but possibly combined with data, that initiates an SP wherein the AP delivers a burst of buffered data (i.e., a batch transmission) to them.

Furthermore, the PSMP defined in the IEEE 802.11n extends the operation of APSD by allowing the AP to begin an SP that includes an uplink and downlink transmission phase in order to minimize the awake time of the STAs in PS mode. Specifically, the AP transmits a PSMP frame addressed to those STAs in PS mode that are awake and containing a schedule of uplink and downlink transmissions for each of them. They only awake at their assigned transmission and reception slots.

PSM, APSD, and PSMP are all based on the same concept of periodic beacons and listen intervals. Although APSD improves some of the limitations of PSM and PSMP improves some of the limitations of APSD, all these PS mechanisms do not work optimally when there exists a large number of STAs with high amounts of bidirectional traffic in the network. This is due to the need to attach identifiers to the beacons, thus suffering from scalability limitations, and the dependencies on the beacon and listen intervals, which may cause performance degradation and additional energy consumption for the STAs.

On the contrary, the TXOP PSM mechanism recently defined in the IEEE 802.11ac is not based on listen intervals and beacons attaching a TIM. STAs in this PS mode may opportunistically go to sleep when the AP transmits to other STAs and when other STAs transmit (i.e., during TXOPs where they are not involved) by exploiting the virtual carrier sense mechanism. More specifically, the NAVs of STAs are set to the duration that the wireless channel will remain busy, based on the duration information carried in the duration field of overheard control and data frames. During this period of time, STAs may enter the sleep state and awake before such waiting time expires. In this case, the available time for sleeping (i.e., the total data transmission time or TXOP duration) must allow the STAs to go to sleep and awake taking into account the duration of the on/off transitions of radio transceivers.

TXOP PSM could significantly improve the energy efficiency of STAs in highly dense networks and with heavy traffic conditions, while also being able to be used in conjunction with other PS mechanisms when the number of STAs and the traffic load in the network are both low. Unfortunately, the regular operation of the DCF may not facilitate the TXOP PSM operation. Typically, a TXOP is reserved/granted for the transmission of a single data packet. Therefore, depending on the duration of the TXOP, which depends on the data length and the data transmission rate, and the duration of on/off radio transitions, which depend on the hardware implementation and may be in the order of hundreds of microseconds [3–5], it may not be possible for a third STA to go to sleep during the transaction.

In order to cope with this limitation, the inspiring work in [53] proposed a new mechanism called EDA that is based on the DCF and exploits a low-power idle state with a very short transition time into transmitting/receiving such that it can be considered as negligible with respect to a packet transmission. The authors selected the Socket Mobile CF WLAN card [77] as a commercial WLAN product that fulfills these requirements. They affirmed that such card provides an idle state characterized by a power consumption of 0.066 W (i.e., 14 times and 9 times lower than the card consumption in transmission and reception state, respectively) and a transition time into transmitting/receiving of 20 μs.

However, it has not been possible to verify what the authors claimed regarding the specific value of the transition time because the resource they cited is currently unavailable on the web. In addition, the datasheet found [77] does not specify the transition duration, only the power consumption of the low-power idle state. The power consumed during the transition from idle to transmitting/receiving is also not specified in [53]. As shown in [3–5], the transition from a low-power state (in this case the sleep state) to a high-power state (in this case the idle state) produces a power peak that consumes significantly more power than the high-power state switching into and so that should not be neglected.

According to the EDA scheme, contending STAs can enter the low-power idle state, after setting their NAVs, while a packet is being transmitted in the wireless channel (in a similar way to TXOP PSM) and are required to remain in this state during the entire backoff period, upon NAV expiration. Then, they sense the wireless channel before transmitting and, if the wireless channel is sensed busy, return to the low-power idle state and wait for an additional random backoff period that exponentially increases until they seize the wireless channel. Results presented in [53] showed that the EDA mechanism can achieve energy savings up to 80% and 28% under UDP and TCP traffic, respectively, when compared to the standard DCF.

Unfortunately, the EDA scheme has two important limitations. Firstly, the fact that EDA requires a radio interface with a low-power state characterized by a very short transition time into transmitting and receiving states introduces a dependency between radio hardware design and energy saving that can be achieved with EDA. Thus, for example, if an STA implements EDA at the MAC layer but the low-power sate of its radio interface does not comply with the timing requirements of EDA, then the STA may end up operating as in the standard DCF mode when a packet transmission is shorter than the radio transition time of the STA. In such a case, EDA would provide no energy savings. Secondly, the fact that EDA requires contending STAs to remain in the low-power state during backoff periods that exponentially increase until getting access to the wireless channel results in throughput degradation and increased access delays due to not being able to perform carrier sensing and receive any data packets when in this state.

Therefore, the EDA mechanism does not represent a general solution to efficiently implement the TXOP PSM strategy taking into account the diversity of STAs with different radio profiles. Instead of identifying a radio transceiver that fulfills the requirements of a given MAC protocol design, an optimal approach should be aware of the radio requirements of STAs and adapt to maximize the efficiency of the TXOP PSM mechanism. For example, this can be achieved by extending the data transmission time according to the timing requirements of the on/off radio transitions of

STAs. In this sense, there exist well-known techniques that could be used for this purpose.

Recently, the use of Reverse Direction (RD) transmissions has been proposed in the IEEE 802.11 Standard to improve the throughput and energy efficiency of WLANs. More specifically, the RDP has been defined in the IEEE 802.11n as a MAC layer enhancement of the legacy DCF to increase channel utilization. The RDP breaks with the basic operation of the DCF where an STA gains a TXOP by competing to get access to the wireless channel in order to transmit data to one arbitrary destination (i.e., unidirectional data flow). In RDP, the holder of a TXOP, once it has seized the channel, can allocate the unused TXOP duration to one or more receivers in order to allow data transmissions in the reverse link (i.e., reverse direction or bidirectional data flow). For scenarios with bidirectional traffic, this approach is very convenient as it reduces contention in the wireless channel.

The concept of reverse direction (or bidirectional) transmission in WLANs was first introduced by [43], prior to the standardization of the RDP. Since then, several works have proposed similar approaches with different purposes. Existing RD-based protocols can be classified into two categories: *(i)* proactive, i.e. RD exchange sequence initiated by the transmitter, or *(ii)* reactive, i.e. RD exchange sequence initiated by the receiver. Proactive RD protocols [46, 78] allow the transmitter to grant the receiver the remaining time of its TXOP for reverse data transfer, in a way similar to RDP. On the other hand, reactive RD protocols [43–45, 47] allow the receiver to reserve the wireless channel for a backward transmission by extending the transmitter's TXOP time, without needing to compete for the channel. This sort of RD protocols can achieve higher performance in some scenarios because they are more adaptive to the actual needs of a network.

In particular, the inspiring work in [47] investigates the feasibility of reactive RD exchange operation in infrastructure WLANs, wherein an AP is connected to a cable network infrastructure and provides wireless Internet access for a number of STAs in its coverage area. Results show that reactive RD approaches can effectively address the unbalanced operation of DCF between uplink and downlink traffic when traffic flows are highly bidirectional. Indeed, DCF provides equal channel access opportunities for all STAs, including the AP. Therefore, the AP only receives an equal share of the wireless channel to deliver downlink traffic to all the STAs, while it has data to transmit to all of them. Note that the case when all STAs route all their traffic through the AP is considered. Thus, by allowing the AP to dynamically initiate RD exchange sequences when receiving data from the STAs, uplink and downlink transmission opportunities can be better balanced, hence improving the overall WLAN performance. Furthermore, the reactive RD operation extends the data transmission time and can be used to allow STAs to efficiently implement the TXOP PSM mechanism taking into account the on/off transitions of radio transceivers.

Motivated by the discussions above, this chapter presents two new energy-efficient MAC protocols, named BidMAC and GreenBid. BidMAC enables reactive RD transmissions between the AP and the STAs with a single channel access invocation, in a way similar to Bidirectional Distributed Coordination Function (BDCF) proposed in [47]. However, an important difference between BidMAC and BDCF is that in BidMAC a reactive RD exchange sequence may include multiple rounds of bidirectional data transmissions between the transmitting STA and the AP or between the AP and several receiving STAs. Moreover, the AP may initiate a multi-sender/receiver RD exchange sequence where the AP and multiple STAs can exchange data in both directions in a contention-free manner. Then, GreenBid extends the BidMAC operation by exploiting the longer duration of bidirectional transmissions to allow those STAs not involved in the communication to go to sleep, in a way similar to TXOP PSM and EDA [53]. In contrast with EDA, GreenBid can achieve energy saving with longer on/off radio transition times by prolonging the time of data transmissions, and not only improve energy efficiency but also the overall network throughput.

It is important to mention that, based on the comprehensive assessment of the state of the art, the work presented in this chapter can be considered as the first research work that investigates the idea of combining RD transmissions and opportunistic sleeping periods through TXOP PSM for high-throughput high-energy-efficient WLANs based on the IEEE 802.11 Standard.

A preliminary description and performance evaluation of BidMAC by means of computer-based simulations have been presented in [79]. A detailed description and comprehensive performance evaluation of BidMAC via computer-based simulations have then been published in [80]. In addition, GreenBid has been introduced and evaluated through computer-based simulations in [81], where BidMAC has been considered for the purpose of comparison with GreenBid. Then, the performance analyses of BidMAC and GreenBid in terms of throughput and energy efficiency have been presented and validated through computer-based simulations in [82]. Finally, an experimental implementation of BidMAC using the 802.11 PHY/MAC reference design of WARP has been described and evaluated through both analytical and experimental results in [83].

The structure of this chapter is detailed as follows.

- Section 3.2 provides an overview of the legacy DCF MAC protocol and comprehensively describes the proposed BidMAC and GreenBid MAC protocols.

- Section 3.3 analyzes the maximum achievable throughputs and energy efficiencies of the protocols under consideration using a simplified approach and through a Discrete Markov Channel model from [84, 85].
- Section 3.4 describes the implementation of the protocols in a Python simulation environment and comprehensively evaluates the performances of the protocols by means of both analytical and simulation results. Important system parameters such as the traffic load, packet length, data rate, number of STAs in the network, wakeup (off-on) radio transition power consumption, and awake/sleep (on/off) radio transitions time have been considered in the evaluation.
- Section 3.5 presents the experimental evaluation of the DCF and BidMAC protocols using the WARP platform in a proof-of-concept network composed of an AP and two STAs. Experimental results have been obtained in terms of throughput and energy efficiency and have been compared to analytical results considering different values for the traffic load, packet length, and data rate.
- Section 3.6 concludes the chapter by summarizing the key contents of the chapter and highlighting the most relevant results.

3.2 Contention-Based Channel Access Methods

This section overviews the DCF MAC protocol of the IEEE 802.11 Standard and provides a detailed description of the proposed BidMAC and GreenBid MAC protocols.

3.2.1 The Legacy Distributed Coordination Function (DCF)

The DCF MAC specification of the IEEE 802.11 Standard defines a basic access method that is based on the CSMA/CA mechanism (i.e., listen before talk) in combination with a BEB algorithm as the collision resolution mechanism. In addition, an optional access mechanism is defined by which a handshake of RTS and CTS control packets (i.e., RTS/CTS) can be performed between source and destination before the transmission of data. The aim of this handshake is to reduce the impact of collisions of data packets and to combat the problem of hidden terminals.

In general, when the DCF is executed, an STA that has a data packet ready to be transmitted (i.e., a source) senses the wireless channel (i.e., the physical carrier sense mechanism) for a time interval called DIFS. If the wireless channel is sensed idle during this period of time, the STA initiates data transfer (or an RTS/CTS handshake) to an intended receiver (i.e., a destination). Otherwise, if the wireless channel is sensed busy at any time instant within this period of time, the STA avoids attempting access to the wireless channel for the time indicated in the duration field of the MAC header of overheard control (either RTS, CTS, or ACK) and data packets (i.e., packets not destined to its address). This information is used by the STA to update its NAV, which is an internal timer that accounts for the time that the wireless channel is expected to be occupied (i.e., the virtual carrier sense mechanism). Note that if no activity is detected in the wireless channel for a guard time shorter than the updated NAV value, the STA may reset its NAV and attempt channel access after a DIFS.

After the NAV expires (or is reset by the STA) and the wireless channel is sensed idle for a DIFS, the STA needs to wait for a random backoff time during which it continues to monitor the channel activity before transmitting in order to minimize the probability of collision with other transmitting STAs. A backoff procedure is then executed by which the STA randomly selects a backoff counter uniformly distributed within a CW. In the backoff stage, the time following a DIFS is slotted and the STA can only transmit at the beginning of each slot. The slot time is set equal to the time required for any STA to detect the transmission of any other STA. Whenever the wireless channel is sensed idle for a slot time, the STA decrements its backoff counter by one unit whereas it halts the backoff countdown whenever the wireless channel is sensed busy, and it resumes decrementing its backoff counter again after a DIFS. Only when the backoff counter reaches zero, can the STA initiate transmission, indicating the expected occupancy time of the wireless channel in the duration field contained in the MAC header of transmitted RTS or data packets.

Upon successful reception of data (or an RTS), the receiver responds with an ACK packet (or a CTS) after a SIFS. If no ACK packet (or CTS if an RTS was transmitted) is received within a given period of time (i.e., CTS/ACK timeout), the STA waits for an EIFS and then executes the BEB procedure for retransmission. The STA's CW size doubles each failed retransmission attempt up to a maximum value (CW_{max}), and is reset down to a minimum value (CW_{min}) after successful transmission (i.e., after receving an ACK packet to its transmitted data packet). Note that if another data packet is to be transmitted, the STA has to wait for a new random backoff time to avoid channel capture, even if the wireless channel is sensed idle for a DIFS.

Fig. 3.2 shows an example of operation of the DCF with the RTS/CTS access mechanism enabled, where STA 1 and the AP exchange a pair of data packets.

STA 1 and the AP receive at their MAC layers a data packet destined to each other at T_0 and T_1, respectively. They sense the wireless channel for a DIFS and then invoke the backoff procedure before attempting to transmit their data packets. Thus, they wait for a random backoff time by randomly choosing a backoff counter uniformly distributed between 0 and CW_{min}. Their backoff counters are decremented by one, down to zero, each slot time that the wireless channel is sensed idle. STA 1 seizes the

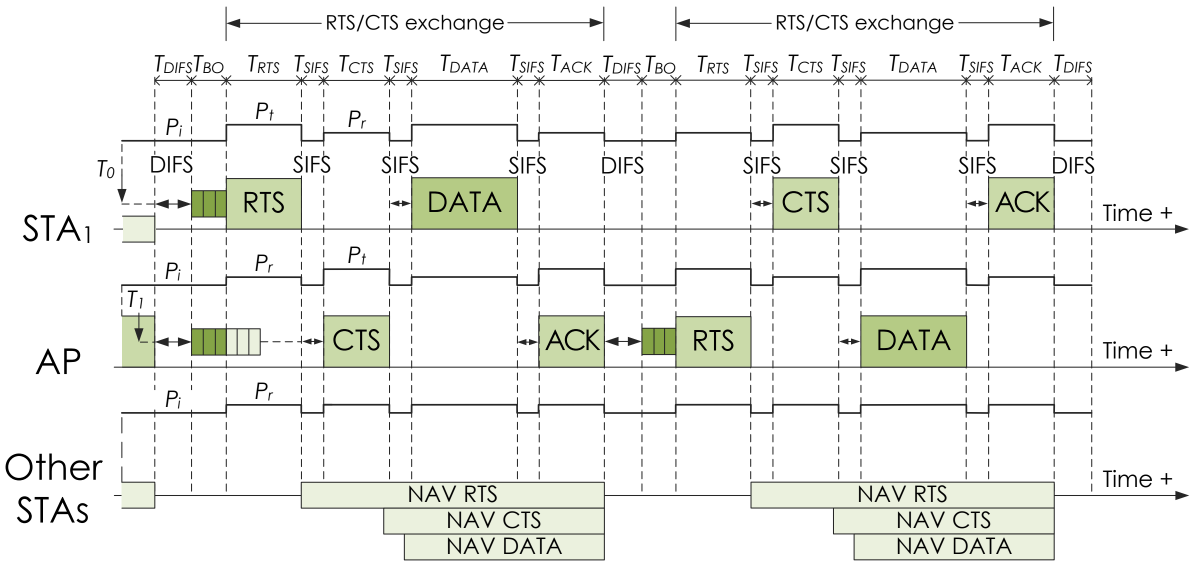


Figure 3.2: Example of operation of the DCF MAC protocol

wireless channel earlier and sends an RTS packet to the AP. Then, the AP freezes its backoff counter and replies with a CTS packet after a SIFS. STA 1 sends its data packet and the AP responds with an ACK packet. After a DIFS, the AP resumes decrementing its backoff counter and then initiates an RTS/CTS exchange to transmit a data packet to STA 1. Other STAs perform the virtual carrier sense mechanism by which their NAVs are updated with the time that the wireless channel will remain busy. This information is carried in the duration field contained in the MAC header of RTS, CTS, DATA, and ACK packets.

3.2.2 The New Bidirectional MAC Protocol (BidMAC)

BidMAC is a reactive (i.e., receiver-initiated) RD MAC protocol that is backwards compatible with the DCF and is aimed at improving the performance of DCF by enabling RD (or bidirectional) transmissions between the AP and the STAs with a single channel access invocation. The operation rules of the DCF only allow the transmission of data from the transmitting STA to the receiving STA (i.e., unidirectional data flow).

The receiving STA is restricted to send back an ACK packet when the data packet is received without errors and needs to contend for channel access if it wishes to transmit a data packet to the transmitting STA of the received data packet, hence increasing access delays and overall contention in the network. Therefore, BidMAC introduces a simple modification into the operation rules of the DCF to allow the receiving STA to initiate RD exchange sequences back to the transmitting STA.

Specifically, the receiver of a valid data packet, either the AP or an STA, is able to transmit, after a SIFS, a data packet of arbitrary length (from 0 to the maximum allowed byte length of payload) with a piggybacked ACK whose destination is the transmitter of the received data packet. As an exception, if the receiver is the AP and has no data ready to be transmitted to the transmitter, the AP is allowed to send a data packet destined to another STA. The transmission rate of the data packet is kept constant for both the forward and reverse transmissions, although it could be reduced for the reverse transmission to increase the probability of successful transmission under bad channel conditions in the reverse link. Also, the value of the duration field in the transmitted data packet is extended to reserve the wireless channel for the duration to complete the RD exchange sequence, including the transmission time of the ACK packet from the transmitter.

Note that the receiver does not know a priori when it is going to receive a data packet and what will be the transmission rate of the received data packet. Hence, as soon as the receiver recognizes a data packet in its transmission buffer, it verifies the data packet length and computes the total transmission time considering all possible data transmission rates. Then, the receiver fetches the data packet and prepares it for transmission as soon as data reception completes. All these steps are necessary to guarantee that the data packet will be transmitted together with the updated duration value and respecting the strict SIFS timing requirement of the legacy DCF. Also, note that when the receiver of a data packet performs a successful RD transmission it cancels its backoff procedure (i.e., the backoff counter is set to zero) and resets its CW size to the minimum value, as it would happen in the legacy DCF when there is a successful transmission.

In the presence of channel errors or collisions, BidMAC follows the same retransmission procedure as that specified in the legacy DCF. So, this means that in BidMAC a failed forward transmission will not follow a reverse transmission. Furthermore, since the reverse transmission is used as an implicit ACK to the forward transmission, a failed RD transmission will result in a failed forward transmission, thus retransmission of the pair of data packets involved will be required. Due to bad channel conditions, the packet loss probability is higher for a data packet than for an ACK packet, since the length of a data packet is usually longer than that of an ACK packet. This is a limitation of BidMAC, when compared to the DCF where the receiver only responds with an ACK packet, that can be overcome by using slower data rates with more robust modulation schemes for the reverse transmissions, as mentioned earlier. Alternatively, if the channel conditions are very bad, the receiver may decide to disable the RD transmission mode until the channel conditions improve.

To protect against hidden terminals, the optional RTS/CTS exchange access method defined in the legacy DCF can be enabled in BidMAC. The operation of BidMAC can also be extended to support batch transmission (i.e., send several data packets together interleaved by a SIFS and its respective ACK packet), aggregation, and block ACK, which are features defined in the IEEE 802.11 Standard. Note that another possible extension of BidMAC is to allow the AP to perform multiple RD transmissions involving one or several receivers whenever it receives data from the STAs. This is feasible because the AP concentrates data in downlink for all the STAs. Also, in this multiple RD exchange sequence process the receiving

STAs can also respond with data, thus boosting the efficiency of BidMAC data transfer.

Fig. 3.3 shows an RD exchange sequence between STA 1 and the AP when BidMAC with RTS/CTS enabled is executed. This example follows the same description as that shown for the DCF in Fig. 3.2. However, when the AP receives the RTS packet from STA 1, it replies with a CTS packet whose duration field includes the additional time required to transmit the data packet buffered for STA 1, based on the information contained in the received RTS packet. Upon successful reception of the CTS packet, STA 1 transmits its data packet to the AP after a SIFS. When the AP receives the data packet, it responds with a data packet after a SIFS. Finally, STA 1 concludes the data exchange by sending an ACK packet to the AP after a new SIFS. As it can be seen, the AP does not need to gain a TXOP to transmit data to STA 1, as it would happen when using the DCF. Instead, with BidMAC the AP uses the TXOP of STA 1 to send its data packet along with the ACK packet to it by extending the TXOP time through the NAV information carried in control and data packets. As a result, access delays can be reduced, hence improving throughput and energy efficiency.

3.2.3 The New Green Bidirectional MAC Protocol (GreenBid)

GreenBid represents an extension of BidMAC to reduce the energy consumed by an STA when it listens to a data transmission where it is not involved. Specifically, the NAV period is used to allow an STA not involved in a data transmission to enter the sleep state to save energy while the wireless channel will be occupied. An essential requirement to accomplish this is that the transmission time is sufficiently long so that an STA can enter the sleep state and return to the awake state before its NAV expires. The transmission time of a single packet may not compensate for the over-

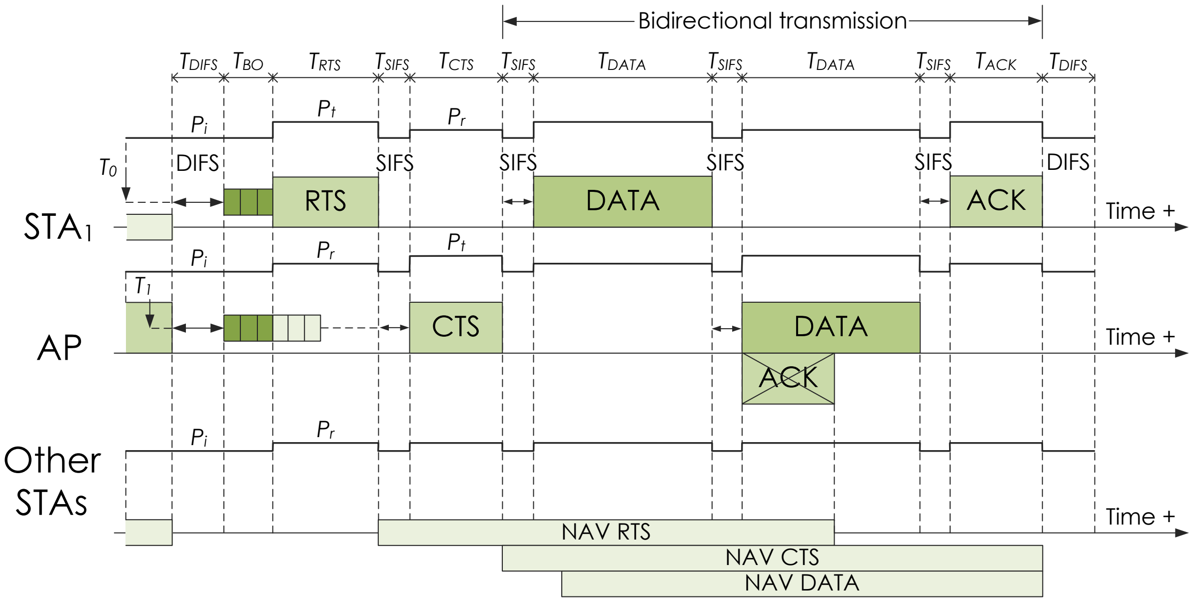


Figure 3.3: Example of operation of the BidMAC protocol

all time required to switch between awake and sleep states, even though the RTS/CTS exchange precedes the transmission of data. Therefore, RD transmissions are used specifically for this purpose, since the exchange of a pair of packets will imply a longer transmission time and thus this may enable the sleep operation.

The use of the RTS/CTS handshake access method is required for the proper operation of GreenBid when the RD exchange sequence involves the transmission of a pair of data packets. The reason is that without RTS and CTS packets the forward transmission is used to set the NAVs of overhearing STAs and, as a result, the available time for sleeping only includes the reverse transmission and the terminating ACK transmission, which may not suffice to compensate for the total awake/sleep transition time and not permit the GreenBid operation. In contrast, the RTS/CTS exchange allows the receiver of an RTS packet to inform overhearing STAs about the total transmission time of an RD exchange sequence through the CTS packet. In this way, those STAs not involved in a data exchange can update their NAVs and based on the available time for sleeping and their awake/sleep radio transitions timing requirements they are able to determine if the sleep operation is possible (i.e., if the available time for sleeping is longer than the total awake/sleep transitions time). If so, they set their wakeup timers such that they will awake when the RD exchange sequence completes (i.e., when their NAVs expire).

Note that if any of the overhearing STAs is unable to update its NAV with any of the transmitted control and data packets or the available time for sleeping does not compensate for the awake/sleep transitions, it will listen to the entire bidirectional communication as it would happen in BidMAC or the legacy DCF. Likewise, if the forward transmission of an RD exchange sequence fails, the reverse transmission will be aborted and retransmission of the forward transmission will be scheduled. However, the STAs that entered the sleep state will not listen to the wireless channel, thus being not aware of this fact. They will be sleeping until their wakeup timers expire, hence loosing the capability of contending for the channel access with the active STAs during some time. Despite this limitation, those STAs will save energy and, in any case, will attempt channel access when they awake after a DIFS.

The operation of GreenBid can also be extended to support batch transmission, aggregation, and block ACK. In these extensions of the protocol, the use of the RTS/CTS exchange access method may not be mandatory since the data packets of the forward and reverse transmission sequences carry the duration information of the complete RD exchange sequence so that overhearing STAs can update their NAVs accordingly. In addition, grouping data packets to be transmitted within the same bidirectional TXOP increases the total transmission time when compared to the case when only a single data packet is allowed in both the forward and reverse transmissions. Therefore, batch transmission and aggregation help to facilitate sleeping processes of longer duration in the STAs not involved in transmissions, hence saving more energy in comparison with the case where only a pair of data packets can be exchanged.

Note that to increase the total transmission time of an RD exchange sequence it is not only possible to increase the amount of transmitted data but also the transmission rate can be adjusted (i.e., reduced). This is reasonable if the main objective is to maximize energy efficiency at the cost of some throughput degradation. Maximizing the energy efficiency of the STAs will depend on the transmission time, which depends on the data length and the transmission rate, and the sleep period, which depends on the transmission time and the awake/sleep transitions time. Thus, given a fixed amount of time for the awake/sleep transitions, a MAC/PHY cross-layer mechanism that determines a proper combination of the amount of data to be transmitted and the data transmission rate to be used in a given RD exchange that results in a sleep period longer than zero could be proposed as an extension of GreenBid. The AP is the ideal executor of this approach because it usually carries much more data than a single STA and is often the responder in an RD exchange sequence when the total traffic load and the number of STAs in the network are both high.

Another possible extension of GreenBid in line with the previous approach is to allow the AP to initiate a contention free period where multiple RD exchanges are performed between the AP and several STAs when the AP receives data from an STA. Then, the STAs not participating in each single RD exchange sequence can go to sleep and save energy.

GreenBid operates as shown in the example of Fig. 3.4 where the RTS/CTS access mechanism is enabled. Following the same description as in Fig. 3.3 for BidMAC, other STAs overhearing the CTS packet or the subsequent data packets read the duration field and update their NAVs. If there exists enough time to go to sleep and wake up before their NAVs

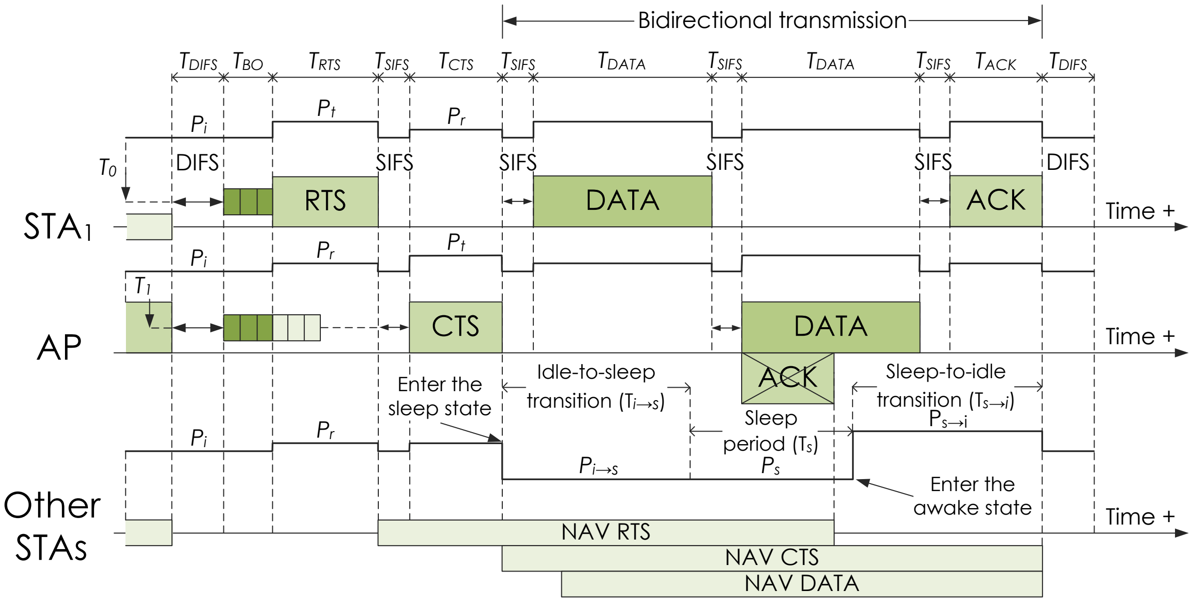


Figure 3.4: Example of operation of the GreenBid MAC protocol

expire (i.e., the available time for sleeping is longer than the duration of the on/off radio transitions), they set their wakeup timers and turn off their radio transceivers. Upon successful reception of the CTS packet, STA 1 transmits a data packet to the AP, which sends back a data packet after a SIFS. The newly received data packet can be interpreted by STA 1 as an implicit ACK for its transmitted data packet. After a SIFS, STA 1 replies with an explicit ACK packet to complete the data exchange. At this point, all other STAs are awake and can resume the channel contention after a DIFS. Therefore, the STAs can save energy without incurring additional delays for the channel access. In addition, both the overhead of control packets and the overall contention of the network can be significantly reduced compared to the case when both the AP and STA 1 need to gain a TXOP to transmit their data packets.

3.3 Theoretical Analysis

This section presents the analysis of the maximum achievable throughputs and energy efficiencies of the protocols under consideration. First, the system model and assumptions made to carry out the analysis are described in detail. Then, a simplified approach for analyzing the protocols is explained. Finally, a complete analytical model for the upper-bound performance of the protocols is presented based on the throughput model of the DCF by Giuseppe Bianchi presented in [84] and its modifications provided in [85].

3.3.1 System Model and Assumptions

A Basic Service Set (BSS) composed of an AP and N associated STAs in the Basic Service Area (BSA) is considered, as shown in Fig. 3.1. All devices are equipped with IEEE 802.11n wireless interfaces enabling a single antenna for communications, i.e., a Single-Input Single-Output (SISO) communications system. Wireless communication within the BSS occurs between the AP and the STAs using a shared radio channel. It is assumed that the size of the BSA allows all the STAs of the BSS to overhear the transmissions between each STA and the AP in both directions. Note that the AP can deliver downlink data to any STA of the BSS.

In order to compute the upper bound of the theoretical throughput and energy efficiency within a BSS in idealistic conditions, the following assumptions are made: *(i)* the wireless channel is ideal, *(ii)* the probability of collision is negligible (only for the simplified approach), *(iii)* the propagation delay is neglected, *(iv)* the transmit queues are never empty, *(v)* no packets are lost due to queue overflow, *(vi)* no management packets, such as beacons and association requests, are transmitted, and *(vii)* fragmentation is not used.

Table 3.1: ERP-OFDM PHY Modes and Transmission Times for RTS, CTS, and ACK Control Packets and DATA Packets (1500-Byte Payload) in IEEE 802.11n

Mode (m)	Modulation	Code Rate	Data Rate	N_{DBPS}	T_{RTS}	T_{CTS}	T_{ACK}	T_{DATA}
1	BPSK	1/2	6 Mbps	24	58 μs	50 μs	50 μs	2078 μs
2	BPSK	3/4	9 Mbps	36	50 μs	50 μs	50 μs	1394 μs
3	QPSK	1/2	12 Mbps	48	42 μs	38 μs	38 μs	1054 μs
4	QPSK	3/4	18 Mbps	72	38 μs	38 μs	38 μs	710 μs
5	16-QAM	1/2	24 Mbps	96	34 μs	34 μs	34 μs	542 μs
6	16-QAM	3/4	36 Mbps	144	34 μs	34 μs	34 μs	370 μs
7	64-QAM	2/3	48 Mbps	192	30 μs	34 μs	34 μs	286 μs
8	64-QAM	3/4	54 Mbps	216	30 μs	34 μs	34 μs	254 μs

Among the possible configurations of the IEEE 802.11n at the PHY layer, the Extended Rate Physical (ERP)-layer Orthogonal Frequency Division Multiplexing (OFDM) specification (i.e., ERP-OFDM) for SISO communications is selected. The ERP-OFDM PHY provides 8 transmission modes with different modulation schemes and coding rates. Table 3.1 summarizes the characteristics of each mode (m), where the supported data rates and the Number of Data Bits Per OFDM Symbol (NDBPS), denoted as N_{DBPS}, are shown.

The structure of an ERP-OFDM packet is shown in Fig. 3.5. Each MAC data packet or MAC Protocol Data Unit (MPDU) consists of a MAC header, frame body or MAC Service Data Unit (MSDU), and Frame Check Sequence (FCS). The MAC header (L_{MAChdr}) and FCS (L_{FCS}) together are up to 34 octets, the RTS packet is 20 octets, and the CTS and ACK packets are 14 octets long.

When an MPDU is to be transmitted, it is passed to the PHY Layer Convergence Protocol (PLCP) sublayer where it is called PLCP Service Data Unit (PSDU). In order to form a PLCP Protocol Data Unit (PPDU), a PLCP preamble and a PLCP header are added to a PSDU. The duration

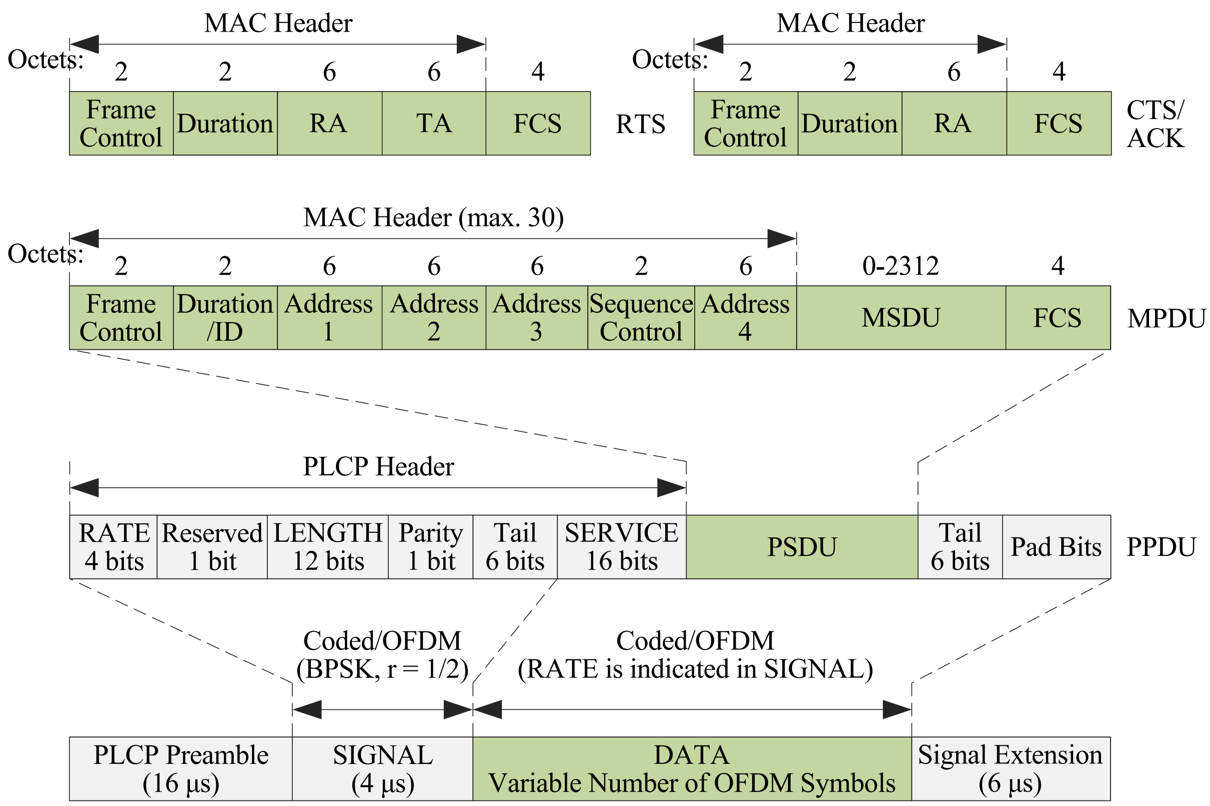


Figure 3.5: IEEE 802.11n ERP-OFDM packet format

of the PLCP preamble field (T_{pre}) is 16 μs. The PLCP header except the SERVICE field constitutes the SIGNAL field whose duration (T_{sig}) equals the duration of a single OFDM symbol (T_{sym}) with 4 μs. The 16-bit SERVICE field (L_{serv}) and the MPDU along with 6 tail bits (L_{tail}) and pad bits, represented by DATA, are transmitted at the data rate specified in the RATE field. Finally, a period of no transmission with a length of 6 μs called the signal extension (T_{sigEx}) follows after the end of the ERP-OFDM transmission. All the above parameters and their values are provided in Table 3.2.

The BSS basic rate set determines the set of data rates that should be supported by all the STAs of a BSS. The mandatory rates are the rates that use 1/2 rate coding, i.e., 6, 12, and 24 Mbps, as shown in Table 3.1. To allow the transmitting STA to calculate the value of the duration field,

Table 3.2: System Parameters

Parameter	Definition	Value
T_{slot}	Slot Time	9 μs
T_{SIFS}	SIFS Interval	10 μs
T_{DIFS}	DIFS Interval	28 μs
T_{EIFS}	EIFS Interval	88 μs
CW_{min}	Minimum Contention Window	15
CW_{max}	Maximum Contention Window	1023
T_{BO}	Average Backoff Time	67.5 μs
T_{pre}	Preamble Time	16 μs
T_{sig}	Signal Time	4 μs
T_{sym}	OFDM symbol Period	4 μs
T_{sigEx}	Signal Extension Period	6 μs
L_{serv}	Service Bits	16 bits
L_{tail}	Tail Bits	6 bits
L_{RTS}	Length of RTS	20 bytes
L_{CTS}	Length of CTS	14 bytes
L_{ACK}	Length of ACK	14 bytes
L_{MAChdr}	MAC Header	30 bytes
L_{FCS}	Frame Check Sequence	4 bytes
$T_{i \to s}$	Transition Time from Idle to Sleep	250 μs
$T_{s \to i}$	Transition Time from Sleep to Idle	250 μs
P_t	Transmission Power Consumption	1.65 W
P_r	Reception Power Consumption	1.4 W
P_i	Idle Power Consumption	1.15 W
P_s	Sleep Power Consumption	0.045 W
$P_{i \to s}$	Idle to Sleep Transition Power Consumption	0.045 W
$P_{s \to i}$	Sleep to Idle Transition Power Consumption	1.725 W

control response packets like CTS and ACK should be transmitted at the highest basic rate that is less than or equal to the rate of the received packet. This means that CTS and ACK packets are transmitted at 6, 12, or 24 Mbps if the RTS and data packets were received at 6 or 9, 12 or 18, and 24, 36, 48 or 54 Mbps, respectively.

Now it is possible to obtain the time to transmit each packet using the ERP-OFDM PHY mode. The transmission times of a data packet with L_{MSDU} octets of data payload (T_{DATA}) and RTS (T_{RTS}), CTS (T_{CTS}) and ACK (T_{ACK}) packets are computed by [16] as

$$\begin{aligned} T_{DATA}\left(m\right) &= T_{pre}+T_{sig}+T_{sym}\left\lceil \frac{L_{serv}+8\cdot\left(L_{MAChdr}+L_{MSDU}+L_{FCS}\right)+L_{tail}}{N_{DBPS}\left(m\right)} \right\rceil \\ &+T_{sigEx}=26+4\cdot\left\lceil \frac{22+8\cdot\left(34+L_{MSDU}\right)}{N_{DBPS}\left(m\right)} \right\rceil \end{aligned} \tag{3.1}$$

$$\begin{aligned} T_{RTS}\left(m\right) &= T_{pre}+T_{sig}+T_{sym}\left\lceil \frac{L_{serv}+8\cdot L_{RTS}+L_{tail}}{N_{DBPS}\left(m\right)} \right\rceil+T_{sigEx} \\ &=26+4\cdot\left\lceil \frac{22+8\cdot 20}{N_{DBPS}\left(m\right)} \right\rceil \end{aligned} \tag{3.2}$$

$$\begin{aligned} T_{CTS}\left(m\right)=T_{ACK}\left(m\right) &= T_{pre}+T_{sig}+T_{sym}\left\lceil \frac{L_{serv}+8\cdot L_{ACK}+L_{tail}}{N_{DBPS}\left(m\right)} \right\rceil+T_{sigEx} \\ &=26+4\cdot\left\lceil \frac{22+8\cdot 14}{N_{DBPS}\left(m\right)} \right\rceil \end{aligned} \tag{3.3}$$

where the ceiling function $\lceil x \rceil$ returns the smallest integer value greater than or equal to x. Table 3.1 shows the transmission time of each packet for each transmission rate. The transmission time of the data packet is given for 1500 octets of data payload.

In the following, T_{SIFS}, T_{DIFS}, and T_{slot} denote the SIFS and DIFS intervals and the slot time, respectively, and T_{DIFS} is given by

$$T_{DIFS}=T_{SIFS}+2\cdot T_{slot}=10+2\cdot 9=10+18=28\mu s \tag{3.4}$$

Since no collisions are considered in the simplified analysis, the backoff period (T_{BO}) is an average value obtained from CW_{min} and T_{slot} as

$$T_{BO}=\left(\frac{CW_{min}}{2}\right)T_{slot}=\frac{15}{2}\cdot 9=7.5\cdot 9=67.5\mu s \tag{3.5}$$

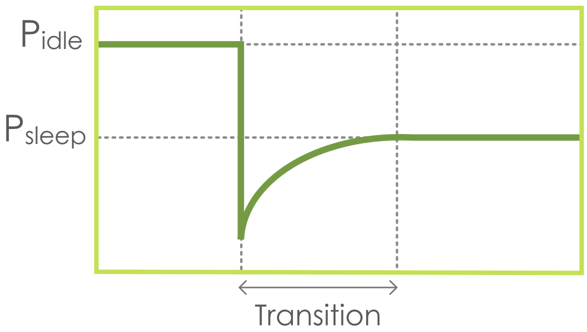
For the same reason, CW_{max} and the EIFS interval (T_{EIFS}) are not considered for the simplified analysis (although they will be used in the Binachi-based analysis and the simulation part), and T_{EIFS} is expressed as

$$T_{EIFS}=T_{SIFS}+T_{DIFS}+T_{ACK}\,(m=1)=10+28+50=88\mu s \tag{3.6}$$

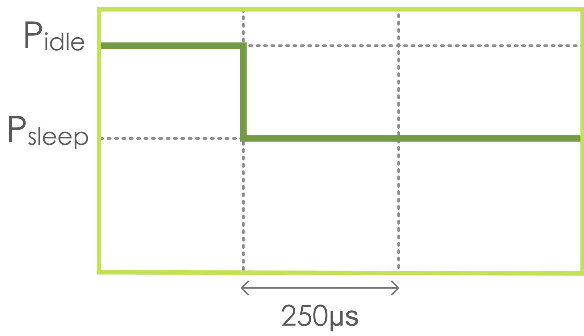
These variables and their values are shown in Table 3.2.

The IEEE 802.11n wireless interface of an STA can be in one of the following operational states: transmitting, receiving or overhearing (i.e., receiving packets not destined to itself), idle, and sleeping. In the first two states, the radio transceiver is actively used to send and receive information. In the idle state, the wireless interface is ready to receive but no signal is received by the radio transceiver. In the sleep state, the radio transceiver is turned off to save energy. Each of these operational states has associated power consumption. In addition, each transition between states incurs a certain switching time that cannot be neglected. These values will vary depending on the product hardware.

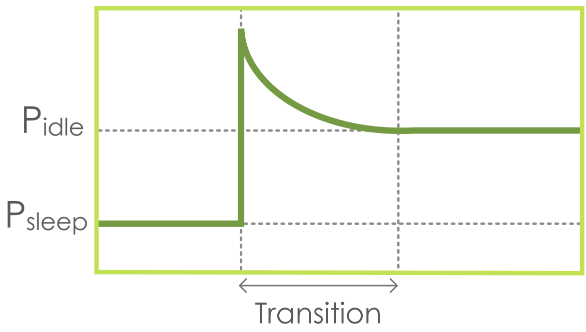
Let P_t, P_r, P_i, and P_s denote the power consumed while transmitting, receiving, idle, and sleeping, respectively. When an idle STA identifies an opportunity to sleep, a transition from idle to sleep takes place. Similarly, a transition from sleep to idle occurs when the STA decides to wake up. Based on [3–5], the transition time from idle to sleep ($T_{i\to s}$) is shown to be similar to the transition time from sleep to idle ($T_{s\to i}$). Hence, it is assumed that $T_{i\to s}$ is equal to $T_{s\to i}$. Regarding the power consumed during these transitions, the works in [3–5] show that the power consumed from idle to sleep ($P_{i\to s}$) is substantially lower than P_s. In contrast, the power



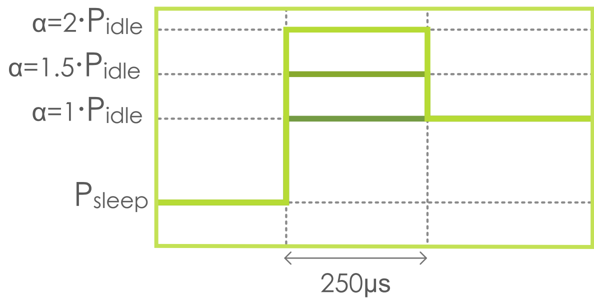
(a) Experiment: Idle to sleep transition



(b) Analysis: Idle to sleep transition



(c) Experiment: sleep to idle transition



(d) Analysis: sleep to idle transition

Figure 3.6: Time and power consumption during the transitions between idle and sleep states: (a) and (c) correspond to an abstraction of real measurements reported in [3–5]) for a WaveLAN interface whereas (b) and (d) represent the adopted awake/sleep transition model for the analysis

consumed from sleep to idle ($P_{s\to i}$) is shown to be significantly higher than P_i. Thus, it is assumed that $P_{i\to s}$ is equal to P_s and $P_{s\to i}$ is modeled as αP_i, where α is defined as the transition coefficient between sleep and idle states, or wakeup transition coefficient, and $\alpha > 1$. Fig. 3.6 illustrates this explanation and Table 3.2 records the variables mentioned above and their values (most of them taken from [3–5]).

3.3.2 Simplified Approach

This approach considers a simplified scenario where the AP and the STAs do not compete concurrently for access to the wireless channel. Instead, in each transmission cycle there is only one active transmitter, either the AP or an STA, that executes the random backoff procedure and then performs an RTS/CTS handshake to send a data packet to the intended receiver. Based on this approach, the mathematical expressions of the maximum achievable throughputs and energy efficiencies of the protocols are derived as follows from three different perspectives: entire network, AP (i.e. downlink), and average per STA (i.e. uplink). Note that similar expressions can be obtained for the basic access method, where the RTS/CTS handshake is not used.

A. Throughput

The throughput of a given protocol x (S_x) is defined as the amount of information contained in an MSDU (L_{MSDU}) divided by the time ratio (T_x) required to transmit the data packet that includes the MSDU. This is expressed as

$$S_x[\text{Mbps}]=\frac{8{\cdot}L_{MSDU}}{T_x} \tag{3.7}$$

where (T_x) is defined as the amount of time spent in transmission over the total amount of transmitted data packets.

The transmission time ratio of each protocol under consideration is described and formulated as follows.

1) DCF – RTS/CTS:

The transmission delay of DCF comprises a DIFS interval, a backoff period, an RTS transmission, a SIFS interval, a CTS transmission, a SIFS interval, a data transmission, a SIFS interval, and an ACK transmission. Thus, the transmission time ratio that corresponds to the saturation network throughput of the DCF is expressed as

$$T_{DCF}^{net\_sat}{=}T_{DIFS}{+}T_{BO}{+}T_{RTS}{+}T_{CTS}{+}T_{DATA}{+}T_{ACK}{+}3{\cdot}T_{SIFS} \tag{3.8}$$

The downlink throughput of DCF shows a maximum value and a lower stable value under saturation. The maximum value is obtained when the AP is able to deliver a data packet to each STA and each STA is able to transmit a data packet to the AP. Given N STAs, the AP is able to perform N channel accesses every N transmissions from the STAs. In total, $2N$ transmissions are required. As a result, the minimum transmission time ratio that leads to the maximum downlink throughput of the DCF is given by

$$T_{DCF}^{dwl\_min}{=}\frac{2N}{N}\left(T_{DIFS}{+}T_{BO}{+}T_{RTS}{+}T_{CTS}{+}T_{DATA}{+}T_{ACK}{+}3{\cdot}T_{SIFS}\right) \tag{3.9}$$

However, when the network enters the saturation state, the AP can only perform one transmission every N transmissions from the STAs, due to the long-term fairness characteristic of the DCF. Therefore, the transmission time ratio that results in the saturation downlink throughput is computed as

$$T_{DCF}^{dwl\_sat}{=}\left(N{+}1\right)\left(T_{DIFS}{+}T_{BO}{+}T_{RTS}{+}T_{CTS}{+}T_{DATA}{+}T_{ACK}{+}3{\cdot}T_{SIFS}\right) \tag{3.10}$$

Similarly, the transmission time ratio that provides the saturation uplink per STA throughput can be obtained by

$$T_{DCF}^{uplpersta\_sat}{=}\left(N{+}1\right)\left(T_{DIFS}{+}T_{BO}{+}T_{RTS}{+}T_{CTS}{+}T_{DATA}{+}T_{ACK}{+}3{\cdot}T_{SIFS}\right) \tag{3.11}$$

2) BidMAC – RTS/CTS:

The transmission delay of BidMAC contains the same as that of DCF but it includes an additional data transmission and a SIFS interval. To compute the maximum throughput of BidMAC, it is assumed that the receiver of a data packet, either the AP or an STA, has a data packet ready to be sent to the transmitter. Therefore, a pair of data packets can be exchanged within a BidMAC transmission.

The transmission time ratio that produces the saturation network throughput of BidMAC is expressed as

$$T_{BidMAC}^{net\_sat}=\frac{1}{2}\left(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+2\cdot T_{DATA}+T_{ACK}+4\cdot T_{SIFS}\right) \quad (3.12)$$

Unlike the DCF, the downlink throughput of BidMAC only shows a maximum value under saturation, since the AP is able to deliver a data packet whenever it has received a data packet. The AP is granted N channel accesses when the STAs send their data packets to it. Similarly, when the AP gets a transmission opportunity, the receiving STA also performs a data transmission to the AP. Thus, the transmission time ratio that corresponds to the saturation downlink throughput of BidMAC is given as

$$T_{BidMAC}^{dwl\_sat}=\frac{N+1}{N+1}\left(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+2\cdot T_{DATA}+T_{ACK}+4\cdot T_{SIFS}\right) \quad (3.13)$$

Given the downlink throughput of BidMAC, the maximum uplink per STA throughput of BidMAC is computed as follows. An STA gets a transmission opportunity to send a data packet to the AP every N bidirectional transmissions from the $N-1$ STAs and the AP. When the AP gets a transmission opportunity, the receiving STA is allowed to transmit a data packet. Hence, a given STA is able to perform an additional data transmission with probability $\frac{1}{N}$. As a result, the transmission time ratio that leads to the saturation uplink per STA throughput of BidMAC is calculated as

$$T_{BidMAC}^{uplpersta\_sat}=\frac{N+1}{1+\frac{1}{N}}\left(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+2\cdot T_{DATA}+T_{ACK}+4\cdot T_{SIFS}\right) \quad (3.14)$$

3) GreenBid – RTS/CTS:

The throughput values of GreenBid are exactly the same as those of BidMAC because GreenBid has been designed to improve energy efficiency.

B. Energy Efficiency

The energy efficiency of a given protocol x (η_x) is defined as the amount of bits contained in an MSDU divided by the energy consumption ratio (E_x) required to transmit the data packet that includes the MSDU:

$$\eta_x[\text{Mb/J}]=\frac{8 \cdot L_{MSDU}}{E_x} \tag{3.15}$$

where L_{MSDU} denotes the byte-length of an MSDU and E_x is defined as the product of power consumed and time spent in transmission over the total amount of transmitted data packets.

The energy consumption ratio of each protocol under consideration is described and formulated as follows.

1) DCF – RTS/CTS:

The energy consumption of DCF is split into three energy consumption components, namely, transmitting (E_t), receiving and overhearing (E_r), and idle (E_i). During a transmission cycle of DCF, the transmitter, either the AP or an STA, consumes energy to transmit the RTS packet and the data packet and to receive the CTS packet and the ACK packet from the receiver. On the other hand, the receiver consumes energy to receive the RTS packet and the data packet from the transmitter and to respond with the CTS packet and the ACK packet. Meanwhile, the $N-1$ STAs not involved in transmission consume energy to overhear the exchange of packets. The N STAs and the AP also consume energy to listen to the wireless channel for a DIFS interval, a backoff period, and all SIFS intervals. Therefore, the energy consumption ratio that results in the saturation network energy efficiency of DCF is given as

$$
\begin{aligned}
E_{DCF}^{net\_sat} &= E_t + E_r + E_i \\
E_t &= (T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK})\, P_t \\
E_r &= N\, (T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK})\, P_r \\
E_i &= (N+1)\, (T_{DIFS} + T_{BO} + 3 \cdot T_{SIFS})\, P_i
\end{aligned} \tag{3.16}
$$

The AP energy efficiency of DCF shows a maximum value and a lower stable value under saturation. The maximum value is obtained when the AP consumes energy to deliver a data packet to each STA and to receive a data packet from each STA. Given N STAs, the AP acts as a transmitter during N transmission slots and as a receiver during N transmission slots. As a result, the minimum energy consumption ratio that produces the maximum AP energy efficiency of DCF is expressed as

$$
\begin{aligned}
E_{DCF}^{ap\_min} &= \frac{1}{N}\, (E_t + E_r + E_i) \\
E_t &= N\, (T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK})\, P_t \\
E_r &= N\, (T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK})\, P_r \\
E_i &= 2N\, (T_{DIFS} + T_{BO} + 3 \cdot T_{SIFS})\, P_i
\end{aligned} \tag{3.17}
$$

However, when the network enters the saturation state, the AP acts as a transmitter once and as a receiver during N transmission slots, due to the fairness of the DCF. Hence, the energy consumption ratio that leads to the saturation AP energy efficiency of DCF is computed as

$$
\begin{aligned}
E_{DCF}^{ap\_sat} &= E_t + E_r + E_i \\
E_t &= (T_{RTS} + T_{DATA} + N\, (T_{CTS} + T_{ACK}))\, P_t \\
E_r &= (N\, (T_{RTS} + T_{DATA}) + T_{CTS} + T_{ACK})\, P_r \\
E_i &= (N+1)\, (T_{DIFS} + T_{BO} + 3 \cdot T_{SIFS})\, P_i
\end{aligned} \tag{3.18}
$$

Similarly, the maximum average per STA throughput of DCF is achieved under saturation and can be computed as follows. An STA acts as a transmitter once every N transmissions from $N{-}1$ STAs and the AP. During $N{-}1$ transmissions, an STA overhears. When the AP gets a transmission opportunity, a given STA can be the actual receiver with probability $\frac{1}{N}$ (assuming a uniform traffic distribution) whereas with probability $1{-}\frac{1}{N}$ a given STA is not the intended destination. Thus, the energy consumption ratio that corresponds to the maximum average per STA energy efficiency of DCF is calculated as

$$
\begin{aligned}
E_{DCF}^{persta\_sat} &= E_t + E_r + E_i \\
E_t &= \left(T_{RTS} + T_{DATA} + \frac{1}{N}\left(T_{CTS} + T_{ACK}\right)\right) P_t \\
E_r &= \left(N\left(T_{RTS} + T_{DATA}\right) + \left(N + 1 - \frac{1}{N}\right)\left(T_{CTS} + T_{ACK}\right)\right) P_r \\
E_i &= (N+1)\left(T_{DIFS} + T_{BO} + 3 \cdot T_{SIFS}\right) P_i
\end{aligned}
\tag{3.19}
$$

2) BidMAC – RTS/CTS:

Within a data exchange through BidMAC, the energy consumed by the AP and the STAs is similar to that of DCF. However, the receiver consumes energy to transmit a data packet and not an ACK packet and to receive an ACK packet from the transmitter. On the contrary, the transmitter consumes energy to receive the data packet and to send back the ACK packet. In addition, the $N{-}1$ STAs consume energy to overhear the data packet from the receiver. The N STAs and the AP also consume energy for being idle during an additional SIFS interval. Consequently, the energy consumption ratio that results in the maximum network energy efficiency of BidMAC is obtained by

$$
\begin{aligned}
E_{BidMAC}^{net\_sat} &= \frac{1}{2}\left(E_t + E_r + E_i\right) \\
E_t &= \left(T_{RTS} + T_{CTS} + 2 \cdot T_{DATA} + T_{ACK}\right) P_t \\
E_r &= N\left(T_{RTS} + T_{CTS} + 2 \cdot T_{DATA} + T_{ACK}\right) P_r \\
E_i &= (N+1)\left(T_{DIFS} + T_{BO} + 4 \cdot T_{SIFS}\right) P_i
\end{aligned}
\tag{3.20}
$$

The maximum AP energy efficiency of BidMAC is computed as follows. The AP consumes energy to transmit a data packet each time it has received a data packet from an STA. Hence, the AP acts as a receiver during N transmission slots. In addition, the AP acts as a transmitter once when it gets a transmission opportunity where the receiving STA can send a data packet. Therefore, the energy consumption ratio that leads to the maximum AP energy efficiency of BidMAC is given as

$$\begin{aligned}
E_{BidMAC}^{ap_sat} &= \frac{1}{N+1}\left(E_t + E_r + E_i\right) \\
E_t &= \left(T_{RTS} + T_{DATA} + T_{ACK} + N\left(T_{CTS} + T_{DATA}\right)\right) P_t \\
E_r &= \left(N\left(T_{RTS} + T_{DATA} + T_{ACK}\right) + T_{CTS} + T_{DATA}\right) P_r \\
E_i &= \left(N+1\right)\left(T_{DIFS} + T_{BO} + 4 \cdot T_{SIFS}\right) P_i
\end{aligned} \tag{3.21}$$

Similarly, the maximum average per STA energy efficiency of BidMAC is calculated as follows. An STA acts as a transmitter once every N bidirectional transmissions from $N-1$ STAs and the AP. During $N-1$ bidirectional transmissions, an STA overhears. When the AP gets a transmission opportunity, a given STA can be the actual receiver with probability $\frac{1}{N}$ and so can send a data packet to the AP. On the other hand, with probability $1-\frac{1}{N}$ a given STA is not the intended destination. Thus, the energy consumption ratio that produces the maximum average per STA energy efficiency of BidMAC is calculated as

$$\begin{aligned}
E_{BidMAC}^{persta_sat} &= \frac{1}{1+\frac{1}{N}}\left(E_t + E_r + E_i\right) \\
E_t &= \left(T_{RTS} + T_{DATA} + T_{ACK} + \frac{1}{N}\left(T_{CTS} + T_{DATA}\right)\right) P_t \\
E_r &= \left(N\left(T_{RTS} + T_{DATA} + T_{ACK}\right) + \left(N + 1 - \frac{1}{N}\right)\left(T_{CTS} + T_{DATA}\right)\right) P_r \\
E_i &= \left(N+1\right)\left(T_{DIFS} + T_{BO} + 4 \cdot T_{SIFS}\right) P_i
\end{aligned} \tag{3.22}$$

3) GreenBid – RTS/CTS:

GreenBid builds on top of BidMAC. In addition to transmit (E_t), receive (E_r), and idle energy components (E_i), the energy consumption of GreenBid introduces two new energy consumption components, namely, switching between idle and sleeping (E_{sw}), and sleeping (E_s). These components are described one by one as follows.

- **Transmission period:** the transmitter consumes energy to send an RTS packet, a data packet, and an ACK packet to the receiver whereas the receiver replies with a CTS packet and a data packet.

- **Reception period:** the transmitter and the receiver consume energy to receive the CTS packet and data packet and the RTS packet, data packet, and ACK packet, respectively. The $N-S$ STAs only consume energy to overhear the RTS and CTS packets as they can switch to the sleep state to save energy. S denotes the number of active STAs, which is just 1 (apart from the AP).

- **Idle period:** all the STAs and the AP consume energy to listen to the wireless channel during a DIFS interval, a backoff period, and a SIFS interval. After that, only the transmitter and the receiver are awake for the remaining SIFS intervals.

- **Switch period:** the $N-S$ sleeping STAs consume energy during the transition from idle to sleep and during the transition from sleep to idle.

- **Sleep period:** the STAs can sleep during the data exchange expect for when they have to switch between idle and sleep states. This happens provided that the sleep period (T_s) is greater than zero. Otherwise, none of the overhearing STAs can sleep and the energy consumed by GreenBid is the same as for BidMAC. The sleep period is computed as

$$T_s = 2 \cdot T_{DATA} + T_{ACK} + 3 \cdot T_{SIFS} - \left(T_{i \to s} + T_{s \to i}\right) \tag{3.23}$$

Based on the above, the energy consumption ratio that corresponds to the saturation network energy efficiency of GreenBid can be expressed as

$$\begin{aligned}
E_{GreenBid}^{net_sat} &= \frac{1}{2}\left(E_t + E_r + E_i + E_{sw} + E_s\right) \\
E_t &= \left(T_{RTS} + T_{CTS} + 2 \cdot T_{DATA} + T_{ACK}\right) P_t \\
E_r &= \left(N\left(T_{RTS} + T_{CTS}\right) + S\left(2 \cdot T_{DATA} + T_{ACK}\right)\right) P_r \\
E_i &= \left(\left(N+1\right)\left(T_{DIFS} + T_{BO} + T_{SIFS}\right) + \left(S+1\right) 3 \cdot T_{SIFS}\right) P_i \\
E_{sw} &= \left(N - S\right)\left(T_{i \to s} P_{i \to s} + T_{s \to i} P_{s \to i}\right) \\
E_s &= \left(N - S\right) T_s P_s
\end{aligned} \tag{3.24}$$

The maximum AP energy efficiency of GreenBid is exactly the same as that of BidMAC, since GreenBid has been designed to improve the energy efficiency of the STAs.

The maximum average per STA energy efficiency of GreenBid is computed as follows. An STA is only awake when it gets a transmission opportunity once every N transmission slots and when it receives a data packet from the AP with probability $\frac{1}{N}$. As a result, the energy consumption ratio that results in the saturation average per STA energy efficiency of

GreenBid is obtained by

$$\begin{aligned}
E_{GreenBid}^{persta_sat} &= \frac{1}{1+\frac{1}{N}} \left(E_t+E_r+E_i+E_{sw}+E_s\right) \\
E_t &= \left(T_{RTS}+T_{DATA}+T_{ACK}+\frac{1}{N}\left(T_{CTS}+T_{DATA}\right)\right) P_t \\
E_r &= \left(NT_{RTS}+\left(N+1-\frac{1}{N}\right) T_{CTS}+\left(1+\frac{1}{N}\right) T_{DATA}+\frac{T_{ACK}}{N}\right) P_r \\
E_i &= \left(\left(N+1\right)\left(T_{DIFS}+T_{BO}+T_{SIFS}\right)+\left(1+\frac{1}{N}\right) 3\cdot T_{SIFS}\right) P_i \\
E_{sw} &= \left(N-\frac{1}{N}\right)\left(T_{i\to s}P_{i\to s}+T_{s\to i}P_{s\to i}\right) \\
E_s &= \left(N-\frac{1}{N}\right) T_s P_s
\end{aligned} \tag{3.25}$$

3.3.3 Bianchi-Based Approach

This approach considers a more realistic scenario where the AP and the STAs compete for access to the wireless channel concurrently. In this case, in each transmission cycle the AP and the STAs perform carrier sensing and execute the random backoff procedure. Once one of them seizes the wireless channel, it performs an RTS/CTS handshake to send a data packet to the intended receiver. To model this behavior, Giuseppe Bianchi presented in [84] a simple but accurate model based on a discrete Markov chain for the computation of the saturation throughput of the DCF. Enhancements to the proposed model were then reported by Bianchi and Tinnirello in [85], although for many years there have been strong efforts in the research community to propose more accurate models of the throughput of the DCF. Herein, the initial Bianchi model [84] together with its modifications [85] represent the basis for the analysis presented as follows where the mathematical expressions of the maximum achievable throughputs and energy efficiencies of the protocols are derived from three different perspectives: entire network, AP (i.e. downlink), and average per STA (i.e. uplink). Note that similar expressions can be obtained for the basic access method, where the RTS/CTS handshake is not used.

A. Packet Transmission Probability

Consider saturation conditions (i.e., the AP and all the STAs have data ready to be transmitted in their buffers) and a given constant number of contending STAs (n). Also, the wireless channel is considered to be error-free and there are no hidden terminals. Although the propagation delay (δ) is neglected for the analytical and simulation results, it is included in the formulas.

To model the backoff rules a contenting STA needs to follow to get access to the wireless channel, two important variables are defined in [84], namely:

- **τ:** it refers to the transmission probability that an STA transmits in a randomly chosen slot time and is expressed as

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \tag{3.26}$$

  where W is defined for convenience as $W = CW_{min}+1 = 15+1 = 16$ because initially the randomly chosen backoff counter value by a contending STA may range from 0 to CW_{min} (i.e., between 0 and 15), leading to a CW size of W possible values (i.e., 16). Then, the CW size doubles after each failed retransmission attempt up to $CW_{max}+1 = 1023+1 = 1024$ according to the BEB algorithm. This is modeled by $W_i = 2^i W$ where $i \in (0, m)$ and m is the maximum backoff stage whose value is 6 by $CW_{max}+1 = 1023+1 = 1024 = 2^m W = 2^m 16$.

- **p:** it denotes the probability of a collision experienced by a packet being transmitted in the wireless channel, named conditional collision probability, and is computed as

$$p{=}1{-}\left(1{-}\tau\right)^{n+1-1}{=}1{-}\left(1{-}\tau\right)^{n} \tag{3.27}$$

Note that p is assumed to be constant and independent from the number of retransmissions already suffered.

As it can been seen, τ and p depend on each other, i.e., (3.26) and (3.27) form a non-linear system that can be solved using numerical techniques (in this case Matlab fzero function has been used to obtain unique solutions of τ and p).

B. Throughput

The throughput S is defined as the fraction of time that the wireless channel is used to successfully transmit payload bits as

$$S{=}\frac{E[\text{payload information transmitted in a slot time}]}{E[\text{length of a slot time}]} \tag{3.28}$$

To compute S, two new variables that represent what can happen in a randomly chosen slot time are defined in [84], namely:

- P_{tr}**:** it refers to the probability that there is at least one transmission in the considered slot time. Since there exist an AP and n STAs all contenting for channel access and each transmits with probability τ, thus

$$P_{tr}{=}1{-}\left(1{-}\tau\right)^{n+1} \tag{3.29}$$

- P_s**:** it denotes the probability that a transmission occurring in the wireless channel is successful and is given by the probability that only one STA transmits in the wireless channel, given that at least one STA transmits, that is,

$$P_s{=}\frac{(n{+}1)\,\tau\left(1{-}\tau\right)^{n+1-1}}{P_{tr}}{=}\frac{(n{+}1)\,\tau\left(1{-}\tau\right)^{n}}{P_{tr}} \tag{3.30}$$

Therefore, the saturation throughput of a given protocol x (S_x) being the DCF or a variation that uses the same backoff rules as those of the DCF (i.e., BidMAC and GreenBid) is expressed by [84] as

$$S_x = \frac{\alpha_x P_s P_{tr} E[P]'}{(1-P_{tr})\,\sigma + P_s P_{tr} T_s'^x + P_{tr}\,(1-P_s)\,T_c'} \tag{3.31}$$

where

- α_x**:** number of data transmissions within a given slot time
- $P_s P_{tr}$**:** probability of successful transmission in a given slot time
- $E[P]'$**:** average packet payload size considering the modification of [85] to more accurately model the backoff freezing operation and is given as

  $$E[P]' = \frac{8 \cdot E[P]}{1-B_0} \tag{3.32}$$

  where B_0 refers to the probability that a successfully transmitting STA may access to the first slot after a DIFS. This occurs when it extracts a new backoff counter value equal to zero, i.e., with probability $B_0 = \frac{1}{W}$.
- $1-P_{tr}$**:** probability that a given slot time is empty
- $P_{tr}\,(1-P_s)$**:** probability that a collision occurs in a given slot time
- T_s'**:** duration of a successful transmission considering the backoff freezing modification and the additional backoff slot σ after a DIFS for a listening STA that will decrement its backoff counter by one unit [85], which is computed as

  $$T_s'^x = \frac{T_s^x}{1-B_0} + \sigma \tag{3.33}$$

  Note that T_s^x will vary depending on the analyzed MAC protocol.
- T_c'**:** duration of a collision considering the updated model and the EIFS interval [85]

  $$T_c' = T_c + \sigma \tag{3.34}$$

where T_c=T_{RTS}+δ+T_{EIFS}.

To complete the throughput analysis for the MAC protocols under consideration, T_s^x and α_x are characterized for each protocol as follows.

1) DCF – RTS/CTS:

The successful transmission duration of DCF (T_s^{DCF}) consists of a DIFS interval, an RTS transmission plus the propagation delay, a SIFS interval, a CTS transmission plus the propagation delay, a SIFS interval, a DATA transmission plus the propagation delay, a SIFS interval, and an ACK transmission plus the propagation delay. As a result, T_s^{DCF} is expressed as

$$T_s^{DCF}=T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK}+T_{DIFS}+3\cdot T_{SIFS}+4\cdot\delta \tag{3.35}$$

¡ For the saturation network throughput of DCF, α is equal to one because from a network perspective in each transmission slot a single data packet is transmitted. From the AP perspective, α is equal to $\frac{1}{2(n+1)}$ for the maximum downlink throughput because $P_{tr}P_s$ considers a successful transmission from any of the n STAs or the AP and not for the AP only. Also, the AP can send n data packets from a total number of $2n$ transmissions from the AP and the n STAs. On the contrary, for the saturation downlink throughput α is equal to $\frac{1}{n+1}$ since the AP can only send once every n transmissions from the STAs. From an STA perspective, α takes the same value for the uplink per STA saturation throughput. All these explanations are formulated as

$$\begin{aligned}
S_{DCF}^{net_sat} &\rightarrow \alpha=1\\
S_{DCF}^{dwl_max} &\rightarrow \alpha=\frac{n}{2n\,(n+1)}=\frac{1}{2\,(n+1)}\\
S_{DCF}^{dwl_sat} &\rightarrow \alpha=\frac{1}{n+1}\\
S_{DCF}^{uplpersta_sat} &\rightarrow \alpha=\frac{1}{n+1}
\end{aligned} \tag{3.36}$$

2) BidMAC – RTS/CTS:

The successful transmission duration of BidMAC (T_s^{BidMAC}) comprises the same as T_s^{DCF} but it adds a DATA transmission plus the propagation delay and a SIFS interval. As a result, T_s^{BidMAC} is written as

$$T_s^{BidMAC}{=}T_{RTS}{+}T_{CTS}{+}2{\cdot}T_{DATA}{+}T_{ACK}{+}T_{DIFS}{+}4{\cdot}T_{SIFS}{+}5{\cdot}\delta \tag{3.37}$$

Since BidMAC involves the transmission of two data packets, from a network perspective the saturation network throughput of BidMAC is obtained when α equals 2. Then, from the AP perspective, α is equal to $\frac{n+1}{n+1}$ because the AP is able to send a data packet whenever it receives a data packet from an STA (i.e., n STAs produces n bidirectional transmission from the AP). Finally, from an STA perspective α equals $\frac{1+\frac{1}{n}}{n+1}$ since an STA can transmit when it gets a transmission opportunity ad when it receives a data packet from the AP with probability $\frac{1}{n}$ provided that uniform downlink traffic distribution is assumed. The collection of these mathematical expressions is summarized as

$$\begin{aligned} S_{BidMAC}^{net\_sat} &\rightarrow \alpha{=}2 \\ S_{BidMAC}^{dwl\_sat} &\rightarrow \alpha{=}\frac{n{+}1}{n{+}1}{=}1 \\ S_{BidMAC}^{uplpersta\_sat} &\rightarrow \alpha{=}\frac{1{+}\frac{1}{n}}{n{+}1}{=}\frac{\frac{n+1}{n}}{n{+}1}{=}\frac{1}{n} \end{aligned} \tag{3.38}$$

3) GreenBid – RTS/CTS:

The throughput formulas of GreenBid are the same as those of BidMAC becasue GreenBid has been designed to improve the energy efficiency of the STAs in the network.

C. Energy Efficiency

The energy efficiency η is defined as the amount of energy consumed during the fraction of time that the wireless channel is used to successfully transmit payload bits as

$$\eta = \frac{E[\text{payload information transmitted in a slot time}]}{E[\text{energy consumed in a slot time}]} \tag{3.39}$$

The computations of the different energy efficiencies of the protocols from the network, AP, and per STA perspectives are formulated as explained next.

Network Energy Efficiency:

Considering the expression of S, the network energy efficiency of a given protocol x (being x the DCF, BidMAC, or GreenBid) can be similarly formulated as

$$\eta_x^{net} = \frac{\alpha_x P_s P_{tr} E[P]'}{(1-P_{tr})\,E_\sigma + P_s P_{tr} E_s^{\prime x} + P_{tr}\,(1-P_s)\,E_c^{\prime}} \tag{3.40}$$

where

- E_σ**:** energy consumed during an empty slot time, that is

  $$E_\sigma = \sigma\,(n+1)\,\rho_i \tag{3.41}$$

  where ρ_i is the power consumed for being idle (takes the same value as P_i in Table 3.2) and all devices consume energy for being idle during a slot time σ.

- $E_s^{\prime x}$**:** energy consumed during a successful transmission considering the updated model [85], which is computed as

  $$E_s^{\prime x} = \frac{E_s^x}{1-B_0} + \sigma\,(n+1)\,\rho_i \tag{3.42}$$

  Note that E_s^x will be different depending on the MAC protocol considered.

- E_c'**:** energy consumed during a collision taking into account the modifications of the Binachi's model and the EIFS interval [85]

$$E_c'=E_c+\sigma\left(n+1\right)\rho_i$$
$$E_c=E_t+E_r+E_i\begin{cases}E_t=T_{RTS}E[K]\rho_t\\E_r=T_{RTS}\left(n+1-E[K]\right)\rho_r\\E_i=\left(T_{EIFS}+\delta\right)\left(n+1\right)\rho_i\end{cases}\tag{3.43}$$

where ρ_t and ρ_r are the power consumption values associated with transmit and receive states, respectively (are equivalent to P_t and Pr in Table 3.2), and $E[K]$ is the average number of devices (including the n STAs and the AP) involved in a collision. Note that in a collision $E[K]$ devices consume energy to transmit the RTS packets whereas the rest of devices consume energy to overhear the collision of the RTS packets. All devices consume energy for being idle during an EIFS interval, the propagation delay, and the additional slot time. To compute $E[K]$, the Bayesian theorem is used by which the summation of the probabilities that two or more devices up to $n+1$ devices (considering all possible combinations) cause a collision conditioned that there is a collision in a given slot with probability P_c or $P_{tr}\left(1-P_s\right)$. Thus, $E[K]$ is expressed as

$$E[k]=\frac{\sum_{m=2}^{n+1}\binom{n+1}{m}\tau^m\left(1-\tau\right)^{n+1-m}}{P_c}=\frac{\sum_{m=2}^{n+1}\binom{n+1}{m}\tau^m\left(1-\tau\right)^{n+1-m}}{1-\left(1-\tau\right)^{n+1}-\left(n+1\right)\tau\left(1-\tau\right)^n}\tag{3.44}$$

In the following, α and E_s^x are characterized for the network energy efficiency of each protocol under consideration.

1) DCF – RTS/CTS:

During a successful transmission of DCF, the transmitter consumes energy to transmit the RTS and DATA packets to the receiver whereas the receiver consumes energy to transmit the CTS and ACK packets to the transmitter. The transmitter consumes energy to receive the CTS and ACK packets from the receiver whereas the receiver consumes energy to receive the RTS and data packets. The rest of devices overhear the RTS, CTS, data, and ACK transmissions and all the devices consume energy for being idle during the SIFS intervals and propagation delays interleaving each transmission. As a result, the saturation network energy efficiency of DCF can be computed as

$$\eta_{DCF}^{net\_sat}: \quad \overset{\alpha=1}{E_s = E_t + E_r + E_i \begin{cases} E_t = (T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK})\,\rho_t \\ E_r = (T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK})\,n\rho_r \\ E_i = (T_{DIFS} + 3 \cdot T_{SIFS} + 4 \cdot \delta)\,(n+1)\,\rho_i \end{cases}} \tag{3.45}$$

2) BidMAC – RTS/CTS:

During a successful transmission of BidMAC, the transmitter and the receiver can exchange a pair of data packets with a single RTS/CTS handshake. Based on the explanations given above for the DCF, the saturation network energy efficiency of BidMAC can be expressed as

$$\eta_{BidMAC}^{net\_sat}: \quad \overset{\alpha=2}{E_s = E_t + E_r + E_i \begin{cases} E_t = (T_{RTS} + T_{CTS} + 2 \cdot T_{DATA} + T_{ACK})\,\rho_t \\ E_r = (T_{RTS} + T_{CTS} + 2 \cdot T_{DATA} + T_{ACK})\,n\rho_r \\ E_i = (T_{DIFS} + 4 \cdot T_{SIFS} + 5 \cdot \delta)\,(n+1)\,\rho_i \end{cases}} \tag{3.46}$$

3) GreenBid – RTS/CTS:

The various energy components of the energy consumption of GreenBid during a successful bidirectional transmission are described as follows.

- **Transmission period:** the transmitter consumes energy to send an RTS packet, a data packet, and an ACK packet to the receiver whereas the receiver consumes energy to transmit a CTS packet and a data packet.

- **Reception period:** the transmitter and the receiver consume energy to receive the CTS packet and data packet and the RTS packet, data packet, and ACK packet, respectively. The $n-s$ STAs only consume energy to overhear the RTS and CTS packets as they can switch to the sleep state to save energy. s denotes the number of active STAs, which is just 1 (apart from the AP).

- **Idle period:** all the STAs and the AP consume energy to listen to the wireless channel during a DIFS interval, a SIFS interval and the propagation delays of the RTS and CTS transmissions. After that, only the transmitter and the receiver are awake for the remaining SIFS intervals and propagation delays of the data and ACK transmissions.

- **Switch period:** the $N-S$ sleeping STAs consume energy during the transition from idle to sleep and during the transition from sleep to idle.

- **Sleep period:** the STAs can sleep during the data exchange expect for when they have to switch between idle and sleep states. This happens provided that the sleep period (T_{sl}) is greater than zero. Otherwise, none of the overhearing STAs can sleep and the energy consumed by GreenBid is the same as for BidMAC. The sleep period is computed as

$$T_{sl}{=}2{\cdot}T_{DATA}{+}T_{ACK}{+}3{\cdot}\left(T_{SIFS}{+}\delta\right)-\left(T_{i\rightarrow sl}{+}T_{sl\rightarrow i}\right) \tag{3.47}$$

Based on the above, saturation network energy efficiency of GreenBid can be computed as

$$\eta_{GreenBid}^{net\_sat}:$$

$$\alpha{=}2$$

$$E_s{=}E_t{+}E_r{+}E_i{+}E_{sw}{+}E_{sl}$$

$$\begin{cases} E_t{=}\left(T_{RTS}{+}T_{CTS}{+}2{\cdot}T_{DATA}{+}T_{ACK}\right)\rho_t \\ E_r{=}\left(\left(T_{RTS}{+}T_{CTS}\right)n{+}\left(2{\cdot}T_{DATA}{+}T_{ACK}\right)s\right)\rho_r \\ E_i{=}\left(\left(T_{DIFS}{+}T_{SIFS}{+}2{\cdot}\delta\right)(n{+}1){+}3{\cdot}\left(T_{SIFS}{+}\delta\right)(s{+}1)\right)\rho_i \\ E_{sw}{=}\left(T_{i\to sl}\rho_{i\to sl}{+}T_{sl\to i}\rho_{sl\to i}\right)(n{-}s) \\ E_{sl}{=}T_{sl}\rho_{sl}\left(n{-}s\right) \end{cases} \tag{3.48}$$

where $T_{i\to sl}$, $T_{sl\to i}$, $\rho_{i\to sl}$, $\rho_{sl\to i}$, T_{sl}, and ρ_{sl} take the same values as $T_{i\to s}$, $T_{s\to i}$, $P_{i\to s}$, $P_{s\to i}$, T_s, and P_s in Table 3.2.

AP Energy Efficiency:

Based on (3.40) for the network energy efficiency, the AP energy efficiency is expressed as

$$\eta_x^{ap}{=}\frac{\alpha_x P_s P_{tr} E[P]'}{\left(1{-}P_{tr}\right)E_\sigma{+}P_s P_{tr}\left(\beta_x E_s^{t'x}{+}\gamma_x E_s^{r'x}\right){+}p\tau E_c^{t'}{+}\left(1{-}\tau\right)P_c E_c^{r'}} \tag{3.49}$$

where

- α_x: number of data packets transmitted during a transmission slot (it will be different among the protocols considered).
- β_x: is related to the probability that the AP acts as a transmitter during a transmission slot (it will be different among the protocols considered).
- γ_x: is related to the probability that the AP acts as a receiver during a transmission slot. (it will be different among the protocols considered)

- E_σ: energy consumed by the AP during an empty slot, that is

$$E_\sigma = \sigma \rho_i \tag{3.50}$$

- $E_s^{t'x}$ and $E_s^{r'x}$: energy consumed by the AP during a successful transmission when the AP is transmitting and when it is receiving, respectively, as

$$\begin{aligned} E_s^{t'x} &= \frac{E_s^{tx}}{1-B_0} + \sigma \rho_i \\ E_s^{r'x} &= \frac{E_s^{rx}}{1-B_0} + \sigma \rho_i \end{aligned} \tag{3.51}$$

  Note that $E_s^{t'x}$ and $E_s^{r'x}$ will vary depending on the analyzed MAC protocol.

- τp: probability that the AP is involved in a collision as a transmitter.

- $(1-\tau) P_c$: probability that the AP overhears a collision occurring in the wireless channel.

- $E_c^{t'}$ and $E_c^{r'}$: AP energy consumption during a collision when it is directly involved in the collision and when it just overhears the collision, respectively, as

$$\begin{aligned} E_c^{t'} &= \frac{E_c^t}{1-B_0} + \sigma \rho_i \\ E_c^t &= E_t + E_i \begin{cases} E_t = T_{RTS} \rho_t \\ E_i = (T_{EIFS} + \delta) \rho_i \end{cases} \\ E_c^{r'} &= \frac{E_c^r}{1-B_0} + \sigma \rho_i \\ E_c^r &= E_r + E_i \begin{cases} E_r = T_{RTS} \rho_r \\ E_i = (T_{EIFS} + \delta) \rho_i \end{cases} \end{aligned} \tag{3.52}$$

To compute the AP energy efficiency of the protocols, α, β, and γ and $E_s^{t'x}$ and $E_s^{r'x}$ are characterized for each protocol as follows.

1) DCF – RTS/CTS:

In DCF, when the AP acts as a transmitter, it consumes energy to transmit the RTS and data packets, to receive the CTS and ACK packets, and for being idle during a DIFS interval and the various SIFS intervals and propagation delays. On the contrary, when the AP acts as a receiver, it consumed energy to transmit the CTS and ACK packets, to receive the RTS and data packets, and for listening to the wireless channel for a DIFS, several SIFS, and the propagation delays. This is expressed as

$$E_s^{tDCF}{=}E_t{+}E_r{+}E_i \begin{cases} E_t{=}\left(T_{RTS}{+}T_{DATA}\right)\rho_t \\ E_r{=}\left(T_{CTS}{+}T_{ACK}\right)\rho_r \\ E_i{=}\left(T_{DIFS}{+}3{\cdot}T_{SIFS}{+}4{\cdot}\delta\right)\rho_i \end{cases}$$
$$E_s^{rDCF}{=}E_t{+}E_r{+}E_i \begin{cases} E_t{=}\left(T_{CTS}{+}T_{ACK}\right)\rho_t \\ E_r{=}\left(T_{RTS}{+}T_{DATA}\right)\rho_r \\ E_i{=}\left(T_{DIFS}{+}3{\cdot}T_{SIFS}{+}4{\cdot}\delta\right)\rho_i \end{cases} \tag{3.53}$$

The AP achieves its maximum energy efficiency when it can transmit n data packets to the STAs whereas its saturation energy efficiency is reported when it can only transmit a singe data packet. Thus, α, β, and γ are written as

$$\begin{aligned} \eta_{DCF}^{ap\_max} &\rightarrow \alpha{=}\beta{=}\gamma{=}\frac{n}{n{+}1} \\ \eta_{DCF}^{ap\_sat} &\rightarrow \alpha{=}\beta{=}\gamma{=}\frac{1}{n{+}1} \end{aligned} \tag{3.54}$$

2) BidMAC – RTS/CTS:

In DCF, when the AP acts as a transmitter, it consumes energy to transmit the RTS, data, and ACK packets, to receive the CTS and data packets, and for being idle during a DIFS interval and the various SIFS intervals and propagation delays. On the contrary, when the AP acts as a receiver, it consumes energy to transmit the CTS and data packets, to receive the RTS, data, and ACK packets, and for listening to the wireless channel for a DIFS, several SIFS, and the propagation delays. This is given as

$$E_s^{tBidMAC}{=}E_t{+}E_r{+}E_i \begin{cases} E_t{=}\left(T_{RTS}{+}T_{DATA}{+}T_{ACK}\right)\rho_t \\ E_r{=}\left(T_{CTS}{+}T_{DATA}\right)\rho_r \\ E_i{=}\left(T_{DIFS}{+}4{\cdot}T_{SIFS}{+}5{\cdot}\delta\right)\rho_i \end{cases}$$

$$E_s^{rBidMAC}{=}E_t{+}E_r{+}E_i \begin{cases} E_t{=}\left(T_{CTS}{+}T_{DATA}\right)\rho_t \\ E_r{=}\left(T_{RTS}{+}T_{DATA}{+}T_{ACK}\right)\rho_r \\ E_i{=}\left(T_{DIFS}{+}4{\cdot}T_{SIFS}{+}5{\cdot}\delta\right)\rho_i \end{cases} \tag{3.55}$$

For the AP energy efficiency of BidMAC, α is equal to $\frac{n+1}{n+1}$ as the AP can transmit n data packets in bidirectional mode when the STAs transmit and one data packet when it gets a transmission opportunity (i.e., n+1) given that it only seizes the wireless channel with probability $\frac{P_{tr}P_s}{n+1}$. Similarly, β equals $\frac{1}{n+1}$ because when it gets access to the wireless channel it transmits a single data packet. Finally, γ is equal to $\frac{n}{n+1}$ because it acts as a receiver when the n STAs transmit.

$$\eta_{BidMAC}^{ap\_sat} \begin{cases} \alpha{=}\frac{n+1}{n+1}{=}1 \\ \beta{=}\frac{1}{n+1} \\ \gamma{=}\frac{n}{n+1} \end{cases} \tag{3.56}$$

3) GreenBid – RTS/CTS:

The AP energy efficiency of GreenBid is the same as that of BidMAC because Greenbid has been designed to improve the energy efficiency of the STAs.

Per STA Energy Efficiency:

Similar to (3.49) for the AP energy efficiency, the saturation average per STA energy efficiency is computed as

$$\eta_x^{persta\_sat}=\frac{\alpha_x P_s P_{tr} E[P]'}{(1-P_{tr})\, E_\sigma+P_s P_{tr}\beta_x E_s^x+p\tau E_c^{t'}+(1-\tau)\, P_c E_c^{r'}} \tag{3.57}$$

where α_x is the number of data packets transmitted by an STA during a transmission slot, β_x is related to the probability that an STA transmits in a given slot time, and E_s^x is the energy consumed by an STA during a successful transmission when it is either the transmitter, the receiver, or just overhears.

To complete the derivation of the closed expressions of the saturation average per STA energy efficiencies for the protocols under evaluation, α_x, β_x, and E_s^x are discussed for each protocol in the next lines.

1) DCF – RTS/CTS:

In DCF, an STA transmits once, receives from the AP with probability $\frac{1}{n}$ considering uniform downlink traffic distribution for all the STAs, and with probability $n-\frac{1}{n}$ it overhears. This is expressed as

$$\begin{aligned}
E_s^{DCF}&=E_s^{t'}+\frac{1}{n}E_s^{r'}+\left(n-\frac{1}{n}\right)E_s^{ov'}\\
E_s^{ov'}&=\frac{E_s^{ov}}{1-B_0}+\sigma\rho_i\\
E_s^{ov}&=E_t+E_r+E_i\begin{cases}E_r=(T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK})\,\rho_r\\ E_i=(T_{DIFS}+3\cdot T_{SIFS}+4\cdot\delta)\,\rho_i\end{cases}
\end{aligned} \tag{3.58}$$

where E_s^t and E_s^r have been introduced earlier and $E_s^o v$ is the energy consumed by an STA during a successful transmission where it is not involved. Thus, an STA consumes energy to overhear the RTS, CTS, data, and ACK transmissions addressed to other destinations and for being idle during a DIFS interval and the various SIFS intervals and propagation delays. In addition, the values of α and β are constant as

$$\eta_{DCF}^{persta\_sat} \rightarrow \alpha{=}\beta{=}\frac{1}{n{+}1} \tag{3.59}$$

2) BidMAC – RTS/CTS:

In BidMAC, an STA transmits once, receives from the AP with probability $\frac{1}{n}$ (assuming uniform downlink traffic) and can respond with a data packet, and with probability $n{-}\frac{1}{n}$ it overhears. Therefore, the energy consumed by an STA when BidMAC is executed is given as

$$\begin{aligned}
&E_s^{BidMAC}{=}E_s^{t'}{+}\frac{1}{n}E_s^{r'}{+}\left(n{-}\frac{1}{n}\right)E_s^{ov'}\\
&E_s^{ov'}{=}\frac{E_s^{ov}}{1{-}B_0}{+}\sigma\rho_i\\
&E_s^{ov}{=}E_t{+}E_r{+}E_i\begin{cases}E_r{=}\left(T_{RTS}{+}T_{CTS}{+}2{\cdot}T_{DATA}{+}T_{ACK}\right)\rho_r\\E_i{=}\left(T_{DIFS}{+}4{\cdot}T_{SIFS}{+}5{\cdot}\delta\right)\rho_i\end{cases}
\end{aligned} \tag{3.60}$$

where an STA consumes energy to overhear the additional bidirectional transmission plus the propagation delay and the SIFS interval. Also, the values of α and β are given as

$$\eta_{BidMAC}^{persta\_sat}\begin{cases}\alpha{=}\frac{1{+}\frac{1}{n}}{n{+}1}{=}\frac{\frac{n+1}{n}}{n{+}1}{=}\frac{1}{n}\\\beta{=}\frac{1}{n{+}1}\end{cases} \tag{3.61}$$

3) GreenBid – RTS/CTS:

The energy consumed by an STA in GreenBid during successful transmissions can be split in various energy components that are described as follows:

- **Transmission period:** an STA gets a transmission opportunity once and consumes energy to transmit the RTS, data, and ACK packets and can also consume energy to transmit the CTS and data packets when it receives the RTS and data packets from the AP with probability $\frac{1}{n}$.

- **Reception period:** an STA consumes energy to receive RTS packets when it overhears the RTS transmissions from the $n-1$ STAs to the AP, when the AP transmits an RTS packet to such STA, and when the AP transmits the RTS packet to any of the other $n-1$ STAs. Similarly, an STA consumes energy to receive CTS packets when it has transmitted the RTS packet to the AP, when it overhears the CTS transmissions from the AP to the $n-1$ STAs and when the AP transmits the RTS packets to the other $n-1$ STAs with probability $1-\frac{1}{n}$. Also, an STA consumes energy to receive a data packet when it is the transmitter and when it receives a data packet from the AP with probability $\frac{1}{n}$, where it also consumes energy to receive an ACK packet.

- **Idle period:** an STA always consumes energy for being idle for the DIFS interval, a SIFS interval, and the propagation delays associated with the RTS and CTS transmissions. Then, it consumes energy for listening to the wireless channel for the additional SIFS intervals and propagation delays when it is the actual transmitter or it receives a data packet from the AP with probability $\frac{1}{n}$.

- **Switch period:** an STA consumes energy for switching between idle and sleep states except for when it transmits a data packet or receives a data packet from the AP.

- **Sleep period:** an STA consumes energy for sleeping except for when it transmits a data packet or receives a data packet from the AP.

This is thus formulated as

$$E_s^{GreenBid}=E_t+E_r+E_i+E_{sw}+E_{sl}$$
$$\begin{cases} E_t=\left(T_{RTS}+T_{DATA}+T_{ACK}+\frac{1}{n}\left(T_{CTS}+T_{DATA}\right)\right)\rho_t \\ E_r=\left(nT_{RTS}+\left(n-\frac{1}{n}\right)T_{CTS}+\left(1+\frac{1}{n}\right)T_{DATA}+\frac{1}{n}T_{ACK}\right)\rho_r \\ E_i=\left(\left(T_{DIFS}+T_{SIFS}+2\cdot\delta\right)(n+1)+\left(1+\frac{1}{n}\right)3\cdot\left(T_{SIFS}+\delta\right)\right)\rho_i \\ E_{sw}=\left(T_{i\to sl}\rho_{i\to sl}+T_{sl\to i}\rho_{sl\to i}\right)\left(n-\frac{1}{n}\right) \\ E_{sl}=T_{sl}\rho_{sl}\left(n-\frac{1}{n}\right) \end{cases} \tag{3.62}$$

Likewise, the values of α and β are given as

$$\eta_{GreenBid}^{persta\_sat}\begin{cases} \alpha=\frac{1+\frac{1}{n}}{n+1}=\frac{\frac{n+1}{n}}{n+1}=\frac{1}{n} \\ \beta=\frac{1}{n+1} \end{cases} \tag{3.63}$$

3.4 Simulations Framework

This section evaluates the performances of the protocols by means of both analytical and simulation results. The expressions derived in the previous section are used to discuss the upper-bound performance of the different protocols. In addition, an event-driven simulator coded in Python has been developed for the model validation, where the protocol rules have been implemented.

3.4.1 MAC Protocols Simulation

The simulated scenario consists of a single BSS with an AP and a finite number of associated STAs. The STAs are static. All the STAs are within the transmission range of each other and so they are not hidden from each other. The AP and the STAs generate data packets of constant length with their arrivals following a Poisson distribution. Infinite packet queues are assumed to avoid packet losses due to buffer overflow. The data packets of the STAs are addressed to the AP. The destination of each data packet arriving at the AP is randomly selected among all the STAs of the BSS with equal probability. All packets are received with no errors. To balance the uplink and the downlink, the AP is assumed to carry the amount of traffic corresponding to the total traffic load of all the STAs. For example, if each STA generates 200 kbps, the aggregated traffic generated by 20 STAs will be 4 Mbps. As a result, the traffic load of the AP will also be 4Mbps.

The simulator is composed of three main scripts according to the protocols under evaluation, i.e. DCF, BidMAC, and GreenBid:

- "DCFMACsimulator.py": This script refers to the DCF MAC protocol.
- "BidMACsimulator.py": This script is related to the BidMAC protocol
- "GreenBidMACsimulator.py": This script deals with the GreenBid MAC protocol.

Each of these scripts contains the input parameters required to run the simulation of each protocol. These input parameters can be the simulation time, the number of simulation runs, the number of STAs, among other parameters included in Table 3.1 and Table 3.2. These main scripts are also used to collect the obtained results in an Excel file. Each main script calls an associated class that can be:

- "dcfmac.py": This class contains the DCF MAC rules.
- "bidmac.py": This class includes the BidMAC rules.

- "greenbidmac.py": This class runs the GreenBid MAC rules.

These classes are connected with three subclasses:

- "node.py": This subclass describes an STA or an AP. It contains attributes like the state of a node, if it has packets, the packet box, the output packet queue, and several timers like the slot timer, the DIFS timer, and the backoff timer.

- "packet.py": This subclass describes a packet. It contains attributes like the arrival time, the departure time, the transmission delay, and the destination.

- "simreport.py": This subclass collects all the output values of the simulation, such as throughput, energy efficiency, delay, and energy consumption.

In each of these classes, the MAC rules of each protocol are implemented. First, all the input parameters passed from the main script are registered. Then, the code enters the main function called Run. In the Run function, the AP and the STAs are created as independent entities. Each STA is appended to a list of STAs. A box of packets is then generated for each STA and the AP according to a Poisson-distributed arrival process and considering the available simulation time. After that, a loop that is running until the simulation time is reached begins.

Inside the loop, the code checks what happens in each microsecond of the simulation. A transmit list that includes the potential transmitting nodes in a given time is created. When this list is empty, it means nothing happens in a given microsecond. In each idle microsecond, the states of the AP and each STA are verified and also if the AP or any STA gets data packets to transmit. When this happens, the data packet is removed from the packet box and inserted in the output queue. At this time, the

AP or any STA executes the protocol rules to transmit the data packet. When the AP or an STA has data packets to send, it can be in one of the following states: waiting a DIFS (state 1), running the backoff procedure (state 2), transmitting (state 3), freezing the backoff counter (state 4), and just performing virtual carrier sensing (state 5). When the wireless channel is idle for a DIFS or the backoff counter reaches zero, the AP or an STA is included in the transmit list. When the length of the transmit list is one, this means that there is only one transmitter and so a successful transmission occurs. When the length of the transmit list is longer than one, this means that there are several transmitters and so a collision occurs.

When there is a successful transmission, it is checked if the transmitter is the AP or an STA. Then, several variables are updated and reinitialized and the transmitted data packet is removed from the output queue of the transmitter. It is also verified if the transmitter gets a new data packet while it is transmitting and also if the AP or an STA has got new data packets during the transmission. Depending on the current state of the AP and each STA, their state value is updated according to the protocol rules.

When there is a collision, a similar procedure is followed, except that each device involved in a collision doubles its CW size and randomly selects a new backoff counter.

When the simulation run is over, the simreport subclass is called to collect all the simulation results and return them to the main script.

3.4.2 Analytical and Simulation Results

The results are shown in terms of throughput, energy efficiency, and energy consumption, considering different values for the traffic load, MSDU length, PHY data rate, number of STAs, wakeup transition coefficient (α) and awake/sleep transition time. All simulation runs were repeated 10

Table 3.3: Maximum Gains vs. Traffic Load

Saturation Traffic Load	BidMAC vs. DCF	GreenBid vs. DCF
Network throughput	0.27	0.27
Downlink throughput	11.97	11.97
Average uplink per STA throughput	-0.33	-0.33
Network energy efficiency	0.26	0.81
AP energy efficiency	11.33	11.33
Average per STA energy efficiency	-0.34	-0.03

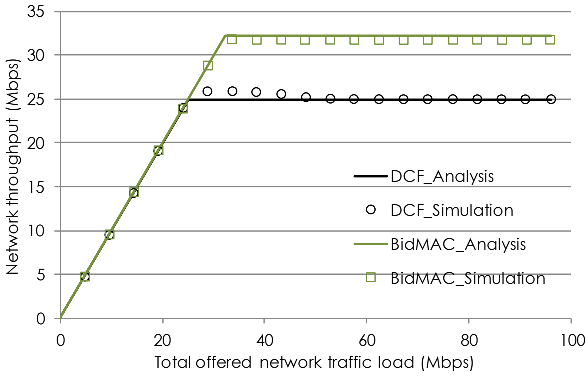
times for the duration of 15 s each. The simulation results in the plots are obtained with a 95% confidence interval lower than 0.03.

Traffic Load

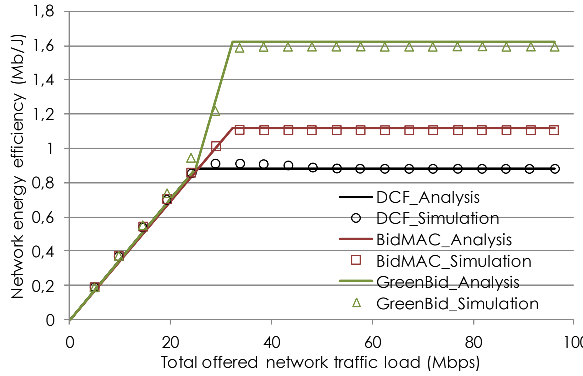
The throughput and energy efficiency versus the traffic load are plotted in Fig. 3.7. The results are obtained for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, a wakeup transition coefficient of 1.5, and an awake/sleep transition time of 500 μs (i.e., 250 μs each transition).

Fig. 3.7a and Fig. 3.7b show the network throughput and energy efficiency, respectively. In general, the throughput and energy efficiency of the protocols increase as the traffic load increases, until a stable value is shown when the network enters the saturation state. It can be seen that the proposed BidMAC and GreenBid protocols outperform DCF when the traffic load is high. Table 3.4 records the maximum gains of the protocols versus DCF in terms of throughput and energy efficiency versus the traffic load. The maximum throughput gain of BidMAC versus DCF is 0.27 whereas the maximum energy efficiency gain of GreenBid versus DCF is 0.81.

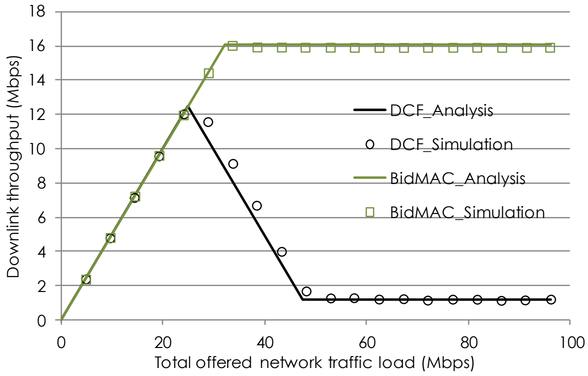
Fig. 3.7c and Fig. 4.6d show the downlink throughput and the AP energy efficiency. The throughput and energy efficiency of DCF increase linearly as the traffic load increases. However, when the traffic load is



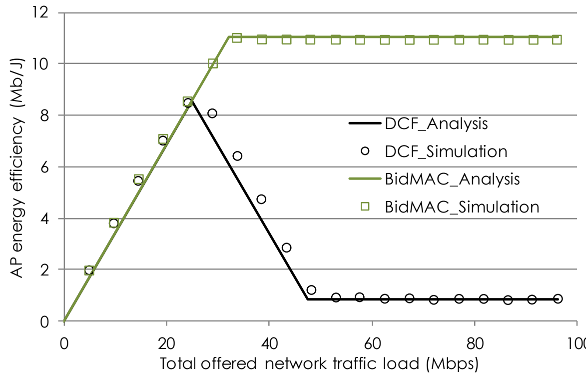
(a) Network throughput



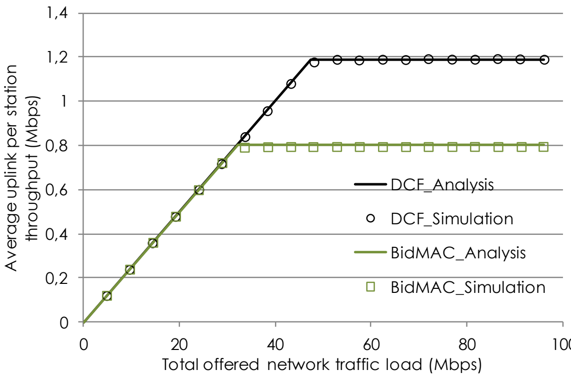
(b) Network energy efficiency



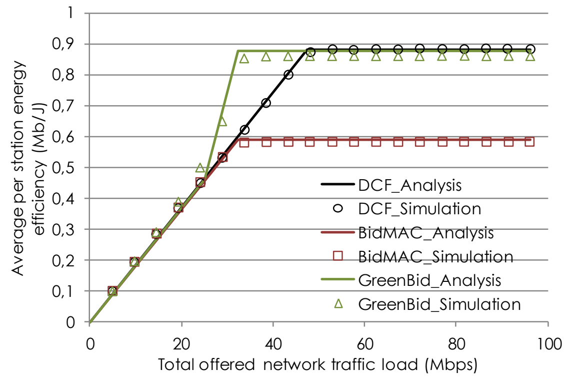
(c) Downlink throughput



(d) AP energy efficiency



(e) Average uplink per STA throughput



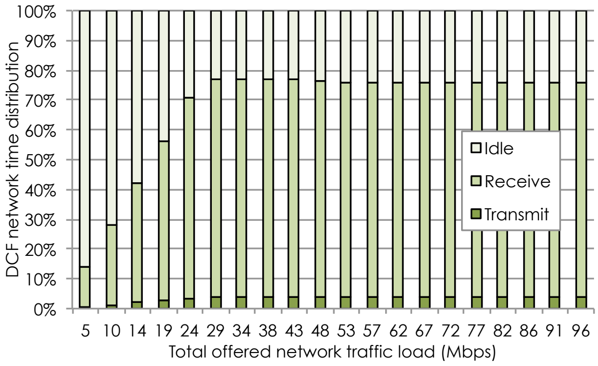
(f) Average per STA energy efficiency

Figure 3.7: Throughput and energy efficiency of the contention-based MAC protocols versus the traffic load

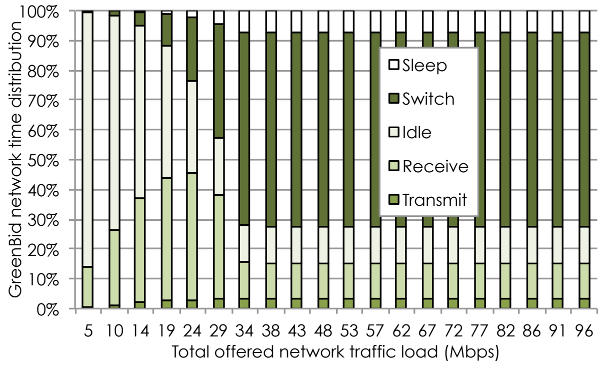
above 20 Mbps, the throughput and energy efficiency of DCF decreases dramatically, due to the DCF fairness, until saturation. On the contrary, BidMAC is able to improve the throughput and energy efficiency of DCF for loads above 20 Mbps. As shown in Table 3.4, the maximum throughput gain of BidMAC versus DCF is 11.97 whereas the energy efficiency gain is 11.33.

Fig. 3.7e and Fig. 3.7f show the average uplink per STA throughput and average per STA energy efficiency. It can be seen that BidMAC reduces the throughput and energy efficiency of DCF for the STAs in uplink in order to balance the uplink and the downlink. However, GreenBid is able to compensate for the reduction of energy efficiency of BidMAC and can almost reach the energy efficiency of DCF. Table 3.4 shows that the maximum throughput gain of BidMAC versus DCF is -0.34 whereas the maximum energy efficiency gain of GreenBid versus DCF is -0.03.

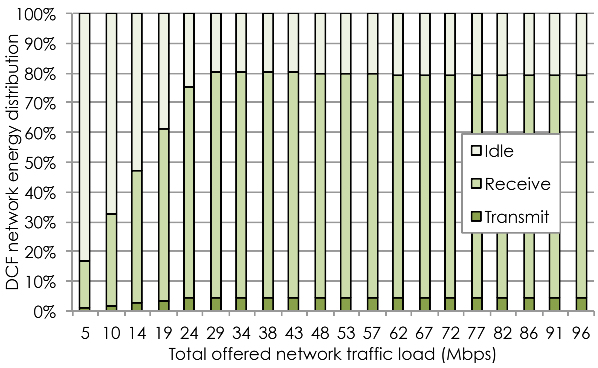
In Fig. 3.8, the contribution of each operational state to the overall energy consumption of the DCF and GreenBid protocols is studied as the traffic load increases. Also, the amount of time that is spent in each of these states is shown. Fig. 3.8a and Fig. 3.8b illustrate the network time distribution of the DCF and GreenBid protocols. In Fig. 3.8c and Fig. 3.8d, the network energy distributions of the DCF and GreenBid protocols are plotted. It can be seen that in DCF most of the time and most of the energy resources (up to 80%) are dedicated to listening activities when the traffic load is low. When the traffic load is high, most of the time and most of the energy resources (up to 75%) are dedicated to receiving and overhearing activities. On the other hand, GreenBid reduces significantly the time and energy consumed for receiving packets. However, it introduces the components of time and energy consumed for sleeping and switching between idle and sleeping. While the time and energy consumed during sleeping periods have a small contribution (up to 7.5%), the energy



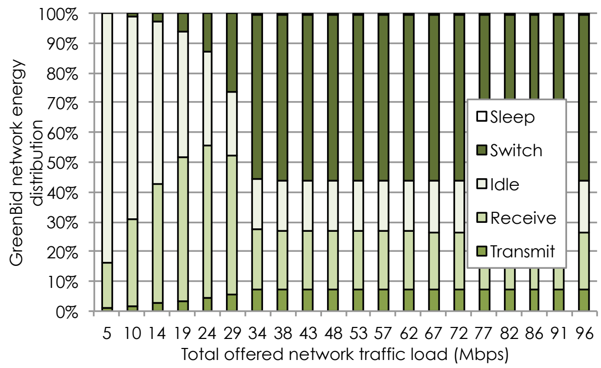
(a) DCF: network time distribution



(b) GreenBid: network time distribution



(c) DCF: network energy distribution



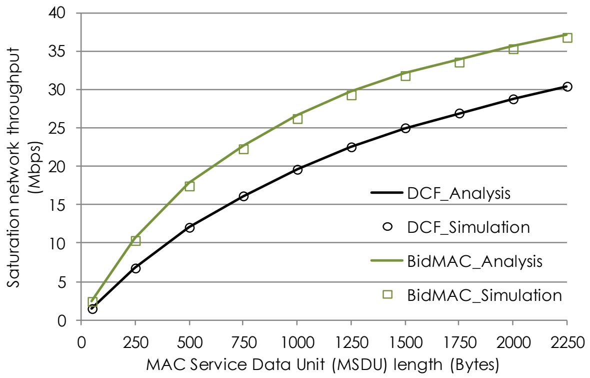
(d) GreenBid: network energy distribution

Figure 3.8: Distribution of time and energy consumption of the contention-based MAC protocols versus the traffic load

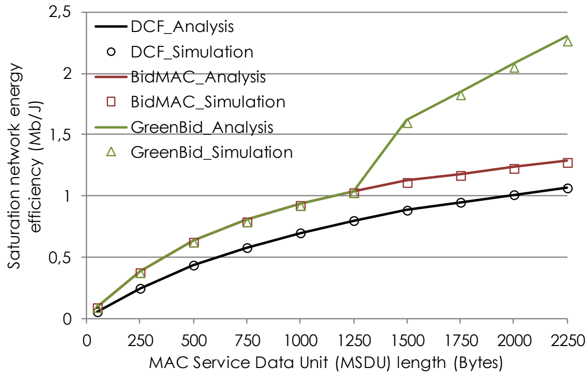
consumed during switch periods has a strong influence on the overall time and energy consumption (up to 65%). These results show the importance of considering the transitions between awake and sleep states in the energy efficiency analysis of energy-efficient MAC protocols based on low-power states.

MSDU Length

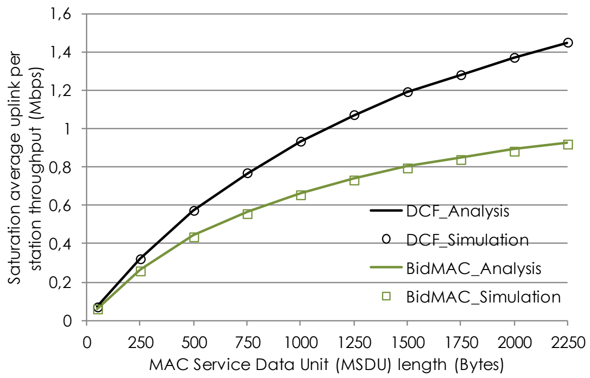
Fig. 3.9 shows the saturation throughput and energy efficiency versus the MSDU length. The results are plotted for a PHY data rate of 54 Mbps, a wakeup transition coefficient of 1.5, and an awake/sleep transition time



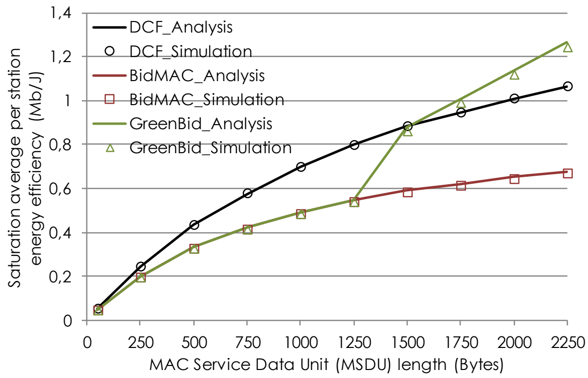
(a) Saturation network throughput



(b) Saturation network energy efficiency



(c) Saturation average uplink per STA throughput



(d) Saturation average per STA energy efficiency

Figure 3.9: Saturation throughput and energy efficiency of the contention-based MAC protocols versus the MSDU length

of 500 μs (i.e., 250 μs each transition).

The saturation network throughput is plotted in Fig. 3.9a. In general, the throughput of the protocols under evaluation increases as the data payload increases since more information is transmitted. It is seen that BidMAC outperforms DCF for all MSDU lengths. However, Table 3.4 shows that the throughput gain decreases as the packet length increases, due to the stronger influence of the data transmission on the overall transmission time. The maximum gain of 0.62 is achieved for an MSDU length of 50 bytes and the minimum gain of 0.21 is shown for an MSDU length of 2250 bytes.

Table 3.4: Maximum Gains vs. MSDU Length

MSDU Length	Throughput BidMAC vs. DCF		Energy Efficiency GreenBid vs. DCF	
	Network	Average Per STA	Network	Average Per STA
50 bytes	0.62	-0.15	0.61	-0.15
250 bytes	0.53	-0.2	0.52	-0.2
500 bytes	0.45	-0.25	0.43	-0.25
750 bytes	0.38	-0.28	0.36	-0.28
1000 bytes	0.34	-0.30	0.32	-0.31
1250 bytes	0.3	-0.32	0.28	-0.33
1500 bytes	0.27	-0.33	0.81	-0.03
1750 bytes	0.25	-0.35	0.92	0.05
2000 bytes	0.23	-0.36	1.03	0.11
2250 bytes	0.21	-0.37	1.12	0.17

The saturation network energy efficiency is plotted in Fig. 3.9b. Similar conclusions can be drawn for the protocols except for GreenBid. The energy efficiency of GreenBid increases as that of BidMAC until the packet length is sufficiently long to let the STAs enter the sleep state within a data exchange. This corresponds to a packet length that makes the sleep period (T_s) be greater than zero. For a data rate of 54 Mbps, the critical MSDU length is 1250 bytes for which the sleep period is zero. For MSDU lengths above this value, the energy efficiency of GreenBid increases significantly showing outstanding gains in comparison with DCF and BidMAC. Table 3.4 shows that the maximum energy efficiency gain of GreenBid versus DCF is achieved for an MSDU length of 2250 bytes, where the gain is 1.12. This is because the available time for sleeping also increases when the MSDU length increases. As a result, the STAs can sleep longer and save more energy.

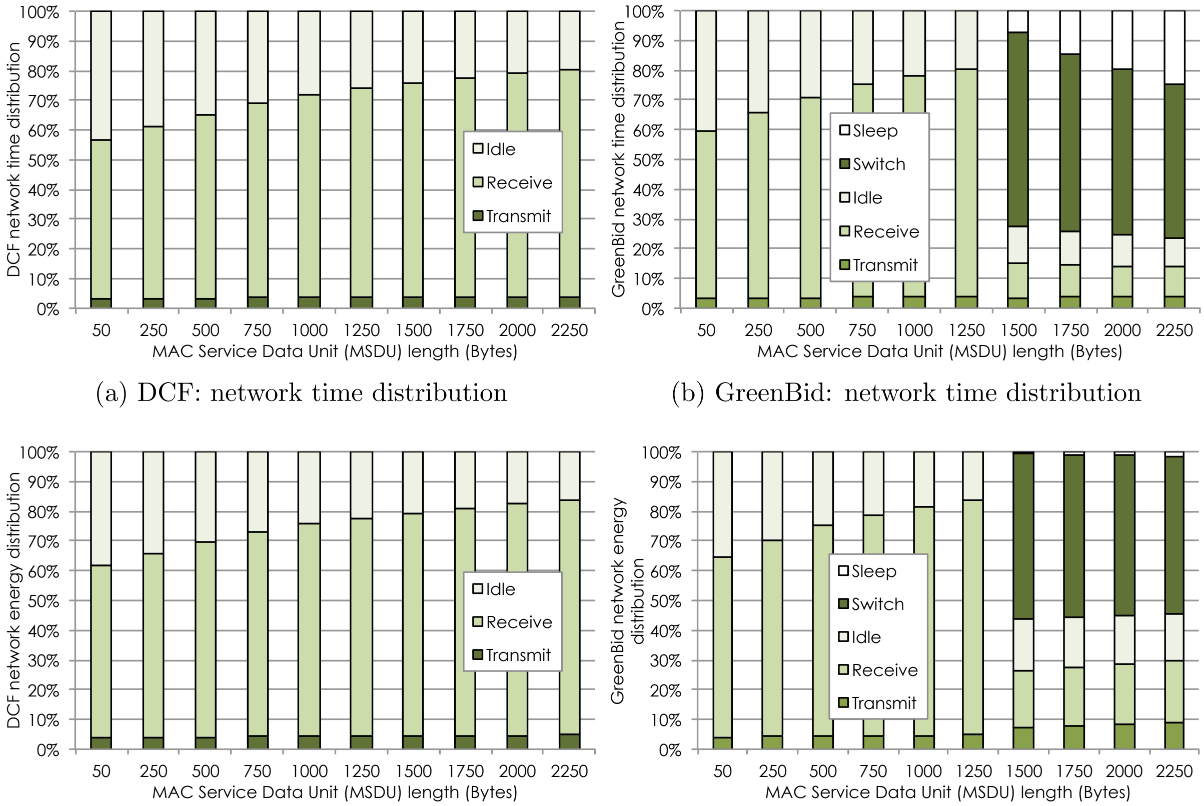
The saturation average uplink per STA throughput is illustrated in Fig. 3.9c. The throughput of BidMAC is always lower than the throughput of DCF in order to balance the uplink and the downlink. It is observed that the difference between the two lines increases as the MSDU length increases. As shown in Table 3.4, the throughput gain of BidMAC versus DCF ranges from -0.15 to -0.37.

Fig. 3.9d presents the average per STA energy efficiency. Similar conclusions can be drawn for the protocols except for GreenBid. The energy efficiency of GreenBid increases as that of BidMAC until the MSDU length is longer than 1250 bytes. In Table 3.4, it can seen that the energy efficiency gain of GreenBid versus DCF is positive for MSDU lengths above 1500 bytes. The maximum gain of 0.17 is achieved for an MSDU length of 2250 bytes.

The network time and energy distributions of the DCF and GreenBid protocols versus the MSDU length are provided in Fig. 3.10. The network time distribution of each protocol is shown in Fig. 3.10a and Fig. 3.10b, respectively. The network energy distribution of each protocol is presented in Fig. 3.10c and Fig. 3.10d. It can seen that for DCF most of the energy and time resources (up to 90%) are spent for receiving and overhearing activities. The share of time and energy consumed during reception periods increases with longer packet lengths. On the contrary, GreenBid shows a similar behavior to that of DCF until the MSDU length is 1250 bytes. Then, the STAs can go to sleep in data exchanges where they are not involved and so the switch and sleep periods play an important role in the overall time and energy consumption. it can be observed that the contribution of switching between idle and sleeping decreases as the MSDU length increases because the contribution of sleeping increases.

PHY Data Rate

Fig. 3.11 shows the throughput and energy efficiency versus the PHY data rate. The results are obtained for an MSDU length of 1500 bytes, a



(a) DCF: network time distribution

(b) GreenBid: network time distribution

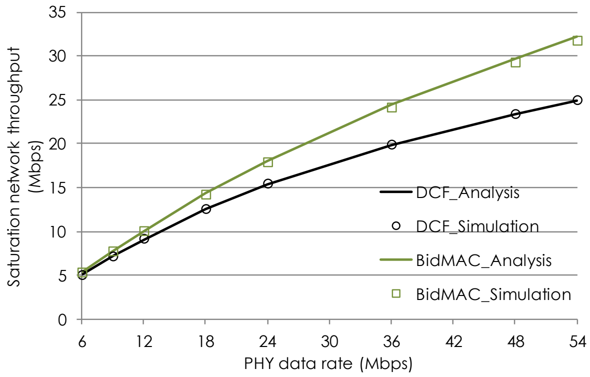
(c) DCF: network energy distribution

(d) GreenBid: network energy distribution

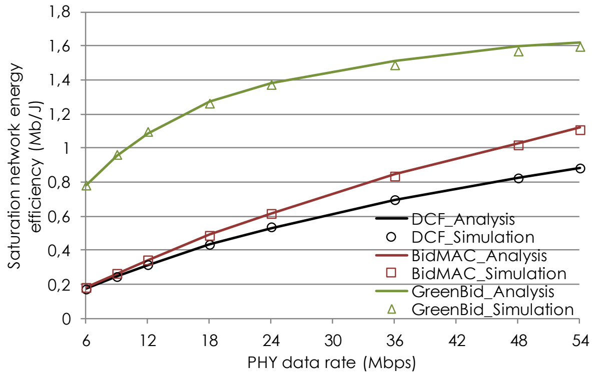
Figure 3.10: Distribution of time and energy consumption of the contention-based MAC protocols versus the MSDU length

wakeup transition coefficient of 1.5, and an awake/sleep transition time of 500 μs (i.e., 250 μs each transition).

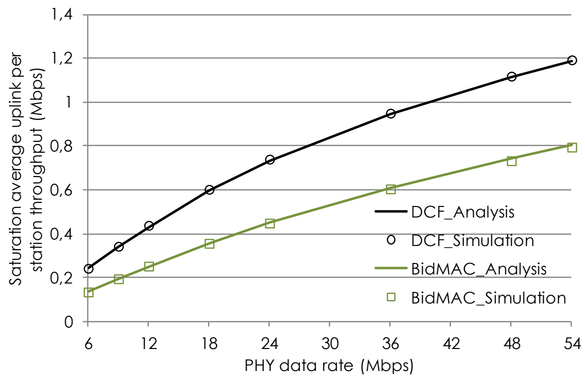
The saturation network throughput is depicted in Fig. 3.11a. The throughput of each protocol increases as the data rate increases since the time to transmit a data packet decreases. The BidMAC protocol outperforms the DCF protocol for all data rates and can achieve higher gains as the data rate increases. This can be understood by the explanations given above for the MSDU length. Table 3.5 records the maximum gains of the proposed protocols versus the PHY data rate. The throughput gain of BidMAC versus DCF ranges from 0.06 to 0.27.



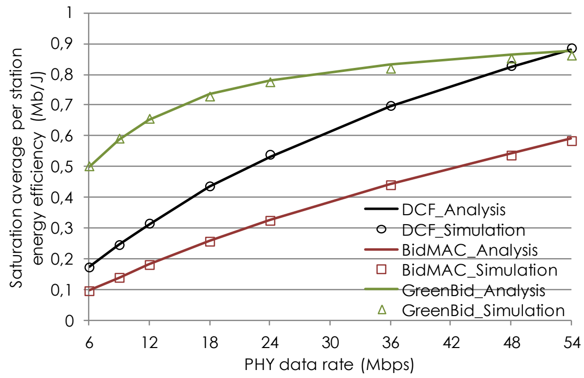
(a) Saturation network throughput



(b) Saturation network energy efficiency



(c) Saturation average uplink per STA throughput



(d) Saturation average per STA energy efficiency

Figure 3.11: Saturation throughput and energy efficiency of the contention-based MAC protocols versus the PHY data rate

The saturation network energy efficiency is plotted in Fig. 3.11b. The energy efficiency of the DCF and BidMAC protocols shows great similarities to what is shown in Fig. 3.11a for the throughput. In contrast, GreenBid significantly improves DCF and BidMAC for all data rates. Furthermore, the highest gain is achieved for the lowest data rate, as shown in Table 3.5. Then, it decreases as the data rate increases. The main reason for this is that the transmission time of each single packet increases as the data rate decreases. Therefore, the STAs can remain longer in the sleep state during data exchanges. The maximum energy efficiency gains of GreenBid versus DCF vary from 3.53 to 0.81.

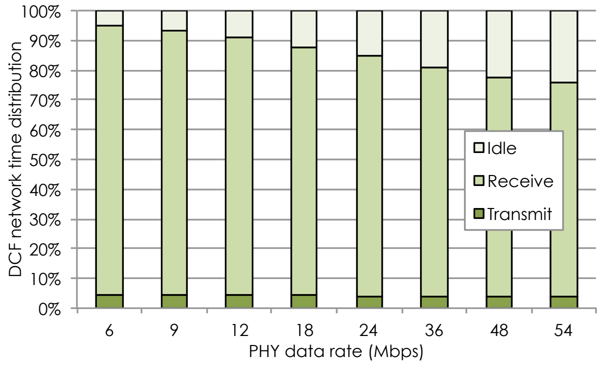
Table 3.5: Maximum Gains vs. PHY Data Rate

PHY Rate	Throughput BidMAC vs. DCF		Energy Efficiency GreenBid vs. DCF	
	Network	Average Per STA	Network	Average Per STA
6 Mbps	0.06	-0.45	3.53	1.9
9 Mbps	0.08	-0.43	2.91	1.41
12 Mbps	0.09	-0.42	2.47	1.08
18 Mbps	0.13	-0.41	1.90	0.67
24 Mbps	0.16	-0.39	1.55	0.44
36 Mbps	0.21	-0.36	1.13	0.17
48 Mbps	0.25	-0.34	0.9	0.03
54 Mbps	0.27	-0.33	0.81	-0.03

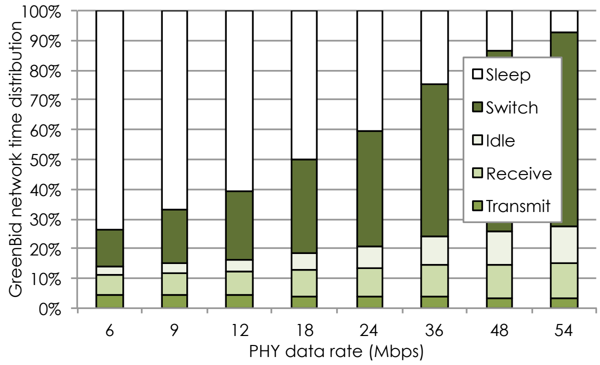
Fig. 3.11c presents the saturation average uplink per STA throughput. The throughput of BidMAC is always lower than that of DCF to provide a balanced share of the channel between the AP in downlink and the STAs in uplink. Table 3.5 shows that the throughput reduction decreases as the data rate increases from -0.45 to -0.33.

In Fig. 3.11d, the saturation average per STA energy efficiency is plotted. The GreenBid protocol is able to outperform the DCF and BidMAC protocols for all data rates except for 54 Mbps. The main reason for this is that the sleep period at 54 Mbps is not long enough to allow GreenBid to compensate for the reduction of energy efficiency in BidMAC. As provided in Table 3.5, the maximum energy efficiency gain of GreenBid is between 1.9 and -0.03 as the data rate increases.

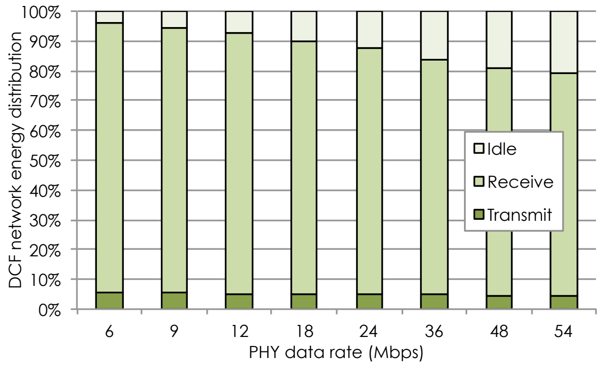
The impact of the PHY data rate on the time spent and energy consumed in the different operational states for the DCF and GreenBid protocols is evaluted in Fig. 3.12. Fig. 3.12a refers to the DCF network time distribution whereas Fig. 3.12b plots the GreenBid network time distribution. In Fig. 3.12c, the DCF network energy distribution is shown and in Figure 3.12d, the GreenBid network energy distribution is presented.



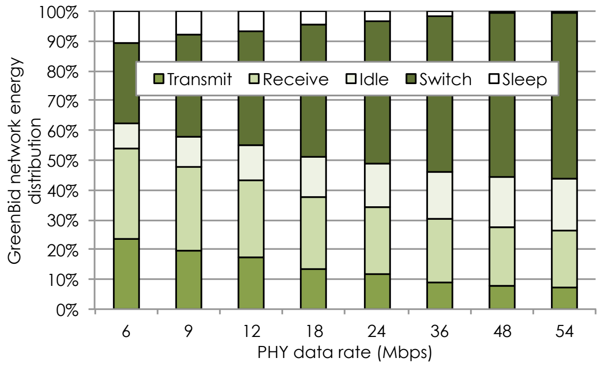
(a) DCF: network time distribution



(b) GreenBid: network time distribution



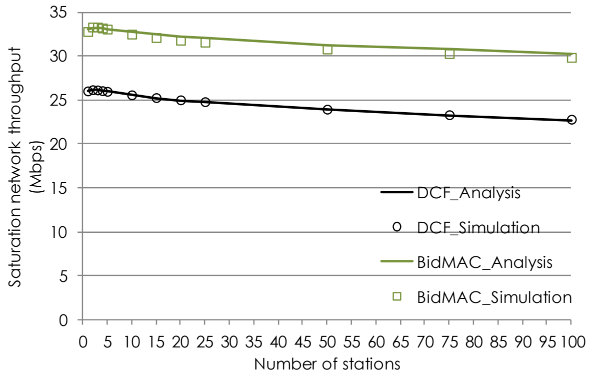
(c) DCF: network energy distribution



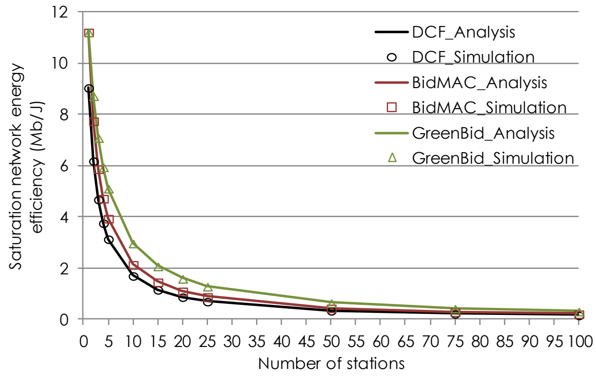
(d) GreenBid: network energy distribution

Figure 3.12: Distribution of time and energy consumption of the contention-based MAC protocols versus the PHY data rate

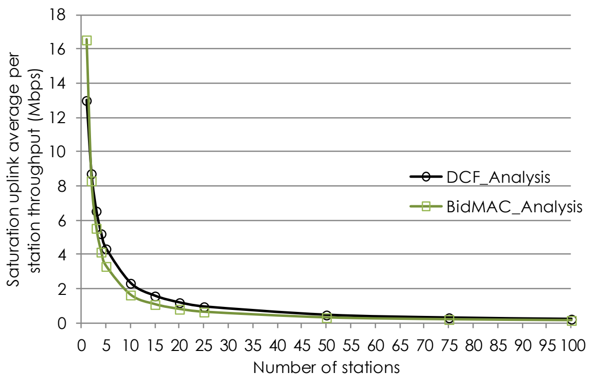
In DCF, the share of time and energy consumed during reception periods decreases as the data rate increases because the data transmission time decreases. In contrast, for GreenBid the network remains in the sleep state for more than 70% of time for a data rate of 6 Mbps and only 12% during switching periods. The share of energy consumption for 6 Mbps is 10% and less than 30% for sleeping and switching. However, when the data rate increases, the share of energy consumption during sleep periods is negligible (less than 1%). In addition, the energy consumed during switching periods can represent up to 55% of the overall network energy consumption and 65% of the overall time.



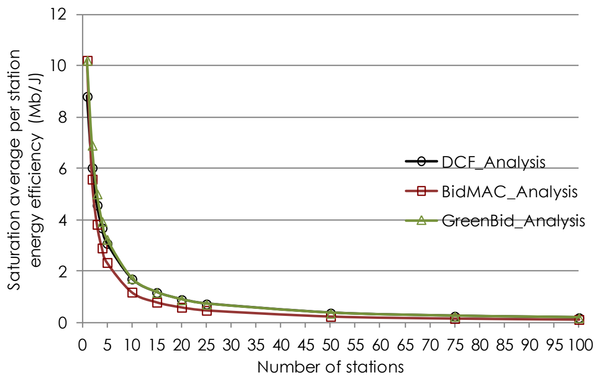
(a) Saturation network throughput



(b) Saturation network energy efficiency



(c) Saturation average uplink per STA throughput



(d) Saturation average per STA energy efficiency

Figure 3.13: Saturation throughput and energy efficiency of the contention-based MAC protocols versus the number of STAs in the network

Number of STAs

The throughput and energy efficiency versus the number of STAs are shown in Fig. 3.13. The results are obtained for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, a wakeup transition coefficient of 1.5, and an awake/sleep transition time of 500 μs (i.e., 250 μs each transition).

Fig. 3.13a shows the saturation network throughput as the number of STAs increases. In general, the throughput of the DCF and BidMAC protocols increases for small numbers of STAs and then decreases as the number of STAs increases. This non-linear behavior is due to the concur-

Table 3.6: Maximum Gains vs. Number of STAs

Num. of STAs	Throughput	Energy Efficiency	
	BidMAC vs. DCF	BidMAC vs. DCF	GreenBid vs. DCF
1	0.26	0.24	0.24
2	0.27	0.25	0.41
3	0.27	0.25	0.51
4	0.27	0.25	0.58
5	0.27	0.25	0.63
10	0.27	0.25	0.74
15	0.27	0.25	0.78
20	0.27	0.26	0.81
25	0.27	0.26	0.82
50	0.27	0.27	0.86
75	0.27	0.28	0.87
100	0.27	0.29	0.88

rent competition of the STAs and the AP for the access to the wireless channel. When the number of STAs is small, there are few collisions and the time between consecutive transmissions is small. When the number of STAs increases, the number of collisions increases and so the throughput decreases. The BidMAC protocol outperforms DCF in all cases. As shown in Table 3.6, the maximum throughput gain of 0.27 is stable across the different numbers of STAs.

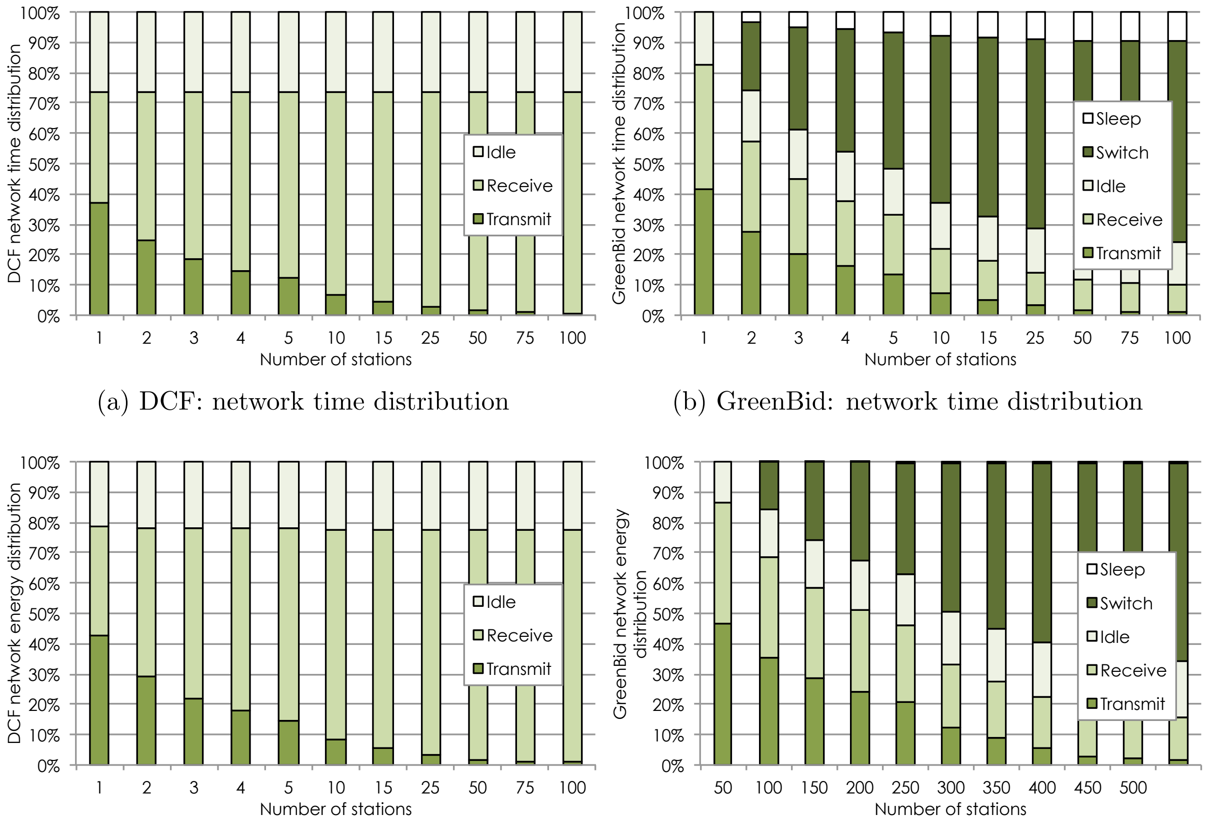
The saturation network energy efficiency versus the number of STAs is presented in Fig. 3.13b. In general, the energy efficiency of the protocols decreases as the number of STAs increases because more STAs are overhearing packets during data transmissions. The BidMAC protocol performs better than DCF whereas the GreenBid protocol achieves the highest energy efficiency. Table 3.6 shows that the maximum energy efficiency gain increases as the number of STAs increases from 0.24 to 0.88.

Fig. 3.13c presents the saturation average uplink per STA throughput. The throughput of BidMAC is always lower than that of DCF to provide a balanced share of the channel between the AP in downlink and the STAs in uplink. In Fig. 3.13d, the saturation average per STA energy efficiency is plotted. The GreenBid protocol is able to outperform the DCF and BidMAC protocols for small to medium numbers of STAs (i.e., up to 20) and then DCF performs slightly better with a gain that increases as the number of STAs increase above 20 STAs. The main reason for this is that the sleep period is not long enough to allow GreenBid to compensate for the reduction of energy efficiency in BidMAC.

In Fig. 3.14, the influence of the number of STAs on the distributions of time and energy consumption of the DCF and GreenBid protocols in the different operational states is analyzed. Fig. 3.14a and Fig. 3.14b show the network time distribution of the DCF and GreenBid protocols, respectively. Fig. 3.14c and Fig. 3.14d illustrate the network energy distribution of the protocols, respectively. In DCF, the share of receiving becomes more significant as the number of STAs increases. When the number of STAs is very high, the share of transmitting is negligible (less than 1%) whereas the share of receiving is around 70%. In GreenBid, when the number of STAs is longer than one, there is at least one STA that can enter the sleep state. As the number of STAs increases, the share of sleeping and the share of switching increase because more STAs go to sleep. The share of sleeping can reach up to 10% and the share of switching can be up to 65%.

Wakeup Transition Coefficient

Fig. 3.15 shows the energy efficiency and time and energy distributions of the protocols versus the wakeup transition coefficient. This coefficient determines the amount of energy consumed in the transition between sleep and idle states having as reference the value of power consumed in the idle state. The higher the value of the wakeup transition coefficient is, the higher the energy consumed in the transition between sleep and idle states



(a) DCF: network time distribution

(b) GreenBid: network time distribution

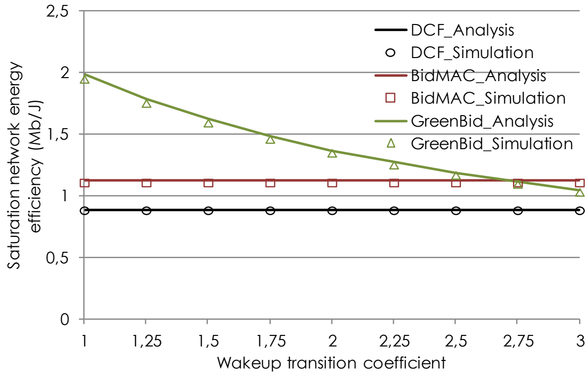
(c) DCF: network energy distribution

(d) GreenBid: network energy distribution

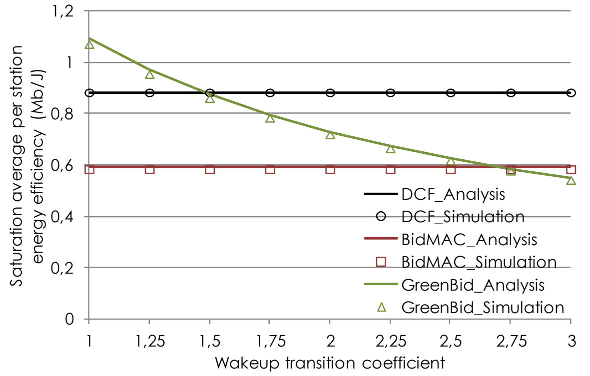
Figure 3.14: Distribution of time and energy consumption of the contention-based MAC protocols versus the number of STAs in the network

is. The results are obtained for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, and an awake/sleep transition time of 500 μs (i.e., 250 μs each transition).

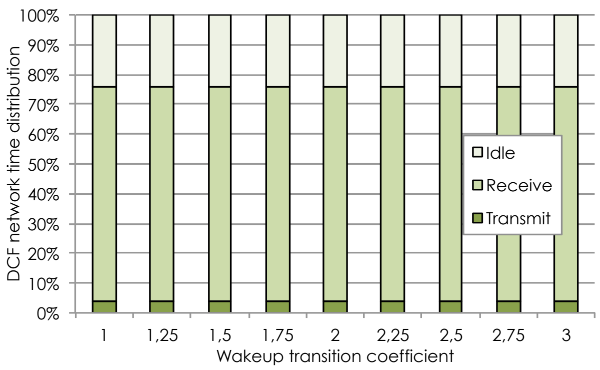
The saturation network energy efficiency is plotted in Fig. 3.15a. The value of the wakeup transition coefficient only affects the energy efficiency of the GreenBid protocol. As the value of the wakeup transition coefficient increases, the energy efficiency of GreenBid decreases and approaches the energy efficiency of BidMAC. The critical value of the wakeup transition coefficient that makes the energy efficiency of GreenBid be the same as that of BidMAC is 2.75. Table 3.7 records the maximum gains versus the



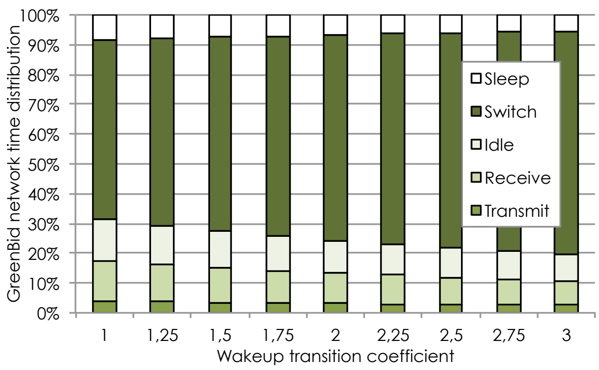
(a) Saturation network energy efficiency



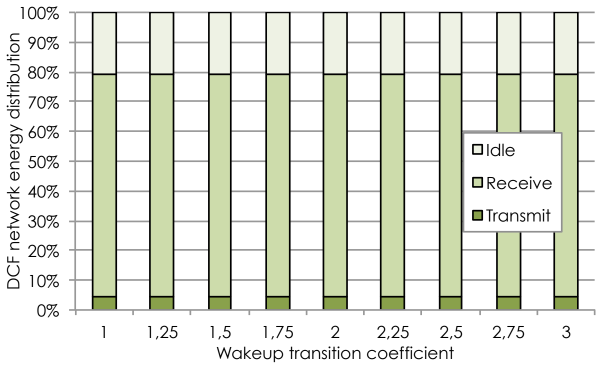
(b) Saturation average per STA energy efficiency



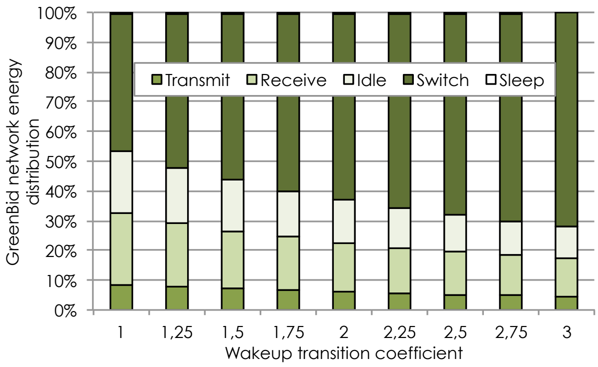
(c) DCF: network time distribution



(d) GreenBid: network time distribution



(e) DCF: network energy distribution



(f) GreenBid: network energy distribution

Figure 3.15: Energy efficiency and time and energy distributions of the contention-based MAC protocols versus the wakeup transition coefficient

Table 3.7: Maximum Energy Efficiency Gains vs. Wakeup Transition Coefficient

Wakeup	Network		Average Per STA	
Transition	GreenBid vs.	GreenBid vs.	GreenBid vs.	GreenBid vs.
Coefficient	DCF	BidMAC	DCF	BidMAC
1	1.21	0.76	0.21	0.83
1.25	0.99	0.58	0.08	0.63
1.5	0.81	0.44	-0.03	0.47
1.75	0.66	0.32	-0.11	0.34
2	0.53	0.22	-0.19	0.23
2.25	0.42	0.13	-0.25	0.14
2.5	0.32	0.06	-0.3	-0.06
2.75	0.24	-0.01	-0.35	-0.01
3	0.17	-0.07	-0.39	-0.07

wakeup transition coefficient. The maximum gain of GreenBid versus DCF ranges from 1.21 to 0.17 whereas the gain versus BidMAC varies between 0.76 and -0.07.

Also, the saturation average per STA energy efficiency is shown in Fig. 3.15b. The energy efficiency of GreenBid also decreases as the wakeup transition coefficient increases, as shown in Fig. 3.15a for the network energy efficiency. However, for the per STA energy efficiency the critical value of the wakeup transition coefficient that makes the energy efficiency be the same as that of DCF is around 1.5. For values above 1.5, the energy efficiency of GreenBid is lower than that of DCF and approaches the energy efficiency of BidMAC as similarly shown for the network energy efficiency in Figure 14a. As shown in Table 3.7, the gain of GreenBid versus DCF is between 0.21 and -0.39 and between 0.83 and -0.07 versus BidMAC.

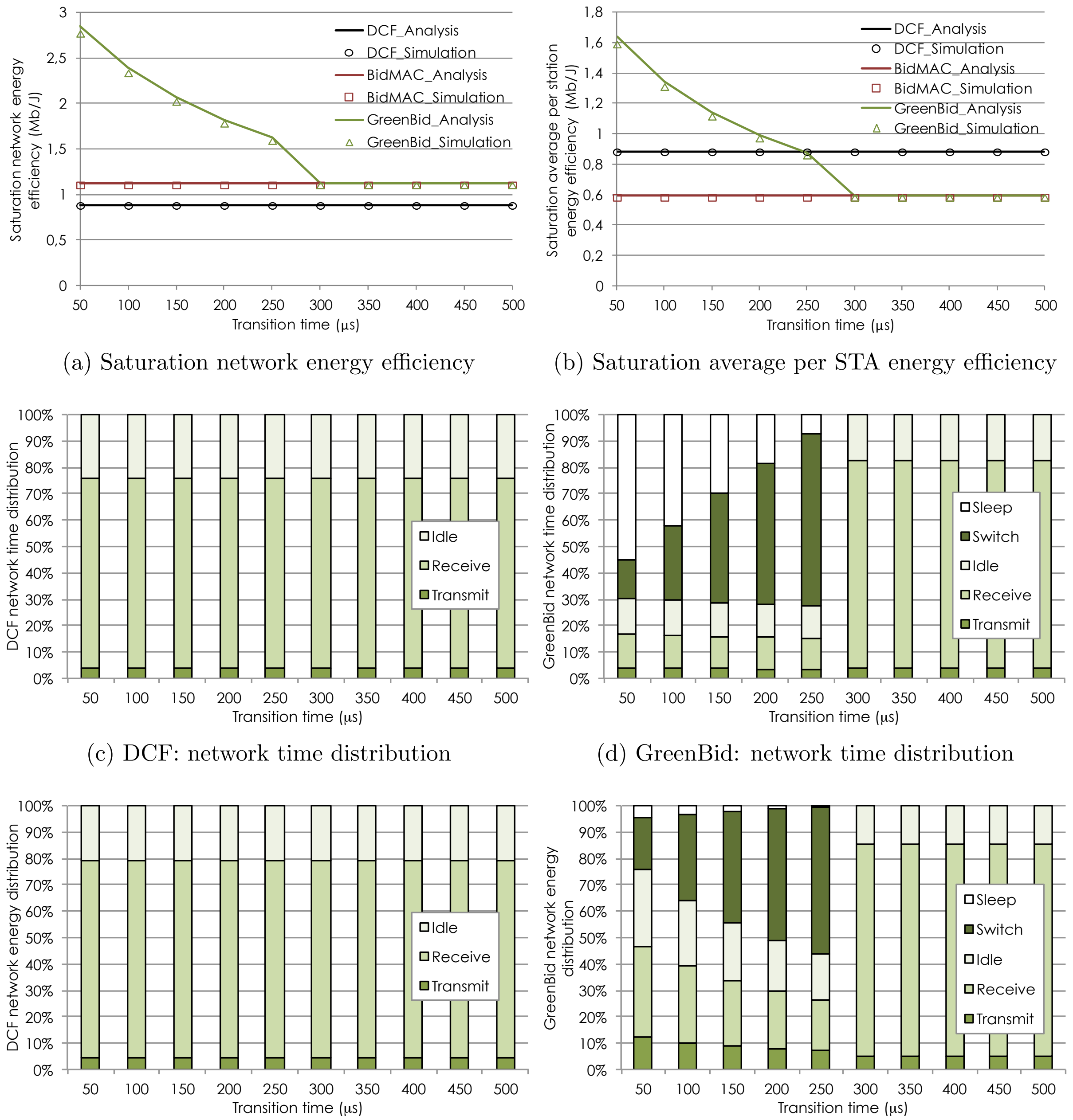
Finally, the evaluation of the impact of the wakeup transition coefficient on the overall time and energy consumption distributions is presented as follows. Fig. 3.15c and Fig. 3.15d show the network time distribution of DCF and GreenBid, respectively. Similarly, Fig. 3.15e and Fig. 3.15f represent the network energy distribution of DCF and GreenBid, respectively. In GreenBid, it can seen that as the wakeup transition coefficient increases more time and energy is consumed during the switching procedure. A maximum value of 75 % of the overall time and energy consumption corresponds to switching.

Awake/Sleep Transitions Time

The energy efficiency and time and energy distributions of the protocols versus the awake/sleep transitions time are shown in Fig. 3.16. The transition time determines how much time is spent in the transition from idle to sleep and the transition from sleep to idle. The longer the transition time is, the longer the data transmission time has to be in order to make the sleep period be greater than zero. The results are obtained for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, and a wakeup transition coefficient of 1.5.

Fig. 3.16a shows the saturation network energy efficiency. The value of the transition time only affects the energy efficiency of the GreenBid protocol. As the transition time increases, the energy efficiency of GreenBid decreases, since the sleep period also decreases. The critical value of the transition time that makes the sleep period be equal to or lower than zero is 300 μs. For transition times above 300 μs, the energy efficiency of GreenBid is the same as that of BidMAC because none of the STAs can go to sleep. The critical value of the transition time may increase or decrease depending on the MSDU length and the PHY data rate. Table 3.8 reports the maximum gains versus the transition time. The gain of GreenBid versus DCF varies between 2.26 and 0.26.

In Fig. 3.16b the saturation average per STA energy efficiency is presented. Similar conclusions to those shown for Fig. 3.16a can be drawn except that the critical value of the transition time that makes the energy



(a) Saturation network energy efficiency

(b) Saturation average per STA energy efficiency

(c) DCF: network time distribution

(d) GreenBid: network time distribution

(e) DCF: network energy distribution

(f) GreenBid: network energy distribution

Figure 3.16: Energy efficiency and time and energy distributions of the contention-based MAC protocols versus the total awake/sleep transitions time

Table 3.8: Maximum Energy Efficiency Gains vs. Awake/Sleep Transitions Time

Transition Time	GreenBid vs. DCF	
	Network	Average per STA
50 μs	2.26	0.88
100 μs	1.72	0.53
150 μs	1.32	0.28
200 μs	1.03	0.11
250 μs	0.81	-0.03
300 μs	0.26	-0.34
350 μs	0.26	-0.34
400 μs	0.26	-0.34
450 μs	0.26	-0.34
500 μs	0.26	-0.34

efficiency of GreenBid be equal to or higher than that of DCF is below 250 μs. As shown in Table 3.8, the maximum gain of GreenBid versus DCF ranges from 0.88 to -0.34.

To conclude, the influence of the transition time on the time and energy distributions of the DCF and GreenBid protocols along the different operational states is studied as follows. Fig. 3.16c illustrates the network time distribution of DCF whereas Fig. 3.16d represents the network energy distribution of GreenBid. Likewise, the DCF network energy distribution is shown in Fig. 3.16d and the GreenBid network energy distribution is plotted in Fig. 3.16e. In GreenBid, when the transition time is very small, with small data packets and fast data rates a positive sleep period can be achieved, thus improving energy efficiency. For example, for a transition time of 50 μs, the STAs remain in the sleep state for more than 50% of time. In addition, the contribution of switching periods to the overall energy consumption is relatively small (around 20%). However, when the transition time increases, the amount of time that the STAs spend in the sleep state decreases whereas the share of energy consumption during switching periods increases. For the critical transition time of 250 μs, the STAs remain in the sleep state for less than 10% of time. In addition, the portion of energy consumed for switching between idle and sleep states is up to 65% of the total energy consumption.

3.5 Experiments Framework

This section describes an experimental implementation of the proposed BidMAC protocol that has been carried out using a programmable wireless platform called WARP [28] and has been tested in a proof-of-concept network formed by an AP and two STAs. There are various available wireless platforms for prototyping at the MAC layer [27]. Among them, WARP (version 3) has been selected because it offers an available open-source reference design that can interoperate with commercial IEEE 802.11a/g devices, acting as either AP or STA. Further details about the WARP platform and its reference design are provided in Appendix A and Appendix B.

The DCF MAC source code of the reference design of WARP has been modified to implement BidMAC. The focus has been put on the evaluation of the experimental throughputs and energy efficiencies of DCF and BidMAC, which have been measured in each node by means of custom-design Python scrips and Energino meters [8] controlled through a custom program developed in Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW). The reader may refer to Appendix C for further details about Energino's hardware and software. In order to validate the accuracy of the experimental implementation, the theoretical throughput and energy efficiency results of DCF and BidMAC presented in the previous section are compared to the experimental results, taking into account various values for relevant system parameters such as the traffic load, packet length, and data rate.

3.5.1 MAC Protocol Implementation

The BidMAC protocol is mainly implemented in the lower-level MAC of the 802.11 reference design of WARP (see Appendix B), i.e., the C code in the CPU Low MicroBlaze core (wlan_mac_dcf.c). The proposed BidMAC implementation allows the AP to transmit a data packet (with an implicit ACK) of the same length and with the same transmission rate as those of the received data packet back to the transmitting STA after a SIFS, upon successful data reception. The main modifications to the existing MAC software of the reference design are described below.

In the wlan_mac_dcf.c file, the static MAC addresses of the WARP v3 nodes considered, namely, an AP and several STAs, are defined to determine which of them is the receiver of a data packet inside the frame_receive function, as follows:

```
<wlan_mac_dcf.c>
// MAC address of WARP node 112 acting as AP
static u8 ap_addr[6] = {0x40, 0xD8, 0x55, 0x04, 0x21, 0x72};
// MAC address of WARP node 339 acting as STA 1
static u8 sta1_addr[6] = {0x40, 0xD8, 0x55, 0x04, 0x22, 0x9E};
// MAC address of WARP node 220 acting as STA 2
static u8 sta1_addr[6] = {0x40, 0xD8, 0x55, 0x04, 0x22, 0x9E};
```

In order to allow the AP to send a data packet together with the ACK packet after valid reception of a data packet after a SIFS, the auto-responder state machine implemented in the reference design of WARP for the transmission of ACK packets is enabled and configured with a new type of data packet called ACK_DATA. This packet type is defined in the wlan_mac_802_11_defs.h (MAC High Framework) as

```
<wlan_mac_802_11_defs.h>
//Define subtype ACK_DATA as type/subtype (10, 1101) as specified in
    table 8.1 of 802.11 2011-2012. The subtype is reserved.
```

```
#define MAC_FRAME_CTRL1_SUBTYPE_ACK_DATA (MAC_FRAME_CTRL1_TYPE_DATA | 0
    xD0)
```

The ACK_DATA packet is created via a new function, called wlan_create_ack_data_frame, that sets the packet type in the frame_control_1 field contained in the header of the packet as

```
<wlan_mac_dcf.c>
int wlan_create_ack_data_frame(void* pkt_buf, mac_header_80211_common*
    common, u8 flags) {
        // Set subtype ACK_DATA
        data_80211_header->frame_control_1 =
            MAC_FRAME_CTRL1_SUBTYPE_ACK_DATA;
        // Set MAC address of AP as the source of the ACK_DATA packet
        memcpy(data_80211_header->address_2,ap_addr,6);
}
```

Since DATA packets are created by the upper-level MAC (wlan_mac_packet_types.c), these modifications are required in the lower-level MAC to prepare an ACK_DATA packet for transmission before reception completes, thus respecting the SIFS requirement.

In the function frame_receive, when a packet of type DATA destined to the AP is received with a valid FCS, an ACK_DATA auto-response is performed with the length and transmission rate of the ACK_DATA packet set equal to those of the received data packet. Also, the reception of an ACK_DATA packet with a valid FCS by an STA involves processing the received packet as an ACK and DATA packet, generating an ACK auto-response and notifying data reception to the upper-level MAC. The function is modified as follows:

```
<wlan_mac_dcf.c>
u32 frame_receive(u8 rx_pkt_buf, u8 rate, u16 length){
        // Check if AP
        if(wlan_addr_eq(eeprom_addr, ap_addr)){
                //Check if a packet of type DATA is received
                if (unicast_to_me && (mpdu_info->state ==
                    RX_MPDU_STATE_FCS_GOOD) && ((rx_header->
                    frame_control_1)==MAC_FRAME_CTRL1_SUBTYPE_DATA)){
```

```
                    // Check source address of the received data
                       packet
                    //Create ACK_DATA
                    txackdata_length = wlan_create_ackdata_frame((
                       void*)(TX_PKT_BUF_TO_ADDR(TX_PKT_BUF) +
                              PHY_TX_PKT_BUF_MPDU_OFFSET), &
                       tx_header_ackdata,
                       MAC_FRAME_CTRL2_FLAG_FROM_DS)
                    // Set LLC header of 8 bytes
                    txackdata_length += sizeof(llc_header);
                    // Set payload from the previous received data
                       packet and subtract mac and llc headers
                       already added before
                    txackdata_length += length - ( sizeof(llc_header
                       ) + sizeof(mac_header_80211) );
                    // Configure auto-responder for ACK_DATA Tx
                    wlan_phy_set_tx_signal(TX_PKT_BUF, ackdata_rate
                       , txackdata_length + WLAN_PHY_FCS_NBYTES);
            }
    }
    // If STA
    else{
            //Check if a packet of type ACK_DATA is received
            if (unicast_to_me && (mpdu_info->state ==
               RX_MPDU_STATE_FCS_GOOD) && ((rx_header->
               frame_control_1)==MAC_FRAME_CTRL1_SUBTYPE_ACK_DATA)){
                    // Create ACK
                    tx_length = wlan_create_ack_frame((void*)(
                       TX_PKT_BUF_TO_ADDR(TX_PKT_BUF_ACK) +
                              PHY_TX_PKT_BUF_MPDU_OFFSET),
                       rx_header->address_2);
                    // Configure auto-responder for ACK Tx
                    wlan_phy_set_tx_signal(TX_PKT_BUF_ACK, tx_rate,
                       tx_length + WLAN_PHY_FCS_NBYTES);
                    // Process ACK_DATA as ACK
                    if((rx_header->frame_control_1) ==
                       MAC_FRAME_CTRL1_SUBTYPE_ACK  || (rx_header->
                       frame_control_1) ==
                       MAC_FRAME_CTRL1_SUBTYPE_ACK_DATA ){
                          return_value |= POLL_MAC_TYPE_ACK;
                    }
                    // Process ACK_DATA as DATA and send it to
                       higher level MAC
```

```
                        if((!WLAN_IS_CTRL_FRAME(rx_header)) || (
                            rx_header->frame_control_1 ==
                            MAC_FRAME_CTRL1_SUBTYPE_ACK_DATA)){
                                wlan_mac_low_frame_ipc_send();
                        }
                }
        }
}
```

Finally, a condition is included in the mpdu_rx_process function contained in the wlan_mac_sta.c file to account received ACK_DATA packets as DATA packets and update reception statistics. This is done as follows:

```
<wlan_mac_sta.c>
void mpdu_rx_process(void* pkt_buf_addr, u8 rate, u16 length) {
        if(((rx_80211_header->frame_control_1 & 0xF) ==
            MAC_FRAME_CTRL1_TYPE_DATA) || ((rx_80211_header->
            frame_control_1 & 0xF) == MAC_FRAME_CTRL1_SUBTYPE_ACK_DATA)
            ) {
                (station_stats->data_num_rx_success)++;
                (station_stats->data_num_rx_bytes) += mpdu_info->length;
        }
}
```

3.5.2 Experimental Setup

An experiment framework called WARPnet [6] is used for the experimental evaluation of the DCF and BidMAC implementations. WARPnet is a Python-coded environment that allows performing real-time experiments with multiple WARP nodes through an experiment controller running on a host PC. Specifically, the WARPnet module implemented for the 802.11 reference design is called wlan_exp. This framework enables low-level visibility and control of MAC and PHY behaviors of the reference design in real-time.

The testbed used to perform the experiments with the wlan_exp module consists of two systems: wireless and wired (see Fig. 3.17). The wireless system implements an IEEE 802.11g WLAN composed of three WARP v3 nodes, an AP and STA 1 and STA 2, that are placed at 1-meter distance from each other, forming an equilateral triangle, in a zone free of wireless interferences. Each WARP v3 node is equipped with a single common Wi-Fi 2.4 GHz antenna and a 12 V power charger. The wired system, instead, implements a Gigabit Ethernet network that connects the WARP v3 nodes to a PC (i.e., the experiment controller) through a switch. The experiment controller launches custom-design Python scripts that exploit various features of the wlan_exp experiment framework. The scripts generate traffic flows between the AP and the STAs through a Local Traffic Generator (LTG) implemented in the upper-level MAC code (see Appendix B) and calculate the throughput as the number of delivered bits of information over a given trial time, using Tx/Rx packet counts at each node.

Specifically, three different scripts have been developed:

1. throughput_traffic.py: This script generates bidirectional symmetric traffic flows of different periodic inter-packet arrival intervals (from long to short) between the AP and each STA with a constant data payload length (i.e., MSDU) of 1400 bytes and a fixed PHY data rate of 54 Mbps. Note that for BidMAC only unidirectional data flows from each STA to the AP are configured, since the AP will automatically generate an ACK_DATA packet for each STA in response to successful data reception.

2. throughput_payload.py: This script varies the MSDU length from 50 to 1500 bytes with a 250-byte interval and considering zero inter-packet arrival interval (i.e., fill up the transmit queues to reach the saturation state) and a fixed PHY data rate of 54 Mbps.

3. throughput_rate.py: This script tunes the PHY data rate from 6 to

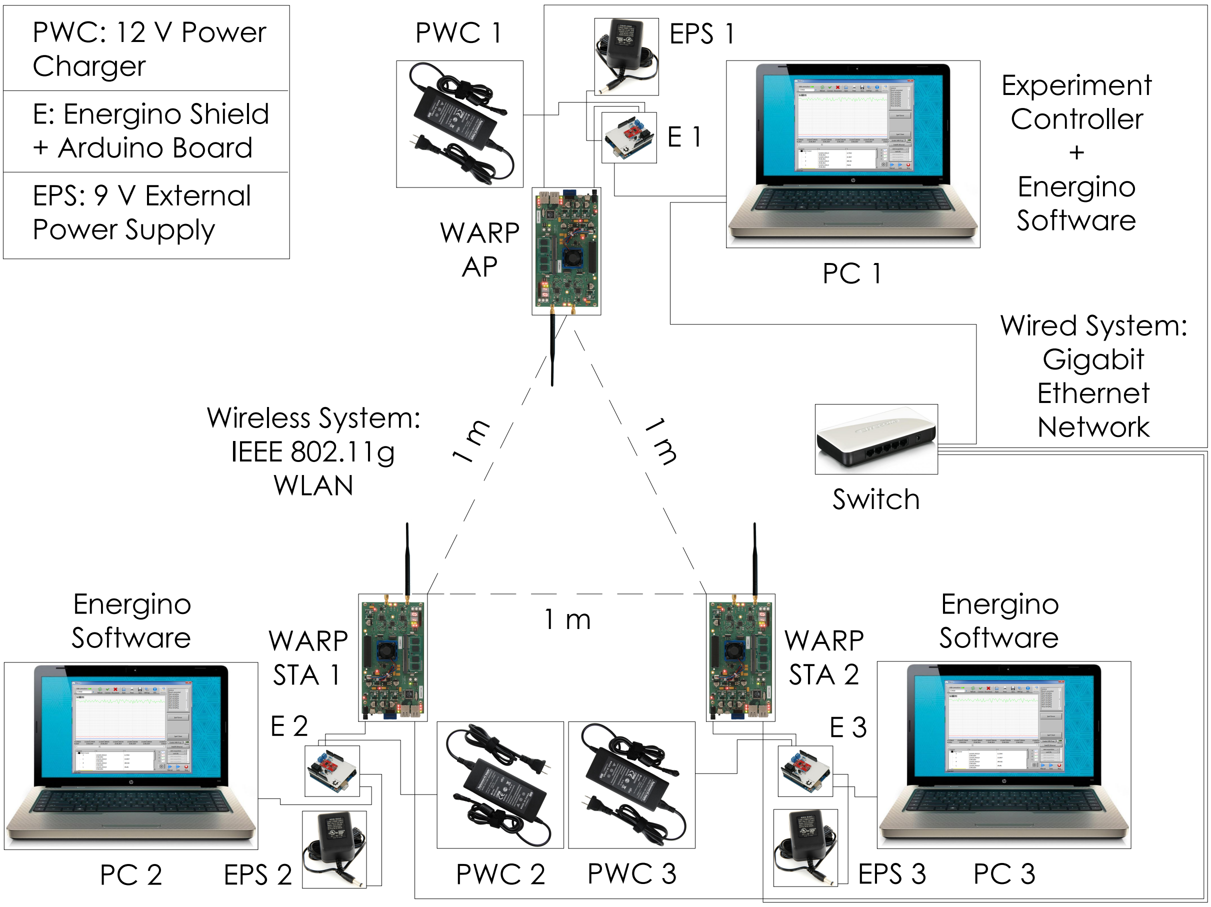


Figure 3.17: The testbed layout.

54 Mbps with zero inter-packet arrival interval and a constant MSDU length of 1500 bytes.

In all these scripts, the trial time for each experiment is set to 30 s and the throughput results are obtained as an average value of 10 repetitions per experiment.

In order to compute the energy efficiency results, the throughput results are divided by the power consumption data of the WARP v3 boards, gathered during the experiments from the Energino meters via custom-design software. Three Energino shields on top of Arduino UNO boards are built following the instructions given in [9] and redesigned in software to achieve sampling rates of 15 kHz. Each Energino shield is connected to the WARP v3 board's power supply and its power charger using the screw terminals. The Arduino UNO board assembled below each Energino shield is connected to a PC using the Universal serial Bus (USB) interface. Also, an additional external power source of 9 V is used to supply the Arduino UNO board (see Fig. 3.17 and Appendix C).

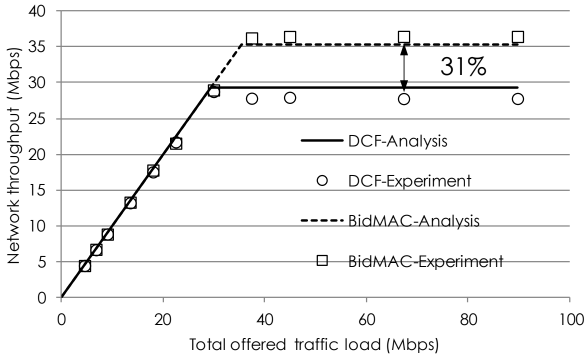
A custom program developed in LabVIEW is executed in each PC to control Energino and acquire samples of voltage, current, and power for each WARP v3 board during a selected period of time. This software allows averaging the samples values, for instance, the average value of power consumption measured in the WARP v3 boards when transmitting (P_t), receiving (P_r), and being idle (P_i) during the experiments is 18.95 W (each board). This value is used in the mathematical expressions derived in previous sections to obtain the theoretical energy efficiency results for the protocols analyzed. Also, note that the Energino meters start sampling 5 s before the beginning of a new experiment in order to gather the power consumption data exactly during the 30 s that each experiment takes.

3.5.3 Analytical and Experimental Results

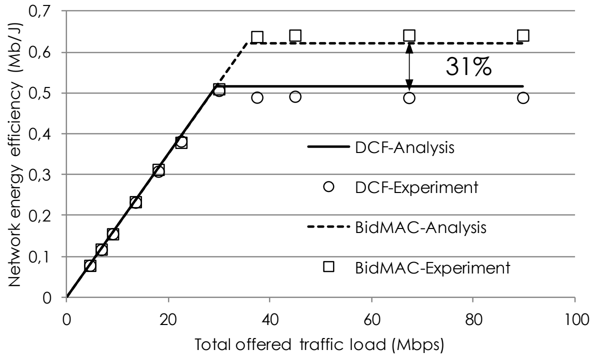
The results of throughput and energy efficiency obtained from the analysis and experiments described in the previous sections for the DCF and BidMAC protocols are presented and discussed as follows. They are summarized in Figs. 3.18 and 3.19. In general, it can be seen that in all the graphs the experimental results are in line with the analytical results for both protocols. The differences between analytical and experimental results in DCF are due to channel errors and collisions that may occur during the experiments. On the contrary, in BidMAC the upper bounds obtained experimentally are slightly higher than those derived analytically. The reason for this variation is that the proposed BidMAC implementation only allows the AP (and not the STAs) to exploit bidirectional transmissions and also does not require the AP to compete for the channel access, in contrast with the general BidMAC operation considered in the analysis.

Fig. 3.18 shows the network, AP, and average per-STA throughputs (Figs. 3.18a, 3.18c, and 3.18e) and energy efficiencies (Figs. 3.18b, 3.18d, and 3.18f) of DCF and BidMAC versus the total offered traffic load. When either the DCF or BidMAC is executed and the traffic load is low, the AP and the two STAs can transmit all their data packets normally. Note that the AP transmits twice more data packets than the STAs as it delivers downlink traffic that is symmetric to the uplink traffic received from them. As the traffic load increases, the AP and the STAs transmit more frequently and so their throughputs and energy efficiencies increase due to the increase of data transmitted and the reduction of idle periods which waste time and energy. The AP achieves the highest throughput and energy efficiency when the total traffic load is almost 30 Mbps (see Figs. 3.18c and 3.18d), where the AP captures half of the channel accesses and each of the two STAs obtains a quarter (half in total) of the channel accesses.

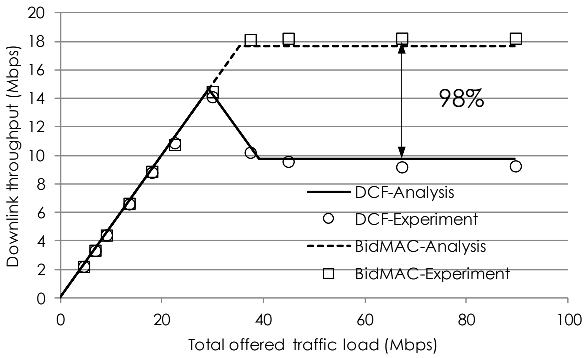
When the traffic load increases above that value until reaching the saturation point (below 40 Mbps), the channel share of the AP is reduced down to one third whereas those of the STAs increase up to two thirds (one third each), due to the DCF MAC fairness. As a result, the AP experiences a significant reduction of its throughput and energy efficiency that affects the STAs in terms of a lower amount of received downlink packets. In contrast, BidMAC allows the AP to initiate contention-free channel accesses to deliver downlink data to the STAs after each successful data reception, thus increasing the amount of downlink packets transmitted and reducing energy consumption due to unnecessary backoff periods. Therefore, when the traffic load is high the throughput and energy efficiency of the AP improves by 98%, as well as the network throughput and energy efficiency by 31%, with minimum impact on the throughputs and energy efficiencies of



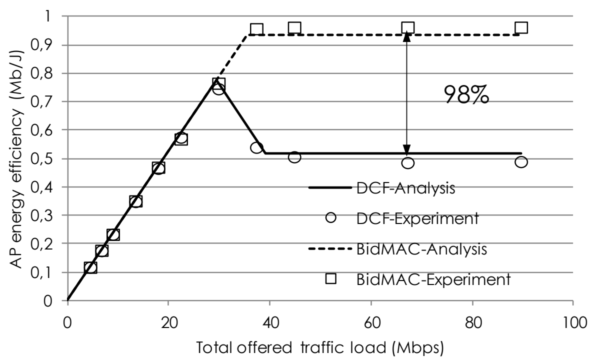
(a) Network throughput



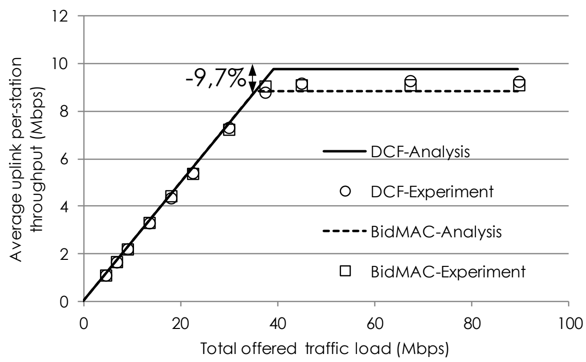
(b) Network energy efficiency



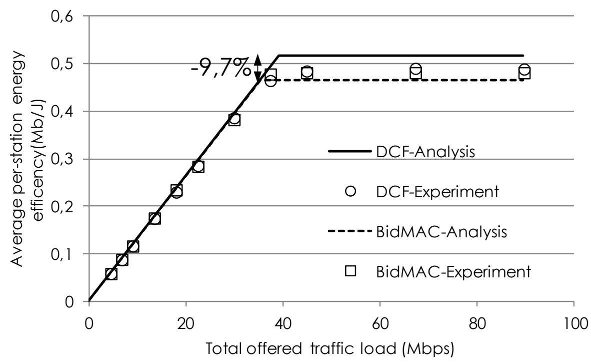
(c) Downlink throughput



(d) AP energy efficiency



(e) Average uplink per STA throughput



(f) Average per STA energy efficiency

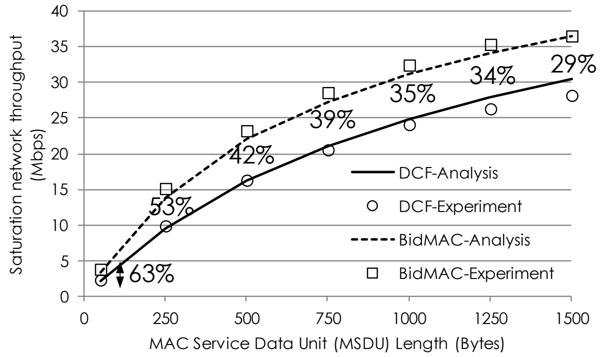
Figure 3.18: Experimental throughput and energy efficiency of the contention-based MAC protocols versus the traffic load

the STAs.

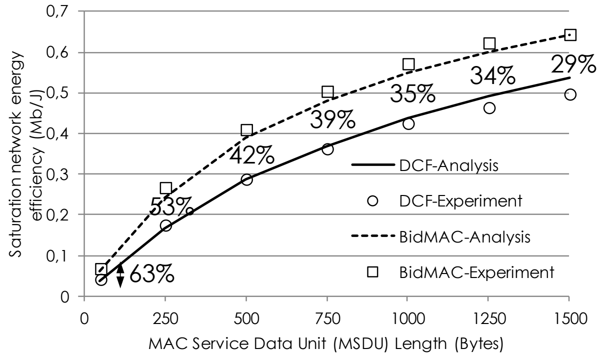
Fig. 3.19 reports the network throughputs and energy efficiencies of DCF and BidMAC under saturation (i.e., the AP and the STAs have always data ready to be transmitted) versus the MSDU length and the PHY data rate in Figs. 3.19a and 3.19b and Figs. 3.19c and 3.19d, respectively. It can be seen in these figures that BidMAC outperforms DCF for all MSDU lengths and PHY data rates considered, showing significant gains. Whereas in Figs. 3.19a and 3.19b the gain of BidMAC versus DCF decreases from 63% to 29% as the MSDU length increases, Figs. 3.19c and 3.19d show that the gain varies between 15% and 29% with increasing PHY data rates. The reason for these behaviors is related to the influence of the data transmission time on the total time required to transmit data in DCF and BidMAC. While faster rates or shorter packet lengths lead to shorter data transmission times with lower impact on the total transmission time, slower rates or longer packet lengths imply longer data transmission times with higher impact.

3.6 Conclusions

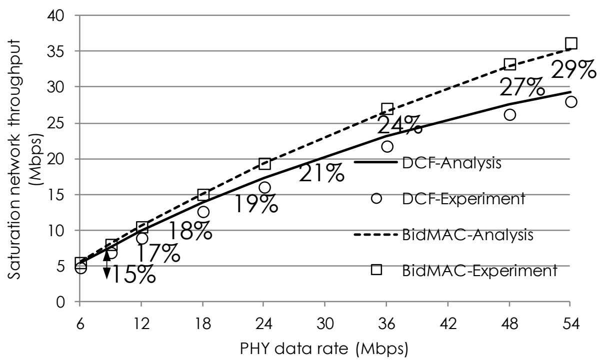
BidMAC and GreenBid have been presented in this chapter as new energy-efficient distributed MAC protocols that have been designed to improve both the throughput and energy efficiency of the DCF of the IEEE 802.11 Standard for WLANs. The basic idea behind BidMAC is to allow the receiver of a valid data packet to perform an RD transmission (with an implicit ACK) back to the transmitter (or to another receiver if the RD executor is the AP) without contending for the channel, as it would be the case in the standard DCF. Then, GreenBid exploits the longer duration of BidMAC transmissions, which include both forward and reverse transmissions, to allow overhearing STAs to turn off their radio transceivers in



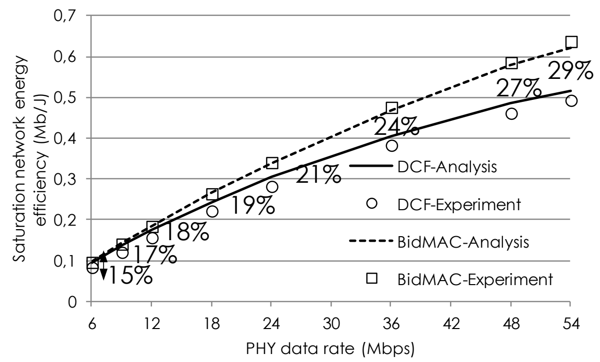
(a) Saturation network throughput vs. MSDU



(b) Saturation network energy efficiency vs. MSDU



(c) Saturation network throughput vs. Rate



(d) Saturation network energy efficiency vs. Rate

Figure 3.19: Experimental throughput and energy efficiency of the contention-based MAC protocols versus the MSDU length and PHY data rate

order to save energy, taking into account the on/off radio transitions of STAs.

The closed expressions of the maximum achievable throughputs and energy efficiencies of DCF, BidMAC, and GreenBid have been derived and a Python simulation environment where the protocol rules have been implemented has been developed for the validation of the proposed analytical model. The performances of the protocols have been evaluated in a WLAN composed of an AP and 20 STAs considering relevant system parameters such as the traffic load, data payload length, data rate, number of STAs in the network, wakeup radio transition coefficient, and awake/sleep radio transitions time. Both analytical and simulation results have shown the high performances of BidMAC and GreenBid when compared to that of DCF for all evaluated parameters.

More specifically, the throughput gains vary from 60% to 20% as the packet length increases and from 6% to 30% as the data rate increases. The throughput gains are stable around 30% as the number of STAs increases. The energy efficiency gains range from 60% to 120% with increasing packet lengths and from 360% to 80% with increasing data rates. Also, they vary between 24% and 88% as the number of STAs increases. Furthermore, the results have shown the importance of taking into account the wakeup radio transitions in the energy efficiency analysis of energy-efficient MAC protocols based on low-power states (i.e., GreenBid), since those transitions represent the 70% of the total energy consumption. In this sense, the energy efficiency gains vary between 120% and 20% as the wakeup radio transition coefficient increases. Similarly, the gains are between 230% and 30% as the awake/sleep radio transition time increases. These parameters will vary depending on the radio hardware design and are critical for the proper operation of GreenBid.

Finally, the proposed BidMAC protocol has been implemented on WARP v3 platforms using a reference design that implements the DCF MAC and OFDM PHY from IEEE 802.11a/g. A testbed composed of three WARP v3 nodes where one acts as an AP and two as STAs have been set up. To perform the experiments and gather the experimental results, several scripts that generate traffic flows between the AP and the STAs and calculate the throughput at each node have been developed. Also, Energino meters and a program developed in labVIEW to control Energino have been used to measure the energy consumption of the WARP v3 nodes and then calculate the energy efficiency. The experimental throughput and energy efficiency results of DCF and BidMAC have been shown versus the traffic load, the packet length, and the data rate. The maximum experimental gain of BidMAC versus DCF at the network level is above 60% whereas the maximum experimental gain from the AP perspective is around 100% with minimum impact on the average per-STA performance.

Therefore, this chapter has demonstrated through analysis, computer-based simulation, and real-life experimentation that the proposed energy-efficient distributed MAC protocols can improve the throughput and energy efficiency of the legacy DCF in WLANs.

Chapter 4

Energy-Efficient Centralized MAC Protocols

4.1 Introduction

The typical deployment of a WLAN based on the IEEE 802.11 Standard [16] is the infrastructure mode shown in Fig. 4.1. In an infrastructure WLAN, an AP is connected to a cable network infrastructure and provides wireless Internet access for a set of WLAN-enabled user devices (referred to as STAs in the terminology of the Standard) in its BSA, all together forming a BSS. Wireless communication within the BSS occurs between the AP and the STAs using a shared radio channel. Therefore, an efficient radio resource management strategy is of paramount importance to fulfill the QoS requirements of STAs for both downlink and uplink traffic flows, while minimizing the energy consumption of STAs in order to prolong their operational times.

The MAC and PHY layer specifications of the IEEE 802.11 Standard define two modes of power management for the STAs of a WLAN. In active mode, STAs are required to remain in awake state to continuously listen to the wireless channel (being ready to either transmit or receive data). This makes the STAs in active mode to consume significant amounts of energy for keeping their radio transceivers always on (i.e., idle-listening)

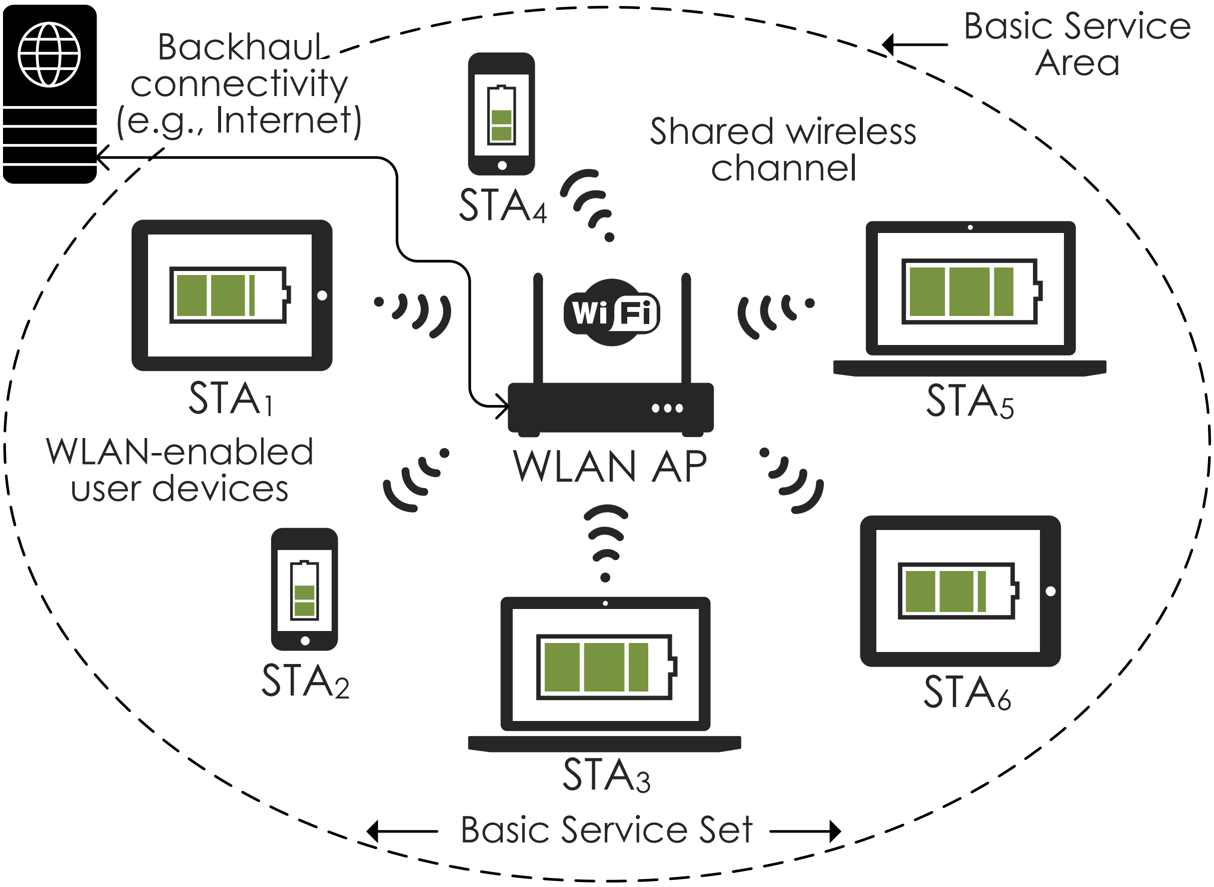


Figure 4.1: Typical deployment of a WLAN in the infrastructure mode where the PCF can be executed.

and receiving packets not addressed to themselves (i.e., overhearing). In contrast, the STAs in PS mode enter a low-power doze (or sleep) state wherein their radio transceivers are turned off. This yields energy savings at the cost of not being able to either transmit or receive while being in this state.

When operating in active mode within an infrastructure WLAN, the AP and the STAs may execute two different methods for sharing access to the wireless channel: DCF or PCF. The DCF method is based on random channel access coordinated in a distributed manner through a contention strategy and can only support best-effort traffic due to the inefficiency induced by collisions and backoff periods. On the contrary, the PCF method is based on deterministic channel access centrally controlled by the AP through a polling strategy, hence completely avoiding collisions, and can provide QoS support for real-time traffic. The AP may announce through periodic beacons the beginning of a CFP repetition interval wherein the AP and the STAs execute the centralized polling-based access method (PCF) during a CFP. After that, the AP and the STAs enter a CP wherein the distributed contention-based access method (DCF) is executed.

On the other hand, the STAs operating in PS mode typically alternate between awake and sleep states periodically to listen to selected beacons broadcasted periodically by the AP (every listen interval is negotiated with the AP). These beacons inform them about data buffered in the AP through a TIM. This TIM consists in a logical list that contains the list of identifiers of the STAs that must remain awake until the AP delivers all their buffered data. In the infrastructure power saving scheme specified in the original version of the IEEE 802.11 (PSM), STAs retrieve buffered data from the AP by transmitting PS-Poll frames using the DCF during a CP (each PS-Poll frame is used to retrieve a single data frame), or otherwise using the PCF without PS-Poll frames during a CFP (i.e., waiting to be polled). In addition, STAs may also wake up at any time to transmit data.

Along the various amendments of the Standard, different methods backwards compatible with the PSM have been specified to optimize the amount of time that the STAs in PS mode spend in awake state for transmitting and receiving data. For instance, the power saving strategy defined in the IEEE 802.11e (APSD) is a mechanism for the delivery of downlink data buffered in the AP, which can be unscheduled or scheduled. In unscheduled APSD, STAs decide when to awake to transmit a trigger frame, similar to the PS-Poll but possibly combined with data, that initiates an SP wherein the AP delivers a burst of buffered data to them (i.e., unscheduled SP). Otherwise, in scheduled APSD STAs awake at fixed intervals determined by the AP to receive the data (i.e., scheduled SP).

Moreover, the power saving mechanism defined in the IEEE 802.11n (PSMP) extends the operation of APSD (both unscheduled and scheduled) by allowing the AP to begin an SP that includes an uplink and downlink transmission phase in order to minimize the awake time of the STAs in PS mode. Specifically, the AP transmits a PSMP frame addressed to those STAs in PS mode that are awake and containing a schedule of uplink and downlink transmissions for each of them. They only awake at their assigned transmission and reception slots.

PSM, APSD, and PSMP are all based on the same concept of periodic beacons and listen intervals. Although APSD improves some of the limitations of PSM and PSMP improves some of the limitations of APSD, all these power saving mechanisms do not work optimally when there exists a large number of STAs with high amounts of bidirectional traffic in the network. This is due to the need to attach identifiers to the beacons, thus suffering from scalability limitations, and the dependencies on the beacon and listen intervals, which may cause performance degradation and additional energy consumption for the STAs.

On the contrary, the new power saving scheme defined in the IEEE 802.11ac (TXOP PSM) is not based on listen intervals and beacons attaching a TIM. STAs in this PS mode opportunistically go to sleep when the AP or other STAs transmit (i.e., during a TXOP), based on the virtual carrier sense (NAV) information carried in management and overheard control and data frames. TXOP PSM is able to significantly improve the energy efficiency of STAs in highly dense networks and with heavy traffic conditions, while also being able to be used in conjunction with other power saving mechanisms when the number of STAs and the traffic load in the network are both low. In this case, the available time for sleeping (i.e., the total data transmission time or TXOP duration) must allow the

STAs to go to sleep and wake up taking into account the duration of the on/off transitions of radio transceivers.

Unfortunately, the regular operation of the DCF may not facilitate the TXOP PSM operation. Typically, a TXOP is reserved/granted for the transmission of a single data packet. Therefore, depending on the duration of the TXOP, which depends on the data length and the data transmission rate, and the duration of on/off radio transitions, which may be in the order of hundreds of microseconds [3–5], it may not be possible for a third STA to go to sleep during the transaction.

In contrast, the PCF concatenates multiple bidirectional TXOPs between the AP and the STAs, thus facilitating the execution of the TXOP PSM. Motivated by this, this chapter investigates two new energy-efficient polling-based MAC protocols, named BidPoll and GreenPoll, aiming to efficiently implement the TXOP PSM operation on top of the PCF. BidPoll allows the AP to initiate through beacons two virtual phases inside the conventional CFP structure. The first phase is specifically reserved for uplink and downlink data transmissions with very low overhead of poll and ACK frames between the AP and the STAs that requested TXOPs in the previous CFP. The second phase is used for dynamic data exchanges between the AP and the rest of STAs that are not served in the first phase until the end of the CFP. Also, in this phase it is possible to achieve low overhead by using downlink data as implicit polls and uplink data as implicit ACKs when there are TXOPs in both directions.

Furthermore, GreenPoll is an extension of BidPoll that combines the TXOP PSM for energy saving and the PCF with reservation and implicit polling/ACK (i.e. BidPoll) for more efficient data transfer. Thus, GreenPoll achieves low overhead and overcomes scalability limitations compared to beacon-based PS mechanisms. The basic idea behind GreenPoll is to allow the STAs involved in the polling activity during the first virtual phase of a BidPoll CFP to save energy by turning off their radio transceivers after exchanging data with the AP, while those not involved can also sleep to save energy during the entire phase. After that, all STAs awake for the second virtual phase where those being not yet granted a TXOP are able to exchange data with the AP until the end of the CFP.

It is important to mention that, based on the comprehensive assessment of the state of the art, the work presented in this chapter can be considered as the first research work that investigates the idea of combining the PCF with virtual reservation, implicit polling/ACK, and opportunistic sleeping periods through TXOP PSM for high-throughput high-energy-efficient WLANs based on the IEEE 802.11 Standard.

A preliminary description and performance evaluation of BidPoll by means of computer-based simulations have been presented in [79]. GreenPoll has been introduced and evaluated through computer-based simulations in [86], where a detailed description and comprehensive performance evaluation of BidPoll has also been presented for the purpose of comparison with GreenPoll. Then, the performance analyses of BidPoll and GreenPoll in terms of energy efficiency have been presented and validated through computer-based simulations in [87].

The structure of this chapter is detailed as follows.

- Section 4.2 summarizes the most relevant existing energy-efficient MAC protocols based on polling and points out the differences between these MAC protocols and the proposed BidPoll and GreenPoll MAC protocols.

- Section 4.3 provides an overview of the legacy PCF MAC protocol and comprehensively describes BidPoll and GreenPoll.

- Section 4.4 analyzes the maximum achievable throughputs and energy efficiencies of the protocols under consideration using a simplified approach.

- Section 4.5 describes the implementation of the protocols in a Python simulation environment and comprehensively evaluates the performances of the protocols by means of both analytical and simulation results. Important system parameters such as the traffic load, packet length, data rate, number of STAs in the network, wakeup (off-on) radio transition power consumption, and awake/sleep (on/off) radio transitions time have been considered in the evaluation. Note that the legacy DCF MAC protocol has also been considered in the evaluation for the purpose of comparison with the polling-based MAC protocols.

- Section 4.6 concludes the chapter by summarizing the key contents of the chapter and highlighting the most relevant results.

4.2 Related Work

In addition to the power saving features defined in the IEEE 802.11 Standards, power saving has received much attention in recent years [19,29]. Particularly related to this paper are the power saving mechanisms based on polling presented in [31, 36, 52]. These inspiring works propose different structures for the beacons. Essentially, they refer to multi-polling packets which poll various STAs at once. These packets contain the access order, the receiver association identifier, the TXOP duration, and other relevant information for each polled STA, in a way similar to the PSMP. Based on the multi-polling packet, the STAs that are not involved in the polling process can immediately return to the sleep state while those involved in the data exchange are only awake for data transmission and reception periods. In [31], the STAs of later order may consume more energy due to overhearing, whereas in [52] this problem is effectively addressed at the cost of certain throughput degradation. The work in [36] copes with the limitations of [31] and [52] in terms of robustness and reliability.

Unfortunately, existing energy-efficient multi-polling protocols show the following drawbacks when the number of STAs in the network and the traffic load increase: *(i)* scalability issues related to the multi-polling packet, i.e. the greater the number of STAs the larger the packet, and *(ii)* complexity issues in terms of TXOP scheduling. Furthermore, all the aforementioned works do not analyze the influence of the on/off radio transitions on the energy consumption of the STAs. These transitions require a certain switching time and extra power consumption that should not be neglected [3–5].

In its turn, GreenPoll differs from the proposed schemes in [31, 36, 52] in the fact that STAs enable sleeping processes based only on the virtual carrier sense information attached to the transmitted beacon, control, and data packets by exploiting BidPoll. Therefore, BidPoll and GreenPoll can avoid explicit scheduling information attached to the beacons and overcome scalability limitations. To reduce the overheard of control packets, BidPoll and GreenPoll employs both implicit polling and ACK through uplink and downlink data packets from [79]. Also, to reduce the energy consumed by the last polled STAs, a cyclic polling order scheduling mechanism, wherein, for example, the last STA will become the first to be polled in the next round, is integrated in the GreenPoll operation.

The following sections will describe, analyze, and evaluate the proposed BidPoll and GreenPoll MAC protocols considering the legacy DCF and PCF MAC protocols for the puprose of comparison.

4.3 Polling-Based Channel Access Methods

This section overviews the PCF MAC protocol of the IEEE 802.11 Standard and provides a detailed description of the proposed BidPoll and GreenPoll MAC protocols.

4.3.1 The Legacy Point Coordination Function (PCF)

The PCF MAC specification of the IEEE 802.11 Standard defines a centrally-controlled access mechanism that employs a polling strategy to enable contention-free data transmissions between the AP (i.e., the central controller or point coordinator) and the STAs with some degree of QoS. The AP initiates a CFP after a PIFS by broadcasting a Beacon (B) that contains the duration of the CFP. All the STAs receiving the beacon update their NAVs and can only transmit when they receive a poll packet from the AP. The AP sequentially polls each STA, possibly in combination with downlink data, based on a polling list that is updated with the identifiers of the STAs registered to it during the association process. A polled STA may respond with a null data packet after a SIFS if no data packets are to be transmitted. If a transmission failure occurs during the polling activity, the AP will wait for a PIFS and will poll the next STA of the polling list. The transmission of a CE packet indicates the end of a CFP, after which a new CFP may begin after a PIFS.

When the PCF interoperates with the DCF, the AP manages through beacons a periodic super structure, called CFP repetition interval, that is divided into a CFP, where the PCF is executed, and a CP, where the DCF is executed (see Fig. 4.2). The beacons are transmitted after a PIFS because a PIFS has a shorter duration than a DIFS, thus allowing the AP to seize the wireless channel earlier than the STAs. Each beacon contains information related to the durations of both the CFP and CP by specifying

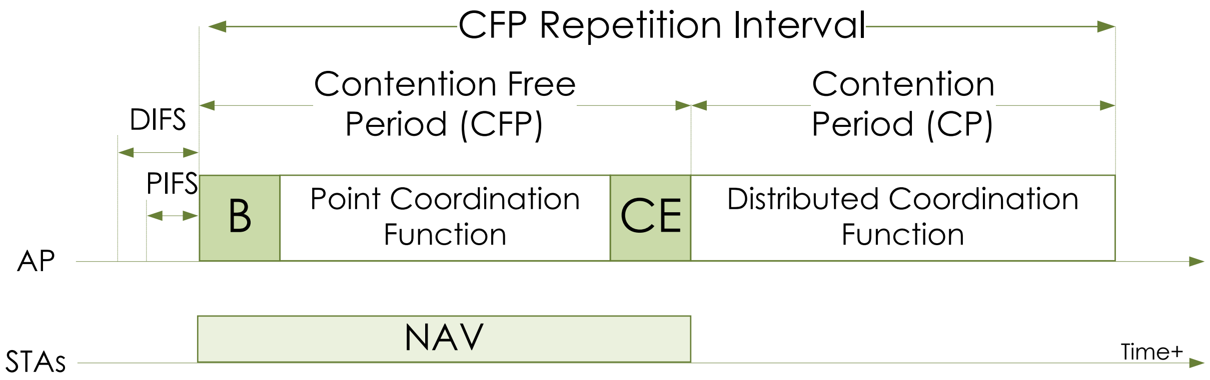


Figure 4.2: Coexistence of the DCF and PCF through a CFP repetition interval

the CFP repetition interval and the maximum allowable duration of a CFP (CFPMaxDuration). The value of CFPMaxDuration should be selected to allow at least one data packet transmission during a CP, as required for the coexistence of contention and contention-free traffic. After the end of a CFP, a CP begins and the AP and the STAs may exchange data by competing for getting access to the wireless channel after a DIFS. Note that the STAs that wish to transmit through CFPs need to register to the polling list by sending re-association requests to the AP during CPs. Similarly, the STAs that wish to unregister from the polling list also need to send re-association requests to the AP during CPs.

Fig. 4.3 shows an example of operation of the PCF where the AP and STA 1, STA 2, and STA 3 exchange data. After a PIFS, the AP broadcasts a beacon and after a SIFS sends a poll packet combined with a data packet to STA 1. After a SIFS, STA 1 responds with an ACK packet in combination with a data packet destined to the AP. The AP then sends an ACK packet to STA 1 after a SIFS and a poll packet together with a data packet to STA 2, which responds after a SIFS with an ACK packet along with a null packet because it has no data ready to be transmitted to the AP. After a SIFS, the AP polls STA 3 and STA 3 responds with a data packet after a SIFS. Finally, the AP sends an ACK packet to STA 3

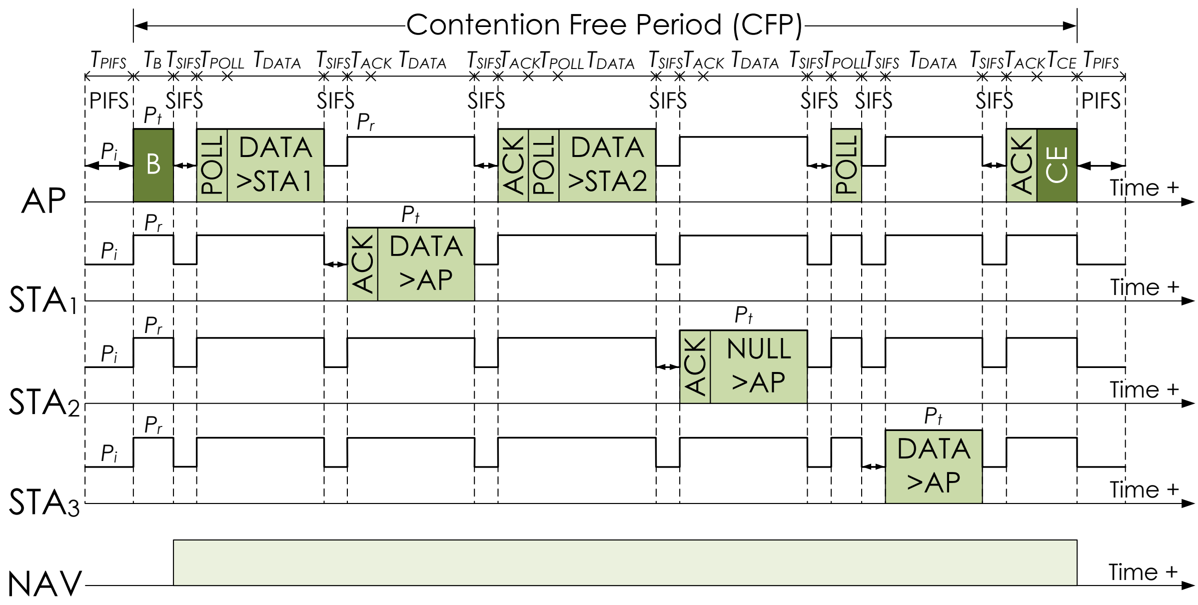


Figure 4.3: Example of operation of the PCF MAC protocol

after a SIFS and broadcasts a CE packet to conclude the CFP.

4.3.2 The New Bidirectional Polling MAC Protocol (BidPoll)

BidPoll is aimed at reducing the high overheard of control packets, such as poll and ACK, during the polling activity of the PCF, when the number of polled STAs and the traffic load increase. In BidPoll, the STAs indicate whether more data packets are to be transmitted and, if any, the required transmission time to be allocated by the AP in the next CFP, by using the more data and duration fields, respectively, contained in the MAC header of transmitted data packets. When the AP receives all this information, it determines for which of the STAs that have requested a transmission slot it has buffered data to be delivered in the next CFP. Then, the AP prepares a virtual list with the identifiers of such STAs along with the time required to perform both uplink and downlink transmissions and, based on these data, computes the total time of all the expected data exchanges. This value will be attached to the next transmitted beacon and also the

AP may add another time value that corresponds to CFPMaxDuration, in case that not all the STAs have requested transmission opportunities before the beginning of the next CFP. Therefore, in BidPoll the actual CFP is virtually split into two phases whose time intervals are adjusted by the AP through two NAV values attached to the beacons according to dynamic traffic requirements.

- In the first phase, the AP serves the STAs of the polling list that informed it about more data packets ready to be transmitted in the previous CFP and that also have downlink data packets in its buffer. In this way, grouping pairs of uplink/downlink transmissions for the STAs the AP can use downlink data packets as implicit poll packets and polled STAs can send back uplink data packets as implicit ACK packets. Note that the received uplink data packets are always acknowledged to ensure the notification of a successful data exchange between the AP and each polled STA.

- In the second phase, the STAs that were not granted transmission opportunities in the first phase are able to transmit and receive data in this phase. Depending on the traffic characteristics of the network in real-time, the AP and the STAs may execute the legacy PCF or otherwise BidPoll (i.e., data exchanges between the AP and the STAs with implicit polling and partial implicit ACK) for more efficient data transfer in this phase.

Fig. 4.4 shows an example of operation of BidPoll when all the STAs have both uplink and downlink transmissions scheduled for the next CFP (i.e., all data exchanges are performed in the first phase, after which the CFP is terminated by the AP). Following the same description as that provided for the PCF in Fig. 4.3, the STAs receiving downlink data packets from the AP can immediately respond with uplink data packets with no

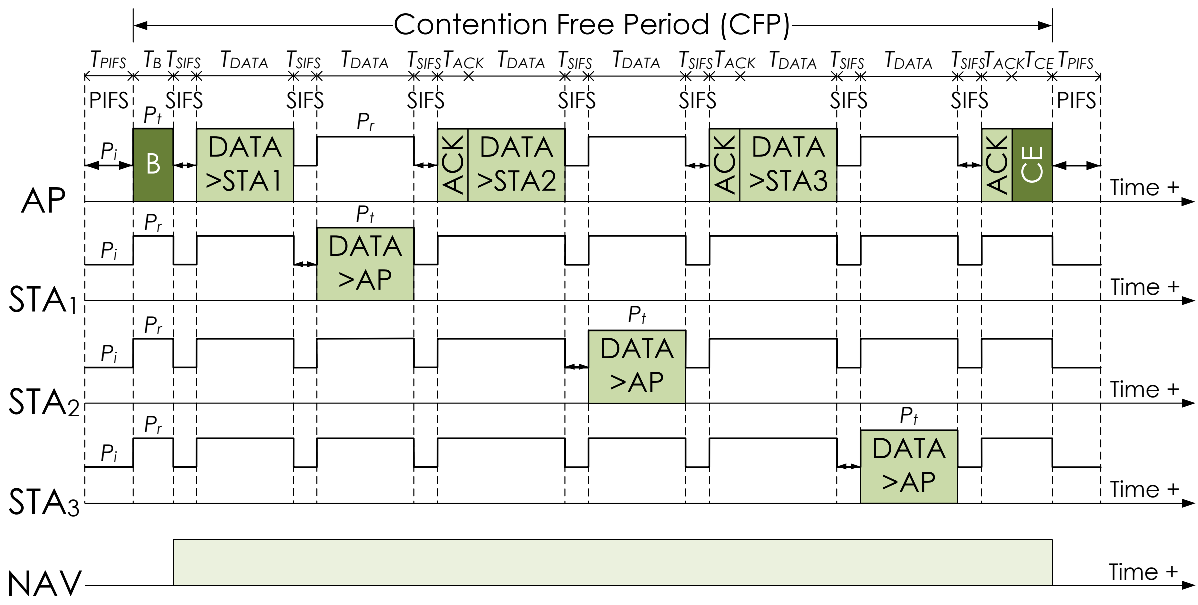


Figure 4.4: Example of operation of the BidPoll MAC protocol

additional combined poll and ACK packets.

4.3.3 The New Green Polling MAC Protocol (GreenPoll)

GreenPoll represents an extension of BidPoll to reduce the energy consumed by the STAs when they listen to data transmissions between the AP and other STAs during the polling activity. In GreenPoll, the general structure of a CFP is split into two virtual phases (similar to BidPoll) whose duration is determined by the AP through two NAV values attached to the beacons.

- In the first phase, the STAs with no data to transmit go to sleep whereas those with data to transmit remain awake until the AP delivers downlink data to them, after which they respond with uplink data. After reception of the ACK packet in response to valid uplink data reception by the AP, these STAs go to sleep until the end of this phase according to one of the NAV values retrieved from the beacon. The STAs can return to the sleep state only if the remaining time allows them to switch between awake and sleep states before their NAV timers expire.

- In the second phase, all the STAs are awake and those that entered the sleep state in the first phase but were not granted transmission opportunities as well as those that remained awake during the entire phase without transmitting data are able to transmit and receive data in this phase. Depending on the traffic characteristics of the network in real-time, the AP and the STAs may execute the legacy PCF or otherwise BidPoll for more efficient data transfer in this phase.

To compute the duration of the first phase, the AP uses own information regarding the downlink buffer status for all the STAs and external information regarding uplink traffic provided by the STAs. Specifically, each polled STA informs the AP about more data packets ready to be transmitted and the required TXOP duration to transmit backlogged packets by using the duration and more data fields, respectively, available in the header of data packets. With this information, the AP will allocate the required time in the first phase for the STAs having both transmission and reception opportunities. When an STA receives the beacon, it returns to sleep if it has sent no request for data transmission in the previous CFP. Otherwise, the STA records the duration of the first phase and then sets a timer to monitor the time elapsed until it successfully performs a data exchange with the AP. Using these two values, a polled STA can compute the remaining polling time and determine if it can go to sleep and wake up before the end of the first phase. If so, it sets its wakeup timer and enters the sleep state.

In addition, the AP will estimate a maximum CFP duration (i.e. the other NAV value in the beacon) in case that not all the STAs have requested a transmission opportunity in the previous CFP or some of those willing to transmit do not have downlink data buffered in the AP. The time left after the end of the first phase will be allocated to those STAs in the second phase, whose nature is unpredictable and where the AP can terminate the CFP at any time (preferably after all the STAs have been served).

Note that the performance of GreenPoll highly depends on the intensity and symmetry of the traffic flows in the network. When the traffic load is heavy and the traffic flows are highly bidirectional, most of the STAs in the network will transmit and receive data in the first phase whereas the impact of the second phase will be marginal. In other cases, the second phase will be predominant and GreenPoll will operate as BidPoll, or in the worst case as the legacy PCF.

Also, to reduce the energy consumed by the last polled STAs, GreenPoll may integrate a cyclic polling order scheduling mechanism, wherein, for example, the last STA will become the first to be polled in the next round.

Fig. 4.5 shows an example of operation of GreenPoll when all the STAs of the polling list (i.e., STA 1, STA 2, and STA 3) have both transmission and reception opportunities at the beginning of the CFP (i.e., illustrating a CFP that entirely operates as in the first phase). It can be seen that once STA 1 receives the ACK packet to its transmitted uplink data packet it can return to the sleep state until the end of the CFP. Similarly, STA 2 remains awake until it transmits its data packet to the AP, after valid reception of a data packet from the AP, and then goes back to sleep when it receives the ACK packet from the AP. In contrast with STA 1 and STA 2, STA 3 cannot enter the sleep state because it is the last polled STA and the available time for sleeping would only be the transmission time of the CE packet from the AP, which would be shorter than the time required by an STA to switch between awake and sleep states.

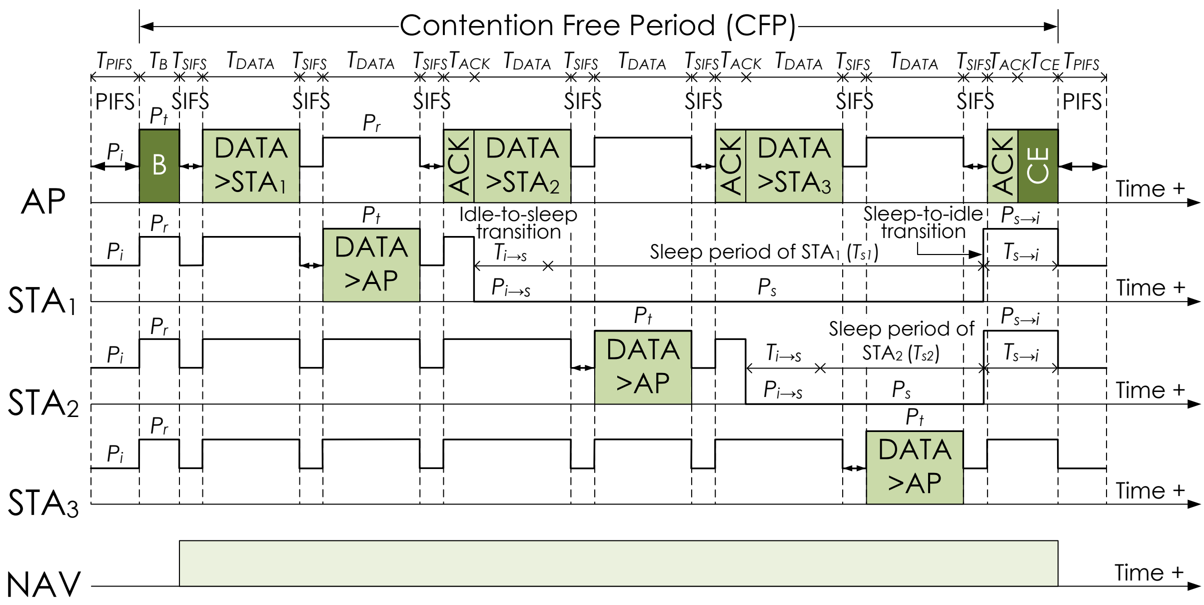


Figure 4.5: Example of operation of the GreenPoll MAC protocol

4.4 Theoretical Analysis

In this section, the expressions of the maximum achievable throughputs energy efficiencies of the protocols considered in this chapter are derived based on the system model and assumptions described as follows and considering three different perspectives: entire, network, AP (i.e., downlink), and average per STA (i.e, uplink).

4.4.1 System Layout and Assumptions

A BSS composed of an AP and N associated STAs in the BSA is considered, as shown in Fig. 4.1. All devices are equipped with IEEE 802.11n wireless interfaces enabling a single antenna for communications, i.e., a SISO communications system. Wireless communication within the BSS occurs between the AP and the STAs using a shared radio channel. It is assumed that the size of the BSA allows all the STAs of the BSS to overhear the transmissions between each STA and the AP in both directions.

Table 4.1: ERP-OFDM PHY Modes and Transmission Times for Management, Control, and Data Packets (1500-Byte Payload) in IEEE 802.11n

Mode (m)	Data Rate	N_{DBPS}	T_B	T_{CE}	T_{POLL}	T_{NULL}	T_{ACK}	T_{DATA}
1	6 Mbps	24	58 μs	58 μs	58 μs	50 μs	50 μs	2078 μs
2	9 Mbps	36	58 μs	58 μs	50 μs	50 μs	50 μs	1394 μs
3	12 Mbps	48	58 μs	58 μs	42 μs	38 μs	38 μs	1054 μs
4	18 Mbps	72	58 μs	58 μs	38 μs	38 μs	38 μs	710 μs
5	24 Mbps	96	58 μs	58 μs	34 μs	34 μs	34 μs	542 μs
6	36 Mbps	144	58 μs	58 μs	34 μs	34 μs	34 μs	370 μs
7	48 Mbps	192	58 μs	58 μs	30 μs	34 μs	34 μs	286 μs
8	54 Mbps	216	58 μs	58 μs	30 μs	34 μs	34 μs	254 μs

Note that the AP can deliver downlink data to any STA of the BSS.

In order to compute the upper bound of the theoretical throughput and energy efficiency within the BSS in idealistic conditions, the following assumptions are made: *(i)* neither collisions nor channel errors occur, *(ii)* the transmit queues are never empty, *(iii)* no packets are lost because of queue overflow, and *(iv)* fragmentation is not used. In addition, constant data packet length and negligible propagation delay due to the short-range transmissions are considered.

Among the possible configurations of the IEEE 802.11n at the PHY layer, the ERP-OFDM specification for SISO communications has been selected. The ERP-OFDM PHY provides 8 transmission modes with different modulation schemes and coding rates. The characteristics of each mode (m) together with the data transmission rate and NDBPS (N_{DBPS}) are reported in Table 4.1.

The expressions to compute the transmission times of Beacon (B), poll and null packets using the ERP-OFDM PHY mode are expressed as

$$\begin{aligned} T_B = T_{CE} = T_{pre} + T_{sig} + T_{sym}\left\lceil \frac{L_{serv} + 8 \cdot L_{POLL} + L_{tail}}{N_{DBPS}\,(m=6)} \right\rceil + T_{sigEx} \\ = 26 + 4 \cdot \left\lceil \frac{22 + 8 \cdot 20}{6} \right\rceil = 58\mu s \end{aligned} \tag{4.1}$$

$$\begin{aligned} T_{POLL}\,(m) = T_{pre} + T_{sig} + T_{sym}\left\lceil \frac{L_{serv} + 8 \cdot L_{POLL} + L_{tail}}{N_{DBPS}\,(m)} \right\rceil + T_{sigEx} \\ = 26 + 4 \cdot \left\lceil \frac{22 + 8 \cdot 20}{N_{DBPS}\,(m)} \right\rceil \end{aligned} \tag{4.2}$$

$$\begin{aligned} T_{NULL}\,(m) = T_{pre} + T_{sig} + T_{sym}\left\lceil \frac{L_{serv} + 8 \cdot L_{NULL} + L_{tail}}{N_{DBPS}\,(m)} \right\rceil + T_{sigEx} \\ = 26 + 4 \cdot \left\lceil \frac{22 + 8 \cdot 14}{N_{DBPS}\,(m)} \right\rceil \end{aligned} \tag{4.3}$$

where all the variables and their values are specified in Table 4.2. Note that control response packets like NULL, and ACK are transmitted using the mandatory rates, i.e., 6, 12, and 24 Mbps, depending on whether the transmission rate of the received packet is 6 or 9, 12 or 18, and 24, 36, 48, or 54 Mbps, respectively [16]. In addition, it is assumed that management packets such as Beacons (B) and CE are transmitted at the lowest basic rate, i.e., 6 Mbps. The transmission times of all packet types for each ERP-OFDM PHY mode are also given in Table 4.1. Note that T_{DATA} and T_{ACK} are computed by (3.1) and (3.3), respectively.

In the following, T_{PIFS} denotes the PIFS interval and is computed as

$$T_{PIFS} = T_{SIFS} + T_{slot} = 10 + 9 = 19\mu s \tag{4.4}$$

The IEEE 802.11n wireless interface of an STA can be in one of the following operational states: transmitting, receiving or overhearing (i.e., receiving packets not destined to itself), idle, and sleeping. In the first

Table 4.2: System Parameters

Parameter	Definition	Value
T_{slot}	Slot Time	9 μs
T_{SIFS}	SIFS Interval	10 μs
T_{PIFS}	PIFS Interval	19 μs
T_{pre}	Preamble Time	16 μs
T_{sig}	Signal Time	4 μs
T_{sym}	OFDM symbol Period	4 μs
T_{sigEx}	Signal Extension Period	6 μs
L_{serv}	Service Bits	16 bits
L_{tail}	Tail Bits	6 bits
L_B	Length of Beacon (B)	20 bytes
L_{CE}	Length of CE	20 bytes
L_{POLL}	Length of POLL	20 bytes
L_{NULL}	Length of NULL	14 bytes
L_{ACK}	Length of ACK	14 bytes
L_{MAChdr}	MAC Header	30 bytes
L_{FCS}	Frame Check Sequence	4 bytes
$T_{i \to s}$	Transition Time from Idle to Sleep	250 μs
$T_{s \to i}$	Transition Time from Sleep to Idle	250 μs
P_t	Transmission Power Consumption	1.65 W
P_r	Reception Power Consumption	1.4 W
P_i	Idle Power Consumption	1.15 W
P_s	Sleep Power Consumption	0.045 W
$P_{i \to s}$	Idle to Sleep Transition Power Consumption	0.045 W
$P_{s \to i}$	Sleep to Idle Transition Power Consumption	1.725 W

two states, the radio transceiver is actively used to send and receive information. In the idle state, the wireless interface is ready to receive but no signal is received by the radio transceiver. In the sleep state, the radio transceiver is turned off to save energy. Each of these operational states has associated power consumption. In addition, each transition between states incurs a certain switching time that cannot be neglected. These values will vary depending on the product hardware.

Let P_t, P_r, P_i, and P_s denote the power consumed while transmitting, receiving, idle, and sleeping, respectively. When an idle STA identifies an opportunity to sleep, a transition from idle to sleep takes place. Similarly, a transition from sleep to idle occurs when the STA decides to wake up. Based on [3–5], the transition time from idle to sleep ($T_{i \to s}$) is shown to be similar to the transition time from sleep to idle ($T_{s \to i}$). Hence, it is assumed that $T_{i \to s}$ is equal to $T_{s \to i}$. Regarding the power consumed during these transitions, the works in [3–5] show that the power consumed from idle to sleep ($P_{i \to s}$) is substantially lower than P_s. In contrast, the power consumed from sleep to idle ($P_{s \to i}$) is shown to be significantly higher than P_i. Thus, it is assumed that $P_{i \to s}$ is equal to P_s and $P_{s \to i}$ is modeled as αP_i, where α is defined as the transition coefficient between sleep and idle states, or wakeup transition coefficient, and $\alpha > 1$. Fig. 3.6 illustrates this explanation and Table 4.2 records the variables mentioned above and their values (most of them taken from [3–5]).

4.4.2 Throughput

The throughput of a given protocol (S_x) is defined as the amount of information contained in an MSDU (L_{MSDU}) divided by the time ratio (T_x) required to transmit the data packet that includes the MSDU. This is expressed as

$$S_x[\mathrm{Mbps}]{=}\frac{8{\cdot}L_{MSDU}}{T_x} \tag{4.5}$$

where T_x is defined as the amount of time spent in transmission over the total amount of transmitted data packets.

The transmission time ratio of each protocol under consideration is described and formulated as follows.

1) PCF:

The transmission delay of PCF comprises a PIFS interval, a B transmission, N poll transmissions (from the AP to N STAs), $2N$ data and ACK transmissions (from the AP to N STAs and from N STAs to the AP), $2N{+}1$ SIFS intervals, and a CE transmission. Thus, the transmission ratio that corresponds to the saturation network throughput of the PCF is expressed as

$$\begin{aligned} T_{PCF}^{net\_sat}{=}&\frac{1}{2N}\left(T_{PIFS}{+}T_B{+}N\left(T_{POLL}{+}2\left(T_{DATA}{+}T_{ACK}\right)\right)\right)\\ &+\frac{1}{2N}\left(\left(2N{+}1\right)T_{SIFS}{+}T_{CE}\right) \end{aligned} \tag{4.6}$$

Considering (4.6) from the AP perspective, the transmission ratio that results in the saturation downlink throughput of the PCF considers that the AP performs N transmissions during a CFP as

$$\begin{aligned} T_{PCF}^{dwl\_sat}{=}&\frac{1}{N}\left(T_{PIFS}{+}T_B{+}N\left(T_{POLL}{+}2\left(T_{DATA}{+}T_{ACK}\right)\right)\right)\\ &+\frac{1}{N}\left(\left(2N{+}1\right)T_{SIFS}{+}T_{CE}\right) \end{aligned} \tag{4.7}$$

Similarly, taking into account (4.6) from an STA perspective, the transmission ratio that leads to the saturation average uplink per STA throughput of the PCF considers that a randomly chosen STA performs a single transmission in a CFP, that is

$$\begin{aligned} T_{PCF}^{uplpersta\_sat}{=}&T_{PIFS}{+}T_B{+}N\left(T_{POLL}{+}2\left(T_{DATA}{+}T_{ACK}\right)\right)\\ &+\left(2N{+}1\right)T_{SIFS}{+}T_{CE} \end{aligned} \tag{4.8}$$

2) BidPoll:

The transmission delay of BidPoll contains the same as that of the PCF except that poll packets and an ACK packet in each data exchange between the AP and each STA (in both directions) are removed. Therefore, the transmission ratio that produces the saturation network throughput of BidPoll is given as

$$T_{BidPoll}^{net\_sat} = \frac{1}{2N}\left(T_{PIFS} + T_B + N\left(2 \cdot T_{DATA} + T_{ACK}\right) + (2N+1)\, T_{SIFS} + T_{CE}\right) \tag{4.9}$$

Using (4.9) from the AP perspective, the AP transmits N data packets during a given CFP and so the transmission ratio that corresponds to the saturation downlink throughput is written as

$$T_{BidPoll}^{dwl\_sat} = \frac{1}{N}\left(T_{PIFS} + T_B + N\left(2 \cdot T_{DATA} + T_{ACK}\right) + (2N+1)\, T_{SIFS} + T_{CE}\right) \tag{4.10}$$

Also, based on (4.9) from an STA perspective, an STA transmits once in a CFP and thus the transmission ratio that leads to the saturation average uplink per STA throughput is computed as

$$T_{BidPoll}^{uplpersta\_sat} = T_{PIFS} + T_B + N\left(2 \cdot T_{DATA} + T_{ACK}\right) + (2N+1)\, T_{SIFS} + T_{CE} \tag{4.11}$$

3) GreenPoll:

The saturation throughputs of GreenPoll from entire network, downlink, and average uplink per STA are expressed as those of BidPoll because GreenPoll has been designed to improve the energy efficiency of the STAs.

4.4.3 Energy Efficiency

The energy efficiency of a given protocol x (η_x) is defined as the amount of bits contained in an MSDU divided by the energy consumption ratio

(E_x) required to transmit the data packet that includes the MSDU:

$$\eta_x[\text{Mb/J}]=\frac{8\cdot L_{MSDU}}{E_x} \tag{4.12}$$

where L_{MSDU} denotes the byte-length of an MSDU and E_x is defined as the product of power consumed and time spent in transmission over the total amount of transmitted data packets.

The energy consumption ratio of each protocol under consideration is described and formulated as follows.

1) PCF:

The energy consumption of PCF during a CFP can be split into three energy consumption components, namely, transmitting (E_t), receiving and overhearing (E_r), and idle (E_i). During the polling activity, the AP and the N STAs of the polling list consume energy to transmit and receive, respectively, both the B and CE packets and a poll packet, a data packet, and an ACK packet for each polled STA. In addition, they consume energy to receive and transmit, respectively, a data packet and an ACK packet by each polled STA. When the AP communicates with an STA, or vice versa, the other $N-1$ STAs consume energy to overhear the exchange of packets. The AP and the N STAs also consume energy for being idle during a PIFS interval and all the SIFS intervals. As a result, $2N$ data transmissions are performed between the AP and the N STAs (in both directions). The energy consumption ratio that results in the saturation network energy efficiency of PCF is thus formulated as

$$\begin{aligned}
E_{PCF}^{net_sat}&=\frac{1}{2N}\left(E_t+E_r+E_i\right)\\
E_t&=\left(T_B+N\left(T_{POLL}+2\left(T_{DATA}+T_{ACK}\right)\right)+T_{CE}\right)P_t\\
E_r&=\left(T_B+N\left(T_{POLL}+2\left(T_{DATA}+T_{ACK}\right)\right)+T_{CE}\right)NP_r\\
E_i&=\left(T_{PIFS}+\left(2N+1\right)T_{SIFS}\right)\left(N+1\right)P_i
\end{aligned} \tag{4.13}$$

When the PCF is executed, the energy consumption of the AP is computed as follows. First, the AP consumes energy to broadcast the beacon and CE packets and to transmit the poll, data, and ACK packets for the N STAs. Then, the AP consumes energy to receive N data packets and N ACK packets from the STAs. Finally, the AP consumes energy for being idle during a PIFS and all the SIFS intervals. In total, the AP performs N data transmissions during a CFP. Hence, the energy consumption ratio that leads to the saturation AP energy efficiency of PCF is given as

$$\begin{aligned}
E_{PCF}^{ap_sat} &= \frac{1}{N}\left(E_t + E_r + E_i\right) \\
E_t &= \left(T_B + N\left(T_{POLL} + T_{DATA} + T_{ACK}\right) + T_{CE}\right) P_t \\
E_r &= N\left(T_{DATA} + T_{ACK}\right) P_r \\
E_i &= \left(T_{PIFS} + \left(2N+1\right) T_{SIFS}\right) P_i
\end{aligned} \tag{4.14}$$

The energy consumption of an STA when operating under the PCF rules is described next. First, an STA consumes energy to transmit an uplink data packet and an ACK packet when it receives a downlink data packet. Then, an STA consumes energy to receive the beacon and CE packets and the poll, data, and ACK packets from the AP and to overhear the $N-1$ poll, data, and ACK transmissions. Finally, an STA consumes energy for listening to a PIFS and all the SIFS intervals. Note that during a CFP an STA can only perform a single data transmission. Therefore, the energy consumption ratio that produces the saturation average uplink per STA throughput is expressed as

$$\begin{aligned}
E_{PCF}^{persta_sat} &= E_t + E_r + E_i \\
E_t &= \left(T_{DATA} + T_{ACK}\right) P_t \\
E_r &= \left(T_B + NT_{POLL} + \left(2N-1\right)\left(T_{DATA} + T_{ACK}\right) + T_{CE}\right) P_r \\
E_i &= \left(T_{PIFS} + \left(2N+1\right) T_{SIFS}\right) P_i
\end{aligned} \tag{4.15}$$

2) BidPoll:

The energy consumption ratios of BidPoll from entire network, AP, and per STA perspectives contain the same as those of PCF but removing all poll packets and an ACK packet in each data exchange, which are expressed as

$$\begin{aligned}
E_{BidPoll}^{net_sat} &= \frac{1}{2N}\left(E_t + E_r + E_i\right) \\
E_t &= \left(T_B + N\left(2 \cdot T_{DATA} + T_{ACK}\right) + T_{CE}\right) P_t \\
E_r &= \left(T_B + N\left(2 \cdot T_{DATA} + T_{ACK}\right) + T_{CE}\right) N P_r \\
E_i &= \left(T_{PIFS} + \left(2N+1\right) T_{SIFS}\right)\left(N+1\right) P_i
\end{aligned} \tag{4.16}$$

$$\begin{aligned}
E_{BidPoll}^{ap_sat} &= \frac{1}{N}\left(E_t + E_r + E_i\right) \\
E_t &= \left(T_B + N\left(T_{DATA} + T_{ACK}\right) + T_{CE}\right) P_t \\
E_r &= N T_{DATA} P_r \\
E_i &= \left(T_{PIFS} + \left(2N+1\right) T_{SIFS}\right) P_i
\end{aligned} \tag{4.17}$$

$$\begin{aligned}
E_{BidPoll}^{persta_sat} &= E_t + E_r + E_i \\
E_t &= T_{DATA} P_t \\
E_r &= \left(T_B + \left(2N-1\right) T_{DATA} + N T_{ACK} + T_{CE}\right) P_r \\
E_i &= \left(T_{PIFS} + \left(2N+1\right) T_{SIFS}\right) P_i
\end{aligned} \tag{4.18}$$

3) GreenPoll:

The energy consumption of GreenPoll is based on that of BidPoll but it introduces two new energy consumption components, namely, switching between idle and sleeping (E_{sw}), and sleeping (E_s). In GreenPoll, each STA of the polling list progressively returns to the sleep state once it successfully performs a data exchange with the AP. Due to the time required to switch between idle and sleep states, the last STA of the polling list may be unable to go to sleep and wake up before a CFP ends. Therefore, in order to compute the closed expression of the energy consumption for GreenPoll it is assumed that the last polled STA does not enter the sleep state. Then, a correction factor that takes into account those STAs of the polling list that cannot go to sleep apart from the last STA is introduced. To express this, M is defined as the number of active STAs during the whole polling period. M can be calculated in the following steps: *(i)* determine the total duration of a CFP to allow a data exchange (in both directions) between the AP and each STA of the polling list, *(ii)* subtract the total transition time between awake and sleep states from the total CFP time, *(iii)* divide by the time required to complete a single bidirectional data exchange between the AP and an STA (T_D), *(iv)* subtract the resulting value from the N STAs of the polling list, and *(v)* apply a ceiling function to the final value. As a result, the formula of M is expressed as

$$M=\left\lceil N-\frac{NT_D+T_{CE}-\left(T_{i\to s}+T_{s\to i}\right)}{T_D}\right\rceil \tag{4.19}$$

where $T_D=2T_{DATA}+T_{ACK}+2T_{SIFS}$.

Therefore, the energy consumption ratio that describes the saturation network energy efficiency of GreenPoll is written as

$$\begin{aligned}
E^{net_sat}_{GreenPoll}=&\frac{1}{2N}\left(E_t+E_r+E_i+E_{sw}+E_s\right)\\
E_t=&\left(T_B+N\left(2\cdot T_{DATA}+T_{ACK}\right)+T_{CE}\right)P_t\\
E_r=&\left(NT_B+\left(\frac{N+1}{2}N+\frac{M-1}{2}M\right)\left(2\cdot T_{DATA}+T_{ACK}\right)+MT_{CE}\right)P_r\\
E_i=&\left(\left(N+1\right)T_{PIFS}+\left(N\left(N+2\right)+M\left(M-1\right)+2N+1\right)T_{SIFS}\right)P_i\\
E_{sw}=&\left(T_{i\to s}P_{i\to s}+T_{s\to i}P_{s\to i}\right)\left(N-M\right)\\
E_s=&T_sP_s
\end{aligned} \tag{4.20}$$

where T_s is the total sleep period considering all the STAs and is expressed as

$$T_s = \left(\frac{N}{2}(N-1) - \frac{M-1}{2}M\right)T_D + \left(T_{CE} - \left(T_{i\to s} + T_{s\to i}\right)\right)(N-M) \quad (4.21)$$

The different energy consumption components of GreenPoll in (4.22) are described as follows.

- **Transmission period:** the AP and the N STAs of the polling list consume the same amounts of energy as those in BidPoll.

- **Reception period:** the AP consumes energy for receiving a data packet from each STA of the polling list. In contrast with the AP, each polled STA consumes energy for receiving the beacon and a data packet and an ACK packet from the AP. Depending on its polling order, an STA also consumes energy for overhearing a number of data and ACK transmissions between the AP and the other STAs before being polled. Note that the last M STAs of the polling list consume energy for overhearing all the transmissions and for receiving the CE packet from the AP.

- **Idle period:** the AP and all the STAs consume energy to listen to the wireless channel for a PIFS interval. Then, each STA of the polling list listens to a number of SIFS intervals until it goes to sleep whereas the AP and the last M STAs of the polling list are idle during all the SIFS intervals.

- **Switch period:** the $N-M$ sleeping STAs consume energy during the transition from idle to sleep and during the transition from sleep to idle.

- **Sleep period:** each STA of the polling list, but the last M STAs, sleeps during the data exchanges between the AP and the rest of STAs until the CFP end, except for when it needs to switch between idle and sleep states.

The saturation AP energy efficiency of GreenPoll is the same as that of BidPoll since GreenPoll has been designed to improve the energy efficiency of the STAs.

To compute the energy consumption ratio that results in the saturation average per STA energy efficiency of GreenPoll, the components related to the energy consumption of the AP are removed from (4.22) and then the resulting expression is divided by N STAs. The specific contributions of the AP to the network energy consumption of GreenPoll are broadcasting the beacon and CE packets, receiving N data and ACK packets from the STAs, and listening to a PIFS and $2N{+}1$ SIFS intervals. As a result, the final expression is given as

$$\begin{aligned}
E_{GreenPoll}^{persta_sat} &= \frac{1}{N}\left(E_t+E_r+E_i+E_{sw}+E_s\right)\\
E_t &= NT_{DATA}P_t\\
E_r &= \left(\left(N^2+M\left(M-1\right)\right)T_{DATA}+\left(\frac{N+1}{2}N+\frac{M-1}{2}M\right)T_{ACK}\right)P_r\\
&\quad+\left(NT_B+MT_{CE}\right)P_r\\
E_i &= \left(NT_{PIFS}+\left(N\left(N+2\right)+M\left(M-1\right)\right)T_{SIFS}\right)P_i\\
E_{sw} &= \left(T_{i\to s}P_{i\to s}+T_{s\to i}P_{s\to i}\right)\left(N-M\right)\\
E_s &= T_sP_s
\end{aligned} \tag{4.22}$$

4.5 Simulations Framework

This section evaluates the performances of the considered protocols (also including the legacy DCF for the purpose of comparison with the polling-based access methods) by means of the analysis presented in the previous sections and computer-based simulations through an event-driven custom-made simulator coded in Python.

4.5.1 MAC Protocols Simulator

A BSS composed of an AP and a finite number of non-hidden static STAs, all of them operating in the ERP-OFDM-only mode. The AP and the STAs generate data packets of constant length through a Poisson arrival process and all data packets are received with no errors. The DCF is implemented with the RTS/CTS enabled and no PCF operating whereas the polling-based MAC protocols are implemented with the DCF not used. The DCF simulator has been described earlier in this thesis. The next lines will describe the implementation of the PCF, BidPoll, and GreenPoll protocols.

The simulator is composed of three main scripts according to the protocols under evaluation, i.e. PCF, BidPoll, and GreenPoll:

- "PCFMACsimulator.py": This script refers to the PCF MAC protocol.
- "BidPollMACsimulator.py": This script is related to the BidPoll MAC protocol
- "GreenPollMACsimulator.py": This script deals with the GreenPoll MAC protocol.

Each of these scripts contains the input parameters required to run the simulation of each protocol. These input parameters can be the simulation time, the number of simulation runs, the number of STAs, among other parameters included in Table 4.1 and Table 4.2. These main scripts are also used to collect the obtained results in an Excel file. Each main script calls an associated class that can be:

- "pcfmac.py": This class contains the PCF MAC rules.
- "bidpollmac.py": This class includes the BidPoll MAC rules.

- "greenpollmac.py": This class runs the GreenPoll MAC rules.

These classes are connected with three subclasses:

- "node.py": This subclass describes an STA or an AP. It contains attributes like association identifier, the state of a node, if it has packets, the packet box, the output packet queue, and several timers.
- "packet.py": This subclass describes a packet. It contains attributes like the arrival time, the departure time, the transmission delay, and the destination.
- "simreport.py": This subclass collects all the output values of the simulation, such as throughput, energy efficiency, delay, and energy consumption.

In each of these classes, the MAC rules of each protocol are implemented. First, all the input parameters passed from the main script are registered. Then, the code enters the main function called Run. In the Run function, the AP and the STAs are created as independent entities. Each STA is appended to a list of STAs. A box of packets is then generated for each STA and the AP according to a Poisson-distributed arrival process and considering the available simulation time. After that, a loop begins that is running until the simulation time is reached.

Inside the loop, there is a new loop that models what happens in a CFP, i.e., the code goes through each STA and checks if the AP has a downlink data packet ready to be transmitted to the STA and if the STA has an uplink data packet ready to be sent to the AP. So, this means that there are four possibilities: both the AP and the STA have data for each other, the AP has data but the STA does not, the STA has data but the AP does not, or neither the AP nor the STA have data to exchange with each other. Thus, the code splits into four conditions in which global and local variables are reinitialized or updated and before moving to the next STA in the polling list it is verified if the AP or any of the STAs has a new data packet.

When the simulation run is over, the simreport subclass is called to collect all the simulation results and return them to the main script.

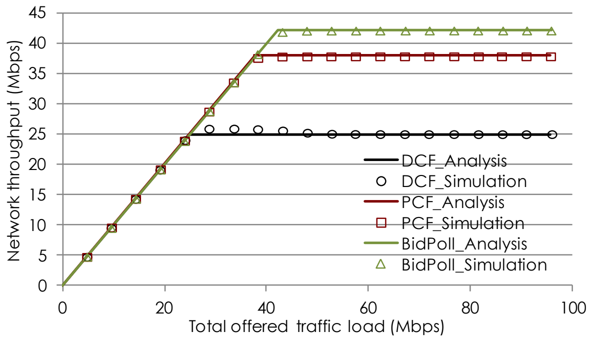
4.5.2 Analytical and Simulation Results

The results are shown in terms of throughput, energy efficiency, and energy consumption, considering different values for the traffic load, MSDU length, PHY data rate, number of STAs, wakeup transition coefficient (α) and awake/sleep transition time. All simulation runs were repeated 10 times for the duration of 15 seconds each and the simulation results in the plots are obtained with a 95% confidence interval lower than 0.01.

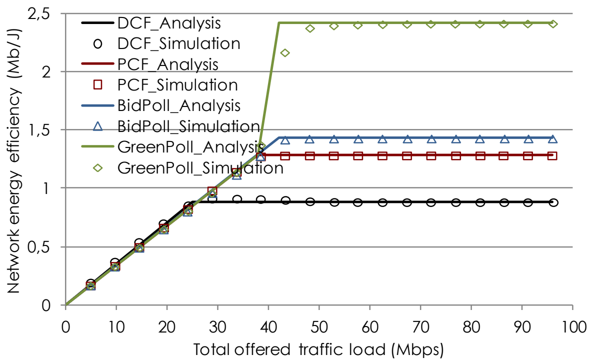
Traffic Load

The throughput and energy efficiency versus the traffic load are plotted in Fig. 4.6a. The results are obtained for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, a wakeup transition coefficient of 1.5, and an awake/sleep transition time of 500 μs (i.e., 250 μs each transition).

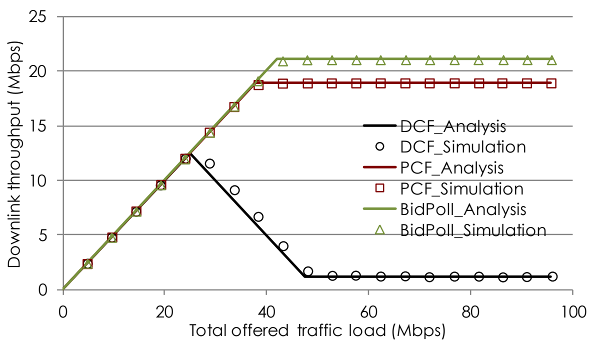
Fig. 4.6a and Fig. 4.6b show the network throughput and energy efficiency, respectively. In general, the throughput and energy efficiency of the protocols increase as the traffic load increases, until a stable value is shown when the network enters the saturation state. It can be seen that the proposed BidPoll and GreenPoll protocols outperform the DCF and PCF when the traffic load is high. Table 4.4 records the maximum gains of the protocols versus the DCF and PCF in terms of throughput and energy efficiency versus the traffic load. The maximum throughput gains of BidPoll versus DCF and PCF are 0.69 and 0.11, respectively, whereas the



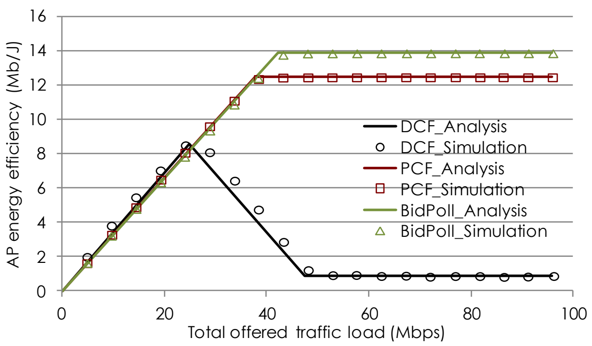
(a) Network throughput



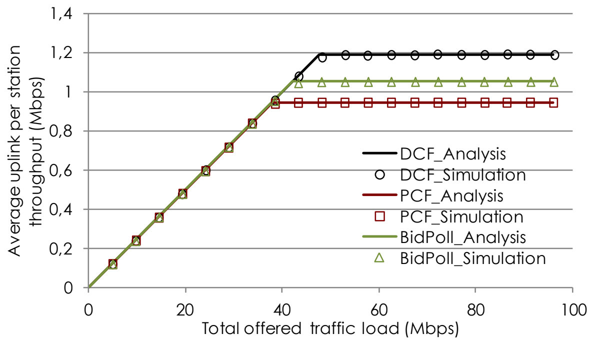
(b) Network energy efficiency



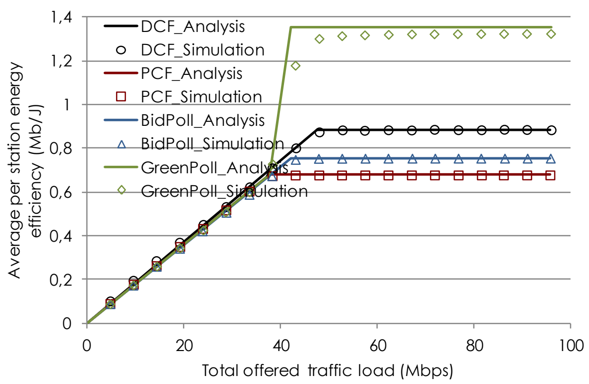
(c) Downlink throughput



(d) AP energy efficiency



(e) Average uplink per STA throughput



(f) Average per STA energy efficiency

Figure 4.6: Throughput and energy efficiency of the polling-based MAC protocols versus the traffic load

Table 4.3: Maximum Gains vs. Traffic Load

Saturation Traffic Load	BidPoll vs.		GreenPoll vs.		
	DCF	PCF	DCF	PCF	BidPoll
Network throughput	0.69	0.11	0.69	0.11	-
Downlink throughput	16.74	0.11	16.74	0.11	-
Average uplink per STA throughput	-0.11	0.11	-0.11	0.11	-
Network energy efficiency	0.63	0.11	1.75	0.89	0.69
AP energy efficiency	15.14	0.11	15.14	0.11	-
Average per STA energy efficiency	-0.14	0.11	0.53	0.99	0.79

maximum energy efficiency gains of GreenPoll versus DCF and PCF are 1.75 and 0.89, respectively.

Fig. 4.6c and Fig. 4.6d show the downlink throughput and the AP energy efficiency. The throughputs and energy efficiencies of DCF and PCF increase linearly as the traffic load increases. However, when the traffic load is above 20 Mbps, the throughput and energy efficiency of DCF decreases dramatically, due to the DCF fairness, until saturation. On the contrary, PCF and BidPoll are able to improve the throughput and energy efficiency of DCF for loads above 20 Mbps. Furthermore, BidPoll performs the best. As shown in Table 4.4, the maximum throughput gains of BidPoll versus DCF and PCF are 16.74 and 0.11, respectively, whereas the energy efficiency gains are 15.14 and 0.11, respectively.

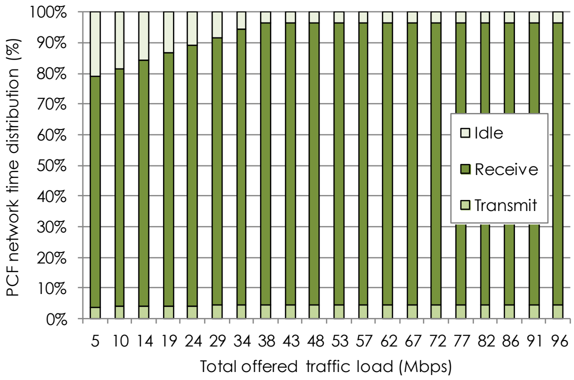
Fig. 4.6e and Fig. 4.6f show the average uplink per STA throughput and average per STA energy efficiency. It can be seen that BidPoll shows lower throughput and energy efficiency than those of DCF for the STAs in uplink in order to balance the uplink and the downlink, but higher than those of PCF. Furthermore, GreenPoll is able to compensate for the reduction of energy efficiency of BidPoll and can significantly improve the energy efficiency of DCF. Table 4.4 shows that the maximum throughput gains of BidPoll versus DCF and PCF are -0.14 and 0.11, respectively, whereas the maximum energy efficiency gains of GreenBid versus DCF and PCF are 0.53 and 0.99.

In Fig. 4.7, the contribution of each operational state to the overall energy consumption of the PCF and GreenPoll protocols is studied as the traffic load increases. Also, the amount of time that is spent in each of these states is shown. Fig. 4.7a and Fig. 4.7b illustrate the network time distribution of the PCF and GreenPoll protocols. In Fig. 4.7c and Fig. 4.7d, the network energy distributions of the PCF and GreenPoll protocols are plotted. It can be seen that in PCF most of the time and most of the energy resources (up to 75%) are dedicated to receiving and overhearing activities even when the traffic load is low. When the traffic load is high, the share increases up to 90%. On the other hand, GreenPoll reduces the time and energy consumed for receiving packets. However, it introduces the components of time and energy consumed for sleeping and switching between idle and sleeping. While the time and energy consumed during switch periods have a small contribution (up to 10%), the time spent during sleeping periods has a strong influence on the overall time (up to 65%) but marginal impact on the energy consumption (less than 5%).

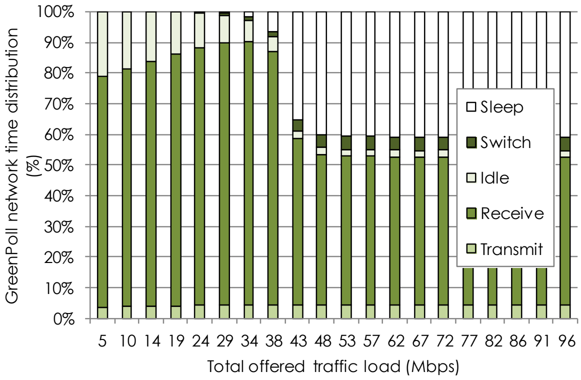
MSDU Length

Fig. 4.8 shows the saturation throughput and energy efficiency versus the MSDU length. The results are plotted for a PHY data rate of 54 Mbps, a wakeup transition coefficient of 1.5, and an awake/sleep transition time of 500 μs (i.e., 250 μs).

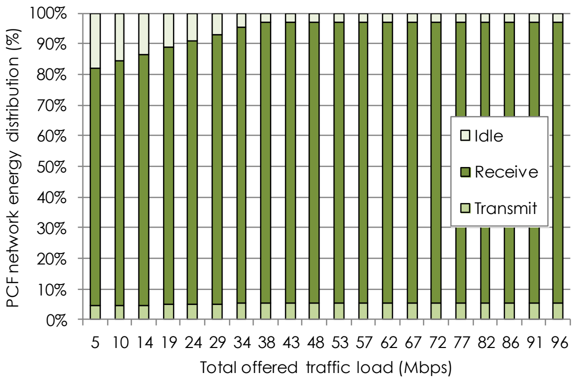
The saturation network throughput is plotted in Fig. 4.8a. In general, the throughput of the protocols under evaluation increases as the data payload increases since more information is transmitted. It is seen that BidPoll outperforms DCF and PCF for all MSDU lengths. However, Table 4.9 shows that the throughput gains of BidPoll versus DCF and PCF decrease as the packet length increases, due to the stronger influence of the



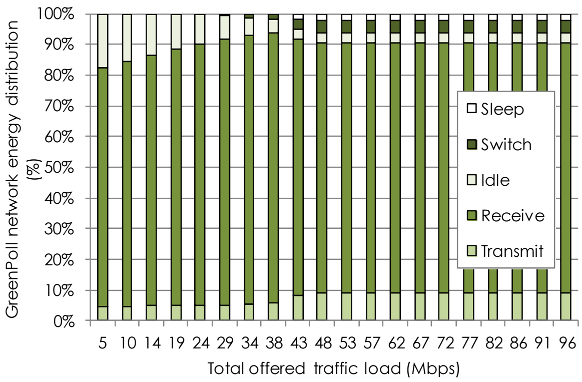
(a) PCF: network time distribution



(b) GreenPoll: network time distribution



(c) PCF: network energy distribution

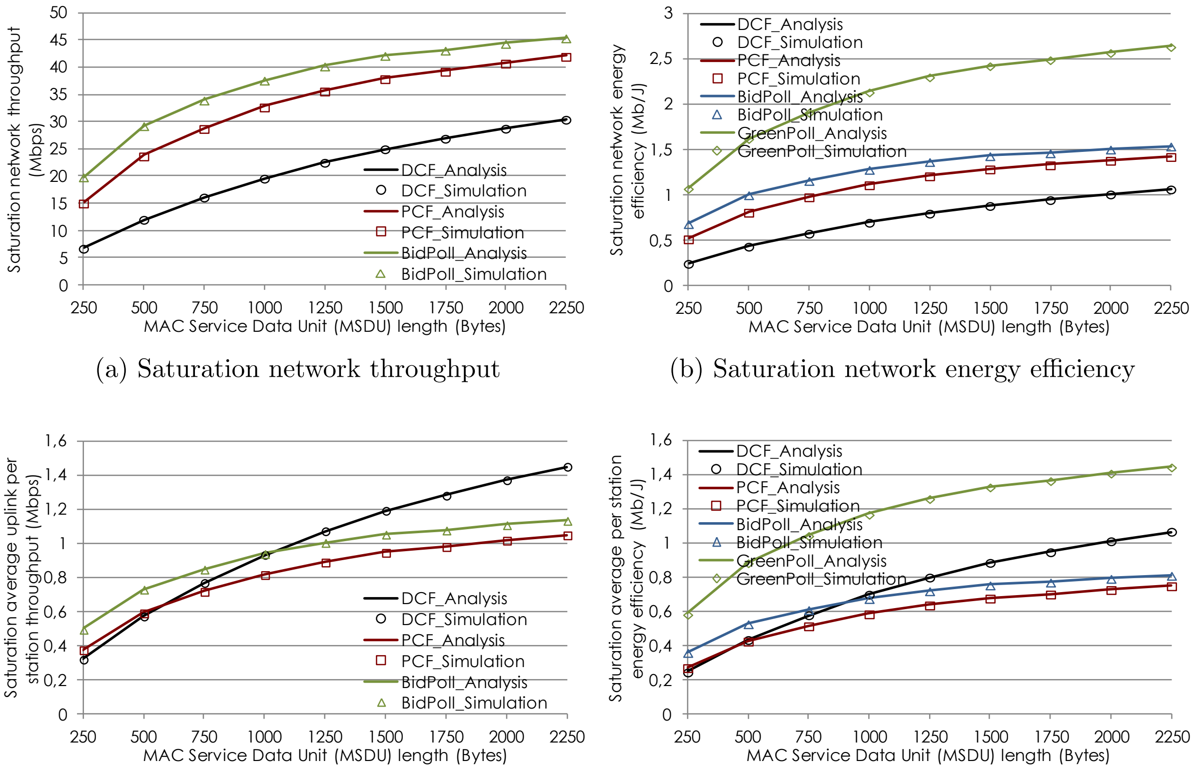


(d) GreenPoll: network energy distribution

Figure 4.7: Distribution of time and energy consumption of the polling-based MAC protocols versus the traffic load

data transmission on the overall transmission time. The maximum gains of 1.92 and 0.31 versus DCF and PCF, respectively, are achieved for an MSDU length of 50 bytes and the minimum gains of 0.49 and 0.09 versus DCF and PCF, respectively, are shown for an MSDU length of 2250 bytes.

The saturation network energy efficiency is plotted in Fig. 4.8b. Similar conclusions can be drawn for the protocols except for GreenPoll, whose energy efficiency is the highest for all MSDU lengths. Table 4.4 shows that the maximum energy efficiency gains of GreenPoll versus DCF and PCF



(a) Saturation network throughput

(b) Saturation network energy efficiency

(c) Saturation average uplink per STA throughput

(d) Saturation average per STA energy efficiency

Figure 4.8: Saturation throughput and energy efficiency of the polling-based MAC protocols versus the MSDU length

are achieved for an MSDU length of 250 bytes, where the gains are 3.38 and 1.08, respectively. Then, the gains decrease as the packet length increases. The reason for this is that the data transmission time has a higher impact on the total transmission time for longer packet lengths, thus increasing the total energy consumption for transmitting, receiving and overhearing, and sleeping.

The saturation average uplink per STA throughput is illustrated in Fig. 4.8c. The throughputs of PCF and BidPoll are higher than the throughput of DCF for lower packet lengths up to an MSDU length of 500 bytes for PCF and of 1000 bytes for BidPoll. For higher values of those MSDU lengths, PCF and BidPoll do not outperform DCF since these protocols

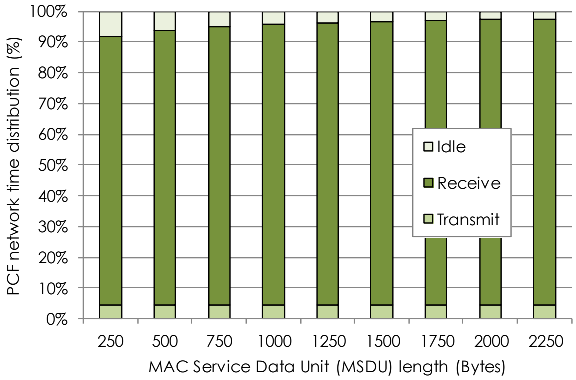
Table 4.4: Maximum Gains vs. MSDU Length

MSDU Length	Throughput BidPoll vs. DCF/PCF				Energy Efficiency GreenPoll vs. DCF/PCF			
	Network		Average Per STA		Network		Average Per STA	
	DCF	PCF	DCF	PCF	DCF	PCF	DCF	PCF
250 bytes	1.95	0.31	0.55	0.32	3.38	1.08	1.38	1.15
500 bytes	1.44	0.23	0.28	0.23	2.74	1.00	1.04	1.07
750 bytes	1.11	0.18	0.11	0.18	2.32	0.95	0.81	1.03
1000 bytes	0.92	0.15	0.01	0.15	2.07	0.92	0.68	1.00
1250 bytes	0.79	0.13	-0.06	0.13	1.89	0.90	0.58	0.97
1500 bytes	0.69	0.11	-0.11	0.11	1.75	0.89	0.51	0.96
1750 bytes	0.60	0.10	-0.16	0.10	1.63	0.87	0.44	0.95
2000 bytes	0.54	0.09	-0.19	0.09	1.55	0.86	0.40	0.94
2250 bytes	0.49	0.08	-0.22	0.08	1.48	0.85	0.36	0.93

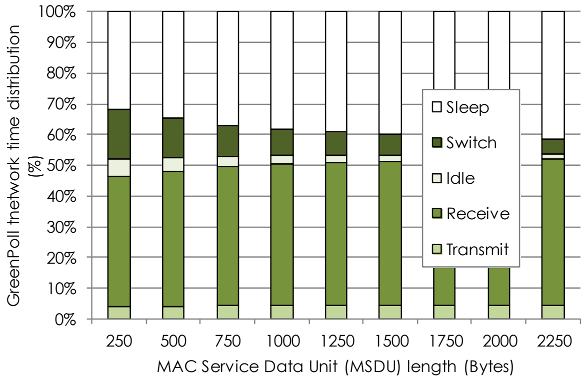
allow balancing the uplink and the downlink. As shown in Table 4.8c, the throughput gains of BidPoll versus DCF and PCF range from 0.55 to -0.22 and from 0.32 to 0.08, respectively.

Fig. 4.8d presents the average per STA energy efficiency. Similar conclusions can be drawn for the protocols except for GreenPoll, which achieves the highest energy efficiency for all MSDU lengths. In Table 4.4, it can be seen that the maximum energy efficiency gains of GreenPoll versus DCF and PCF are 1.08 and 1.38, respectively, for an MSDU length of 250 bytes.

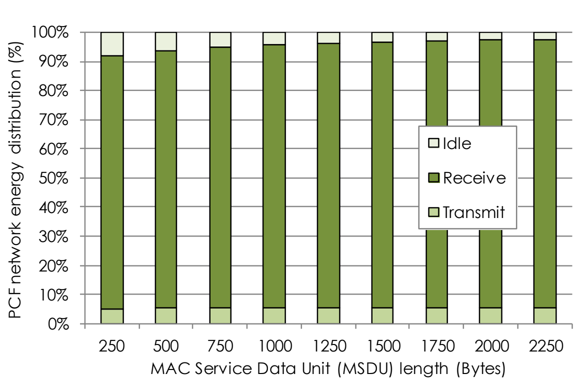
The network time and energy distributions of the PCF and GreenPoll protocols versus the MSDU length are provided in Fig. 4.9. The network time distribution of each protocol is shown in Fig. 4.9a and Fig. 4.9b, respectively. The network energy distribution of each protocol is presented in Fig. 4.9c and Fig. 4.9d. It can be seen that for PCF most of the energy and time resources (up to 90%) are spent for receiving and overhearing activities. The shares of time and energy consumption during reception periods increase with longer packet lengths. In GreenPoll the amount of



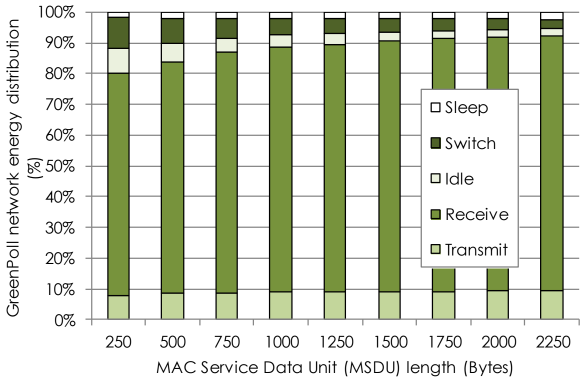
(a) PCF: network time distribution



(b) GreenPoll: network time distribution



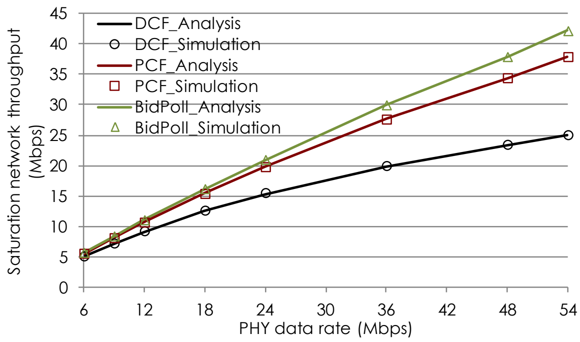
(c) PCF: network energy distribution



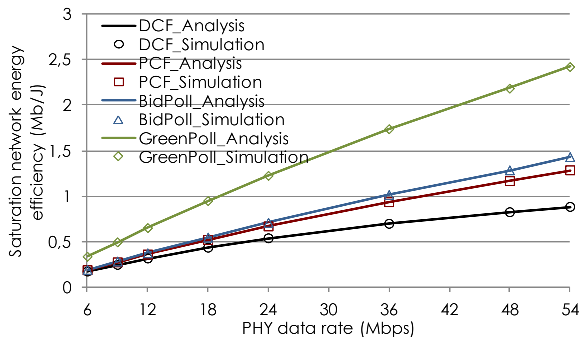
(d) GreenPoll: network energy distribution

Figure 4.9: Distribution of time and energy consumption of the polling-based MAC protocols versus the MSDU length

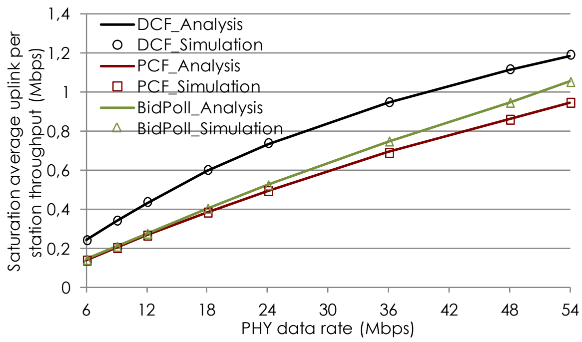
time spent in the sleep state increases as the MSDU length increases from 30% to 40% whereas the switch periods contribute from 20% to less than 10%. Regarding the energy distribution, the contributions of sleeping (less than 5%) and switching (up to 10%) show similar behaviors to those shown for the time distribution, although the overall impact is significantly lower.



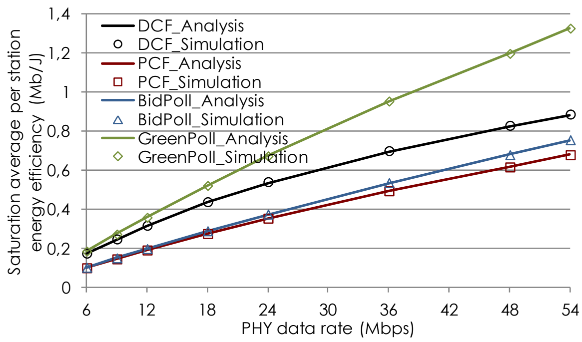
(a) Saturation network throughput



(b) Saturation network energy efficiency



(c) Saturation average uplink per STA throughput



(d) Saturation average per STA energy efficiency

Figure 4.10: Saturation throughput and energy efficiency of the polling-based MAC protocols versus the PHY data rate

PHY Data Rate

Fig. 4.10 shows the throughput and energy efficiency versus the PHY data rate. The results are obtained for an MSDU length of 1500 bytes, a wakeup transition coefficient of 1.5, and an awake/sleep transition time of 500 μs (i.e., 250 μs each transition).

The saturation network throughput is depicted in Fig. 4.10a. The throughput of each protocol increases as the data rate increases since the time to transmit a data packet decreases. The BidPoll protocol outperforms the DCF and PCF protocols for all data rates and can achieve higher gains as the data rate increases. This can be understood by the explanations given above for the MSDU length. Table 4.5 records the maximum

Table 4.5: Maximum Gains vs. PHY Data Rate

PHY Rate	Throughput				Energy Efficiency			
	BidPoll vs. DCF/PCF				GreenPoll vs. DCF/PCF			
	Network		Average Per STA		Network		Average Per STA	
	DCF	PCF	DCF	PCF	DCF	PCF	DCF	PCF
6 Mbps	0.12	0.02	-0.41	0.02	0.96	0.79	0.08	0.87
9 Mbps	0.17	0.03	-0.38	0.03	1.03	0.80	0.12	0.88
12 Mbps	0.22	0.03	-0.36	0.03	1.11	0.80	0.16	0.88
18 Mbps	0.29	0.05	-0.32	0.05	1.19	0.82	0.20	0.89
24 Mbps	0.37	0.06	-0.28	0.06	1.31	0.83	0.27	0.90
36 Mbps	0.50	0.08	-0.21	0.08	1.50	0.85	0.37	0.93
48 Mbps	0.62	0.10	-0.15	0.10	1.66	0.87	0.45	0.95
54 Mbps	0.69	0.11	-0.11	0.11	1.75	0.89	0.51	0.96

gains of the proposed protocols versus the PHY data rate. The throughput gains of BidPoll versus DCF and PCF range from 0.12 to 0.69 and from 0.02 to 0.11, respectively.

The saturation network energy efficiency is plotted in Fig. 4.10b. The energy efficiencies of DCF, PCF, and BidPoll show great similarities to what is shown in Fig. 4.10a for the throughput. In contrast, GreenPoll significantly improves DCF and PCF for all data rate. As shown in Table 4.5, the maximum gains of GreenPoll versus DCF and PCF vary from 0.96 to 1.75 and from 0.79 to 0.89, respectively.

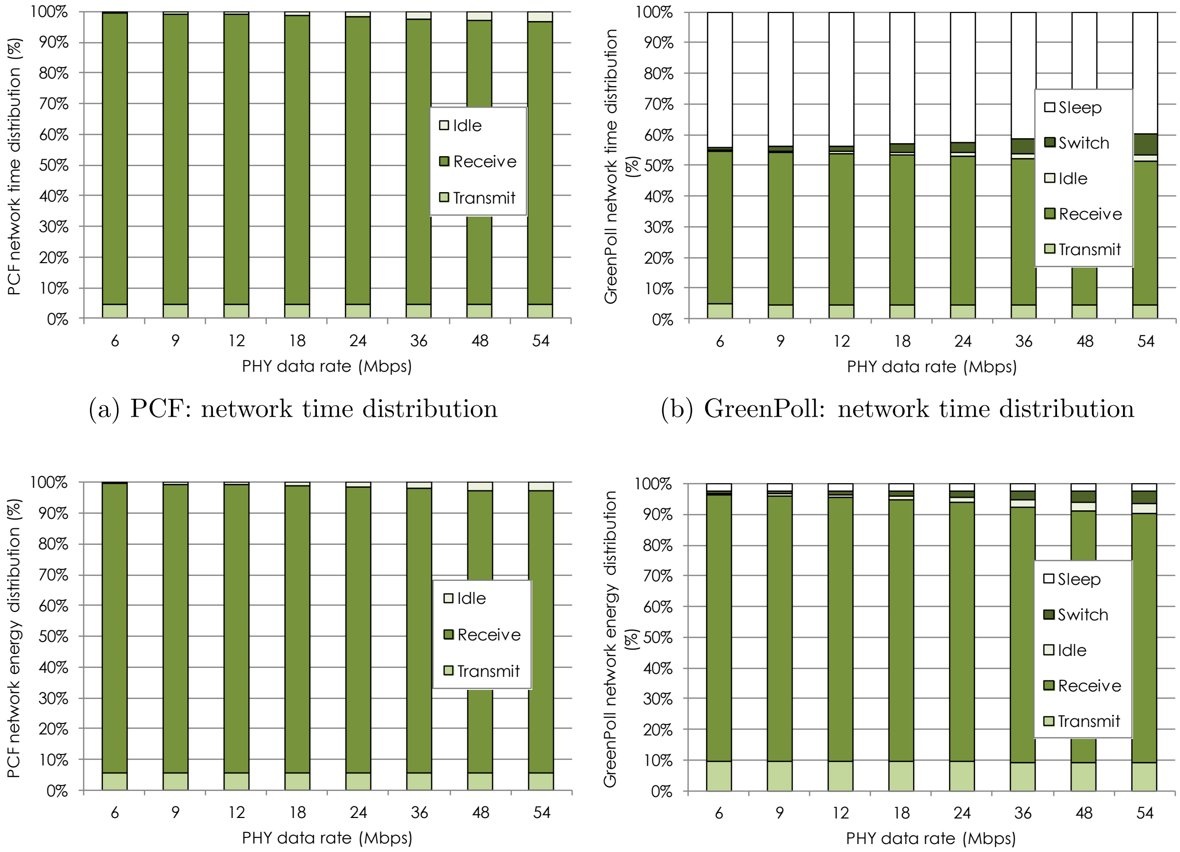
Fig. 4.10c presents the saturation average uplink per STA throughput. The throughputs of PCF and BidPoll are always lower than that of DCF to provide a balanced share of the wireless channel between the AP in downlink and the STAs in uplink, although BidPoll performs closer to DCF. Table 4.5 shows that the throughput reduction of BidPoll versus DCF decreases from -0.41 to -0.11 as the data rate increases and that the throughput improvement of BidPoll versus PCF increases from 0.02 to 0.11 as the data rate increases.

In Fig. 4.10d, the saturation average per STA energy efficiency is plotted. The GreenPoll protocol is able to outperform the DCF and PCF protocols for all data rates. As provided in Table 4.5, the maximum energy efficiency gains of GreenPoll versus DCF and PCF are between 0.08 and 0.51 and between 0.87 and 0.96 as the data rate increases.

The impact of the PHY data rate on the time spent and energy consumed in the different operational states for the PCF and GreenPoll protocols is evaluted in Fig. 4.11. Fig. 4.11a refers to the PCF network time distribution whereas Fig. 4.11b plots the GreenPoll network time distribution. In Fig. 4.11c, the PCF network energy distribution is shown and in Figure 4.11d the GreenPoll network energy distribution is presented. In PCF, the shares of time and energy consumed during reception periods slightly decrease as the data rate increases because the data transmission time decreases. In contrast, for GreenPoll the network remains in the sleep state for more than 40% of time for a data rate of 6 Mbps and less than 5% during switching periods. The share of energy consumption for 6 Mbps is less than 5% for sleeping and switching. However, when the data rate increases, the share of time spent during sleep periods is reduced down to 40 % whereas the contribution for sleeping energy consumption is very small (less than 5%). In addition, the time and energy consumed during switching periods do not represent more than 10% of the overall network energy consumption and time.

Number of STAs

The throughput and energy efficiency versus the number of STAs are shown in Fig. 4.12. The results are obtained for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, a wakeup transition coefficient of 1.5, and an awake/sleep transition time of 500 μs (i.e., 250 μs each transition).



(a) PCF: network time distribution

(b) GreenPoll: network time distribution

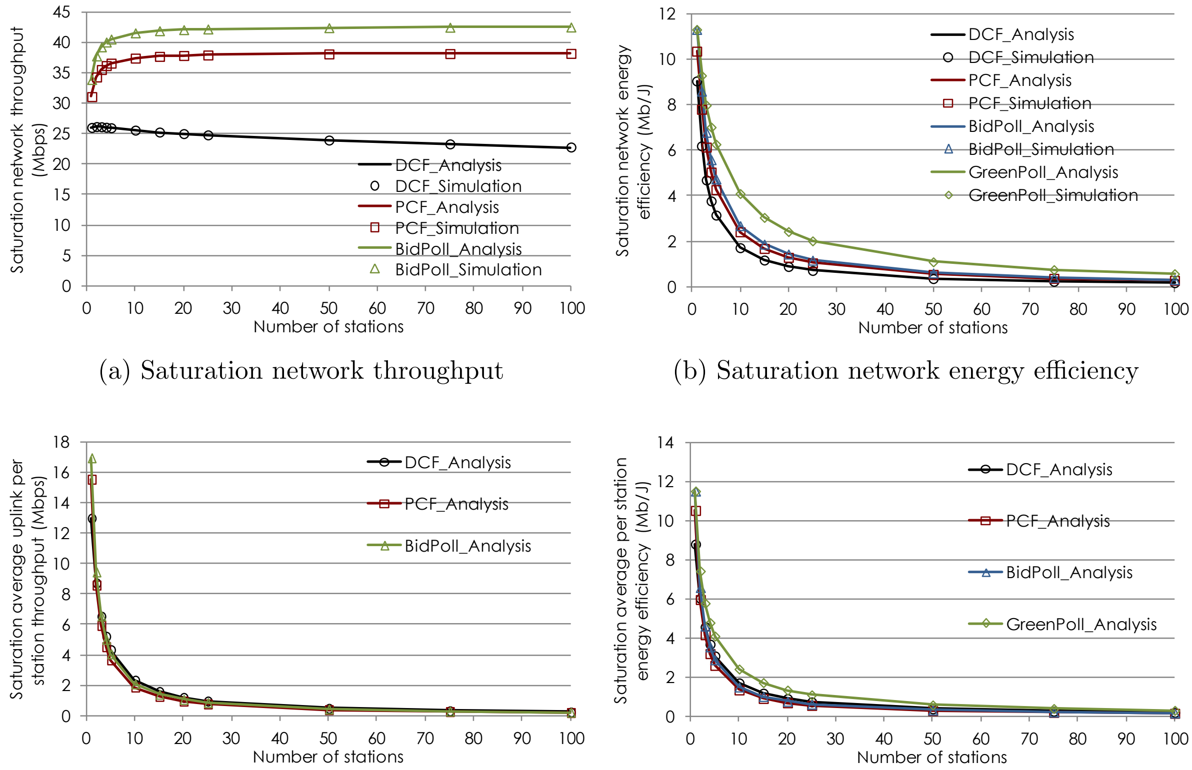
(c) PCF: network energy distribution

(d) GreenPoll: network energy distribution

Figure 4.11: Distribution of time and energy consumption of the polling-based MAC protocols versus the PHY data rate

Fig. 4.12a shows the saturation network throughput as the number of STAs increases. The throughputs of PCF and BidPoll increase as the number of STAs increases whereas that of DCF increases for small numbers of STAs and then decreases when the number of STAs becomes bigger. BidPoll achieves the highest throughput in all cases, showing outstanding gains when compared to DCF. As shown in Table 4.6, the maximum throughput gains of BidPoll versus DCF and PCF vary between 0.31 and 0.88 and between 0.09 and 0.11.

The saturation network energy efficiency versus the number of STAs is



(a) Saturation network throughput

(b) Saturation network energy efficiency

(c) Saturation average uplink per STA throughput

(d) Saturation average per STA energy efficiency

Figure 4.12: Saturation throughput and energy efficiency of the polling-based MAC protocols versus the number of STAs in the network

presented in Fig. 4.13. In general, the energy efficiency of the protocols decreases as the number of STAs increases because more STAs are overhearing during data transmissions. The BidPoll protocol performs better than DCF and PCF whereas the GreenPoll protocol achieves the highest energy efficiency when there are two STAs or more in the network. Table 4.6 shows that the maximum energy efficiency gains of GreenPoll versus DCF and PCF increase from 0.25 to 2.37 and from 0.09 to 1.09, respectively, as the number of STAs increases.

Fig. 4.10c presents the saturation average uplink per STA throughput. The throughputs of PCF and BidPoll are higher than that of DCF for few STAs (up to 3 STAs) whereas for many STAs they become lower due to the

Table 4.6: Maximum Gains vs. Number of STAs

Num. of STAs	Throughput BidPoll vs. DCF/PCF				Energy Efficiency GreenPoll vs. DCF/PCF			
	Network		Average Per STA		Network		Average Per STA	
	DCF	PCF	DCF	PCF	DCF	PCF	DCF	PCF
1	0.31	0.09	0.31	0.09	0.25	0.09	0.31	0.09
2	0.44	0.10	0.08	0.10	0.50	0.19	0.23	0.24
3	0.50	0.10	0.00	0.10	0.70	0.30	0.26	0.39
4	0.54	0.11	-0.04	0.11	0.86	0.39	0.30	0.50
5	0.56	0.11	-0.06	0.11	0.99	0.47	0.33	0.58
10	0.63	0.11	-0.10	0.11	1.40	0.70	0.42	0.80
15	0.66	0.11	-0.11	0.11	1.61	0.82	0.46	0.90
20	0.69	0.11	-0.11	0.11	1.75	0.89	0.49	0.96
25	0.71	0.11	-0.11	0.11	1.85	0.93	0.51	1.00
50	0.78	0.11	-0.09	0.11	2.12	1.04	0.54	1.07
75	0.83	0.11	-0.07	0.11	2.26	1.07	0.55	1.10
100	0.88	0.11	-0.05	0.11	2.37	1.09	0.56	1.11

balanced share of the channel between the AP in downlink and the STAs in uplink. Note that BidPoll always outperforms PCF. Table 4.5 shows that the throughput gain of BidPoll versus DCF decreases from 0.31 to -0.11 as the number of STAs increases whereas that of BidPoll versus PCF increases from 0.09 to 0.11.

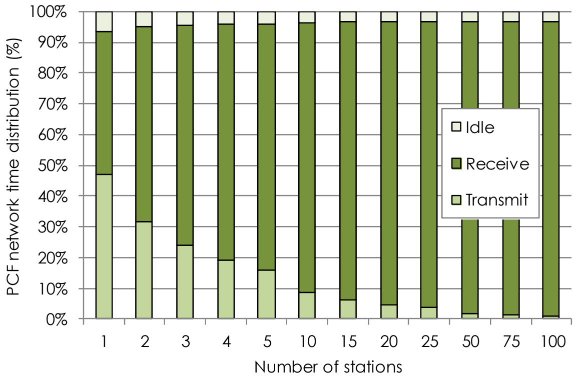
In Fig. 4.10d, the saturation average per STA energy efficiency is plotted. The GreenPoll protocol is able to outperform the DCF and PCF protocols for all the numbers of STAs considered in the network. As provided in Table 4.5, the maximum energy efficiency gains of GreenPoll versus DCF and PCF are between 0.31 and 0.56 and between 0.09 and 1.11, respectively, as the number of STAs increases.

In Fig. 4.13, the influence of the number of STAs on the distribution of time and energy consumption of the PCF and GreenPoll protocols in the different operational states is analyzed. Fig. 4.13a and Fig. 4.13b show the network time distributions of PCF and GreenPoll, respectively. Fig. 4.13c and Fig. 4.13d illustrate the network energy distributions of the protocols, respectively. In PCF, when the number of STAs is small the network time and energy is mainly dedicated to transmitting (more than 45%) and receiving and overhearing activities (more than 45%) whereas less than 10% goes to idle periods. For large numbers of STAs, most of time and energy is dedicated to receiving and overhearing activities (more than 95%). Regarding the network time distribution of GreenPoll, most of time (more than 95%) is roughly equally distributed between sleeping (around 47.5%) and receiving and overhearing activities (around 47.5%), whereas transmitting, idle, and switching periods have an overall small contribution. In contrast, the network energy distribution of GreenPoll shows that the switching energy contribution is higher (more than 10%) than the sleep energy contribution (less than 10%) when there are few STAs. Then, when there are many STAs both contribute little in favor of the receiving energy contribution, which represents most of the total energy consumption (around 90%).

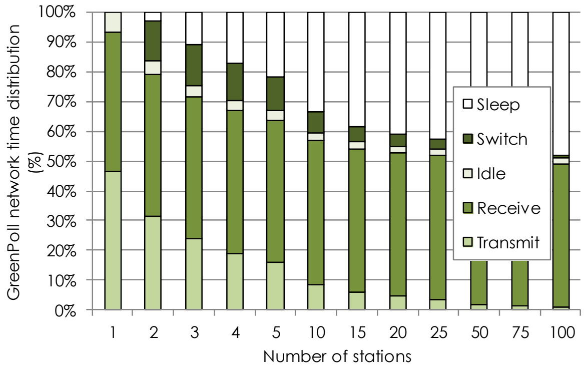
Wakeup Transition Coefficient

Fig. 4.14 shows the energy efficiency and time and energy distributions of the protocols versus the wakeup transition coefficient. This coefficient determines the amount of energy consumed in the transition between sleep and idle states having as reference the value of power consumed in the idle state. The higher the value of the wakeup transition coefficient is, the higher the energy consumed in the transition between sleep and idle states is. The results are obtained for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, and an awake/sleep transition time of 500 μs (i.e., 250 μs each transition).

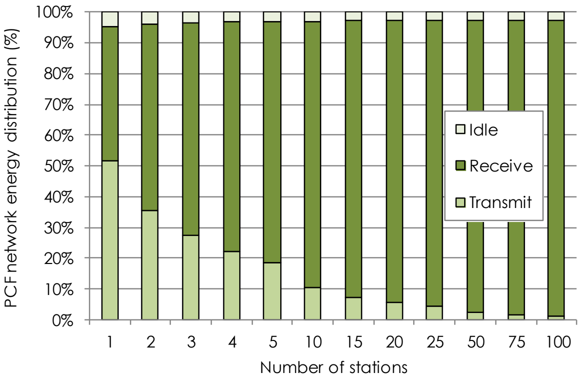
The saturation network energy efficiency is plotted in Fig. 4.14a. The



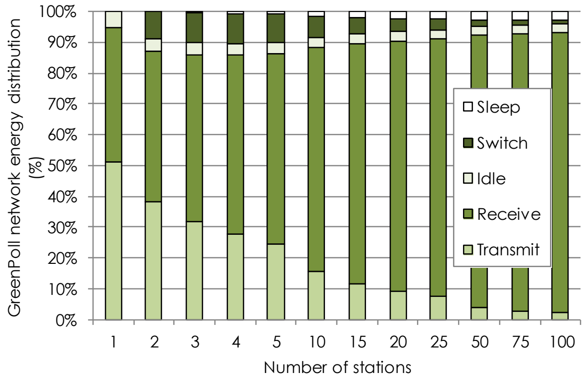
(a) PCF: network time distribution



(b) GreenPoll: network time distribution



(c) PCF: network energy distribution

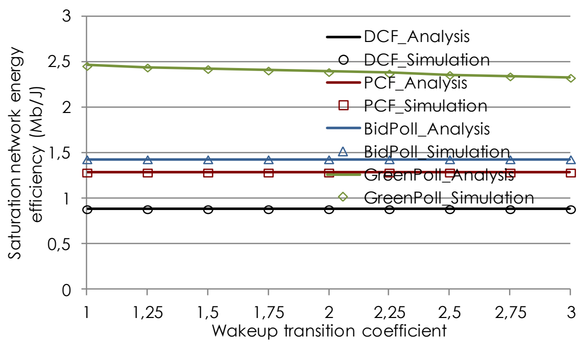


(d) GreenPoll: network energy distribution

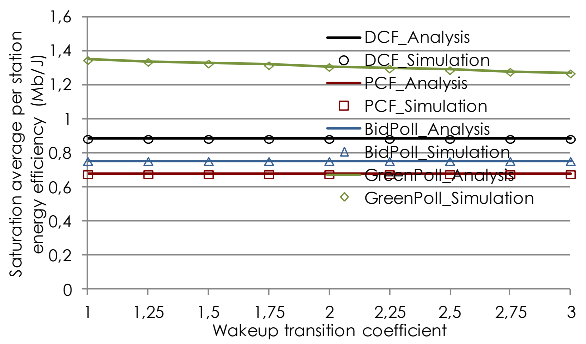
Figure 4.13: Distribution of time and energy consumption of the polling-based MAC protocols versus the number of STAs in the network

value of the wakeup transition coefficient only affects the energy efficiency of the GreenPoll protocol. As the value of the wakeup transition coefficient increases, the energy efficiency of GreenPoll decreases slightly but still GreenPoll achieves the highest energy efficiency when compared to the DCF and PCF. Table 4.7 records the maximum gains of GreenPoll versus the wakeup transition coefficient. The maximum gain of GreenPoll versus DCF ranges from 1.79 to 1.64 whereas that of GreenPoll versus PCF varies between 0.91 and 0.81.

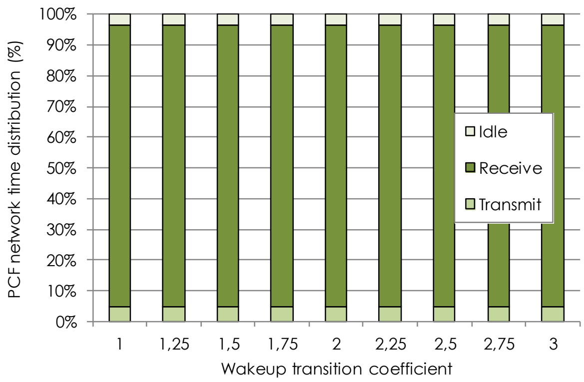
In addition, the saturation average per STA energy efficiency is shown in



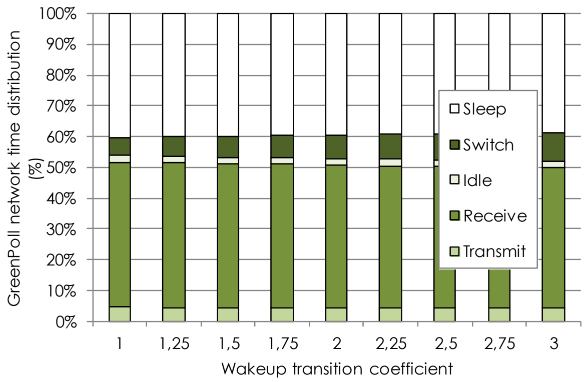
(a) Saturation network energy efficiency



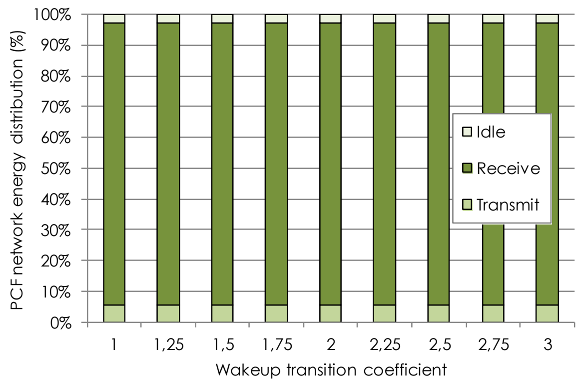
(b) Saturation average per STA energy efficiency



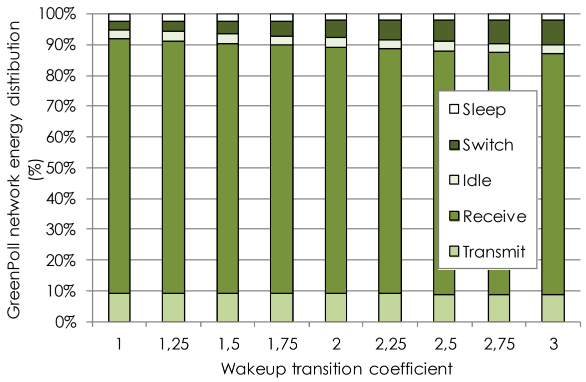
(c) PCF: network time distribution



(d) GreenPoll: network time distribution



(e) PCF: network energy distribution



(f) GreenPoll: network energy distribution

Figure 4.14: Energy efficiency and time and energy distributions of the polling-based MAC protocols versus the wakeup transition coefficient

Table 4.7: Maximum Energy Efficiency Gains vs. Wakeup Transition Coefficient

Wakeup Transition Coefficient	Network GreenPoll vs. DCF	PCF	BidPoll	Average Per STA GreenPoll vs. DCF	PCF	BidPoll
1	1.79	0.91	0.72	0.53	0.99	0.79
1.25	1.77	0.90	0.71	0.52	0.97	0.77
1.5	1.75	0.89	0.69	0.51	0.96	0.76
1.75	1.73	0.87	0.68	0.49	0.94	0.75
2	1.71	0.86	0.67	0.48	0.93	0.73
2.25	1.70	0.85	0.66	0.47	0.92	0.72
2.5	1.68	0.83	0.65	0.46	0.90	0.71
2.75	1.66	0.82	0.64	0.45	0.89	0.70
3	1.64	0.81	0.63	0.44	0.87	0.68

Fig. 4.14b. The energy efficiency of GreenPoll also decreases as the wakeup transition coefficient increases, as shown in Fig. 4.14a for the network energy efficiency. However, GreenPoll always performs better than the rest of protocols. As shown in Table 4.7, the maximum energy efficiency gains of GreenPoll versus DCF and PCF are between 0.53 and 0.44 and between 0.99 and 0.87, respectively.

Finally, the evaluation of the impact of the wakeup transition coefficient on the overall time and energy consumption distributions of the protocols is presented as follows. Fig. 4.15c and Fig. 4.15c show the network time distributions of PCF and GreenPoll, respectively. Similarly, Fig. 4.15d and Fig. 4.15f represent the network energy distribution of PCF and GreenPoll, respectively. In GreenPoll, it can be seen that as the wakeup transition coefficient increases more time and energy are consumed during the switching procedure. A maximum value of 10% of the overall time and energy consumption corresponds to switching.

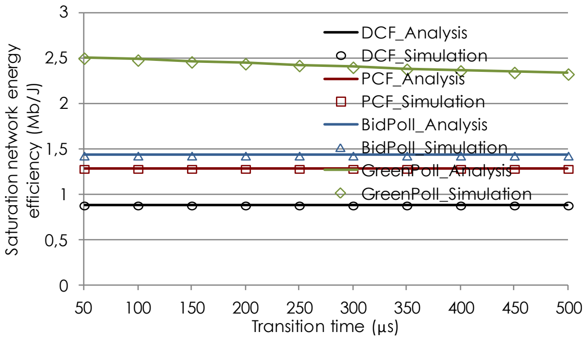
Awake/Sleep Transitions Time

The energy efficiency and time and energy distributions of the protocols versus the awake/sleep transitions time are shown in Fig. 4.15. The transition time determines how much time is spent in the transition from idle to sleep and the transition from sleep to idle. The longer the transition time is, the longer the data transmission time has to be in order to make the sleep period be greater than zero. The results are obtained for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, and a wakeup transition coefficient of 1.5.

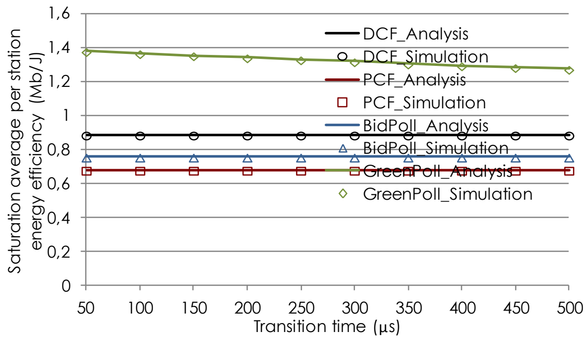
Fig. 4.15a shows the saturation network energy efficiency. The value of the transition time only affects the energy efficiency of the GreenPoll protocol. As the transition time increases the energy efficiency of GreenPoll decreases since the sleep period also decreases and so the STAs can remain in the sleep state less time. For all the cases studied, GreenPoll always achieves the highest energy efficiency. Table 4.8 reports the maximum gains of GreenPoll versus the transition time. The gains of GreenPoll versus DCF and PCF vary between 1.84 and 1.65 and between 0.95 and 0.81, respectively.

In Fig. 4.15a the saturation average per STA energy efficiency is presented. Similar conclusions to those extracted from Fig. 4.15a can be drawn, i.e., GreenPoll always achieves the highest energy efficiency in spite of increasing the transition time, which reduces the sleep period and so the STAs consume more energy. As shown in Table 4.8, the maximum gains of GreenPoll versus DCF and PCF range from 0.56 to 0.44 and from 1.03 to 0.88, respectively.

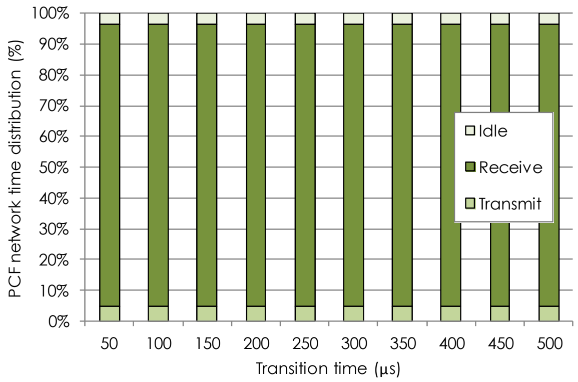
To conclude, the influence of the transition time on the time and energy distributions of the PCF and GreenPoll protocols along the different operation states is studied as follows. Fig. 4.15c illustrates the network time distribution of DCF whereas Fig. 4.15d represents the network energy



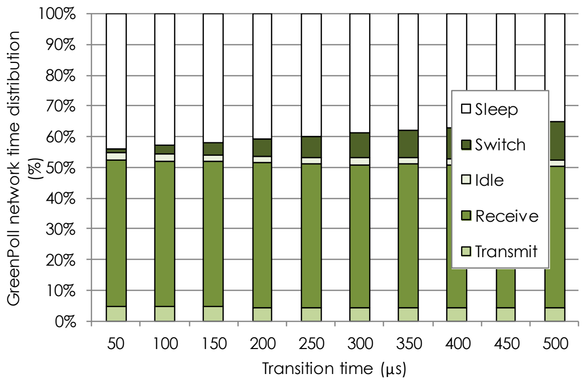
(a) Saturation network energy efficiency



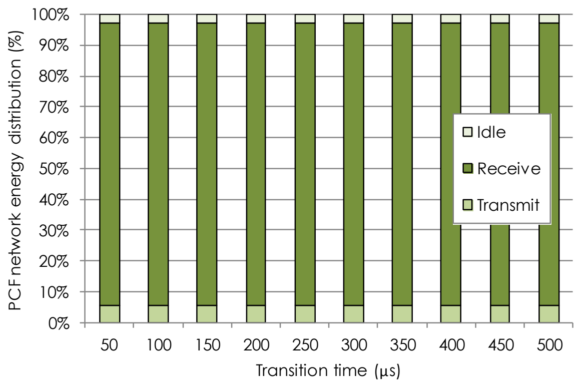
(b) Saturation average per STA energy efficiency



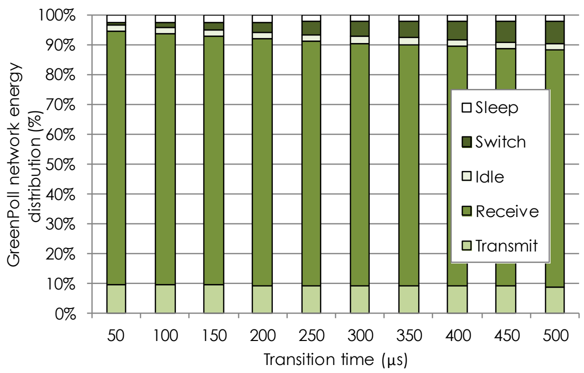
(c) PCF: network time distribution



(d) GreenPoll: network time distribution



(e) PCF: network energy distribution



(f) GreenPoll: network energy distribution

Figure 4.15: Energy efficiency and time and energy distributions of the polling-based MAC protocols versus the total awake/sleep transitions time

Table 4.8: Maximum Energy Efficiency Gains vs. Awake/Sleep Transitions Time

Awake/Sleep Transition Time	Network GreenPoll vs.			Average Per STA GreenPoll vs.		
	DCF	PCF	BidPoll	DCF	PCF	BidPoll
50 μs	1.84	0.95	0.75	0.56	1.03	0.82
100 μs	1.82	0.93	0.74	0.55	1.01	0.81
150 μs	1.80	0.92	0.72	0.53	0.99	0.79
200 μs	1.77	0.90	0.71	0.52	0.98	0.78
250 μs	1.75	0.89	0.69	0.51	0.96	0.76
300 μs	1.73	0.87	0.68	0.49	0.94	0.74
350 μs	1.71	0.86	0.67	0.48	0.92	0.73
400 μs	1.69	0.84	0.65	0.47	0.91	0.71
450 μs	1.67	0.83	0.64	0.45	0.89	0.70
500 μs	1.65	0.81	0.63	0.44	0.88	0.69

distribution of GreenPoll. Likewise, the DCF network energy distribution is shown in Fig. 4.15e and the GreenPoll network energy distribution is plotted in Fig. 4.15f. In GreenPoll, the sleeping time contribution varies from 45% to 35% whereas the switching time contribution ranges from 5% to 10%. In contrast, the sleeping energy contribution is very small (up to 5%) whereas the switching energy contributes up to 10%.

4.6 Conclusions

BidPoll and GreenPoll have been presented in this chapter as new energy-efficient centralized MAC protocols that have been designed to improve both the throughput and energy efficiency of the PCF of the IEEE 802.11 Standard for WLANs. The basic idea behind BidPoll is to split the CFP into two virtual phases. The first phase is reserved for low-overhead uplink and downlink transmissions between the AP and the STAs that requested TXOPs in the previous CFP. The second phase is used for dynamic (possibly low-overhead) data exchanges between the AP and the rest of STAs that did not take part in the first phase. The first phase of BidPoll is deterministic (i.e., the duration is announced through beacons). Thus, GreenPoll exploits the duration information of the first phase to allow the STAs involved in this phase to enter the sleep state from the time instants at which they receive the ACK packets to their transmitted data packets until the end of the first phase. In addition, those STAs not involved in the first phase can also enter the sleep state until this phase completes.

The closed expressions of the maximum achievable throughputs and energy efficiencies of PCF, BidPoll, and GreenPoll have been derived taking into account the influence of the on/off radio transitions. Also, a Python simulation environment where the protocol rules have been implemented has been developed for the validation of the proposed analytical model. The performances of the protocols have been evaluated in a WLAN composed of an AP and 20 STAs considering relevant system parameters such as the traffic load, data payload length, data rate, number of STAs in the network, wakeup radio transition coefficient, and awake/sleep radio transitions time. Both analytical and simulation results have shown the high performances of BidPoll and GreenPoll when compared to those of the DCF and PCF for all evaluated parameters.

More specifically, the throughput gains versus DCF and PCF vary from 195% to 49% and from 31% to 8%, respectively, as the packet length increases and from 12% to 69% and from 2% to 11%, respectively, as the data rate increases. The throughput gains versus DCF and PCF are between 31% and 88% and between 9% and 11%, respectively, as the number of STAs increases. The energy efficiency gains versus DCF and PCF range from 338% to 148% and from 108% to 85%, respectively with increasing packet lengths and from 96% to 175% and from 79% to 89%, respectively, with increasing data rates. Also, they vary between 25% and 237% and 9% and 109%, respectively, with larger numbers of STAs. In addition, the results have shown that the impact of the on/off radio transitions represents the 10% of the total energy consumption of GreenPoll from the entire network perspective. In this sense, the energy efficiency gains versus DCF and PCF vary between 179% and 164% and between 91% and 81%, respectively, as the wakeup radio transition coefficient increases. Similarly, the gains versus DCF and PCF are between 184% and 165% and between 75% and 63%, respectively, as the awake/sleep radio transition time increases. These parameters will vary depending on the radio hardware design and are important for the proper operation of GreenPoll.

Therefore, this chapter has demonstrated through analysis and computer-based simulation that the proposed energy-efficient centralized MAC protocols can improve the throughput and energy efficiency of the legacy DCF and PCF in WLANs.

Chapter 5

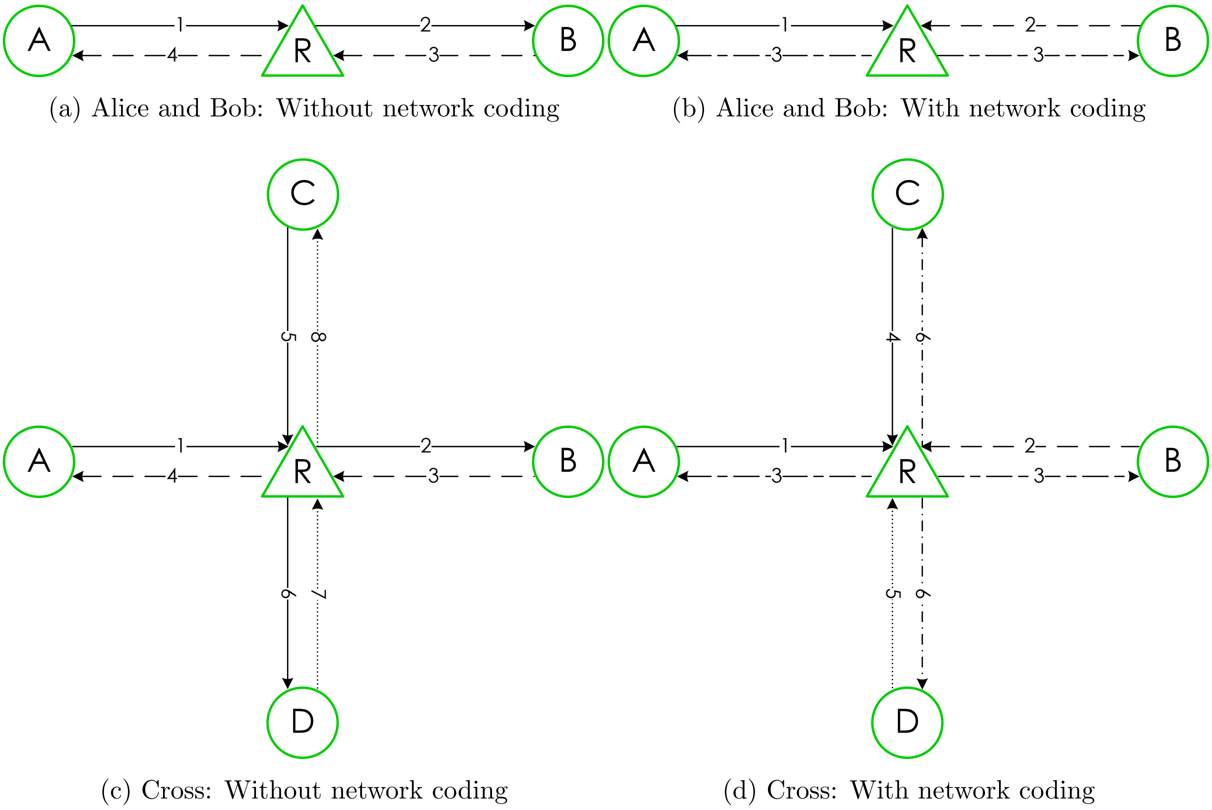
Network Coding-Aware Energy-Efficient MAC Protocols

5.1 Introduction and Related Work

The NC paradigm has been widely recognized as a powerful mean for improving throughput and energy efficiency in wireless networks [17]. The basic idea behind the NC approach is to allow intermediate (or relay) nodes to simultaneously transmit combined information from one or several sources to multiple destinations by exploiting the broadcast channel. This operation implies a reduction of the total number of channel accesses and thus results in less time spent and energy consumed per delivered bit of information.

Depending on whether combined data are composed of data from a single source or from several sources, the NC operation is classified into intra-session (i.e., combining packets from the same data flow) or inter-session (i.e., combining packets from different data flows). Fig. 5.1 shows the advantages of inter-session NC schemes over traditional store and forward schemes in two canonical scenarios, namely, the Alice and Bob and cross topologies.

In the Alice Bob scenario, two sources nodes (i.e., nodes A and B) exchange a pair of data packets through a relay node (i.e., node R). As



(a) Alice and Bob: Without network coding

(b) Alice and Bob: With network coding

(c) Cross: Without network coding

(d) Cross: With network coding

Figure 5.1: Reference scenarios: Alice and Bob and cross topologies, (a) and (c) without network coding and (b) and (d) with network coding

it can be seen in Fig. 5.1a, without NC the relay node forwards the data packets from nodes A and B to their respective destinations. In total, 4 transmissions are required for the exchange of a pair of data packets from end to end. However, when NC is enabled (see Fig. 5.1b), the relay node is able to combine the two data packets using the XOR operation and broadcast the new coded data packet. Then, nodes A and B can subtract the packet of each other by performing the XOR operation for decoding with the received coded data packet and their own data packet. In this case, 3 transmissions, instead of 4, are required. Therefore, NC can improve the network throughput, since 1 transmission out of 4 can be used to send new data. In addition, it reduces the amount of redundant transmissions, hence improving energy efficiency.

In the cross scenario, two pairs of sources nodes (i.e., nodes A and B and nodes C and D) exchange a pair of data packets through a common relay node (i.e., node R). As it can be see in Fig. 5.1c, without NC the relay node forwards the data packets from nodes A and B and nodes C and D, respectively, to their respective destinations. In total, 8 transmissions are required for the exchange of a pair of data packets from end to end. When NC is enabled (see Fig. 5.1d), instead, the relay node is able to combine pairs of data packets from nodes A and B and nodes C and D, respectively, and broadcast the new coded data packets. As a results, 6 transmissions, instead of 8, are required.

Despite the potential throughput gains and energy savings of NC, the authors of [18] showed that there exist important practical considerations that should be taken into account for the proper implementation of NC in currently operating wireless networks. This inspiring work introduced the first system architecture of a practical inter-session NC protocol, named COPE, for real Wi-Fi networks (i.e., based on the IEEE 802.11 Standard [16]). COPE seamlessly integrates an NC layer into the current protocol stack between MAC and IP layers, which is responsible for identifying coding opportunities in order to forward multiple data packets in a single transmission.

One of the main contributions of [18] was to show the impact of the MAC protocol on the performance of NC. COPE employs the widely used distributed channel access method of the IEEE 802.11 Standard (DCF). This MAC protocol is a variation of CSMA/CA by which nodes sense the shared wireless channel before transmitting and get random access to it through channel contention. Indeed, NC awareness of the MAC protocol is essential to allow assigning different channel access priorities to multiple nodes based on the NC operation. Unfortunately, the standard DCF does not represent a suitable MAC protocol as it was designed to provide equal channel access opportunities for all contenting nodes on average.

To illustrate this problem, let us consider the Alice and Bob and cross scenarios shown in Fig. 5.1. In the Alice and Bob topology (see Fig. 5.1a), relay node R would always capture 1/2 of the wireless channel to send twice more data packets than source nodes A and B. However, node R will get 1/3, due to the DCF fairness, when nodes A and B increase their transmission rates, which together will capture up to 2/3 of the wireless channel. Therefore, node R will not be able to forward data packets to nodes A and B with the same rate as they arrive. On the contrary, when node R enables NC operations (see Fig. 5.1b), it uses 1/3 of the wireless channel to send coded data packets that contain pairs of data packets from nodes A and B, reaching 2/3 and thus matching the incoming and outgoing rates.

Furthermore, in the cross topology (see Fig. 5.1c), node R gets 1/5 of the wireless channel, because there are four source nodes around it, while it is receiving data packets from nodes A, B, C, and D with 4/5 rate. Even though NC is enabled (see Fig. 5.1d), node R can only send a coded data packet composed of a pair of data packets from nodes A and B or nodes C and D, respectively, reaching 2/5, which is not sufficient to match the incoming rate. Therefore, providing additional transmission priority for congested relay nodes is essential to fully exploit the advantages of NC.

Existing NC-aware MAC protocols presented in [26, 72, 74, 75] are based on tuning the CW size used in the DCF backoff procedure considering the level of congestion, the state of channel contention, and NC information, aiming to assign different channel access priorities to several nodes. Unfortunately, these approaches assume that relay nodes ready to transmit coded data packets will compete for channel access as if they were regular nodes. Hence, probabilistic channel access priority can only be provided for congested relay nodes, which does not guarantee an immediate (i.e., contention-free) channel access when they actually need it.

Another essential requirement for the proper operation of NC in wireless networks is that nodes must enable the promiscuous mode in order to overhear all wireless transmissions, seeking for coding and decoding opportunities. However, not all overheard data packets may be useful for a given node. For example, in Fig. 5.1d, when node R sends a coded data packet containing data packets from nodes A and B, nodes C and D do not benefit from overhearing that coded data packet. Overhearing requires nodes to keep their radio transceivers always on, hence consuming significant amounts of energy. To reduce energy consumption, nodes may enter a low-power doze (or sleep) state where their radio transceiver are turned off for some periods of time (i.e., duty cycling), thus not being able to either transmit or receive when in this state and cutting overhearing. Therefore, if nodes can determine when it is worth listening to an upcoming data transmission, then they may go to sleep to save energy when a data transmission is not expected to provide any new information for them.

The inspiring work in [76] proposes to combine NC and duty cycling for more aggressive energy savings in wireless sensor networks. Duty cycling is a technique that increases energy efficiency by allowing a node to turn off part or all of its systems for some periods of time. The focus of this work is on applications such as data dissemination or flooding where, due to the redundancy of coding, there are periods of time when a node does not benefit from overhearing coded data packets being transmitted. The proposed solution, named DutyCode, supports streaming to predict packet arrival and introduces random sleep periods using elastic intervals based on the NC operation. However, DutyCode may lead to wrong predictions that may affect the performance of NC by increasing access delays or sleeping when useful coded data packets are being transmitted.

According to the legacy DCF MAC rules, nodes back off for random periods of time during which they continuously monitor the channel activity when the wireless channel is sensed busy. The inspiring work in [53] proposes that contending nodes shall enter a low-power idle state when another node is transmitting (i.e., during NAV periods) and during subsequent backoff periods in order to save energy. However, the proposed mechanism, named EDA, requires a low-power idle state with a negligible radio transition time into transmitting and receiving states with respect to a packet transmission time, and may degrade throughput and increase access delays.

The basic idea behind EDA has been standardized in the recently-published IEEE 802.11ac amendment under the term of TXOP PSM. This new mechanism allows nodes to enter the sleep state when they listen to data transmissions where they are not involved (i.e., during overhearing TXOPs). More specifically, nodes execute the virtual carrier sense mechanism of the standard DCF by which they update their NAV timers with the duration information contained in overheard control and data packets. This information indicates the time that the wireless channel will be occupied by a TXOP. Then, if the available time for sleeping (i.e., the total data transmission time or TXOP duration) is longer than the duration of the awake/sleep (or on/off) transitions of radio transceivers, overhearing nodes can go to sleep during a TXOP.

Unfortunately, the regular operation of the DCF may not facilitate the TXOP PSM operation. Typically, a TXOP is reserved/granted for the transmission of a single data packet. Therefore, depending on the duration of the TXOP, which depends on the data length and the data transmission rate, and the duration of on/off radio transitions, which may be in the order of hundreds of microseconds [3–5], it may not be possible for a third node to go to sleep during the transaction. Therefore, new strategies need to be investigated to extend the data transmission time being aware of the on/off radio transitions, thus efficiently implementing TXOP PSM in combination with the NC approach.

Recently, the use of RD transmissions has been proposed in the IEEE 802.11 Standard to improve the throughput and energy efficiency of WLANs. More specifically, the RDP has been defined in the IEEE 802.11n as a MAC layer enhancement of the legacy DCF to increase channel utilization. The RDP breaks with the basic operation of the DCF where a node gains a TXOP by competing to get access to the wireless channel in order to transmit data to one arbitrary destination (i.e., unidirectional data flow). In RDP, the holder of a TXOP, once it has seized the channel, can allocate the unused TXOP duration to one or more receivers in order to allow data transmissions in the reverse link (i.e., reverse direction or bidirectional data flow). For scenarios with bidirectional traffic, this approach is very convenient as it reduces contention in the wireless channel.

The concept of reverse direction (or bidirectional) transmission in WLANs was first introduced by [43], prior to the standardization of the RDP. Since then, several works have proposed similar approaches with different purposes. Existing RD-based protocols can be classified into two categories: *(i)* proactive, i.e. RD exchange sequence initiated by the transmitter, or *(ii)* reactive, i.e. RD exchange sequence initiated by the receiver. Proactive RD protocols [46, 78] allow the transmitter to grant the receiver the remaining time of its TXOP for reverse data transfer, in a way similar to RDP. On the other hand, reactive RD protocols [43–45, 47] allow the receiver to reserve the wireless channel for a backward transmission by extending the transmitter's TXOP time, without needing to compete for the channel. This sort of RD protocols can achieve higher performance in some scenarios because they are more adaptive to the actual needs of a network.

In particular, the inspiring work in [47] investigates the feasibility of reactive RD exchange operation in infrastructure WLANs, wherein an AP is connected to a cable network infrastructure and provides wireless Internet access for a number of STAs in its coverage area. Results show that reactive RD approaches can effectively address the unbalanced operation of DCF between uplink and downlink traffic when traffic flows are highly bidirectional. Indeed, DCF provides equal channel access opportunities for all STAs, including the AP. Therefore, the AP only receives an equal share of the wireless channel to deliver downlink traffic to all the STAs, while it has data to transmit to all of them. Note that the case when all STAs route all their traffic through the AP is considered. Thus, by allowing the AP to dynamically initiate RD exchange sequences when receiving data from the STAs, uplink and downlink transmission opportunities can be better balanced, hence improving the overall WLAN performance. Furthermore, the reactive RD operation extends the data transmission time and can be used to allow STAs to efficiently implement the TXOP PSM mechanism taking into account the on/off transitions of radio transceivers.

The previous scenario with the problem of unfairness between downlink and uplink data flows in infrastructure WLANs shows great similarities to the scenario where a relay node needs to forward the received data packets from several source nodes to their respective destinations. Thus, applying the reactive RD transmission method to this scenario may allow congested relay nodes to significantly increase their forwarding capacities. Moreover, if the reactive RD operation is exclusively implemented in those relay nodes with NC capabilities, the overall network performance can be further improved. Note that this kind of relay nodes can provide more information for other nodes in the network than those that forward single data packets.

Furthermore, the fact that bidirectional transmissions extend the total data transmission time due to both forward and reverse transmissions may facilitate the execution of the TXOP PSM operation. As a result, those nodes overhearing bidirectional coded data transmissions where they are not involved can save energy by switching off their radio transceivers during the time that the wireless channel will remain busy.

Motivated by the discussions above, this chapter presents two new NC-aware energy-efficient MAC protocols, named BidCode and GreenCode. BidCode enables reactive RD transmissions between pairs of nodes with a single channel access invocation, in a way similar to BidMAC introduced earlier in Chapter 3 and BDCF proposed in [47]. However, an important difference between BidCode and BidMAC and BDCF is that in BidCode a reactive RD exchange sequence may include multiple rounds of bidirectional coded data transmissions between a pair of sender and receiver or between a relay node and several receivers. Then, GreenCode extends the BidCode operation by exploiting the longer duration of bidirectional coded data transmissions to allow those nodes not involved in the communication to go to sleep in a way similar to GreenBid presented earlier in Chapter 3, TXOP PSM, and EDA [53]. In contrast with EDA, GreenCode can achieve energy saving with longer on/off radio transition times by prolonging the time of data transmissions, and not only improve energy efficiency but also the overall network throughput. In addition, GreenCode allows nodes to determine whether the next transmitted coded packet will be a useful data packet based on control signaling information that is provided right before the transmission of data. Thus, unlike DutyCode [76] GreenCode can achieve energy saving without incurring packet losses and additional access delays.

It is important to mention that, based on the comprehensive assessment of the state of the art, the work presented in this chapter can be considered as the first research work that investigates the idea of combining RD transmissions, opportunistic sleeping periods through TXOP PSM, and NC for high-throughput high-energy-efficient wireless networks based on the IEEE 802.11 Standard.

A preliminary description and performance evaluation of a variation of BidCode for underwater acoustic networks by means of computer-based simulations have been presented in [88]. Then, BidCode has been introduced and evaluated in the Alice and Bob scenario through computer-based simulations for wireless networking applications in [89], where the performance of BidCode has been compared to those of DCF, COPE, and BidMAC. The performance analyses of BidMAC and BidCode in terms of throughput and energy efficiency in both Alice and Bob and cross scenarios have been presented and validated through numerical results in [90], where DCF and COPE have been considered for the purpose of comparison with BidMAC and BidCode. Similarly, a comprehensive performance evaluation of BidMAC and BidCode in terms of energy efficiency via analysis and computer-based simulations have been presented in [91].

The structure of this chapter is detailed as follows.

- Section 5.2 provides an overview of the reference COPE protocol and comprehensively describes the proposed BidCode and GreenCode MAC protocols together with BidMAC and GreenBid both presented in Chapter 3.

- Section 5.3 analyzes the maximum achievable throughputs and energy efficiencies of the protocols under consideration using a simplified approach in both Alice and Bob and cross scenarios and in a generalized scenario.

- Section 5.4 describes the implementation of the protocols in a Python simulation environment and comprehensively evaluates the performances of the protocols in the Alice and Bob and cross scenarios by means of both analytical and simulation results. Important system parameters such as the traffic load, packet length, data rate, wakeup (off-on) radio transition power consumption, and awake/sleep (on/off) radio transitions time have been considered in the evaluation.

- Section 5.5 presents the experimental evaluation of the DCF, COPE, and BidCode protocols using the WARP platform in a proof-of-concept network composed of a relay node and two source nodes (i.e., the Alice and Bob scenario). Experimental results have been obtained in terms of throughput and energy efficiency and have been compared to analytical results considering different values for the traffic load, packet length, and data rate.

- Section 5.6 concludes the chapter by summarizing the key contents of the chapter and highlighting the most relevant results.

5.2 NC-Aware Channel Access Mechanisms

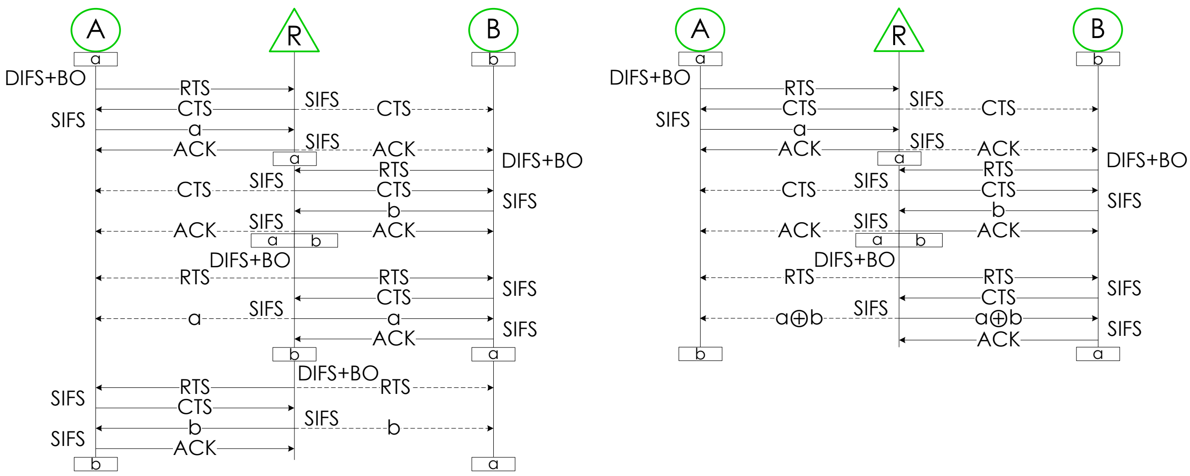
This section overviews the reference COPE (DCF+NC) protocols and provides a detailed description of the proposed BidCode and GreenCode MAC protocols.

5.2.1 The Reference COPE Protocol (DCF+NC)

COPE inserts an NC layer on top of the MAC layer that is responsible for performing linear combinations of several received packets from different flows using XOR operations. In order to allow proper NC operation, COPE introduces a number of modifications in the network stack architecture. First, nodes enable the promiscuous mode to process and store overheard packets for a limited time. Second, nodes opportunistically produce coded packets and send them to one of the intended receivers with an additional header that includes a complete list of the next-hop receivers. Third, upon successful decoding, receiving nodes schedule ACK events that are to be sent together with data packets or periodic control packets. Finally, nodes periodically inform their neighbors about the packets they have stored for coding and decoding opportunities, through reception reports.

The MAC protocol operation of COPE is according to the DCF of the IEEE 802.11. As it has been mentioned earlier in this thesis, this MAC protocol employs the CSMA/CA mechanism in combination with a BEB algorithm and an optional RTS/CTS handshake. Examples of the DCF MAC operation with the RTS/CTS enabled are presented for both the Alice and Bob and cross scenarios in Figs. 5.2a and 5.3a, respectively. In each data transmission, the transmitting node waits for a DIFS and a random backoff period (BO) based on a CW size that starts from CW_{min} after successful transmission and doubles up to CW_{max} after a failed transmission. Then, the transmitting and receiving nodes exchange the RTS and CTS packets, interleaved by a SIFS, before the transmission of data. Upon successful reception of data, the receiving node responds with an ACK packet after a SIFS. Other nodes overhearing the exchange of packets read the duration field of control and data packets and update their NAVs.

The COPE protocol operates as shown in Fig. 5.2b for the Alice and Bob scenario. When node A seizes the channel, it sends packet a to node R using the RTS/CTS mechanism while node B stops the backoff procedure upon overhearing the CTS packet destined to node A. After a DIFS, node B resumes the backoff procedure and obtains a transmission opportunity earlier than node R. After receiving packet b, node R combines packets a and b into packet $a \oplus b$, completes the backoff procedure, and randomly sends packet $a \oplus b$ to node B, which immediately replies with an ACK packet. Both nodes A and B can retrieve packets b and a, respectively, by using their own packets and the received coded packet. Similarly shown for



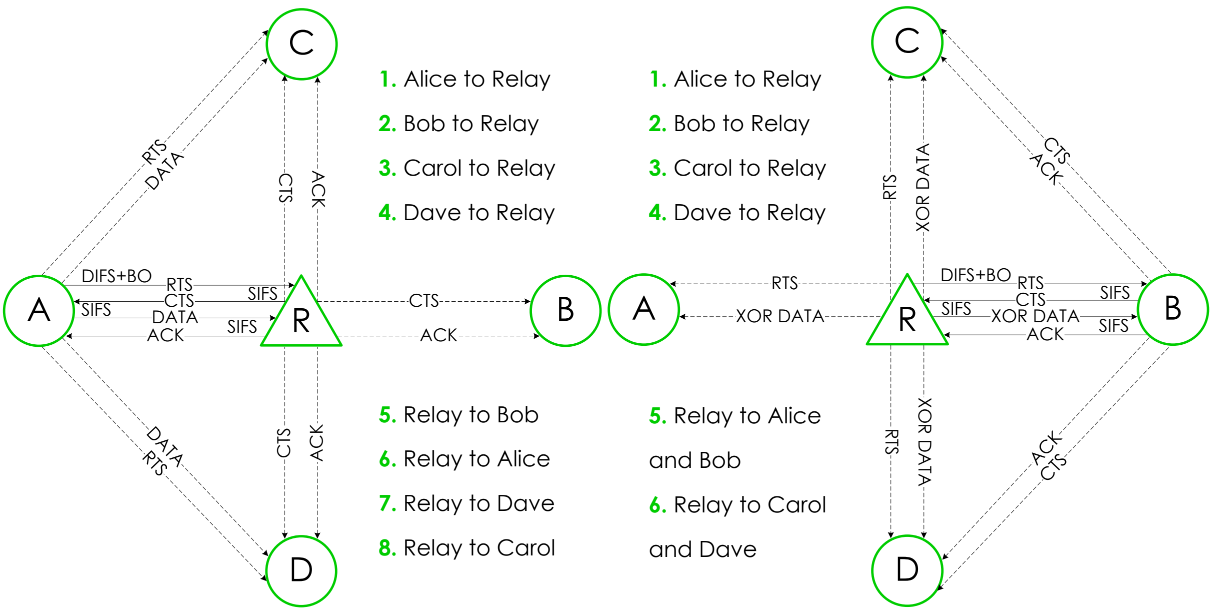
(a) DCF

(b) COPE

Figure 5.2: Examples of operation of the DCF and COPE (DCF+NC) protocols in the Alice and Bob scenario

the cross scenario (see Fig. 5.3b), node R sends a coded packet from nodes A and B to node B. The RTS and coded packets are overheard by nodes A, C, and D whereas the CTS and ACK packets are only overheard by nodes C and D. A similar operation will follow when node R encodes the packets from nodes C and D and transmits the coded packet to one of them. By exploiting NC, node R only needs two transmission slots to forward four packets from nodes A, B, C, and D to their respective destinations.

Fig. 5.4 details the MAC activities and time and energy behaviors of DCF (see Fig. 5.4a) and COPE (see Fig. 5.4b) in the Alice and Bob scenario. It can be seen that nodes R, A, and B consume significant amounts of energy for transmitting and receiving and overhearing packets and listening to DIFS and SIFS intervals and random backoff periods during the exchange of a pair of data packets from end to end.

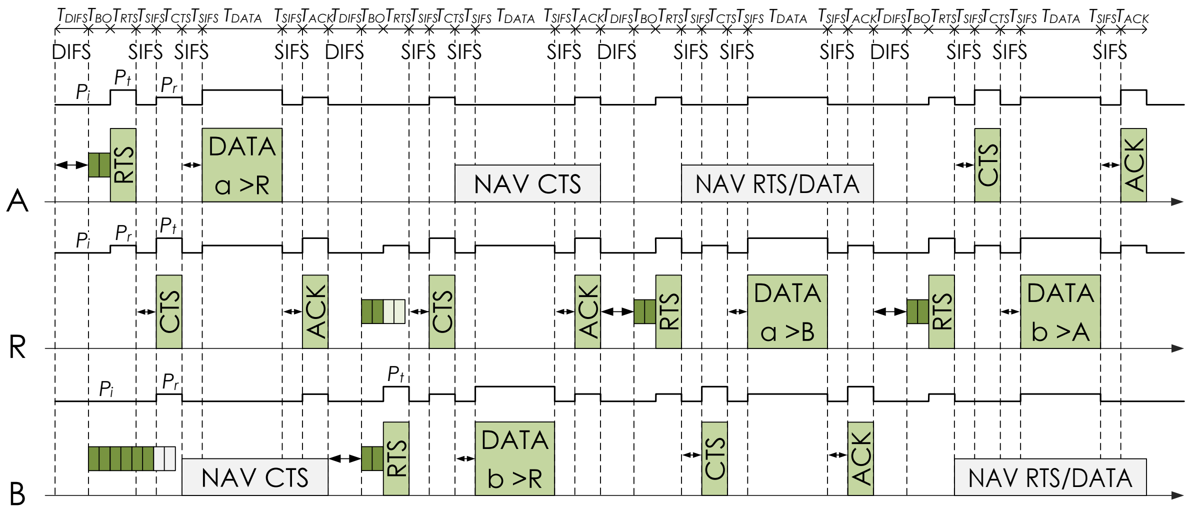


(a) DCF (b) COPE

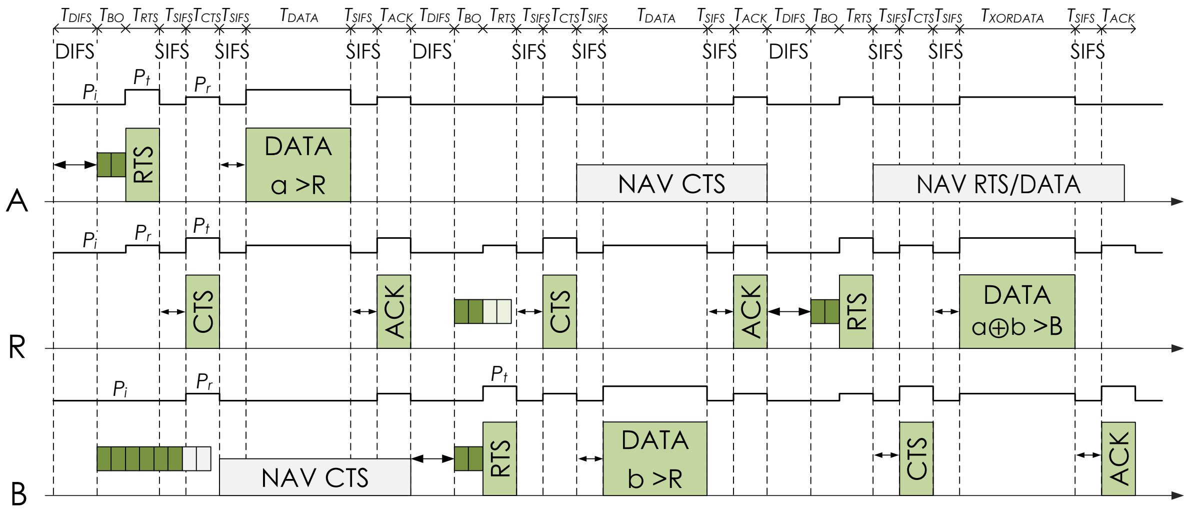
Figure 5.3: Examples of operation of the DCF and COPE (DCF+NC) protocols in the Cross scenario

5.2.2 The New Bidirectional NC-Aware MAC Protocol (BidCode)

BidCode is a new NC-aware MAC protocol that exploits NC as specified in COPE and enables reactive reverse direction transmissions involving network-coded data between wireless nodes as defined in the BidMAC (which has been introduced earlier in this thesis). Figs. 5.5a and 5.6a illustrate the operation of BidMAC in the Alice and Bob and cross scenarios. When node R receives the RTS packet from node A, it replies with a CTS packet whose duration field is updated with the additional time required to enable a transmission in the reverse direction. Thus, node B can update its NAV with the longer duration of the transmission and node R can immediately forward packet a to node B upon receiving it, after a SIFS. Then, node B acknowledges data reception with an explicit ACK packet and node A can interpret the newly received data packet as an implicit

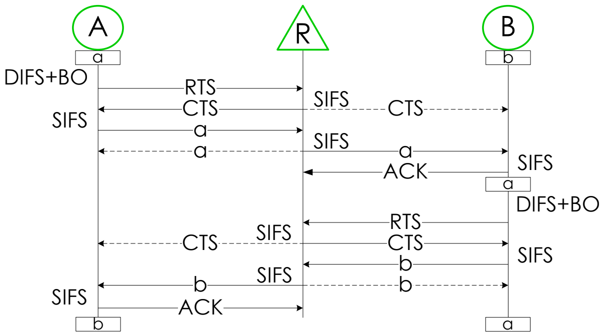


(a) DCF

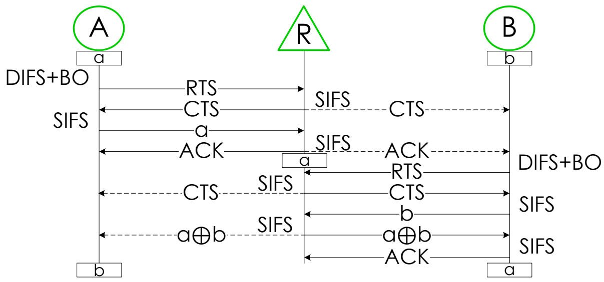


(b) COPE

Figure 5.4: MAC activities and time and energy behaviors of DCF and COPE (DCF+NC) in the Alice and Bob scenario



(a) BidMAC



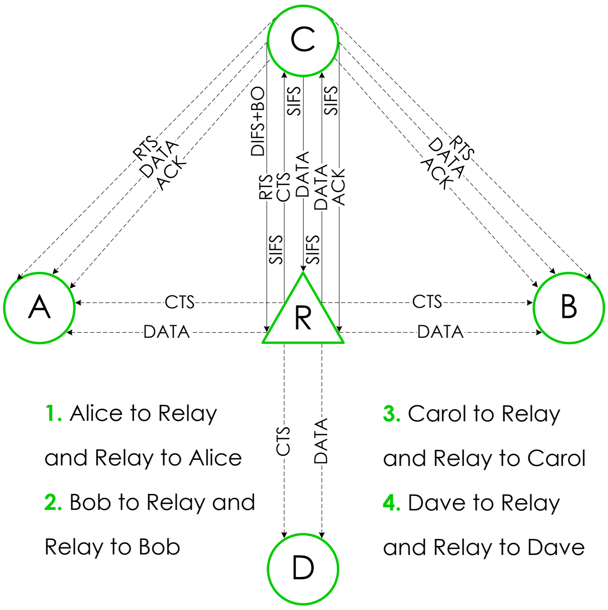
(b) BidCode

Figure 5.5: Examples of operation of the BidMAC and BidCode protocols in the Alice and Bob scenario

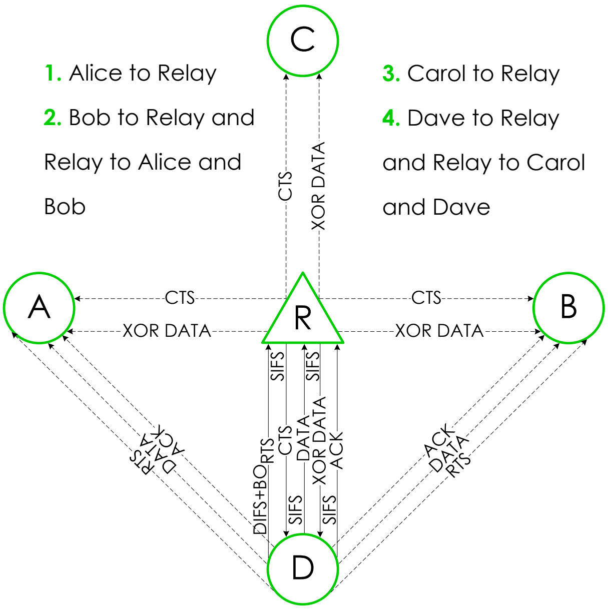
ACK packet for its transmitted data packet. In a similar way, packet b is forwarded from node B to node A.

Following the description of BidMAC, BidCode operates as depicted in Fig. 5.5b for the Alice and Bob scenario. Node A transmits packet a to node R by using standard DCF rules. However, when node R receives the RTS packet from node B, it identifies a coding opportunity with packet a and sends back a CTS packet with the value of the duration field extended to cover the transmission of the possible coded packet. Then, node B sends packet b and node R responds with packet $a \oplus b$. Node B completes the data exchange by sending an ACK packet and both nodes A and B can retrieve the original packets as explained above for COPE. Similarly shown for the cross scenario in Fig. 5.6b, node R can send a coded packet when it receives a data packet from node D. This must precede the transmission of a data packet from node C to node R using DCF, in a way similar to node A in Fig. 5.4a. In this example, nodes A and B overhear the entire communication while node C can only overhear the CTS and coded packets.

To increase coding opportunities, a nonzero time that a relay node can store the received packets before forwarding them without coding is de-



(a) BidMAC

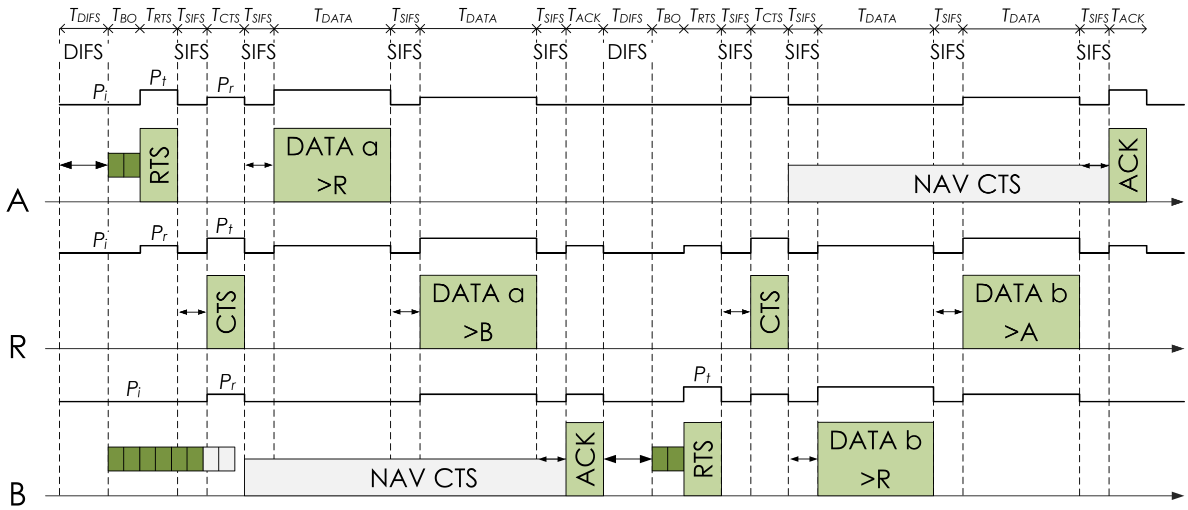


(b) COPE

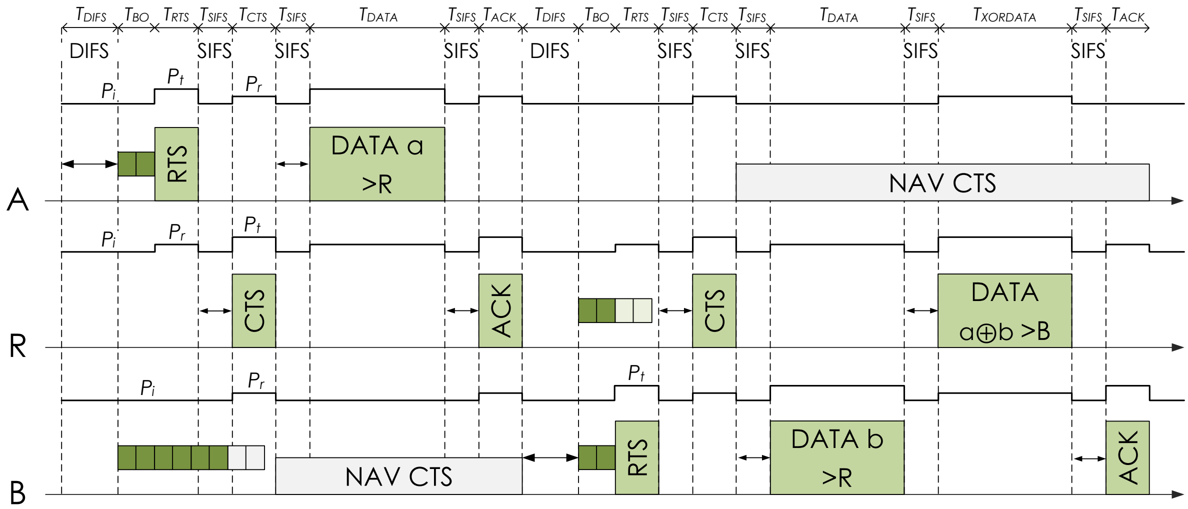
Figure 5.6: Examples of operation of the BidMAC and BidCode protocols in the Cross scenario

fined. This period of time is referred to as holding time, denoted as T_H. In this particular case, the relay node is allowed to send a non-coded packet upon successful reception of a data packet only if the holding time of such packet has expired. Otherwise, it can only reply with the ACK packet and must follow the basic access rules of DCF to transmit non-coded packets.

Fig. 5.7 details the MAC activities and time and energy behaviors of BidMAC (see Fig. 5.7a) and BidCode (see Fig. 5.7b) in the Alice and Bob scenario. It can be seen that nodes R, A, and B consume significant amounts of energy for transmitting and receiving and overhearing packets and listening to DIFS and SIFS intervals and random backoff periods during the exchange of a pair of data packets from end to end.



(a) BidMAC



(b) BidCode

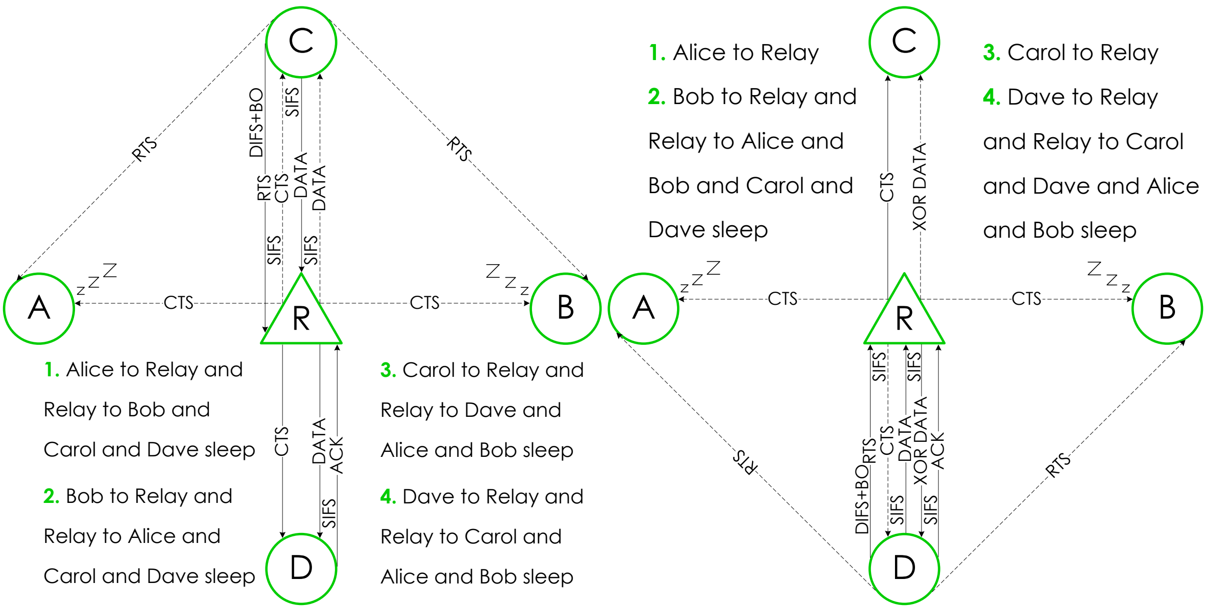
Figure 5.7: MAC activities and time and energy behaviors of BidMAC and BidCode in the Alice and Bob scenario

5.2.3 The New Green Bidirectional NC-Aware MAC Protocol (GreenCode)

GreenCode represents an extension of BidCode to reduce the energy consumed by a node when it listens to a coded data transmission where it is not involved. GreenCode is based on GreenBid, which is an extension of BidMAC and has been introduced earlier in this thesis. GreenBid and GreenCode cannot be used in the Alice and Bob scenario because the hidden source node (e.g, node B when node A is transmitting to node R) needs to be always awake to receive data from the relay node (i.e., node R). Thus, the operations of the protocols are described for the cross topology, where, for example nodes A and B can sleep when nodes C and D are exchanging data through node R.

Fig. 5.8 shows examples of operation of GreenBid and GreenCode, respectively, in the cross topology. The use of the RTS/CTS mechanism is mandatory for the proper operation of the protocols.

In Fig. 5.8a, when node R receives an RTS packet from node C, it identifies an opportunity for bidirectional transmissions. Then, node R responds with a CTS packet whose duration field includes the additional time required to transmit the received data packet to node D. The CTS packet is sent to node D to avoid that it goes to sleep when it overhears the CTS packet, which would be sent to node C by default, and updates its NAV. When node D overhears the CTS packet, it interprets this as a transmission grant from node R because node D is waiting for a CTS packet destined to its address. Meanwhile, nodes A and B read the duration field of the overheard CTS packet, set their NAVs and wakeup timers, and enter the sleep state. The wakeup timers are calculated to allow the transition from the sleep state to the awake state before their NAVs expire. When the bidirectional data transmission among nodes C, R, and D concludes, the sleeping nodes enter the awake state and may contend for the access



(a) GreenBid (b) GreenCode

Figure 5.8: Examples of operation of the GreenBid and GreenCode protocols in the Cross scenario

to the wireless channel after a DIFS. Therefore, the nodes can save energy during channel contention without increasing access delays.

Similarly, in Fig. 5.8b, node R identifies a coding opportunity when it receives an RTS packet from node D. It is assumed that node R has stored a data packet from node C in its transmit queue. Thus, node R sends a CTS packet to node C to force it to stay awake to receive/overhear the potentially coded packet. When node R receives the data packet from node D, it combines the received packet with a stored packet from node C and immediately sends back the coded packet. Node C is able to overhear the coded packet while nodes A and B are sleeping to save energy using a similar procedure described above for GreenBid.

5.3 Theoretical Analysis

This section presents a simple analysis of the maximum theoretical throughputs and energy efficiencies of the proposed NC-aware MAC protocols considering the reference scenarios and assumptions detailed as follows.

5.3.1 Reference Scenarios and Assumptions

Two network scenarios are considered for the analysis of the protocols, namely the Alice and Bob topology and the cross topology, both shown in Fig. 5.1. The Alice and Bob scenario is a chain topology composed of two source nodes that are located at the ends of the chain and communicate with each other through a relay node. The cross scenario is composed of two independent Alice and Bob topologies interconnected through a common relay node. For both scenarios, it is assumed that the relay node does not generate own traffic but only forwards the received data packets from the source nodes to their respective destinations. All nodes are equipped with IEEE 802.11n radio interfaces enabling ad hoc communication mode with a single omnidirectional antenna for communications, i.e., a SISO communications system. Also, the transmission range of each node is assumed to be one hop. This means that each source node is hidden from its source node pair. However, it is assumed that each source node is able to detect the transmission of its source node pair (i.e., it can perform carrier sensing), although it cannot correctly receive the packets transmitted by its source node pair.

Since the aim is to compute the upper bounds of the theoretical throughputs and energy efficiencies of the protocols in idealistic conditions, the following assumptions are made: *(i)* the probability of collision is zero, *(ii)* the wireless channel is ideal, *(iii):* all nodes always have data packets ready to be transmitted in their buffers, *(iv)* no data packets are lost due

Table 5.1: ERP-OFDM PHY Modes and Transmission Times for Control, Data and XOR Data Packets (1500-Byte Payload) in IEEE 802.11n

Mode (m)	Data Rate	N_{DBPS}	T_{RTS}	T_{CTS}	T_{ACK}	T_{DATA}	$T_{XORDATA}$
1	6 Mbps	24	58 μs	50 μs	50 μs	2078 μs	2130 μs
2	9 Mbps	36	50 μs	50 μs	50 μs	1394 μs	1430 μs
3	12 Mbps	48	42 μs	38 μs	38 μs	1054 μs	1078 μs
4	18 Mbps	72	38 μs	38 μs	38 μs	710 μs	730 μs
5	24 Mbps	96	34 μs	34 μs	34 μs	542 μs	554 μs
6	36 Mbps	144	34 μs	34 μs	34 μs	370 μs	378 μs
7	48 Mbps	192	30 μs	34 μs	34 μs	286 μs	290 μs
8	54 Mbps	216	30 μs	34 μs	34 μs	254 μs	262 μs

to buffer overflow, and (v) the propagation delay is neglected. In addition, fragmentation is not used and all data packets have a constant bit length. For the NC-enabled protocols, it is also assumed that coding and decoding XOR operations consume negligible time and energy.

Among the possible configurations of the IEEE 802.11n at the PHY layer, the ERP-OFDM specification for SISO communications has been selected. The ERP-OFDM PHY mode provides 8 transmission modes with different modulation schemes and coding rates. Table 5.1 provides each mode (m) together with its associated transmission data rate and NDBPS (N_{DBPS}).

The expression to compute the time to transmit an XOR coded data packet (i.e., XORDATA) using the ERP-OFDM PHY mode is expressed as

$$\begin{aligned} &T_{XORDATA}(m)= \\ &T_{pre}+T_{sig}+T_{sym}\left\lceil \frac{L_{serv}+8\cdot(L_{MAChdr}+L_{XORhdr}+L_{MSDU}+L_{FCS})+L_{tail}}{N_{DBPS}(m)} \right\rceil \\ &+T_{sigEx}=26+4\cdot\left\lceil \frac{22+8\cdot(74+L_{MSDU})}{N_{DBPS}(m)} \right\rceil \end{aligned} \tag{5.1}$$

where an XOR header of 40 octets (L_{XORhdr}) is added after the MAC header as specified in [18]. All the variables and their values are reported in Table 5.2, where T_{DIFS}, T_{BO}, and T_{EIFS} are calculated by (3.4), (3.5), and (3.6). Note that the EIFS interval and the CW_{max} size are related to the BEB procedure, which is executed when a collision occurs in the wireless channel and retransmission is required. Since for the simplified analysis it is assumed that there are no collisions, these variables are only considered in the simulation part, where collisions are considered. Also, it should be noted that control response packets such as CTS and ACK are transmitted using the basic rates 6, 12, and 24 Mbps, based on the rate selection rules specified in [16]. The transmission times of all packet types for each ERP-OFDM PHY mode are also shown in Table 5.1. Note that T_{RTS}, T_{CTS} and T_{ACK}, and T_{DATA} are computed by (3.2), (3.3), and (3.1), respectively.

The IEEE 802.11n wireless interface of a node can be in one of the following operational states: transmitting, receiving or overhearing (i.e., receiving packets not destined to itself), idle, and sleeping. In the first two states, the radio transceiver is actively used to send and receive information. In the idle state, the wireless interface is ready to receive but no signal is received by the radio transceiver. In the sleep state, the radio transceiver is turned off to save energy. Each of these operational states has associated power consumption. In addition, each transition between states incurs a certain switching time that cannot be neglected. These values will vary depending on the product hardware.

Let P_t, P_r, P_i, and P_s denote the power consumed while transmitting, receiving, idle, and sleeping, respectively. When an idle STA identifies an opportunity to sleep, a transition from idle to sleep takes place. Similarly, a transition from sleep to idle occurs when the STA decides to wake up. Based on [3–5], the transition time from idle to sleep ($T_{i \to s}$) is shown to

Table 5.2: System Parameters

Parameter	Definition	Value
T_{slot}	Slot Time	9 μs
T_{SIFS}	SIFS Interval	10 μs
T_{DIFS}	DIFS Interval	28 μs
T_{EIFS}	EIFS Interval	88 μs
CW_{min}	Minimum Contention Window	15
CW_{max}	Maximum Contention Window	1023
T_{BO}	Average Backoff Time	67.5 μs
T_{pre}	Preamble Time	16 μs
T_{sig}	Signal Time	4 μs
T_{sym}	OFDM symbol Period	4 μs
T_{sigEx}	Signal Extension Period	6 μs
L_{serv}	Service Bits	16 bits
L_{tail}	Tail Bits	6 bits
L_{RTS}	Length of RTS	20 bytes
L_{CTS}	Length of CTS	14 bytes
L_{ACK}	Length of ACK	14 bytes
L_{MAChdr}	MAC Header	30 bytes
L_{XORhdr}	XOR Header	40 bytes
L_{FCS}	Frame Check Sequence	4 bytes
T_H	Holding Time	10 ms
$T_{i\to s}$	Transition Time from Idle to Sleep	250 μs
$T_{s\to i}$	Transition Time from Sleep to Idle	250 μs
P_t	Transmission Power Consumption	1.65 W
P_r	Reception Power Consumption	1.4 W
P_i	Idle Power Consumption	1.15 W
P_s	Sleep Power Consumption	0.045 W
$P_{i\to s}$	Idle to Sleep Transition Power Consumption	0.045 W
$P_{s\to i}$	Sleep to Idle Transition Power Consumption	1.725 W

be similar to the transition time from sleep to idle ($T_{s\to i}$). Hence, it is assumed that $T_{i\to s}$ is equal to $T_{s\to i}$. Regarding the power consumed during these transitions, the works in [3–5] show that the power consumed from idle to sleep ($P_{i\to s}$) is substantially lower than P_s. In contrast, the power consumed from sleep to idle ($P_{s\to i}$) is shown to be significantly higher than P_i. Thus, it is assumed that $P_{i\to s}$ is equal to P_s and $P_{s\to i}$ is modeled as αP_i, where α is defined as the transition coefficient between sleep and idle states, or wakeup transition coefficient, and $\alpha > 1$. Fig. 3.6 illustrates this explanation and Table 5.2 records the variables mentioned above and their values (most of them taken from [3–5]).

5.3.2 Alice and Bob Scenario

The throughputs and energy efficiencies of the protocols under consideration are analyzed in the Alice and Bob scenario as follows.

A. Throughput

The throughput of a given protocol x (S_x) is defined as the amount of information contained in an MSDU (L_{MSDU}) divided by the time ratio (T_x) required to transmit the data packet that includes the MSDU. This is expressed as

$$S_x[\text{Mbps}]=\frac{8{\cdot}L_{MSDU}}{T_x} \tag{5.2}$$

where (T_x) is defined as the amount of time spent in transmission over the total amount of transmitted data packets.

The transmission time ratio of each protocol under consideration is described and formulated as follows.

1) DCF:

The transmission delay of DCF consists of a DIFS interval, a backoff period, an RTS transmissions, a SIFS interval, a CTS transmission, a SIFS interval, a DATA transmission, a SIFS interval, and an ACK transmissions. The maximum throughput of DCF is achieved when Alice and Bob transmit a data packet and the relay node is able to transmit the two data packets to their respective destinations. In total, 4 transmissions are necessary to deliver 2 data packets from end to end. Given the transmission delay of DCF, the minimum transmission time ratio that results in the maximum network throughput of DCF is expressed as

$$T_{DCF}^{net\_min}=\frac{1}{2}\cdot 4\left(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK}+3\cdot T_{SIFS}\right) \quad (5.3)$$

However, when the network enters the saturation state, the relay node can only send a single data packet due to the fairness of DCF. Therefore, the transmission time ratio that corresponds to the saturation network throughput is given as

$$T_{DCF}^{net\_sat}=3\left(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK}+3\cdot T_{SIFS}\right) \quad (5.4)$$

2) COPE (DCF+NC):

The transmission delay of COPE contains the same as that of DCF but there is a transmission of a coded packet from the relay node once every two data transmissions from Alice and Bob. In total, 3 transmissions are required to send two data packets from end to end. As a result, the transmission time ratio that describes the saturation network throughput of COPE is written as

$$\begin{aligned} T_{COPE}^{net\_sat}=&\frac{1}{2}\cdot 3\left(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+T_{ACK}+3\cdot T_{SIFS}\right) \\ &+\frac{1}{2}\left(2\cdot T_{DATA}+T_{XORDATA}\right) \end{aligned} \quad (5.5)$$

3) BidMAC:

The transmission delay of BidMAC is similar to that of DCF but it includes an additional data transmission and a SIFS interval whenever the relay node receives a data packet from either Alice or Bob. Hence, only two transmission slots are required to deliver two data packets from end to end. Therefore, the transmission time ratio that leads to the saturation network throughput of BidMAC is computed as

$$T^{net\_sat}_{BidMAC}=\frac{1}{2}\cdot 2\left(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+2\cdot T_{DATA}+T_{ACK}+4\cdot T_{SIFS}\right) \tag{5.6}$$

4) BidCode:

The transmission delay of BidCode combines the transmission delays of DCF and BidMAC. There is one data transmission from either Alice or Bob to the relay node through a DCF transmission slot. Once every two data transmissions from Alice and Bob, there is a bidirectional transmission similar to BidMAC but there is a coded data transmission from the relay node instead of a single data transmission. Hence, the transmission time ratio that results in the saturation network throughput of BidCode is expressed as

$$\begin{aligned} T^{net\_sat}_{BidCode}=&\frac{1}{2}\cdot 2\left(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK}\right)\\ &+\frac{1}{2}\left(T_{XORDATA}+7\cdot T_{SIFS}\right) \end{aligned} \tag{5.7}$$

B. Energy Efficiency

The energy efficiency of a given protocol x (η_x) is defined as the amount of bits contained in an MSDU divided by the energy consumption ratio (E_x) required to transmit the data packet that includes the MSDU:

$$\eta_x[\text{Mb/J}]=\frac{8\cdot L_{MSDU}}{E_x} \tag{5.8}$$

where L_{MSDU} denotes the byte-length of an MSDU and E_x is defined as the product of power consumed and time spent in transmission over the total amount of transmitted data packets and is split into three energy consumption components, namely, transmitting (E_t), receiving and overhearing (E_r), and idle (E_i).

The energy consumption ratio of each protocol under consideration is described and formulated as follows.

1) DCF:

During the transmission cycle of DCF, the transmitting node, either Alice, Bob, or the relay node, consumes energy to transmit the RTS packet and the data packet and to receive the CTS packet and the ACK packet from the receiver. On the other hand, the receiving node consumes energy to receive the RTS packet and the data packet from the transmitting node and to respond with the CTS packet and the ACK packet. Meanwhile, the node not involved in the communication (or the hidden node), either Alice or Bob, consumes energy to overhear the exchange of packets. When the transmitting node is a source node, the hidden node consumes energy to receive the CTS and ACK packets from the receiving node and to be idle during the RTS and data transmissions. Otherwise, when the transmitting node is the relay node, the hidden node consumes energy to receive the RTS and data packets from the transmitting node and to be idle during the CTS and ACK transmissions. In addition, the three nodes consume energy to listen to the wireless channel for a DIFS interval, a backoff period, and all SIFS periods.

The energy efficiency of DCF shows a maximum value and a lower stable value under saturation. The maximum value is obtained when Alice and Bob transmit a data packet to the relay node and the relay node transmits the two data packets to their respective destinations. There are 4 transmissions in which the relay node transmits and receives twice, the 2 source nodes transmit and receive twice, the hidden node is idle twice, and the 3 nodes are idle 4 times. Given the energy consumed during the transmission cycle of DCF from end to end, the minimum energy consumption ratio that corresponds to the maximum network energy efficiency of DCF is given as

$$\begin{aligned} E_{DCF}^{net\_min} &= \frac{1}{2}\left(E_t + E_r + E_i\right) \\ E_t &= 4\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) P_t \\ E_r &= 6\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) P_r \\ E_i &= \left(12\left(T_{DIFS} + T_{BO} + 3 \cdot T_{SIFS}\right) + 2\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right)\right) P_i \end{aligned} \tag{5.9}$$

However, when the network enters the saturation state, the relay node can only perform one data transmission when Alice and Bob transmit, due to the fairness of DCF. There are 3 transmissions in which the 2 source nodes transmit twice, the relay node receives twice, a source node receives once, the hidden node is idle twice, and the 3 nodes are idle 3 times. Therefore, the energy consumption ratio that leads to the saturation network energy efficiency of DCF is computed as

$$\begin{aligned} E_{DCF}^{net\_sat} &= E_t + E_r + E_i \\ E_t &= 3\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) P_t \\ E_r &= \left(4\left(T_{RTS} + T_{DATA}\right) + 5\left(T_{CTS} + T_{ACK}\right)\right) P_r \\ E_i &= \left(9\left(T_{DIFS} + T_{BO} + 3 \cdot T_{SIFS}\right) + 2\left(T_{RTS} + T_{DATA}\right) + T_{CTS} + T_{ACK}\right) P_i \end{aligned} \tag{5.10}$$

2) COPE:

In the transmission cycle of COPE, the amounts of energy consumed by Alice, Bob, and the relay node are the same as those shown for the transmission cycle of DCF when Alice and Bob transmit. However, when the relay node gets a transmission opportunity, it consumes energy to transmit a coded packet while Alice and Bob consume energy to receive it. There are 3 transmissions in which the 2 source nodes transmit once, the relay node transmits once and receives twice, a source node receives once, the hidden node is idle twice, and the 3 nodes are idle 3 times. As a result, the energy consumption ratio that describes the saturation network energy efficiency of COPE is written as

$$
\begin{aligned}
E_{COPE}^{net\_sat} = & \frac{1}{2}\left(E_t + E_r + E_i\right) \\
& E_t = \left(3\left(T_{RTS} + T_{CTS} + T_{ACK}\right) + 2 \cdot T_{DATA} + T_{XORDATA}\right) P_t \\
& E_r = \left(4 \cdot T_{RTS} + 2\left(T_{DATA} + T_{XORDATA}\right) + 5\left(T_{CTS} + T_{ACK}\right)\right) P_r \\
& E_i = \left(9\left(T_{DIFS} + T_{BO} + 3 \cdot T_{SIFS}\right) + 2\left(T_{RTS} + T_{DATA}\right) + T_{CTS} + T_{ACK}\right) P_i
\end{aligned} \tag{5.11}
$$

3) BidMAC:

In the data exchange through BidMAC, the energy consumed by Alice, Bob, and the relay node is similar to that of DCF. However, the receiving node, i.e. the relay node, consumes energy to transmit a data packet and not an ACK packet and to receive an ACK packet from the hidden node. On the contrary, the hidden node consumes energy to receive the data packet and to send back the ACK packet. Otherwise, the transmitting node consumes energy for overhearing the data packet and for being idle during the ACK transmission. In addition, all nodes are idle for an additional SIFS interval. There are 2 bidirectional transmissions in which the source nodes transmit and receive twice, the relay node transmits and receives twice, and the 3 nodes are idle twice. Therefore, the energy consumption ratio that leads to the saturation network energy efficiency of BidMAC is given as

$$
\begin{aligned}
E_{BidMAC}^{net\_sat} = & \frac{1}{2}\left(E_t + E_r + E_i\right) \\
& E_t = 2\left(T_{RTS} + T_{CTS} + 2 \cdot T_{DATA} + T_{ACK}\right) P_t \\
& E_r = 2\left(T_{RTS} + 2 \cdot T_{CTS} + 3 \cdot T_{DATA} + T_{ACK}\right) P_r \\
& E_i = \left(6\left(T_{DIFS} + T_{BO} + 4 \cdot T_{SIFS}\right) + 2\left(T_{RTS} + T_{DATA} + T_{ACK}\right)\right) P_i
\end{aligned} \tag{5.12}
$$

4) BidCode:

The energy consumed in BidCode combines the energy consumption in a data transmission through DCF and a bidirectional transmission with a coded packet similar to BidMAC. There are 2 transmission slots in which the 2 source nodes transmit twice, the relay node transmits once and receives twice, a source node receives once, the hidden node is idle twice, and the 3 nodes are idle twice. Hence, the energy consumption ratio that corresponds to the saturation network energy efficiency is expressed as

$$\begin{aligned} E_{BidCode}^{net\_sat} &= \frac{1}{2}\left(E_t + E_r + E_i\right) \\ E_t &= \left(2\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) + T_{XORDATA}\right) P_t \\ E_r &= \left(2\left(T_{RTS} + 2 \cdot T_{CTS} + T_{DATA} + T_{XORDATA}\right) + 3 \cdot T_{ACK}\right) P_r \\ E_i &= \left(3\left(2\left(T_{DIFS} + T_{BO}\right) + 7 \cdot T_{SIFS}\right) + 2\left(T_{RTS} + T_{DATA}\right) + T_{ACK}\right) P_i \end{aligned} \tag{5.13}$$

5.3.3 Cross Scenario

The throughputs and energy efficiencies of the protocols under evaluation are now analyzed for the cross scenario.

A. Throughput

The throughput definition is given by (5.2) and the transmission time ratio of each protocol is described and formulated as follows.

1) DCF:

The maximum throughput of DCF is achieved when the four source nodes transmit once and the relay node is able to forward the four received data packets to their respective destinations. In total, 8 transmissions are required to send 4 data packets from end to end. Thus, the minimum transmission time ratio that results in the maximum network throughput of DCF is computed as

$$T_{DCF}^{net\_min} = \frac{1}{4} \cdot 8 \left(T_{DIFS} + T_{BO} + T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK} + 3 \cdot T_{SIFS} \right) \quad (5.14)$$

On the other hand, under saturation the four source nodes and the relay node get a transmission opportunity, due to the DCF fairness. Hence, there are 5 transmissions and only one data packet is forwarded to its destination. As a result, the transmission time ratio that corresponds to the saturation network throughput of DCF is given as

$$T_{DCF}^{net\_sat} = 5 \left(T_{DIFS} + T_{BO} + T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK} + 3 \cdot T_{SIFS} \right) \quad (5.15)$$

2) COPE (DCF+NC):

The maximum throughput of COPE is achieved when the four source nodes transmit once and the relay node transmits two coded data packets, one for each bidirectional flow. In total, 6 transmissions are required to send 4 data packets from end to end. Therefore, the minimum transmission time ratio that corresponds to the maximum network throughput of COPE is represented as

$$\begin{aligned} T_{COPE}^{net\_min} = & \frac{1}{4} \cdot 6 \left(T_{DIFS} + T_{BO} + T_{RTS} + T_{CTS} + T_{ACK} + 3 \cdot T_{SIFS} \right) \\ & + \frac{1}{4} \left(4 \cdot T_{DATA} + 2 \cdot T_{XORDATA} \right) \end{aligned} \quad (5.16)$$

However, when the network enters the saturation state, the relay node can only get a transmission opportunity to send a coded packet. In total, there are 5 transmissions and 2 data packets can be delivered from end to end. Hence, the transmission time ratio that produces the saturation network throughput of COPE is formulated as

$$\begin{aligned} T_{COPE}^{net\_sat} = & \frac{1}{2} \cdot 5 \left(T_{DIFS} + T_{BO} + T_{RTS} + T_{CTS} + T_{ACK} + 3 \cdot T_{SIFS} \right) \\ & + \frac{1}{2} \left(4 \cdot T_{DATA} + T_{XORDATA} \right) \end{aligned} \quad (5.17)$$

3) BidMAC:

The maximum throughput of BidMAC is achieved when each source node transmits to the relay node and the relay node immediately sends the received data packet to its destination. In total, there are 4 bidirectional transmissions and 4 data packets are sent from end to end. Therefore, the transmission time ratio that describes the saturation network throughput of BidMAC is calculated as

$$T_{BidMAC}^{net\_sat}=\frac{1}{4}\cdot 4\left(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+2\cdot T_{DATA}+T_{ACK}+4\cdot T_{SIFS}\right) \tag{5.18}$$

4) BidCode:

The maximum throughput of BidCode is achieved when the four source nodes transmit and the relay node uses the reception of two data packets from different bidirectional flows to transmit two coded packets. In total, there are 2 unidirectional data transmissions and 2 bidirectional coded data transmissions in order to forward 4 data packets from end to end. As a result, the transmission time ratio that leads to the saturation network throughput of BidCode is expressed as

$$\begin{aligned} T_{BidCode}^{net\_sat}=&\frac{1}{4}\cdot 4\left(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK}\right)\\ &+\frac{1}{4}\left(2\cdot T_{XORDATA}+14\cdot T_{SIFS}\right) \end{aligned} \tag{5.19}$$

5) GreenBid – 6) GreenCode:

Note that the throughput values of GreenBid and GreenCode are the same as those of BidMAC and BidCode, respectively, because GreenBid and GreenCode have been designed to improve energy efficiency.

B. Energy Efficiency

Based on the definition of energy efficiency (5.8), the energy consumption ratio of each protocol is described and formulated as follows.

1) DCF:

The maximum energy efficiency of DCF is calculated as follows. There are 4 transmissions in which each source node transmits once and the rest overhear expect one, the hidden node, which is idle and the relay node receives four times. Then, there are 4 additional transmissions in which the relay node transmits four times and each source node receives once and the rest overhear. In total, there are 8 transmissions and 4 data packets delivered from end to end. Therefore, the minimum energy consumption ratio that produces the maximum network throughput energy efficiency of DCF is computed as

$$
\begin{aligned}
E_{DCF}^{net\_min} &= \frac{1}{4}\left(E_t + E_r + E_i\right)\\
E_t &= 8\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) P_t\\
E_r &= 28\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) P_r\\
E_i &= \left(40\left(T_{DIFS} + T_{BO} + 3\cdot T_{SIFS}\right) + 4\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right)\right) P_i
\end{aligned}
\tag{5.20}
$$

On the contrary, under saturation each source node gets a transmission opportunity, and the relay node as well. There are 4 transmissions in which each source node transmits once while the rest overhear except the hidden node. Then, there is an additional transmission where the relay node transmits and one source node receives and the rest overhear. As a result, the energy consumption ratio that leads to the saturation network throughput of DCF is given as

$$
\begin{aligned}
E_{DCF}^{net\_sat} &= E_t + E_r + E_i\\
E_t &= 5\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) P_t\\
E_r &= \left(16\left(T_{RTS} + T_{DATA}\right) + 19\left(T_{CTS} + T_{ACK}\right)\right) P_r\\
E_i &= \left(25\left(T_{DIFS} + T_{BO} + 3\cdot T_{SIFS}\right) + 4\left(T_{RTS} + T_{DATA}\right) + T_{CTS} + T_{ACK}\right) P_i
\end{aligned}
\tag{5.21}
$$

2) COPE:

In COPE, the maximum energy efficiency is obtained when each source node transmits to the relay node and the relay node transmits two coded data packets. There are 4 transmissions in which each source node transmits while the rest overhear except the hidden node, which is idle, and the relay node receives. In addition, there are 2 coded data transmissions in which the relay node transmits, one source node receives and the rest overhear. In total, there are 6 transmissions and 4 data packets delivered from end to end. Thus, the minimum energy consumption ratio that results in the maximum network throughput of COPE is obtained as

$$\begin{aligned} E_{COPE}^{net\_min} &= \frac{1}{4}\left(E_t + E_r + E_i\right) \\ E_t &= \left(6\left(T_{RTS} + T_{CTS} + T_{ACK}\right) + 4 \cdot T_{DATA} + 2 \cdot T_{XORDATA}\right) P_t \\ E_r &= \left(20 \cdot T_{RTS} + 12 \cdot T_{DATA} + 8 \cdot T_{XORDATA} + 22\left(T_{CTS} + T_{ACK}\right)\right) P_r \\ E_i &= \left(30\left(T_{DIFS} + T_{BO} + 3 \cdot T_{SIFS}\right) + 4\left(T_{RTS} + T_{DATA}\right)\right) P_i \\ &\quad + 2\left(T_{CTS} + T_{ACK}\right) P_i \end{aligned} \tag{5.22}$$

However, when the network enters the saturation state, each source node transmits to the relay node and the relay node transmits one coded data packet. In total, there are 5 transmissions and 2 data packets delivered from end to end. The energy consumption ratio that corresponds to the saturation network energy efficiency is written as

$$\begin{aligned} E_{COPE}^{net\_sat} &= \frac{1}{2}\left(E_t + E_r + E_i\right) \\ E_t &= \left(5\left(T_{RTS} + T_{CTS} + T_{ACK}\right) + 4 \cdot T_{DATA} + T_{XORDATA}\right) P_t \\ E_r &= \left(16 \cdot T_{RTS} + 12 T_{DATA} + 4 T_{XORDATA} + 19\left(T_{CTS} + T_{ACK}\right)\right) P_r \\ E_i &= \left(25\left(T_{DIFS} + T_{BO} + 3 \cdot T_{SIFS}\right) + 4\left(T_{RTS} + T_{DATA}\right) + T_{CTS} + T_{ACK}\right) P_i \end{aligned} \tag{5.23}$$

3) BidMAC:

In BidMAC, each source node transmits and the relay node forwards the received packet to its destination. In total, there are 4 bidirectional transmissions and 4 data packets delivered from end to end. When a source node transmits, the relay node receives, three source nodes overhear, and the hidden node is idle. When the relay node transmits, a source node receives and the rest overhear. Hence, the energy consumption ratio that produces the saturation network energy efficiency of BidMAC is computed as

$$\begin{aligned} E_{BidMAC}^{net\_sat} &= \frac{1}{4}\left(E_t + E_r + E_i\right) \\ E_t &= 4\left(T_{RTS} + T_{CTS} + 2 \cdot T_{DATA} + T_{ACK}\right) P_t \\ E_r &= \left(12 \cdot T_{RTS} + 16 \cdot T_{CTS} + 28 \cdot T_{DATA} + 12 \cdot T_{ACK}\right) P_r \\ E_i &= \left(20\left(T_{DIFS} + T_{BO} + 4 \cdot T_{SIFS}\right) + 4\left(T_{RTS} + T_{DATA} + T_{ACK}\right)\right) P_i \end{aligned} \tag{5.24}$$

4) BidCode:

In BidCode, each source node transmits once and the relay node transmits two coded data packets. In total, there are 2 unidirectional transmissions and 2 bidirectional transmissions and 2 data packets delivered from end to end. Therefore, the energy consumption ratio that results in the saturation network throughput of BidCode is given as

$$\begin{aligned} E_{BidCode}^{net\_sat} &= \frac{1}{4}\left(E_t + E_r + E_i\right) \\ E_t &= \left(4\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) + 2 \cdot T_{XORDATA}\right) P_t \\ E_r &= \left(12\left(T_{RTS} + T_{DATA}\right) + 16 \cdot T_{CTS} + 8 \cdot T_{XORDATA} + 14 \cdot T_{ACK}\right) P_r \\ E_i &= \left(20\left(T_{DIFS} + T_{BO}\right) + 70 \cdot T_{SIFS} + 4\left(T_{RTS} + T_{DATA}\right) + 2 \cdot T_{ACK}\right) P_i \end{aligned} \tag{5.25}$$

5) GreenBid:

The energy consumption of GreenBid introduces two new energy consumption components, namely, switching between idle and sleep states

(E_{sw}) and sleeping (E_s). The energy consumed by the nodes in a data transmission when using GreenBid is described as follows.

- **Transmission period:** the transmitting source node consumes energy to send an RTS packet and a data packet to the relay node. The relay node consumes energy to send a CTS packet and the received data packet from the transmitting source node to the hidden source node. The hidden source node consumes energy to transmit the ACK packet.

- **Reception period:** the transmitting source node consumes energy to overhear the CTS and data packets addressed to the hidden source node. The relay node consumes energy to receive the RTS and data packets from the transmitting node and the ACK packet from the hidden source node. The hidden node consumes energy to receive the CTS and data packets. The rest of nodes (two) only consume energy to overhear the RTS and CTS packets as they can switch to the sleep state to save energy.

- **Idle period:** In the idle period, the source nodes and the relay node consume energy to listen to the wireless channel during a DIFS interval, a backoff period, and a SIFS interval. After that, only the transmitting source node, the relay node, and the hidden source node are awake for the remaining SIFS intervals.

- **Switch period:** the two sleeping nodes consume energy during the transition from idle to sleep and during the transition from sleep to idle.

- **Sleep period:** the two overhearing nodes can sleep during the data exchange expect for when they have to switch between idle and sleep states. This happens provided that the sleep period (T_s) is greater than zero. Otherwise, none of the overhearing nodes can sleep and the energy consumed by GreenBid is the same as for BidMAC. The sleep period (T_s) is given by

$$T_s{=}2{\cdot}T_{DATA}{+}T_{ACK}{+}3{\cdot}T_{SIFS}{-}\left(T_{i\to s}{+}T_{s\to i}\right) \tag{5.26}$$

Based on the above, 4 channel access are required in GreenBid to exchange 4 data packets from end to end. This means that there are 4 RTS, CTS, and ACK transmissions and 8 data transmissions. 4 RTS are transmitted by the source nodes and are received by 2 source nodes and the relay node while one source node (i.e., the hidden node) is idle during the RTS transmissions. 4 CTS packets are transmitted by the relay node and are received by the 4 source nodes. 4 data packets are transmitted by the source nodes and are received only by the relay node, because 2 source nodes are sleeping, while the hidden source node is idle during the data transmissions. 4 data packets are transmitted by the relay node and are received by the intended source nodes while the respective hidden source nodes also overhear the data transmissions. 4 ACK packets are transmitted by the receiving source nodes and are received by the relay node while the respective hidden source node are idle during the ACK transmissions. In the 4 transmissions, the 5 nodes listen to the DIFS interval, the backoff period, and a SIFS interval, and then only 2 source nodes and the relay node listen to the remaining 3 SIFS intervals. Also, there are 2 source nodes that switch to sleep during the 4 transmissions. Thus, the energy consumption ratio that produces the saturation network energy efficiency of GreenBid is computed as

$$
\begin{aligned}
E_{GreenBid}^{net\_sat} &= \frac{1}{4}\left(E_t + E_r + E_i + E_{sw} + E_s\right)\\
E_t &= 4\left(T_{RTS} + T_{CTS} + 2 \cdot T_{DATA} + T_{ACK}\right) P_t\\
E_r &= \left(12\left(T_{RTS} + T_{DATA}\right) + 16 \cdot T_{CTS} + 4 \cdot T_{ACK}\right) P_r\\
E_i &= \left(20\left(T_{DIFS} + T_{BO}\right) + 56 \cdot T_{SIFS} + 4\left(T_{RTS} + T_{DATA} + T_{ACK}\right)\right) P_i\\
E_{sw} &= 8\left(T_{i \to s} P_{i \to s} + T_{s \to i} P_{s \to i}\right)\\
E_s &= 8 T_s P_s
\end{aligned}
\tag{5.27}
$$

6) GreenCode:

The energy consumption of GreenCode is similar to that of GreenBid except that two source nodes transmit two data packets in unidirectional mode and then two source nodes transmit two data packets in bidirectional mode where the relay node transmits two XOR coded packets. This means that overhearing nodes can only go to sleep when there are bidirectional XOR coded data transmissions. Thus, the sleep period is recalculated as

$$T_s' = T_{DATA} + T_{XORDATA} + T_{ACK} + 3 \cdot T_{SIFS} - \left(T_{i \to s} + T_{s \to i}\right) \tag{5.28}$$

First, there are two normal data transmissions where 2 RTS and data transmissions are performed by the source nodes and are received by 2 source nodes and the relay node while the hidden node is idle during the RTS and data transmissions. Then, 2 CTS and 2 ACK transmissions are performed by the relay node and are received by the 4 source nodes. In the 2 transmissions, the 5 nodes listen to the DIFS interval, the backoff period, and the 3 SIFS intervals.

Then, there are two bidirectional XOR coded data transmissions where 2 RTS transmissions are performed by the source nodes and are received by 2 source nodes and the relay node while the hidden node is idle during the RTS transmissions. After that, 2 CTS transmissions are performed by the relay node and received by the 4 source nodes. 2 data transmissions are performed by the source nodes and are received only by the relay node because there are 2 source nodes sleeping while the hidden node is idle during the data transmission. Then, 2 XOR coded data transmissions are performed by the relay node and are received by 2 source nodes, which respond with 2 ACK packets that are received by the relay node while the hidden node is idle during the ACK transmissions. In the 2 bidirectional XOR coded data transmissions the 5 nodes listen to a DIFS interval, the backoff period, and a SIFS interval and then only 2 nodes and the relay node listen to the remaining 3 SIFS intervals. Also, there are 2 source nodes that switch to sleep during the 2 bidirectional XOR coded data transmissions. Therefore, the energy consumption ratio that describes the saturation network energy efficiency of GreenCode is formulated as

$$
\begin{aligned}
E_{GreenCode}^{net\_sat} &= \frac{1}{4}\left(E_t + E_r + E_i + E_{sw} + E_s\right) \\
E_t &= \left(4\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) + 2 \cdot T_{XORDATA}\right) P_t \\
E_r &= \left(12 \cdot T_{RTS} + 16 \cdot T_{CTS} + 8 \cdot T_{DATA} + 4 \cdot T_{XORDATA} + 10 \cdot T_{ACK}\right) P_r \\
E_i &= \left(20\left(T_{DIFS} + T_{BO}\right) + 58 \cdot T_{SIFS} + 4\left(T_{RTS} + T_{DATA}\right) + 2 \cdot T_{ACK}\right) P_i \\
E_{sw} &= 4\left(T_{i \to s} P_{i \to s} + T_{s \to i} P_{s \to i}\right) \\
E_s &= 4 T_s' P_s
\end{aligned}
\tag{5.29}
$$

5.3.4 Generalized Scenario

The analysis of the protocols in the Alice and Bob and cross scenarios allows the derivation of general expressions of the throughput and energy efficiency for a given number of source nodes (N). This can be very useful to analyze the performance of the protocols in bigger scenarios, such as the chain topology and the wheel topology.

A. Throughput

Given the throughput definition in (5.2), the transmission time ratio of each protocol is described and formulated as follows.

1) DCF:

To compute the maximum throughput of DCF, the source nodes should send N data packets to the relay node, which should forward them to their respective destinations. In total, $2N$ transmission slots are required to forward N data packets from end to end. Thus, the minimum transmission time ratio that corresponds the maximum network throughput of DCF is given as

$$T_{DCF}^{net\_min} = \frac{2N}{N} \left(T_{DIFS} + T_{BO} + T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK} + 3 \cdot T_{SIFS} \right) \quad (5.30)$$

However, the saturation throughput of DCF will be lower than the maximum throughput. Due to the long-term fairness of DCF, the relay node will only get a transmission opportunity once every N transmissions from the source nodes. Hence, the transmission time ratio that describes the saturation network throughput of DCF is expressed a

$$T_{DCF}^{net\_sat} = (N+1) \left(T_{DIFS} + T_{BO} + T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK} + 3 \cdot T_{SIFS} \right) \quad (5.31)$$

2) COPE (DCF+NC):

To compute the maximum throughput of COPE, the relay node should forward $\frac{N}{2}$ XOR coded packets for every N received data packets from the source nodes. As a result, N data transmissions and $\frac{N}{2}$ XOR coded data transmissions are required to forward N data packets from end to end. Therefore, the minimum transmission time ratio that leads to the maximum network throughput of COPE is computed as

$$T_{COPE}^{net\_min}=\frac{1}{N}\left(N+\frac{N}{2}\right)(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+T_{ACK}+3\cdot T_{SIFS})$$
$$+\frac{1}{N}\left(NT_{DATA}+\frac{N}{2}T_{XORDATA}\right) \tag{5.32}$$

However, under saturation, the relay node will only be able to send a single coded packet every N transmissions from the source nodes, due to the DCF fairness. Thus, two data packets will be delivered form end to end. The transmission time ratio that results in the saturation network throughput of COPE is written as

$$T_{COPE}^{net\_sat}=\frac{1}{2}(N+1)(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+T_{ACK}+3\cdot T_{SIFS})$$
$$+\frac{1}{2}(NT_{DATA}+T_{XORDATA}) \tag{5.33}$$

3) BidMAC:

To compute the maximum throughput of BidMAC, the relay node should forward N data packets from end to end every N transmissions from the source nodes. Therefore, the transmission time ratio that produces the saturation network throughput of BidMAC is formulated as

$$T_{BidMAC}^{net\_sat}=\frac{N}{N}(T_{DIFS}+T_{BO}+T_{RTS}+T_{CTS}+2\cdot T_{DATA}+T_{ACK}+4\cdot T_{SIFS}) \tag{5.34}$$

4) BidCode:

To compute the maximum throughput of BidCode, the source nodes should perform $\frac{N}{2}$ data transmissions in unidirectional mode and $\frac{N}{2}$ data transmissions in bidirectional mode where the relay node should perform $\frac{N}{2}$ bidirectional XOR coded data transmissions. Each unidirectional transmission sequence consists of a DIFS interval, a backoff period, the RTS, CTS, data, and ACK transmissions, and 3 SIFS intervals. Each bidirectional transmission sequence involves a DIFS interval, a backoff period, the

RTS, CTS, data, XOR coded data, and ACK transmissions, and 4 SIFS intervals. Hence, N transmissions slots are required to exchange N data packets from end to end. The transmission time ratio that corresponds to the saturation network throughput of BidCode is obtained as

$$\begin{aligned} T_{BidCode}^{net\_sat} = & \frac{N}{N}\left(T_{DIFS} + T_{BO} + T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) \\ & + \frac{N}{2N}\left(T_{XORDATA} + 7 \cdot T_{SIFS}\right) \end{aligned} \tag{5.35}$$

B. Energy Efficiency

Based on the definition of the energy efficiency in (5.8), the energy consumption ratio of each protocol is described and formulated as follows.

1) DCF:

The energy consumption of DCF during a data transmission is calculated as follows. The transmitting node, either a source node or the relay node, consumes energy to transmit the RTS and data packets and to receive the CTS and ACK packets from the receiving node. On the other hand, the receiving node consumes energy to receive the RTS and data packets from the transmitting node and to respond with the CTS and ACK packets. The $N-1$ source nodes not involved in transmission consume energy to overhear the exchange of packets except one that can only overhear the packets sent from the relay node. The N source nodes and the relay node (i.e., $N+1$) also consume energy to listen to the wireless channel for a DIFS interval, a backoff period, and all SIFS intervals. In addition, one source node is idle when one of the source nodes is transmitting to the relay node.

Based on the above, the maximum network energy efficiency of DCF is achieved when $2N$ transmissions are performed by the AP and the source nodes and N data packets are exchanged from end to end. N transmissions are performed by the source nodes and are received by $N-2$ source nodes (the hidden source nodes is idle) and the relay node (i.e., $N-1$ nodes).

Then, N transmissions are performed by the relay node and are received by N source nodes. In the $2N$ transmissions, $N+1$ nodes listen to a DIFS interval, average backoff period, and 3 SIFS intervals and the hidden nodes listen to N transmissions. Therefore, the transmission time ratio that produces the maximum energy efficiency of DCF is written as

$$\begin{aligned} E_{DCF}^{net\_min} &= \frac{1}{N}\left(E_t+E_r+E_i\right) \\ E_t &= 2N\left(T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK}\right)P_t \\ E_r &= N\left(2N-1\right)\left(T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK}\right)P_r \\ E_i &= 2N\left(N+1\right)\left(T_{DIFS}+T_{BO}+3\cdot T_{SIFS}\right)P_i \\ &\quad +N\left(T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK}\right)P_i \end{aligned} \tag{5.36}$$

In contrast, the saturation network energy efficiency of DCF is computed when $N+1$ transmissions are performed, N by the source nodes and one by the relay node, and so a single data packet is exchanged from end to end. Given the energy consumption of DCF in a transmission, the energy consumption ratio the corresponds to the saturation network energy efficiency of DCF is obtained as

$$\begin{aligned} E_{DCF}^{net\_sat} &= E_t+E_r+E_i \\ E_t &= \left(N+1\right)\left(T_{RTS}+T_{CTS}+T_{DATA}+T_{ACK}\right)P_t \\ E_r &= \big(\left(N\left(N-1\right)+N\right)\left(T_{RTS}+T_{DATA}\right)+\left(N^2+N-1\right)\left(T_{CTS}+T_{ACK}\right)\big)P_r \\ E_i &= \Big(\left(N+1\right)^2\left(T_{DIFS}+T_{BO}+3\cdot T_{SIFS}\right)+N\left(T_{RTS}+T_{DATA}\right)\Big)P_i \\ &\quad +\left(T_{CTS}+T_{ACK}\right)P_i \end{aligned} \tag{5.37}$$

2) COPE:

The energy consumption of COPE during an XOR coded data transmission is similar to that of DCF. However, the relay node consumes energy to transmit an XOR coded packet and the N source nodes consume energy to receive it. Given the energy consumption of COPE, the maximum energy efficiency of COPE is achieved when the source nodes perform N data transmissions and the relay node performs $\frac{N}{2}$ XOR coded data transmissions, hence exchanging N data packets from end to end. N RTS and data transmissions performed by the source nodes are received by $N{-}1$ nodes and N CTS and ACK transmissions performed by the relay node are received by N nodes. Then, $\frac{N}{2}$ RTS and XOR coded data transmissions performed by the relay node are received by N source nodes and $\frac{N}{2}$ CTS and ACK transmissions performed by the source nodes are received by $N{-}1$ nodes. During the N data transmissions and $\frac{N}{2}$ XOR coded data transmissions, the $N{+}1$ nodes listen to a DIFS interval, an average backoff period, and 3 SIFS intervals and the hidden source nodes listen to N RTS and data transmissions and $\frac{N}{2}$ CTS and ACK transmissions. Thus, the minimum energy consumption ratio that produces the maximum energy efficiency of COPE is given as

$$
\begin{aligned}
E_{COPE}^{net\_min} &= \frac{1}{N}\left(E_t+E_r+E_i\right)\\
E_t &= \left(\left(N+\frac{N}{2}\right)\left(T_{RTS}+T_{CTS}+T_{ACK}\right)+NT_{DATA}+\frac{N}{2}T_{XORDATA}\right)P_t\\
E_r &= \left(\left(N\left(N-1\right)+\frac{N^2}{2}\right)T_{RTS}+N\left(N-1\right)T_{DATA}+\frac{N^2}{2}T_{XORDATA}\right)P_r\\
&\quad+\left(N^2+\frac{N}{2}\left(N-1\right)\right)\left(T_{CTS}+T_{ACK}\right)P_r\\
E_i &= \left(N+1\right)\left(N+\frac{N}{2}\right)\left(T_{DIFS}+T_{BO}+3\cdot T_{SIFS}\right)P_i\\
&\quad+\left(N\left(T_{RTS}+T_{DATA}\right)+\frac{N}{2}\left(T_{CTS}+T_{ACK}\right)\right)P_i
\end{aligned}
\tag{5.38}
$$

On the contrary, the saturation network energy efficiency is computed when $N{+}1$ transmissions are performed, N data transmissions by the source nodes and one XOR coded data transmissions by the relay node, and so 2 data packets can be exchanged from end to end. Given the energy consumption of DCF in a data transmission and that of COPE in an XOR coded data transmission, the energy consumption ratio the corresponds to the saturation network energy efficiency of COPE is obtained as

$$\begin{aligned}
E_{COPE}^{net_sat} &= \frac{1}{2}\left(E_t + E_r + E_i\right)\\
E_t &= \left(\left(N+1\right)\left(T_{RTS}+T_{CTS}+T_{ACK}\right)+NT_{DATA}+T_{XORDATA}\right)P_t\\
E_r &= \left(\left(N\left(N-1\right)+N\right)T_{RTS}+N\left(N-1\right)T_{DATA}+NT_{XORDATA}\right)P_r\\
&\quad+\left(N^2+N-1\right)\left(T_{CTS}+T_{ACK}\right)P_r\\
E_i &= \Big(\left(N+1\right)^2\left(T_{DIFS}+T_{BO}+3\cdot T_{SIFS}\right)+N\left(T_{RTS}+T_{DATA}\right)\Big)P_i\\
&\quad+\left(T_{CTS}+T_{ACK}\right)P_i
\end{aligned} \tag{5.39}$$

3) BidMAC:

The energy consumption of BidMAC during a bidirectional transmission is calculated as follows. The transmitting source node consumes energy to perform the RTS and data transmissions that are received by the relay node and overheard by $N-2$ source node while the hidden source node is idle. Then the relay node consumes energy to perform the CTS and data transmissions that are received by the hidden source node of the transmitting source node and are overheard by $N-1$ source nodes. The hidden source node consumes energy to perform an ACK transmission that is received by the relay node and overheard by $N-2$ source nodes while the transmitting source node is idle. During the transmission, the $N+1$ nodes consume energy to listen to a DIFS interval, an average backoff period, and 4 SIFS intervals and the hidden source nodes consume energy to listen to the RTS, data, and ACK transmissions.

Based on the computation of the energy consumption of BidMAC, N bidirectional transmissions are performed to exchange N data packets from end to end. As a result, the energy consumption ratio that leads to the saturation network energy efficiency of BidMAC is written as

$$
\begin{aligned}
E_{BidMAC}^{net\_sat} = & \frac{1}{N}\left(E_t+E_r+E_i\right) \\
& E_t = N\left(T_{RTS}+T_{CTS}+2\cdot T_{DATA}+T_{ACK}\right)P_t \\
& E_r = \big(N\left(N-1\right)\left(T_{RTS}+T_{ACK}\right)+N^2T_{CTS}+N\left(N+N-1\right)T_{DATA}\big)P_r \\
& E_i = N\left(N+1\right)\left(T_{DIFS}+T_{BO}+4\cdot T_{SIFS}\right)P_i \\
& \quad +N\left(T_{RTS}+T_{DATA}+T_{ACK}\right)P_i
\end{aligned}
\tag{5.40}
$$

4) BidCode:

The energy consumption of BidCode during a unidirectional transmission is the same as that of DCF whereas the energy consumption of BidCode during a bidirectional transmission is similar to that of BidMAC. In a bidirectional transmission of BidCode, while the forward transmission from a source node to the relay node contains a normal data packet, the reverse transmission from the relay node to a source node includes an XOR coded data packet. This is different from a BidMAC bidirectional transmission where the reverse transmission also contains a normal data packet. To transmit N data packets from end to end, $\frac{N}{2}$ unidirectional data transmissions and $\frac{N}{2}$ bidirectional XOR coded data transmissions are required in BidCode. As a result, the energy consumption ratio that results in the saturation network energy efficiency of BidCode is formulated as

$$
\begin{aligned}
E_{BidCode}^{net\_sat} &= \frac{1}{N}\left(E_t + E_r + E_i\right) \\
E_t &= \left(N\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) + \frac{N}{2} T_{XORDATA}\right) P_t \\
E_r &= \left(N\left(N-1\right)\left(T_{RTS} + T_{DATA}\right) + N^2 T_{CTS} + \frac{N^2}{2} T_{XORDATA}\right) P_r \\
&\quad + \left(\frac{N^2}{2} + \frac{N}{2}\left(N-1\right)\right) T_{ACK} P_r \\
E_i &= \left(N+1\right)\left(N\left(T_{DIFS} + T_{BO}\right) + \frac{N}{2} 7 \cdot T_{SIFS}\right) P_i \\
&\quad + \left(N\left(T_{RTS} + T_{DATA}\right) + \frac{N}{2} T_{ACK}\right) P_i
\end{aligned}
\tag{5.41}
$$

5) GreenBid:

The energy consumption of GreenBid during a bidirectional transmissions is similar to that of BidMAC but there are the following changes.

- **Transmission period:** the transmitting source node consumes energy to transmit an RTS packet and a data packet to the relay node whereas the relay node consumes energy to transmit a CTS packet and a data packet to the hidden source node of the transmitting source node (i.e., the final destination), which consumes energy to transmit an ACK packet to the relay node.

- **Reception period:** $N-S$ overhearing source nodes only consume energy to overhear the RTS and CTS packets because they can switch to the sleep state to save energy. S denotes the number of active nodes, which is 2. Then, the RTS and data transmissions by the transmitting source node are received by the relay node and the CTS and data transmissions by the relay node are received by the hidden source node and overheard by the transmitting node. Finally, the ACK transmission by the hidden source is received by the relay node.

- **Idle period:** the N+1 nodes consume energy to listen to a DIFS interval, an average backoff period, and a SIFS interval. Then, S+1 nodes consume energy to listen to the remaining 3 SIFS intervals.
- **Switch period:** the $N-S$ overhearing source nodes consume energy during the transition from idle to sleep and during the transition from sleep to idle.
- **Sleep period:** the $N-S$ overhearing source nodes can sleep during the data exchange expect for when they have to switch between idle and sleep states.

Given the energy consumption of GreenBid during a bidirectional transmissions, N bidirectional transmissions are needed to exchange N data packets from end to end. Hence, the energy consumption ratio that corresponds to the saturation network energy efficiency of GreenBid is obtained as

$$\begin{aligned}
E_{GreenBid}^{net_sat} = & \frac{1}{N}\left(E_t + E_r + E_i + E_{sw} + E_s\right) \\
E_t = & N\left(T_{RTS} + T_{CTS} + 2 \cdot T_{DATA} + T_{ACK}\right) P_t \\
E_r = & \left(N\left(N-1\right) T_{RTS} + N^2 T_{CTS} + N\left(S+1\right) T_{DATA} + N T_{ACK}\right) P_r \\
E_i = & \left(N\left(N+1\right)\left(T_{DIFS} + T_{BO}\right) + N\left(N+1+3\left(S+1\right)\right) T_{SIFS}\right) P_i \\
& + N\left(T_{RTS} + T_{DATA} + T_{ACK}\right) P_i \\
E_{sw} = & N\left(N-S\right)\left(T_{i\to s} P_{i\to s} + T_{s\to i} P_{s\to i}\right) \\
E_s = & N\left(N-S\right) T_s P_s
\end{aligned} \tag{5.42}$$

6) GreenCode:

The energy consumption of GreenCode during a unidirectional transmission is the same as that of DCF whereas the energy consumption of GreenCode during a bidirectional transmission is similar to that of GreenBid. In a bidirectional transmission of GreenCode, while the forward transmission from a source node to the relay node contains a normal data packet, the reverse transmission from the relay node to a source node includes an XOR coded data packet, as specified in BidCode. This is different from a GreenBid bidirectional transmission, which is based on BidMAC, where the reverse transmission also contains a normal data packet. To transmit N data packets from end to end, $\frac{N}{2}$ unidirectional data transmissions and $\frac{N}{2}$ bidirectional XOR coded data transmissions are required in GreenCode. This means that during $\frac{N}{2}$ bidirectional XOR coded data transmissions $N-S$ overhearing source node can sleep, hence saving energy. Therefore, the energy consumption ratio that corresponds to the saturation network energy efficiency of GreenCode is computed as

$$\begin{aligned}
E_{GreenCode}^{net_sat} &= \frac{1}{N}\left(E_t + E_r + E_i + E_{sw} + E_s\right) \\
E_t &= \left(N\left(T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}\right) + \frac{N}{2} T_{XORDATA}\right) P_t \\
E_r &= \left(N\left(N-1\right) T_{RTS} + N^2 T_{CTS} + \frac{N}{2}\left(N-1+1\right) T_{DATA}\right) P_r \\
&+ \left(\frac{N}{2} S T_{XORDATA} + \frac{N}{2}\left(N+1\right) T_{ACK}\right) P_r \\
E_i &= \left(N\left(N+1\right)\left(T_{DIFS} + T_{BO}\right) + \frac{N}{2}\left(4\left(N+1\right) + 3\left(S+1\right)\right) T_{SIFS}\right) P_i \\
&+ \left(N\left(T_{RTS} + T_{DATA}\right) + \frac{N}{2} T_{ACK}\right) P_i \\
E_{sw} &= \frac{N}{2}\left(N-S\right)\left(T_{i\to s} P_{i\to s} + T_{s\to i} P_{s\to i}\right) \\
E_s &= \frac{N}{2}\left(N-S\right) T_s' P_s
\end{aligned} \tag{5.43}$$

5.4 Simulations Framework

This section evaluates the performances of the protocols by means of both analytical and simulation results in the Alice and Bob and cross scenarios. The expressions derived in the previous section are used to discuss the upper-bound performances of the different protocols. In addition, an event-driven simulator coded in Python has been developed for the model validation, where the protocol rules have been implemented.

5.4.1 NC-Aware MAC Protocols Simulation

The simulation scenario consists of a relay node and a finite number of source nodes in its coverage area. Basically, the Alice and Bob and cross scenarios are simulated. The nodes are static. All the nodes are within the transmission range of each other except one that is hidden from each source node. However, it is assumed that, although it cannot properly receive data, the hidden node is at least able to perform carrier sensing when one source node gets a transmission opportunity. The source nodes generate data packets of constant length with their arrivals following a Poisson distribution. The relay node does not generate own traffic but only forwards the packets from the source nodes to their respective destinations. Infinite packet queues are assumed to avoid packet losses due to buffer overflow. All packets are received with no errors.

The simulator is composed of six main scripts according to the protocols under evaluation, i.e. DCF, COPE, BidMAC, BidCode, GreenBid and GreenCode:

- "DCFMACsimulator.py": This script refers to the IEEE 802.11 DCF MAC protocol.
- "COPEsimulator.py": This script is related to the COPE protocol.
- "BidMACsimulator.py": This script deals with the BidMAC protocol.
- "BidCodesimulator.py": This script deals with the BidCode protocol.
- "GreenBidsimulator.py": This script deals with the GreenBid protocol.

- "GreenCodesimulator.py": This script deals with the GreenCode protocol.

Each of these scripts contains the input parameters required to run the simulation of each protocol. These input parameters can be the simulation time, the number of simulation runs, the number of nodes among other parameters included in Table 5.1 and Table 5.2. These main scripts are also used to collect the obtained results in an Excel file.

Each main script calls an associated class that can be:

- "dcfmac.py": This class contains the DCF MAC rules.
- "bidmac.py": This class includes the BidMAC rules.
- "bidcode.py": This class includes the BidCode rules.
- "greenbid.py": This class runs the GreenBid rules.
- "greencode.py": This class runs the GreenCode rules.

These classes are connected with three subclasses:

- "node.py": This subclass describes a source node. It contains attributes like the state of a node, if it has packets, the packet box, the output packet queue, and several timers like the slot timer, the DIFS timer, and the backoff timer.
- "relay.py": This subclass describes the relay node. It contains the same attributes as those in node.py but it also includes the holding timer and the transmit queue of coded packets for the NC-aware protocols.
- "packet.py": This subclass describes a packet. It contains attributes like the arrival time, the departure time, the transmission delay, and the destination.

- "simreport.py": This subclass collects all the output values of the simulation, such as throughput, energy efficiency, delay, and energy consumption.

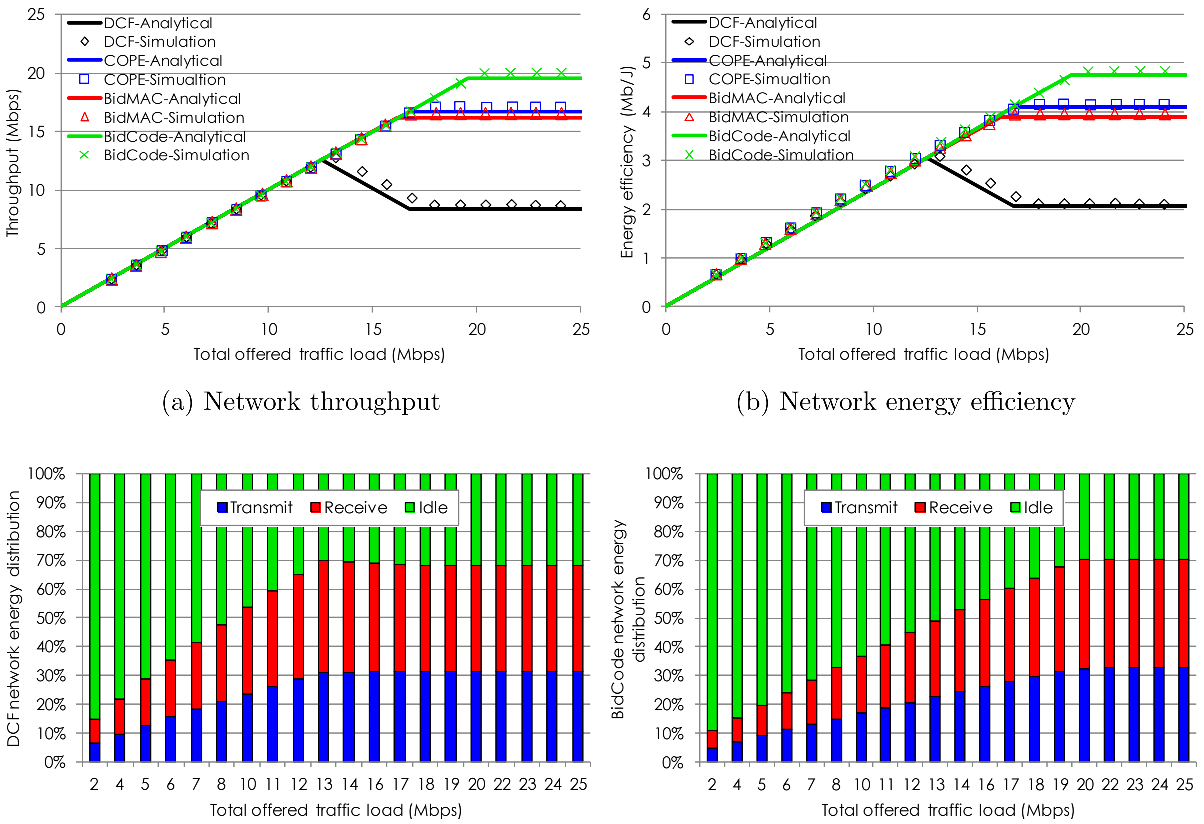
In each of these classes, the rules of each protocol are implemented. First, all the input parameters passed from the main script are registered. Then, the code enters the main function called Run. In the Run function, the relay node and the source nodes are created and the network is set up by defining the neighborhood of each node. Each source node is appended to a list of nodes. A box of packets for each source node is generated according to a Poisson-distributed arrival process and considering the available simulation time. After that, the code enters a loop that is running until the simulation time is reached.

Inside the loop, what happens in each microsecond of the simulation is verified. A transmit list that includes the potential transmitting nodes in a given time is created. When this list is empty, it means nothing happens in a given microsecond. In each idle microsecond, the states of the relay node and each source node are verified and also if the relay node or any source node gets data packets to transmit. When this happens, the data packet is removed from the packet box and inserted in the transmit queue. At this time, the relay node or any source node executes the protocol rules to transmit the data packet. When the relay node or a source node has data packets to send, it can be in one of the following states: waiting for a DIFS (state 1), running the backoff procedure (state 2), transmitting (state 3), freezing the backoff counter (state 4), just performing virtual carrier sensing (state 5), and waiting for the holding time to expire (state 6). When the wireless channel is idle for a DIFS or the backoff counter reaches zero, the relay node or a source node is included in the transmit list. When the length of the transmit list is one, this means that there is only one transmitter and so a successful transmission occurs. When the length of the transmit list is longer than one, this means that there are several transmitters and so a collision occurs.

When there is a successful transmission, it is checked if the transmitter is the relay node or a source node. Then, several variables are updated and reinitialized and the transmitted data packet from the output queue of the transmitter are removed. It is checked if the transmitter gets a new data packet while it is transmitting and also if the relay node or a source node has got new data packets during the transmission. Depending on the current state of the relay node and each source node, its state value is updated according to the protocol rules. If the relay node is the receiver of a data packet, it is verified if the newly received data packet can be coded with other data packets with the queue. If so, a new coded packet is appended to the coded packet queue and the two native data packets are removed from the non-coded packet queue. When there is a collision, a similar procedure is followed, except that each colliding node doubles its CW size and randomly selects a new backoff counter. When the simulation run is over, the simreport subclass is called to collect all the simulation results and return it to the main script.

5.4.2 Analytical and Simulation Results

The analytical and simulation results of the protocols are shown for both the Alice and Bob and cross scenarios in terms of throughput, energy efficiency, and energy distributions, considering different values for the traffic load, MSDU length and PHY data rate. In addition, for the cross scenario the energy efficiencies and time and energy distributions of GreenBid and GreenCode are evaluated considering the wakeup transition coefficient (α) and awake/sleep transition time. All simulation runs were repeated 10 times for the duration of 20 s each. The simulation results in the plots are obtained with a 95% confidence interval lower than 0.02.



(a) Network throughput

(b) Network energy efficiency

(c) DCF: network energy distribution

(d) BidCode: network energy distribution

Figure 5.9: Throughput, energy efficiency, and energy consumption of the NC-aware MAC protocols versus the traffic load in the Alice and Bob topology

Alice and Bob Scenario

The results of the protocols in the Alice and Bob scenario are presented and discussed as follows.

1) Traffic Load:

The throughput, energy efficiency, and energy distributions of the protocols versus the traffic load in the Alice and Bob scenario are plotted in Fig. 5.9. The results are obtained for an MSDU length of 1500 bytes and a PHY data rate of 54 Mbps.

Fig. 5.9a shows the throughput from end to end. Likewise, the energy efficiency is reported in Fig. 5.9b. In general, the performance of the protocols increases linearly as the traffic load from Alice and Bob increases, since the relay node needs to forward more packets. The performance of DCF reaches a maximum value and then decreases until a stable value under saturation. The maximum value corresponds to 1/2 of the traffic load from Alice (1/4) and Bob (1/4). Since the relay node needs to forward twice as many packets as Alice and Bob, it will use half of the channel accesses. Otherwise, the saturation value corresponds to 2/3 of the traffic load from Alice (1/3) and Bob (1/3). Due to the DCF fairness, when Alice and Bob attempt to transmit at a higher rate, the relay node is unable to increase its capacity and can only get 1/3 of the channel. When NC is enabled, COPE allows the relay node to send twice as fast as Alice and Bob, although it will still get 1/3 of the channel. The relay node is able to send two packets in a single transmission and will be able to increase its capacity as Alice and Bob do. The maximum throughput of COPE will be around 2/3 of the channel throughput due to the additional overhead required for coding. Similarly, BidMAC can almost achieve the throughput of COPE because the relay node is able to send a packet when it receives a packet from either Alice or Bob. However, the relay node will have to transmit twice as many packets as Alice and Bob and so the nodes will consume higher amounts of energy. In contrast, BidCode achieves the highest performance as it allows the relay node to send a coded packet as soon as it receives a data packet from either Alice or Bob.

The contribution of each operational state to the overall energy consumptions of the DCF and BidCode protocols as the traffic load increases are studied n Fig. 5.9c and Fig. 5.9d. In general, when the traffic load is low, most of energy (up to 90%) is consumed for being idle since the nodes are inactive most of the time. As the traffic load increases, the energy consumed for transmitting and receiving packets increases significantly, which

Table 5.3: Alice and Bob: Maximum Gains vs. Traffic Load

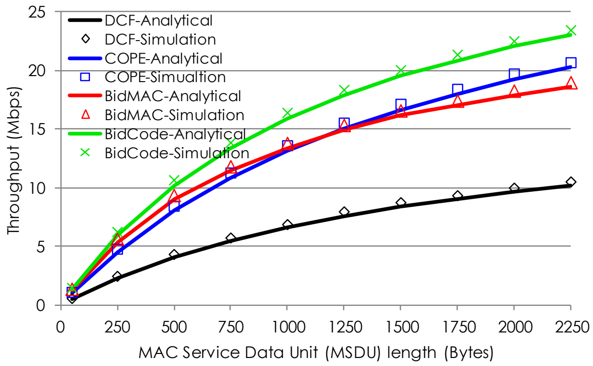
Saturation Traffic Load	BidCode vs. DCF	BidCode vs. COPE
Network throughput	1.33	0.17
Network energy efficiency	1.31	0.17

implies a reduction of the energy consumed during idle periods down to 30%. It can be seen that in DCF the energy dedicated to transmitting and receiving increases faster when compared to BidCode. Under saturation, the two protocols show similar results. The largest amount of the energy resources (up to 40%) is dedicated to receiving and overhearing activities.

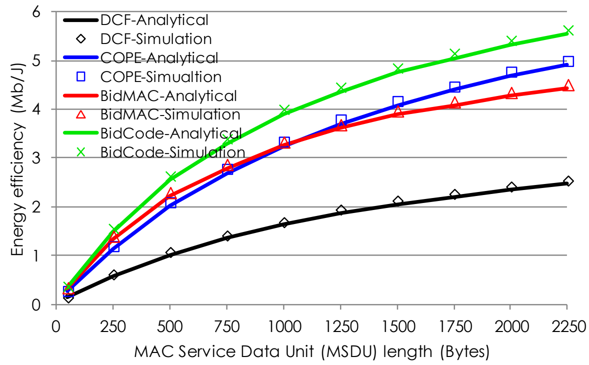
Table 5.3 records the maximum gains of BidCode in the Alice and Bob scenario. The maximum gains of BidCode versus DCF are 1.33 and 1.31 in terms of throughput and energy efficiency, respectively. In addition, the maximum gains of BidCode versus COPE are 0.17 and 0.16 in terms of throughput and energy efficiency, respectively.

2) MSDU Length:

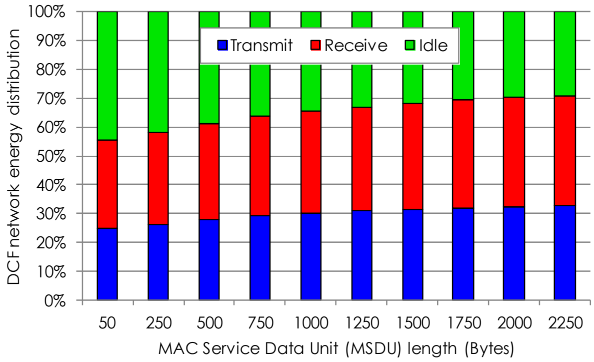
The throughputs, energy efficiencies, and energy consumptions of the protocols versus the MSDU length are reported in Fig. 5.10. The results are plotted for a PHY data rate of 54 Mbps. Fig. 5.10a shows the throughput from end to end whereas Fig. 5.10b shows the energy efficiency. In general, the performance of the protocols increases as the data payload increases since more information is transmitted. BidCode achieves the highest performance whereas DCF achieves the lowest performance. The performance of BidMAC is higher than the performance of COPE for low MSDU lengths. The critical MSDU length that makes COPE and BidMAC perform the same is 1250 bytes. For MSDU lengths above this value, COPE performs better than BidMAC. The main reason for this is that the time of data transmission has a certain influence on the overall performance of the protocols. In BidMAC and BidCode, two data packets are transmitted within the same RTS/CTS handshake. When the packet



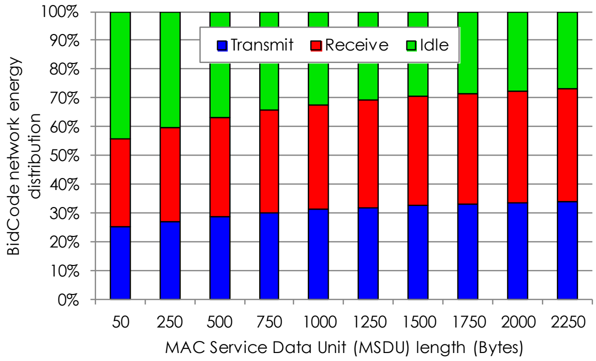
(a) Network throughput



(b) Network energy efficiency



(c) DCF: network energy distribution



(d) BidCode: network energy distribution

Figure 5.10: Throughput, energy efficiency, and energy consumption of the NC-aware MAC protocols versus the MSDU length in the Alice and Bob topology

length is short, the impact of data transmission on the overall transmission time is small. As the packet length increases, its contribution to the overall transmission time becomes more significant. Therefore, the BidCode and BidMAC protocols are more efficient when the MSDU length is shorter.

Fig. 5.10c and Fig. 5.10d show the energy distributions of the DCF and BidCode protocols versus the MSDU length, respectively. It can be seen that both protocols show similar results as the MSDU length increases. For small MSDU lengths, around 45% of the total energy consumption is due to idle periods whereas the remaining 30% and 25% are due to reception and transmission periods, respectively. As the MSDU length increases,

Table 5.4: Alice and Bob: Maximum Gains vs. MSDU Length

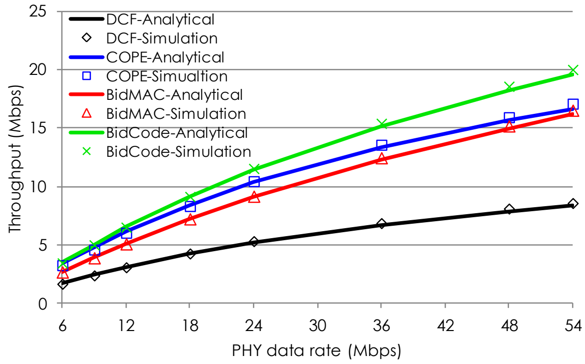
MSDU	Throughput		Energy efficiency	
Length	BidCode vs.	BidCode vs.	BidCode vs.	BidCode vs.
(Bytes)	DCF	COPE	DCF	COPE
50	1.71	0.36	1.70	0.36
250	1.61	0.32	1.59	0.31
500	1.52	0.27	1.50	0.26
750	1.47	0.24	1.44	0.23
1000	1.42	0.21	1.39	0.20
1250	1.37	0.19	1.34	0.18
1500	1.33	0.17	1.31	0.16
1750	1.31	0.16	1.29	0.15
2000	1.29	0.15	1.27	0.14
2250	1.26	0.14	1.24	0.13

the amounts of energy resources consumed for transmitting and receiving increase. For an MSDU length of 2250 bytes, the distribution of energy consumption is 30% for transmitting, 40% for receiving, and 30% for being idle.

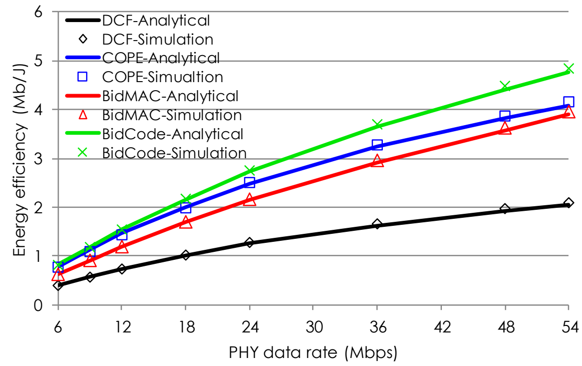
The maximum gains of BidCode vs. the MSDU length are reported in Table 5.4. The gain of BidCode versus DCF ranges from 1.71 to 1.26 as the MSDU length increases. Likewise, the gain of BidCode versus COPE varies between 0.36 and 0.13 as the MSDU length increases.

3) PHY Data Rate:

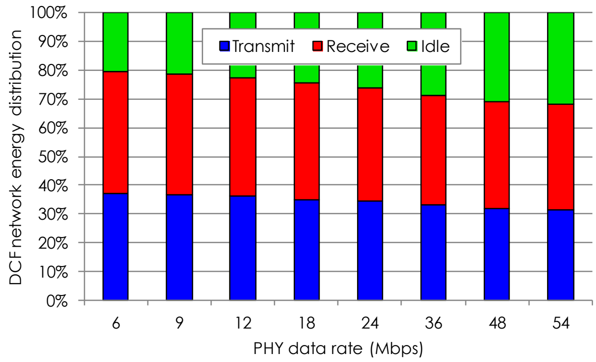
The impact of the PHY data rate on the performances of the protocols is evaluated in Fig. 5.11. The results are obtained for an MSDU length of 1500 bytes. Fig. 5.11a reports the throughput whereas Fig. 5.11b shows the energy efficiency. In general, the performance of the protocols increases as the PHY data rate increases, since the time to transmit data decreases and more data packets can be transmitted. It can be seen that BidCode achieves the highest performance for all the PHY data rates. BidMAC



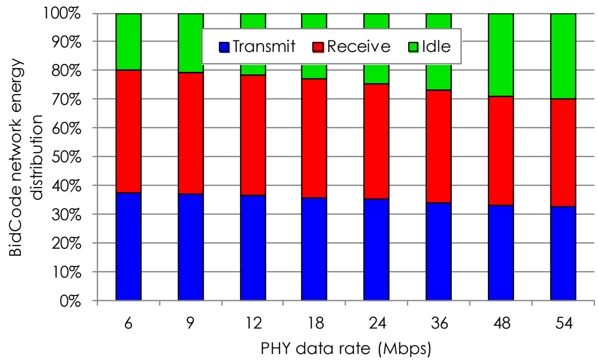
(a) Network throughput



(b) Network energy efficiency



(c) DCF: network energy distribution



(d) BidCode: network energy distribution

Figure 5.11: Throughput, energy efficiency, and energy consumption of the NC-aware MAC protocols versus the PHY data rate in the Alice and Bob topology

performs worse than COPE but it becomes more efficient as the PHY data rate increases and can almost perform the same for 54 Mbps.

Fig. 5.11c and Fig. 5.11d show the energy consumption of DCF and BidCode in the different operational states, namely, transmit, receive, and idle. The results are similar for both protocols. When the data rate is small, the data transmission time increases and so more energy is consumed for transmitting and receiving. It can seen that for the PHY data rate of 6 Mbps the energy consumption is distributed in 35% for transmitting, 45% for receiving, and 20% for being idle. However, when the data rate increases, the data transmission time becomes shorter and less energy is

Table 5.5: Alice and Bob: Maximum Gains vs. PHY Data Rate

PHY	Throughput		Energy efficiency	
Data Rate (Mbps)	BidCode vs. DCF	BidCode vs. COPE	BidCode vs. DCF	BidCode vs. COPE
6	1.06	0.04	1.06	0.04
9	1.10	0.06	1.09	0.05
12	1.11	0.06	1.10	0.06
18	1.16	0.09	1.15	0.08
24	1.20	0.10	1.18	0.10
36	1.26	0.14	1.25	0.13
48	1.32	0.16	1.30	0.15
54	1.33	0.17	1.31	0.16

consumed for transmitting and receiving. For the PHY data rate of 54 Mbps, the energy consumption is split into 30% for transmitting, 40% for receiving, and 30% for being idle.

Table 5.5 reports the maximum gains versus the PHY data rate. The gain of BidCode versus DCF is between 1.06 and 1.33 as the PHY data rate increases, whereas the gain of BidCode versus COPE ranges from 0.04 to 0.16.

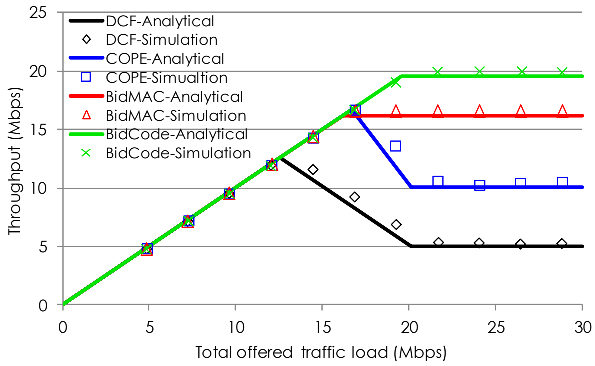
Cross Scenario

The results of the protocols in the cross scenario are presented and discussed as follows.

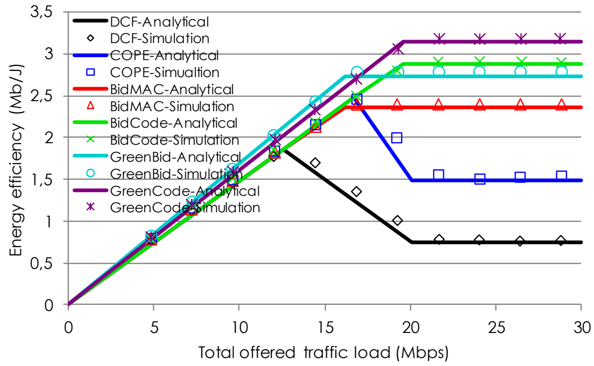
1) Traffic Load:

Fig. 5.12 summarizes the throughputs, energy efficiencies, and energy and time distributions of the protocols versus the traffic load in the cross topology. The results are obtained for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, a wakeup transition coefficient of 1.5, and an awake/sleep transition time of 500 μs (i.e. 250 μs for each transmission).

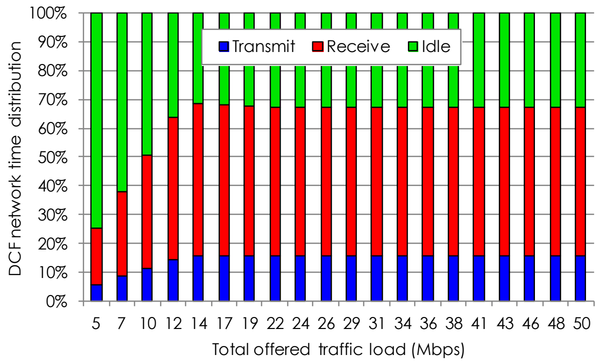
Fig. 5.12a shows the throughput from end to end whereas Fig. 5.12b



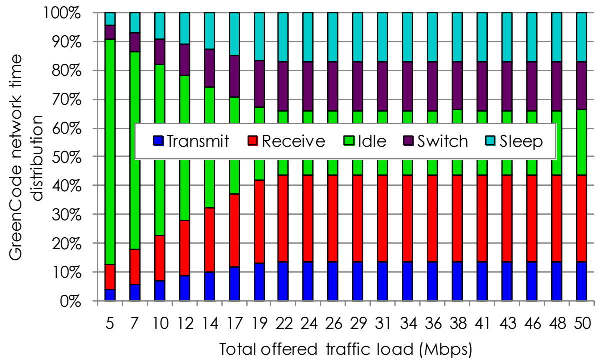
(a) Network throughput



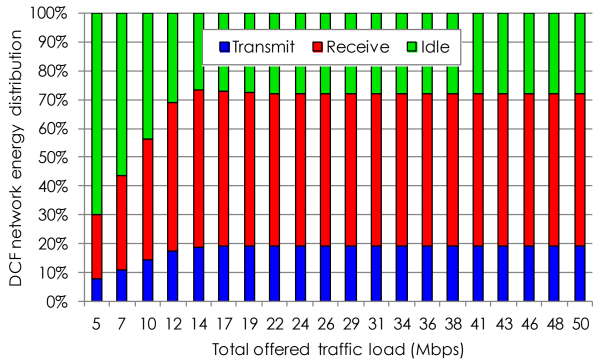
(b) Network energy efficiency



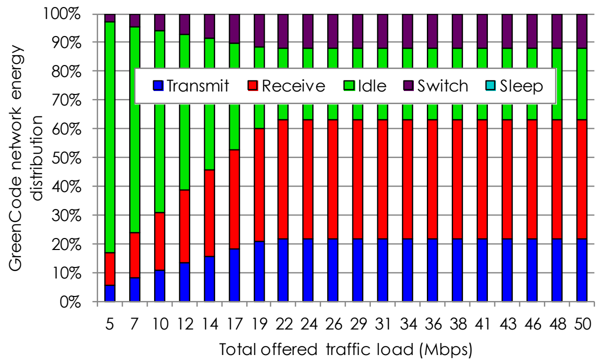
(c) DCF: network time distribution



(d) GreenCode: network time distribution



(e) DCF: network energy distribution



(f) GreenCode: network energy distribution

Figure 5.12: Throughput, energy efficiency, and time and energy distributions of the NC-aware MAC protocols versus the traffic load in the Cross topology

shows the energy efficiency of the network. It can be seen that BidMAC and BidCode can achieve significantly higher gains when compared to DCF and COPE, since they can maintain the same performance as that shown in the Alice and Bob topology. However, the saturation performance of DCF is significantly lower than that shown in the Alice and bob topology. The relay node needs to compete for the channel access with 4 source nodes. Thus, it can only get 1/5 of the channel whereas the source nodes get 4/5 of the channel. When NC is used, the maximum performance of COPE is shown at 2/3 of the load where each source node gets 1/6 of the channel and the relay node gets 1/3. However, the saturation performance of COPE will be reduced to 2/5 of the channel capacity at 4/5 of the load, since the relay node will only get 1/5 of the channel to transmit coded packets. BidMAC and BidCode guarantee half of the channel accesses for the relay node. Therefore, their performances will be stable across the increasing number of nodes in the relay's coverage.

Furthermore, the energy efficiencies of GreenBid and GreenCode are compared to those of BidMAC and BidCode. It can be seen that GreenBid and GreenCode not only achieve the highest energy efficiency for high traffic loads, but also for low traffic loads. GreenBid provides slightly higher energy efficiency than GreenCode for low to medium loads until GreenBid reaches the saturation point around 16.5 Mbps. After that, GreenCode shows higher energy efficiency. The main reason for this is that GreenBid performs more bidirectional transmissions where overhearing nodes can sleep, hence saving more energy. However, GreenCode is more efficient and allows the relay node to send data packets at a higher rate. Under saturation, GreenBid achieves higher energy efficiency than BidMAC but lower than BidCode whereas GreenCode achieves the highest energy efficiency.

In Fig. 5.12c and Fig. 5.12d, the amount of time spent by the DCF

Table 5.6: Cross: Maximum Gains vs. Traffic Load

Saturation	BidCode vs.		GreenCode vs.	
Traffic Load	DCF	COPE	DCF	COPE
Network throughput	2.89	0.95	2.89	0.89
Network energy efficiency	2.85	0.93	3.21	1.11

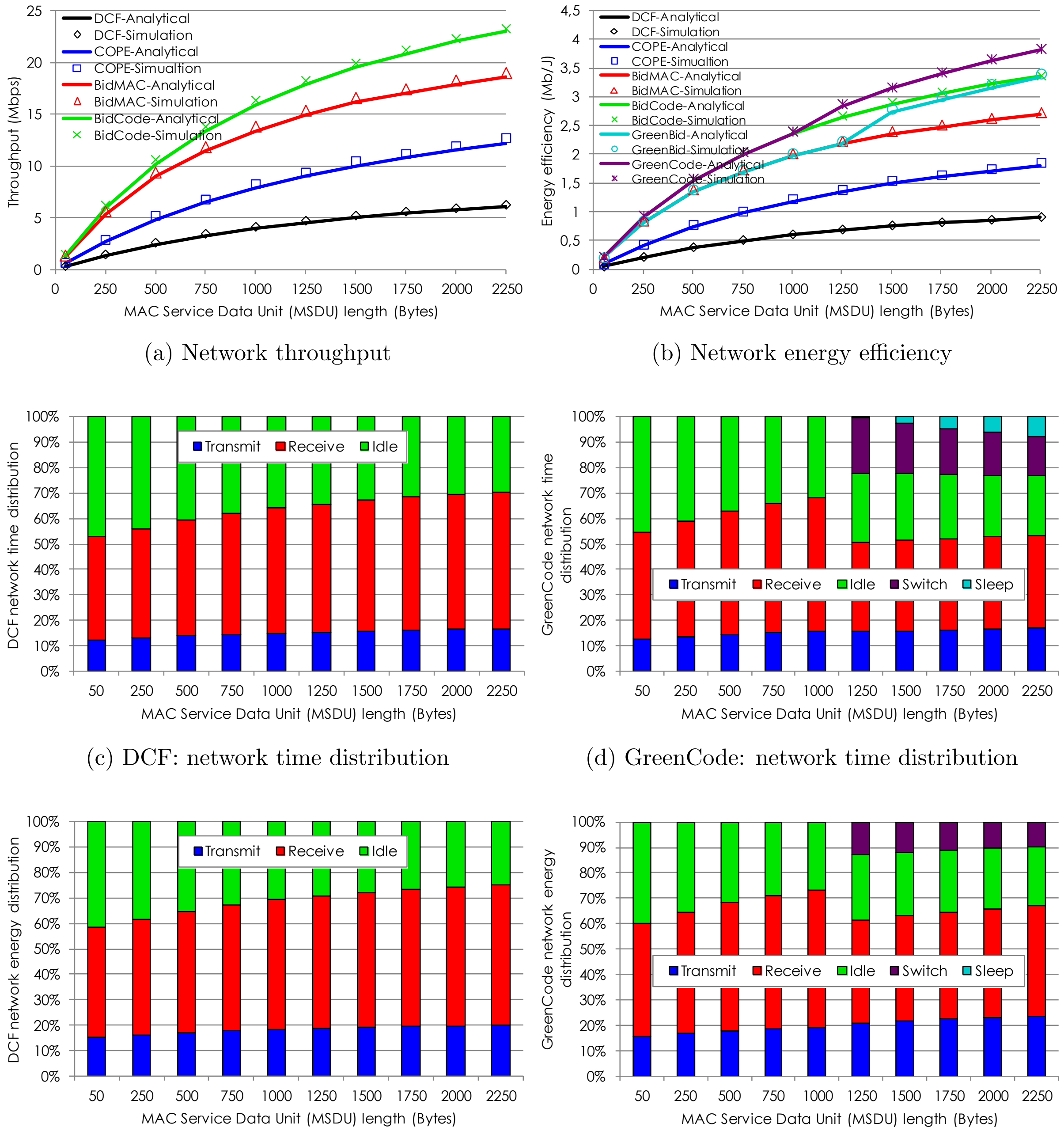
and GreenCode protocols in the different operational states is studied as the traffic load increases. The contribution of each operational state to the overall energy consumption of the protocols is also shown in Fig. 5.12e and Fig. 5.12f. In DCF, most of the time and most of the energy resources (around 75%) are dedicated to listening activities when the traffic load is low. When the traffic load is high, most of the time and most of the energy resources (up to 55%) are dedicated to receiving and overhearing activities. On the other hand, GreenCode reduces the time and energy consumed for receiving packets. However, it introduces the components of time and energy consumed for sleeping and switching between idle and sleeping. The time spent for sleeping is around 20% of the total time whereas the time spent for switching between idle and sleeping is around 20% as well. In addition, the energy consumed for sleeping represents less than 1% of the total energy consumption whereas the energy consumed for switching between idle and sleeping is above 10%.

Table 5.6 presents the maximum gains of BidCode and GreenCode versus DCF and COPE, respectively, in the cross topology. As for BidCode, the maximum gains versus DCF are up to 2.89 whereas those versus COPE are up to 0.95. Regarding GreenCode, the throughput gains are the same as those shown for BidCode whereas the energy efficiency gains are up to 3.21 versus DCF and 1.11 versus COPE.

2) MSDU Length:

Fig 5.13 analyzes the impact of the MSDU length on the overall performances of the protocols for a PHY data rate of 54 Mbps, a wakeup transition coefficient of 1.5, and an awake/sleep transition time of 500 μs (i.e., 250 μs for each transition). Fig. 5.13a reports the throughput and Fig. 5.13c captures the energy efficiency. The throughput values are similar to those shown for the Alice and Bob scenario, except that DCF and COPE experience lower bounds. BidCode achieves the highest throughput, followed by BidMAC, COPE, and DCF. As for the energy efficiency, similar conclusions can be drawn for the protocols, except for GreenBid and GreenCode. The energy efficiency of GreenBid increases as that of BidMAC until the packet length is sufficiently long to let the nodes enter the sleep state within a data exchange. This corresponds to a packet length that makes the sleep period be greater than zero. For a data rate of 54 Mbps, the critical MSDU length is 1250 bytes for which the sleep period is equal to or lower than zero. For MSDU lengths above this value, the energy efficiency of GreenBid outperforms BidMAC showing higher gains as the MSDU length increases. For an MSDU of 2250 bytes, GreenBid reaches BidCode, which is always better than GreenBid for lower MSDU lengths. Similar to GreenBid, GreenCode outperforms BidCode when the MSDU length is greater than 1000 bytes. The critical MSDU length of GreenCode is lower than that of GreenBid because the bidirectional transmission in GreenCode involves a packet that is coded and so it contains an additional header of 40 bytes, thus increasing the transmission time.

Fig. 5.13c and Fig. 5.13c illustrate the time distributions of DCF and GreenCode, respectively, versus the MSDU length. Similarly, Fig. 5.13e and Fig. 5.13f show the energy distribution of the protocols. In DCF, the share of time and energy consumed is 15% for transmitting, 35% for receiving, and 40% for being idle when the MSDU length is 50 bytes. As the MSDU length increases, the share of time and energy consumed during reception periods increases significantly. For an MSDU of 2250 bytes, more than half of the energy and time resources are spent for receiving



(a) Network throughput

(b) Network energy efficiency

(c) DCF: network time distribution

(d) GreenCode: network time distribution

(e) DCF: network energy distribution

(f) GreenCode: network energy distribution

Figure 5.13: Throughput, energy efficiency, and time and energy distributions of the NC-aware MAC protocols versus the MSDU length in the Cross topology

and overhearing activities. On the contrary, GreenCode shows a similar behavior to DCF until the MSDU length is 1000 bytes. Then, the nodes can go to sleep during data exchanges where they are not involved and so the switch and sleep periods have a certain influence on the overall time and energy consumption of GreenCode. It can be seen that the contribution of switching between idle and sleeping decreases as the MSDU length increases whereas the contribution of sleeping increases. The main reason for this is that nodes can sleep longer with longer MSDU lengths, thus consuming more energy during sleep periods. As a result, the contribution of switching between idle and sleeping, which is constant, decreases.

Table 5.7 summarizes the maximum gains of BidCode and GreenCode in the cross scenario as the MSDU length increases. The throughput gain of BidCode versus DCF ranges from 3.52 to 2.77 and from 1.27 and 0.89 when compared to COPE. The energy efficiency gain of BidCode versus DCF varies between 3.50 and 2.98 and then between 3.19 and 3.25. Likewise, the energy efficiency gain of BidCode versus COPE ranges from 1.26 to 1.00 and from 1.10 to 1.13.

3) PHY Data Rate:

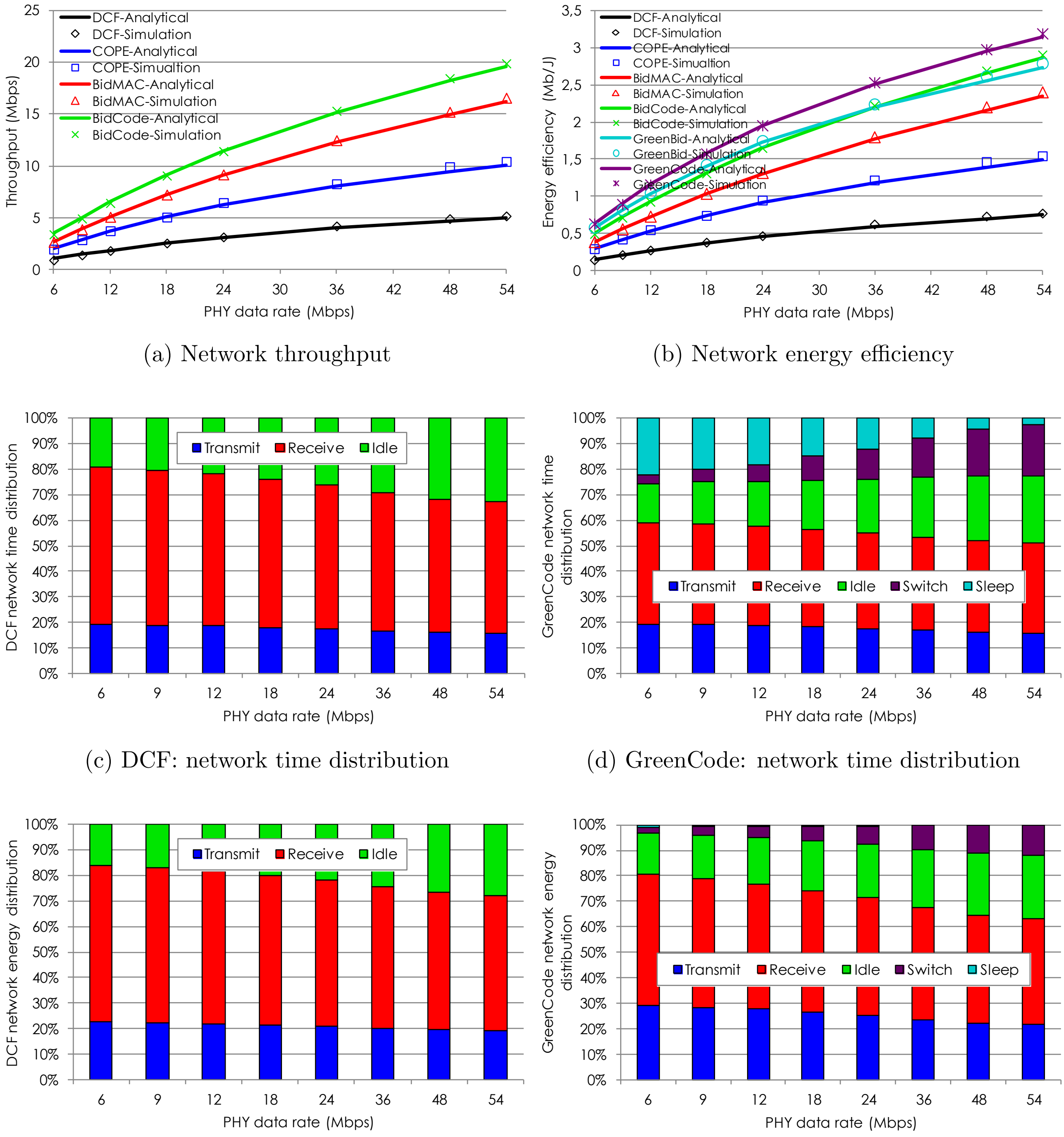
Fig. 5.14 shows the throughputs and energy efficiencies of the protocols versus the PHY data rate in Fig. 5.14b and Fig. 5.14c, respectively. The results are shown for an MSDU length of 1500 bytes, a wakeup transition coefficient of 1.5, and an awake/sleep transition time of 500 μs (i.e., 250 μs for each transition). The throughputs of the protocols increase as the data rate increases since the time to transmit a data packet decreases and so more information can be transmitted. BidCode and BidMAC outperform DCF and COPE for all data rates but BidCode performs better than BidMAC. The energy efficiencies of these protocols show great similarities to what is shown for the throughputs. In contrast, GreenBid significantly improves BidMAC for all data rates and can improve BidCode for rates

Table 5.7: Cross: Maximum Gains vs. MSDU Length

MSDU	Throughput		Energy efficiency	
Length	BidCode vs.	BidCode vs.	GreenCode vs.	GreenCode vs.
(Bytes)	DCF	COPE	DCF	COPE
50	3.52	1.27	3.50	1.26
250	3.35	1.19	3.31	1.17
500	3.21	1.11	3.17	1.09
750	3.11	1.06	3.07	1.04
1000	3.03	1.02	2.98	1.00
1250	2.95	0.98	3.19	1.10
1500	2.89	0.95	3.21	1.11
1750	2.85	0.93	3.24	1.12
2000	2.81	0.91	3.25	1.13
2250	2.77	0.89	3.25	1.13

up to 36 Mbps. Moreover, GreenCode always shows the highest energy efficiency as the data rate increases.

Fig. 5.14e and Fig. 5.14d, respectively, evaluate the impact of the PHY data rate on the time spent in the different operational states for the DCF and GreenCode protocols whereas Fig. 5.14e and Fig. 5.14f show the energy distributions of the protocols. In DCF, the share of time and energy consumed during reception periods decreases as the data rate increases because the data transmission time decreases. In contrast, for GreenCode the network remains in the sleep state for more than 20% of the time for a data rate of 6 Mbps and only 4% during switching periods. The share of energy consumption for 6 Mbps is less than 1% and 2% for sleeping and switching, respectively. However, when the data rate increases, the share of energy consumption during sleep periods is almost 0% while the amount of time represents only the 2%. In addition, the energy consumed during switching periods is around 10% of the overall energy consumption and 20% of the total time.



(a) Network throughput

(b) Network energy efficiency

(c) DCF: network time distribution

(d) GreenCode: network time distribution

(e) DCF: network energy distribution

(f) GreenCode: network energy distribution

Figure 5.14: Throughput, energy efficiency, and time and energy distributions of the NC-aware MAC protocols versus the PHY data rate in the Cross topology

Table 5.8: Cross: Maximum Gains vs. PHY Data Rate

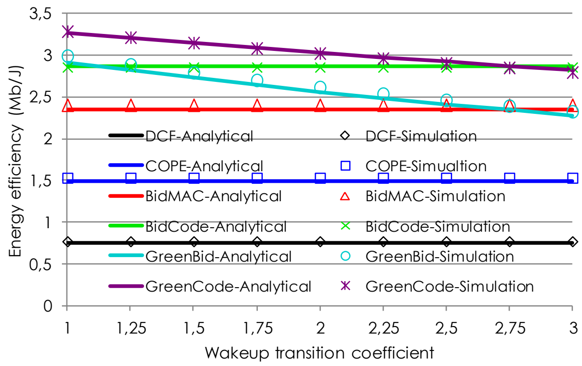
PHY	Throughput		Energy efficiency	
Data Rate (Mbps)	BidCode vs. DCF	BidCode vs. COPE	GreenCode vs. DCF	GreenCode vs. COPE
6	2.44	0.73	3.40	1.21
9	2.49	0.75	3.98	1.19
12	2.52	0.77	3.36	1.19
18	2.60	0.81	3.33	1.17
24	2.66	0.84	3.30	1.16
36	2.77	0.89	3.26	1.14
48	2.86	0.93	3.23	1.12
54	2.89	0.95	3.21	1.11

Table 5.8 records the maximum gains versus the PHY data rate in the cross scenario. The maximum throughput gains versus DCF are between 2.44 and 2.89 as the PHY data rate increases whereas when compared to COPE they range from 0.73 to 0.95. In addition, the maximum gains vary from 3.49 and 3.21 and from 1.21 to 1.11 versus DCF and COPE, respectively.

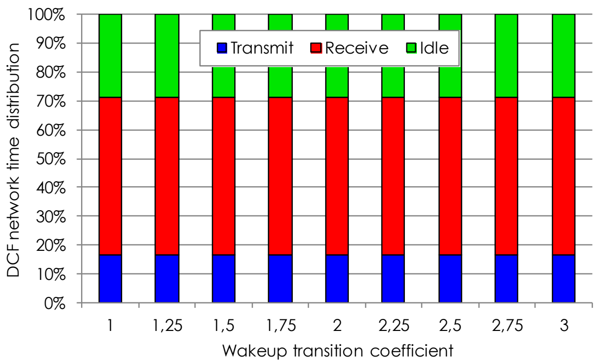
4) Wakeup Transition Coefficient:

Fig. 5.15 shows the energy efficiencies and time and energy distributions of the protocols versus the wakeup transition coefficient. This coefficient determines the amount of energy consumed in the transition between sleep and idle states having as reference the value of power consumed in the idle state. The higher the value of the wakeup transition coefficient is, the higher the energy consumed in the transition between sleep and idle states is. The results are obtained for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, and an awake/sleep transition time of 500 μs (i.e., 250 μs for each transition).

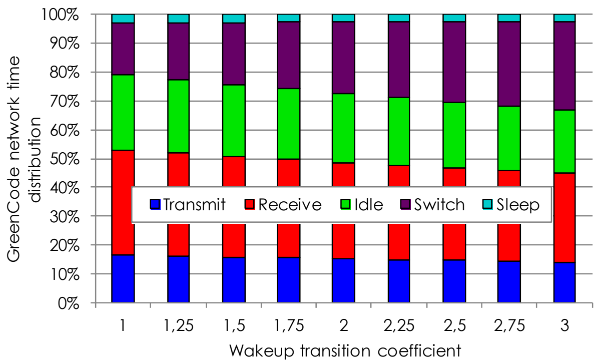
The energy efficiency is plotted in Fig. 5.15a. The value of the wakeup transition coefficient only affects the energy efficiency of the GreenBid and



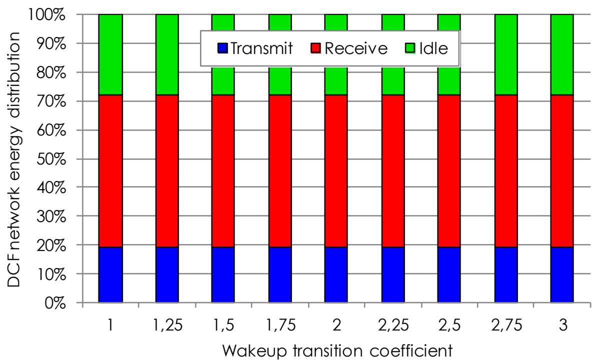
(a) Network energy efficiency



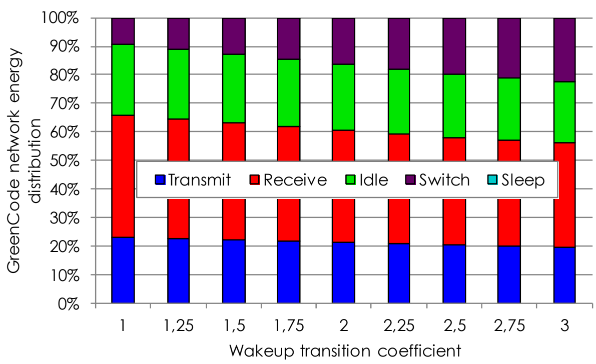
(b) DCF: network time distribution



(c) GreenCode: network time distribution



(d) DCF: network energy distribution



(e) GreenCode: network energy distribution

Figure 5.15: Throughput, energy efficiency, and energy consumption of the NC-aware MAC protocols versus the wakeup transition coefficient in the Cross topology

GreenCode protocols. As the value of the wakeup transition coefficient increases, the energy efficiency of GreenBid and GreenCode decreases and approaches to the energy efficiency of BidMAC and BidCode, respectively. The critical value of the wakeup transition coefficient that makes the energy efficiency of GreenBid and GreenCode be the same as that of BidMAC and BidCode, respectively, is 2.75.

Also, the impact of the wakeup transition coefficient on the time and energy distributions in the different operational states is evaluated as follows. Fig. 5.15b and Fig. 5.15c show the time distribution of DCF and GreenCode, respectively. Similarly, Fig. 5.15d and Fig. 5.15e represent the energy distribution of DCF and GreenCode, respectively. In GreenCode, it can be seen that as the wakeup transition coefficient increases more time and energy are dedicated to the switching procedure. A maximum value of 30% of the overall time and a maximum value of 22% of the overall energy consumption correspond to switching.

In Table 5.9, the maximum energy efficiency gains of GreenCode versus DCF, COPE, BidMAC, and BidCode are reported as a function of the wakeup transition coefficient. The gain versus DCF ranges from 3.37 to 2.78. The gain versus COPE is between 1.19 and 0.9. GreenCode shows 0.39 to 0.20 of gains when compared with BidMAC. The lowest gains are shown for BidCode, which vary between 0.14 and -0.02.

5) Awake/Sleep Transition Time:

Fig. 5.16 presents the energy efficiencies and time and energy distributions of the protocols versus the awake/sleep transition time. The transition time determines how much time is spent in the transition from idle to sleep and the transition from sleep to idle. The longer the transition time is, the longer the data transmission time has to be in order to make the sleep period be greater than zero. The results are plotted for an MSDU length of 1500 bytes, a PHY data rate of 54 Mbps, and a wakeup transition

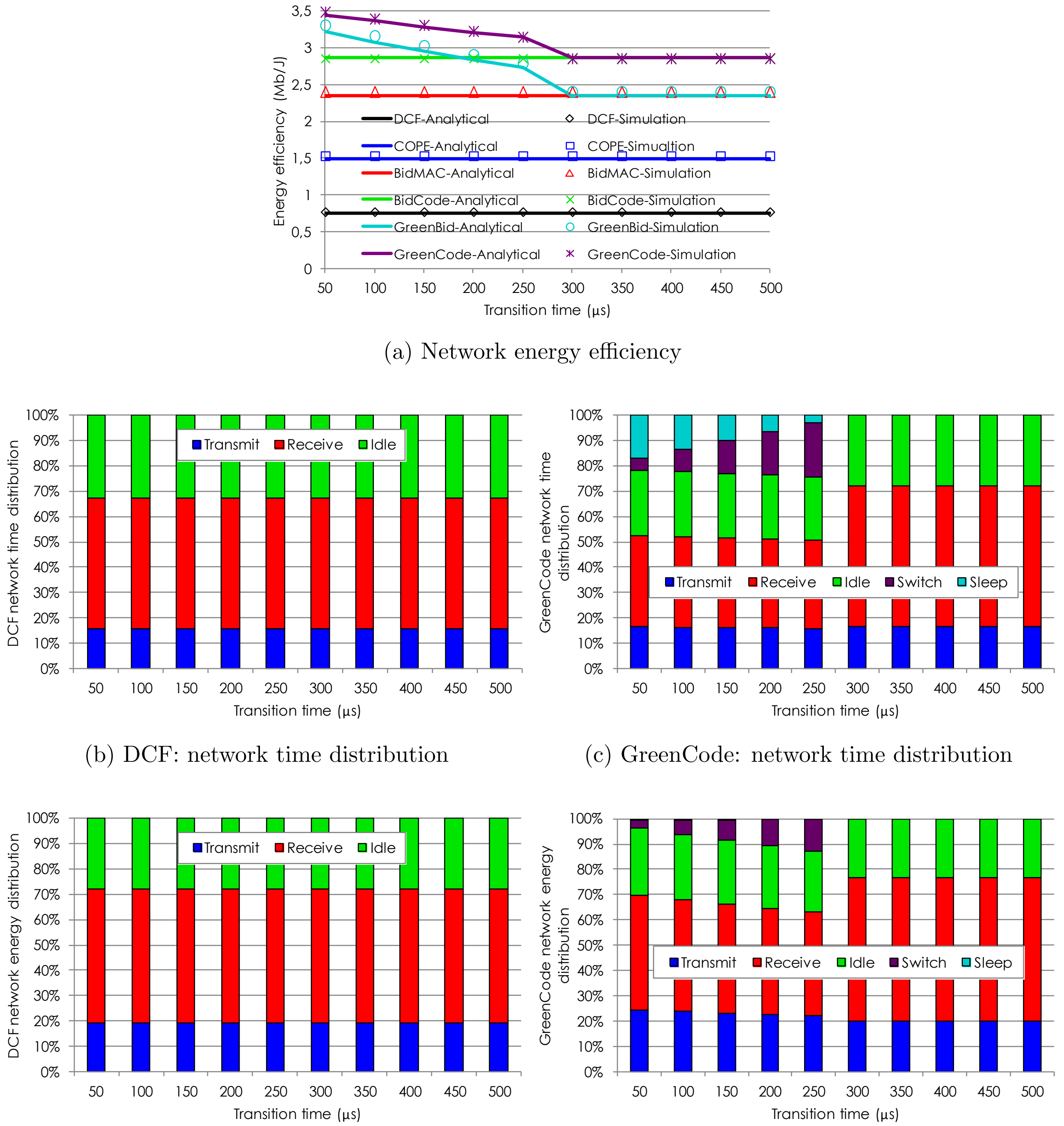
Table 5.9: Cross: Maximum Gains vs. Wakeup Transition Coefficient

Wakeup Transition Coefficient	Energy efficiency GreenCode vs. DCF	GreenCode vs. COPE	GreenCode vs. BidMAC	GreenCode vs. BidCode
1	3.37	1.19	0.39	0.14
1.25	3.29	1.15	0.36	0.11
1.5	3.21	1.11	0.33	0.09
1.75	3.13	1.07	0.31	0.07
2	3.06	1.03	0.29	0.05
2.25	2.98	1.00	0.26	0.04
2.5	2.91	0.96	0.24	0.02
2.75	2.85	0.93	0.22	0.00
3	2.78	0.90	0.20	-0.02

coefficient of 1.5.

Fig. 5.16a shows the energy efficiency. The value of the transition time only affects the energy efficiencies of the GreenBid and GreenCode protocols as they enable sleeping processes. As the transition time increases, the energy efficiency of GreenBid and GreenCode decreases, since the sleep period also decreases. The critical value of the transition time that makes the sleep period be equal to or lower than zero is 300 μs. For transition times above 300 μs, the energy efficiencies of GreenBid and GreenCode are the same as those of BidMAC and BidCode, respectively, because none of the nodes can go to sleep. The critical value of the transition time may increase or decrease depending on the MSDU length and the PHY data rate.

To conclude, the influence of the transition time on the overall energy consumption and time spent in the different operation states for the DCF and GreenCode protocols is analyzed as follows. Fig. 5.16b illustrates the time distribution of DCF whereas Fig. 5.16c represents the time distribution of GreenCode. Likewise, the DCF energy distribution is shown



(a) Network energy efficiency

(b) DCF: network time distribution

(c) GreenCode: network time distribution

(d) DCF: network energy distribution

(e) GreenCode: network energy distribution

Figure 5.16: Throughput, energy efficiency, and energy consumption of the NC-aware MAC protocols versus the awake/sleep transition time in the Cross topology

in Fig. 5.16d and the GreenCode energy distribution is plotted in Fig. 5.16e. In GreenCode, when the transition time is small, with short data packets and fast data rates a positive sleep period can be achieved, thus improving energy efficiency. For example, for a transition time of 50 μs, the nodes remain in the sleep state for almost 20% of the time. In addition, the contribution of switching periods to the overall energy consumption is relatively small (below 10%). However, when the transition time increases, the amount of time that the nodes spend in the sleep state decreases whereas the share of energy consumption during switching periods increases. For the critical transition time of 250 μs, the nodes remain in the sleep sate for less than 3% of the time. In addition, the portion of energy consumed for switching between idle and sleep states is around 20% of the total energy consumption.

Table 5.10 summarizes the maximum gains of GreenCode versus the transition time in the cross scenario. The gain of GreenCode versus DCF and COPE is between 3.61 and 2.85 and between 1.32 and 0.93, respectively, as the transition coefficient increases. GreenCode shows a gain from 0.46 to 0.22 over BidMAC. When compared to BidCode, the gain of GreenCode varies between 0.20 and 0.09 up to the critical transition time of 250 μs.

5.5 Experiments Framework

This section describes experimental implementations of the reference COPE and proposed BidCode protocols carried out on a programmable wireless platform called WARP [28] and tested in a proof-of-concept network formed by two source nodes and a relay node (i.e., the Alice and Bob scenario). There are various available wireless platforms for prototyping at the MAC layer [27]. Among them, WARP (version 3) has been selected

Table 5.10: Cross: Maximum Gains vs. Awake/Sleep Transition Time

Awake/Sleep Transition Time (μs)	Energy efficiency GreenCode vs. DCF	GreenCode vs. COPE	GreenCode vs. BidMAC	GreenCode vs. BidCode
50	3.61	1.32	0.46	0.20
100	3.51	1.26	0.43	0.17
150	3.40	1.21	0.40	0.14
200	3.30	1.16	0.36	0.12
250	3.21	1.11	0.33	0.09
300	2.85	0.93	0.22	0.00
300	2.85	0.93	0.22	0.00
300	2.85	0.93	0.22	0.00
300	2.85	0.93	0.22	0.00
300	2.85	0.93	0.22	0.00

because it offers an available open-source reference design that can interoperate with commercial IEEE 802.11a/g devices, acting as either AP or STA. Further details about the WARP platform and its reference design are provided in Appendix A and Appendix B.

The DCF MAC source code of the reference design of WARP has been modified to implement COPE and BidCode. The focus has been put on the evaluation of the experimental throughputs and energy efficiencies of DCF, COPE, and BidCode, which have been measured in each node by means of custom-design Python scrips and Energino meters [8] controlled through a custom program developed in LabVIEW. The reader may refer to Appendix C for further details about Energino's hardware and software. In order to validate the accuracy of the experimental implementation, the theoretical throughput and energy efficiency results of DCF, COPE, and BidCode presented in the previous section are compared to the experimental results, taking into account various values for relevant system parameters such as the traffic load, packet length, and data rate.

5.5.1 NC-Aware MAC Protocols Implementation

The COPE protocol is mainly implemented in the upper-level MAC of the 802.11 reference design of WARP (see Appendix B), i.e., the C code in the CPU High MicroBlaze core (wlan_mac_ap.c). The AP is a key element as it acts as a network coding relay node for STA 1 and STA 2, which both act as the source nodes (i.e., Alice and Bob). Ideally, the implementation of virtual queues at the AP to manage data packets received from STA 1 and STA 2 would be required, hence easily identifying coding opportunities. However, the aim is to implement the protocol from a conceptual approach (as a proof of concept), i.e. emulating the protocol operation with minimum changes in the code because deep modification of the code would require advanced knowledge of Field Programmable Gate Array (FPGA) programming. Therefore, two global variables that represent the amount of data packets received from STA 1 and STA 2, respectively, (i.e., flow1 and flow2) are defined in the aforementioned file as follows:

```
<wlan_mac_ap.c>
//Counter for the number of received data packets from STA 1
static int flow1 = 0;
//Counter for the number of received data packets from STA 2
static int flow2 = 0;
// Random variable used to determine the actual receiver of a coded data
    packet between STA 1 and STA 2
static int randomold = 1;
// Static MAC address of WARP node 339 acting as STA 1
static u8 sta1_addr[6] = {0x40, 0xD8, 0x55, 0x04, 0x22, 0x9E};
//Static MAC address of WARP node 220 acting as STA 2
static u8 sta2_addr[6] = {0x40, 0xD8, 0x55, 0x04, 0x21, 0x25};
```

As it can been seen in the box above, the static MAC addresses of STA 1 and STA 2 are also defined in order to determine the source of the received data packet by the AP and update the corresponding counter, flow1 or flow2. Also, a random variable randomold is defined to randomly selected the destination of an XOR coded data packet when there is a coding opportunity (i.e., when both flow1 and flow2 are greater than zero). These global variables are reset to zero between consecutive tests through a piece of code that is included in the Python experiment framework. This code calls a function that connects the external Python code with the lower-level MAC of the WARP v3 nodes. A function called ap.reset_flow() has been defined in the Python script for this purpose. This function sends a command to the lower-level MAC, which calls a void function called reset_flow() as follows:

```
<throughput_traffic.py> and <throughput_payload.py> and <throughput_rate
    .py>
// Command to reset flows from the python experiment script
ap.reset_flow()
<node.py>
def reset_flow(self,):
        // Command in the node script that sends an inter-process
            communication message to reset flows to lower-level MAC
        self.set_low_param(cmds.CMD_PARAM_LOW_PARAM_RESET_FLOWS,0)
<wlan_mac_dcf.c>
//Function to reset flows at lower-level MAC
void reset_flow(){
        flow1 = 0;
        flow2 = 0;
}
```

Then, a number of modifications are introduced in the function mpdu_rx_process that, as its name indicates, is responsible for handling wireless receptions at the upper-level MAC. When a packet of type DATA is received by the AP from an STA with a valid FCS (i.e., without errors), counters flow1 and flow2 are updated according to the MAC address of each STA. Then, it is checked if both counters are greater than zero. If so, this means that there is a coding opportunity. Both counters are then decremented by one. A data packet with a new field, called coded, set to 1 is generated and appended to one of the output queues as if it was a normal data packet generated from the LTG framework (see Appendix B).

The length of the XOR coded data packet is set the same as that of the received data packet with an additional header of 40 bytes that emulates coding information needed for decoding at destination. The destination of the XOR coded packet is randomly chosen using the randomold variable with equal probability between the two STAs. Once the packet is inserted in the output queue, it will be sent using the standard DCF MAC rules. Otherwise, if one or both counters are zero, the AP will send an ACK packet upon successful reception of the data packet through the lower-level MAC. All this operation is expressed as follows:

```
<wlan_mac_ap.c>
void mpdu_rx_process(void* pkt_buf_addr, u8 rate, u16 length){
        // Define a variable for the length of the coded data packet
        u32 codedpayload_length;
        // Set codepayload_length to the length of the received data
           packet plus 40 bytes for the coding header
        codedpayload_length = ((mpdu_info->length) - (sizeof(llc_header)
           + sizeof(mac_header_80211)))+40;
        // Check the source address of the received data packet, if STA
           1 or STA 2, and update flow1 or flow2
        if (wlan_addr_eq(rx_80211_header->address_2,sta1_addr)){
                flow1+=1;
        }
        else{
                if (wlan_addr_eq(rx_80211_header->address_2,sta2_addr)){
                        flow2+=1;
                }
                if ((flow1 > 0)&&(flow2 > 0)){
                        // Coding opportunity TRUE
                        //Decrement counters
                        flow1 -= 1;
                        flow2 -= 1;
                        // Create and configure the coded data packet
                        tx_header_common.coded = 1;
                        int random = randomold;
                        randomold = (random % 2)+1;
                        if (random == 1){
                                randdestination =
                                   wlan_mac_high_find_station_info_ADDR
                                   (&association_table, sta1_addr);
```

```
        }
else{
        randdestination =
            wlan_mac_high_find_station_info_ADDR(&
            association_table, sta2_addr);
}
//Append the coded data packet in an output queue
if(queue_num_queued(AID_TO_QID(randdestination->AID)) <
    max_queue_size){
        queue_checkout(&checkout,1);
        if(checkout.length == 1){
                //There was at least 1 free queue
                    element
                tx_queue = (packet_bd*)(checkout.first);
                wlan_mac_high_setup_tx_header( &
                    tx_header_common, randdestination->
                    addr, eeprom_mac_addr );
                mpdu_ptr_u8 = (u8*)((tx_packet_buffer*)(
                    tx_queue->buf_ptr))->frame;
                tx_length = wlan_create_data_frame((void
                    *)((tx_packet_buffer*)(tx_queue->
                    buf_ptr))->frame,
                                &tx_header_common,
                    MAC_FRAME_CTRL2_FLAG_FROM_DS);
                mpdu_ptr_u8 += sizeof(mac_header_80211);
                llc_hdr = (llc_header*)(mpdu_ptr_u8);
                //Prepare the MPDU LLC header
                llc_hdr->dsap = LLC_SNAP;
                llc_hdr->ssap = LLC_SNAP;
                llc_hdr->control_field =
                    LLC_CNTRL_UNNUMBERED;
                bzero((void *)(llc_hdr->org_code), 3);
                //Org Code 0x000000: Encapsulated
                    Ethernet
                llc_hdr->type = LLC_TYPE_CUSTOM;
                tx_length += sizeof(llc_header);
                tx_length += codedpayload_length;
                wlan_mac_high_setup_tx_frame_info (
                    tx_queue, (void*)randdestination,
                    tx_length, MAX_RETRY,
                    default_tx_gain_target,(
                    TX_MPDU_FLAGS_FILL_DURATION |
                    TX_MPDU_FLAGS_REQ_TO) );
```

```
                                enqueue_after_end(AID_TO_QID(
                                   randdestination->AID), &checkout);
                                check_tx_queue();
                                }
                        }
                }
        }
}
```

In order to allow an STA receiving or overhearing a coded data packet to process it as a normal data packet, the operation of the lower-level MAC, i.e., the C code in the CPU Low MicroBlaze core (wlan_mac_dcf.c), has been modified as follows. Specifically, some modifications have been introduced in the frame_receive function. Basically, when an STA receives a coded data packet destined to its address, it should reply with an ACK packet through the auto-responder module and pass it to the upper-level MAC. Otherwise, when an STA receives a coded data packet not destined to its address, it should treat it as a received data packet, passing it to the upper-level MAC. However, in the case of an overhearing STA, the auto-responder is not enabled. So, this means that there is no reliability for overheard coded packets. The code lines are as follows:

```
<wlan_mac_dcf.c>
u32 frame_receive(u8 rx_pkt_buf, u8 rate, u16 length){
        if (unicast_to_me && !WLAN_IS_CTRL_FRAME(rx_header)) {
                // The data packet is sent to my address
                // Configure and sent the ACK packet after a SIFS
                wlan_phy_set_tx_signal(TX_PKT_BUF_ACK, tx_rate,
                   tx_length + WLAN_PHY_FCS_NBYTES);
        }
        if (unicast_to_me || to_broadcast || rx_header->coded == 1){
                // If the packet is unicasted to me, broadcast, or coded
                    (rx_header->coded == 1), passed it to the upper-
                   level MAC
                wlan_mac_low_frame_ipc_send();
        }
}
```

The implementation of BidCode has mainly been carried out in the lower-level MAC, i.e., the C code in the CPU Low MicroBlaze core (wlan_mac_dcf.c). As in the implementation of COPE, the goal is to implement the protocol with minimum changes in the code. In the wlan_mac_dcf.c file, the global variables flow1 and flow 2 and the static MAC address of the 3 WARP nodes, namely, AP, STA 1, and STA 2, are defined as follows:

```
<wlan_mac_dcf.c>
//Counter for the number of received data packets from STA 1
static int flow1 = 0;
//Counter for the number of received data packets from STA 2
static int flow2 = 0;
// MAC address of WARP node 112 acting as AP
static u8 ap_addr[6] = {0x40, 0xD8, 0x55, 0x04, 0x21, 0x72};
// MAC address of WARP node 339 acting as STA 1
static u8 sta1_addr[6] = {0x40, 0xD8, 0x55, 0x04, 0x22, 0x9E};
// MAC address of WARP node 220 acting as STA 2
static u8 sta1_addr[6] = {0x40, 0xD8, 0x55, 0x04, 0x22, 0x9E};
```

In order to allow the AP acting as a relay node to send a coded data packet upon reception of a data packet after a SIFS, the auto-responder state machine is enabled and configured with a data packet of subtype CODED. This new packet type is created through a new function called wlan_create_coded_frame that assigns subtype CODED in the frame_control_1 field of the packet header. This function is necessary because data packets are created at the upper-level MAC. However, a coded data packet needs to be prepared before reception completes, thus respecting the SIFS timing requirement. The function is similar to that used to create a normal data packet contained in wlan_mac_packet_types.c (MAC High Framework). The function is defined as follows:

```
<wlan_mac_dcf.c>
int wlan_create_coded_frame(void* pkt_buf, mac_header_80211_common*
    common, u8 flags) {
        // Set subtype CODED
        data_80211_header->frame_control_1 =
            MAC_FRAME_CTRL1_SUBTYPE_CODED;
```

```
        // Set MAC address of AP as the source of the CODED packet
        memcpy(data_80211_header->address_2,ap_addr,6);
}
```

To create a data packet of type CODED, the new packet type also needs to be defined in the file wlan_mac_802_11_defs.h (MAC High Framework) as follows:

```
<wlan_mac_802_11_defs.h>
//Define subtype CODED as type/subtype (10, 1101) as specified in
//table 8.1 of 802.11 2011-2012. The subtype is reserved.
#define MAC_FRAME_CTRL1_SUBTYPE_CODED (MAC_FRAME_CTRL1_TYPE_DATA | 0xD0)
```

In the function frame_receive, when a packet of type DATA destined to the AP is received with a valid FCS, counters flow1 or flow2 are increased by one depending on whether the source address of the received packet is STA 1 or STA 2. When one of the counters is zero, the AP simply responds with an ACK packet after a SIFS. However, when both counters are greater than zero, this means that there is a coding opportunity and the AP can transmit a coded packet in response to the received data packet. Both counters are decremented by one and the auto-responder is enabled and configured with a packet of type CODED. The length and rate of the CODED packet is set to those of the received data packet plus 40 bytes for the XOR header used to decode the coded packet. Otherwise, when a packet of type CODED destined to an STA is received with a valid FCS, the auto-responder is enabled and configured with a packet of type ACK. The packet of type CODED will be processed as a normal data packet. In addition, when an STA receives a packet of type CODED not destined to its address, the STA will process it as if it was a normal data packet, passing it to the upper-level MAC. The function is modified as follows:

```
<wlan_mac_dcf.c>
u32 frame_receive(u8 rx_pkt_buf, u8 rate, u16 length){
        // Check if AP
        if(wlan_addr_eq(eeprom_addr, ap_addr)){
```

```
//Check if a packet of type DATA is received
if (unicast_to_me && (mpdu_info->state ==
    RX_MPDU_STATE_FCS_GOOD) && ((rx_header->
    frame_control_1)==MAC_FRAME_CTRL1_SUBTYPE_DATA)){
        // Check source address of the received data
           packet
        if(wlan_addr_eq(rx_header->address_2, sta1_addr)
           )
                // Source is STA1 then increment by 1
                   flow1
                flow1 += 1;
        else if (wlan_addr_eq(rx_header->address_2,
           sta2_addr))
                // Source is STA2 then increment by 1
                   flow2
                flow2 += 1;
        //Check if there is a coding opportunity
        if ((flow1 > 0) && (flow2 > 0)){
                // A coding opportunity exists
                // Decrement flow1 and flow2 by 1
                flow1 -=1;
                flow2 -=1;
                //Create CODED
                txcoded_length = wlan_create_coded_frame
                   ((void*)(TX_PKT_BUF_TO_ADDR(
                   TX_PKT_BUF_AP) +

                   PHY_TX_PKT_BUF_MPDU_OFFSET), &
                   tx_header_coded,
                   MAC_FRAME_CTRL2_FLAG_FROM_DS,
                   n_dbps_coded);
                // Set LLC header of 8 bytes
                txcoded_length += sizeof(llc_header);
                // Add XOR header of 40 bytes and set
                   payload form the previous received
                   data packet and subtract mac and llc
                   headers already added before
                txcoded_length += length + 40 - ( sizeof
                   (llc_header) + sizeof(
                   mac_header_80211) );
                // Configure auto-responder for ACKDATA
                   Tx
```

```
                                wlan_phy_set_tx_signal(TX_PKT_BUF_AP,
                                    coded_rate , txcoded_length +
                                    WLAN_PHY_FCS_NBYTES);
                }
                else{
                        //No opportunity for transmitting a coded packet
                            , send an ACK packet
                }
        }
        // If STA
        else{
                //Check if a packet of type CODED is received
                if (unicast_to_me && (mpdu_info->state ==
                    RX_MPDU_STATE_FCS_GOOD) && ((rx_header->
                    frame_control_1)==MAC_FRAME_CTRL1_SUBTYPE_CODED)){
                        // Create ACK
                        tx_length = wlan_create_ack_frame((void*)(
                            TX_PKT_BUF_TO_ADDR(TX_PKT_BUF_ACK) +
                            PHY_TX_PKT_BUF_MPDU_OFFSET), rx_header->
                            address_2);
                        // Configure auto-responder for ACK Tx
                        wlan_phy_set_tx_signal(TX_PKT_BUF_ACK, tx_rate,
                            tx_length + WLAN_PHY_FCS_NBYTES);
                        // Process CODED as ACK
                        if((rx_header->frame_control_1) ==
                            MAC_FRAME_CTRL1_SUBTYPE_ACK  || (rx_header->
                            frame_control_1) ==
                            MAC_FRAME_CTRL1_SUBTYPE_CODED ){
                                return_value |= POLL_MAC_TYPE_ACK;
                        }
                        // Process CODED as DATA and send it to higher
                            level MAC
                        if((!WLAN_IS_CTRL_FRAME(rx_header)) || (
                            rx_header->frame_control_1 ==
                            MAC_FRAME_CTRL1_SUBTYPE_CODED)){
                                wlan_mac_low_frame_ipc_send();
                        }
                }
        }
}
```

Finally, in the wlan_mac_sta.c file, a condition is introduced inside the mpdu_rx_process function to account received CODED packets as DATA packets and update reception statistics. This is done as follows:

```
<wlan_mac_sta.c>
void mpdu_rx_process(void* pkt_buf_addr, u8 rate, u16 length) {
        if(((rx_80211_header->frame_control_1 & 0xF) ==
            MAC_FRAME_CTRL1_TYPE_DATA) || ((rx_80211_header->
            frame_control_1 & 0xF) == MAC_FRAME_CTRL1_SUBTYPE_CODED)  ) {
                (station_stats->data_num_rx_success)++;
                (station_stats->data_num_rx_bytes) += mpdu_info->length;
        }
}
```

5.5.2 Experimental Setup

An experiment framework called WARPnet [6] is used for the experimental evaluation of the DCF and BidMAC implementations. WARPnet is a Python-coded environment that allows performing real-time experiments with multiple WARP nodes through an experiment controller running on a host PC. Specifically, the WARPnet module implemented for the 802.11 reference design is called wlan_exp. This framework enables low-level visibility and control of MAC and PHY behaviors of the reference design in real-time.

The testbed used to perform the experiments with the wlan_exp module consists of two systems: wireless and wired (see Fig. 5.17). The wireless system implements an IEEE 802.11g WLAN composed of three WARP v3 nodes, an AP and STA 1 and STA 2, that are placed at 1-meter distance from each other, forming an equilateral triangle, in a zone free of wireless interferences. Each WARP v3 node is equipped with a single common Wi-Fi 2.4 GHz antenna and a 12 V power charger. The wired system, instead, implements a Gigabit Ethernet network that connects the WARP v3 nodes to a PC (i.e., the experiment controller) through a switch. The experiment controller launches custom-design Python scripts that exploit various features of the wlan_exp experiment framework. The scripts generate traffic

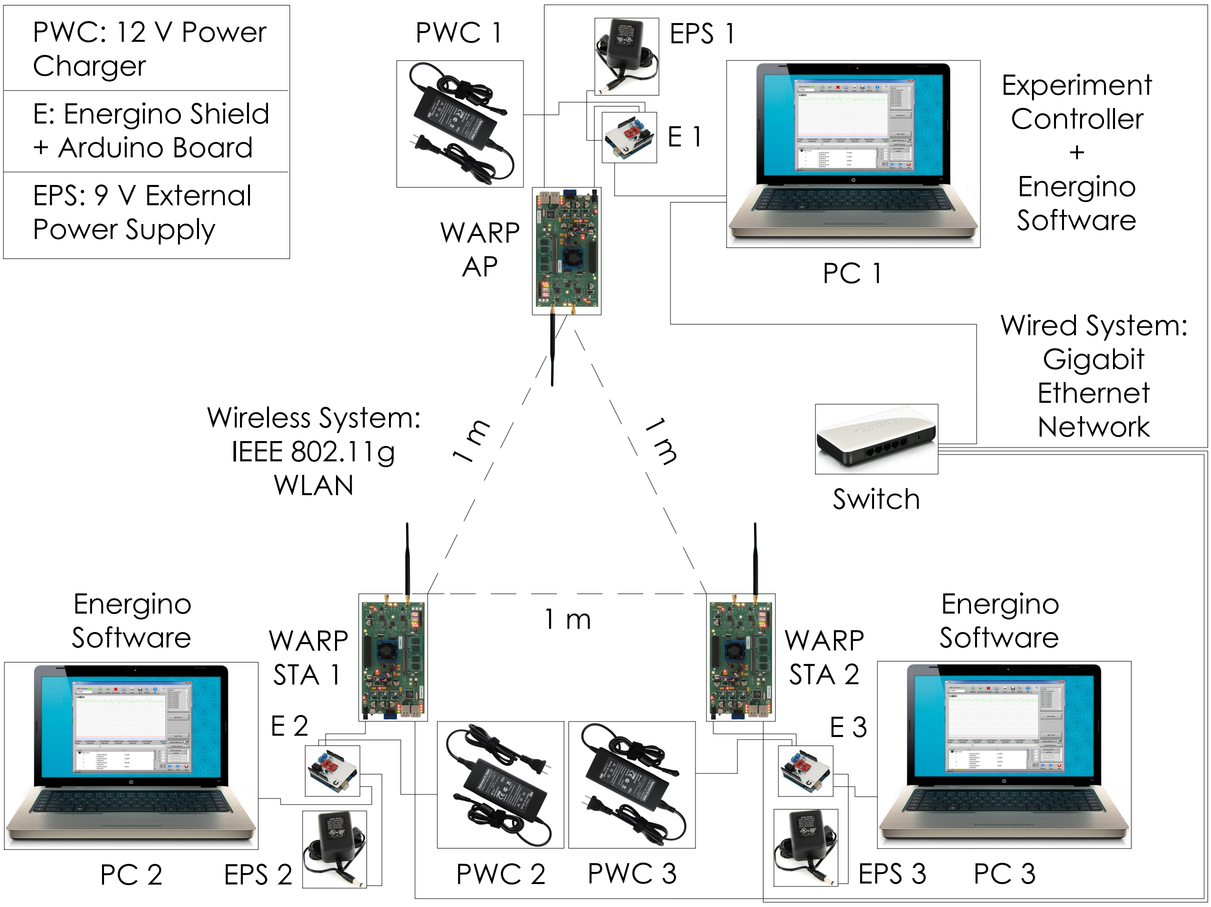


Figure 5.17: The testbed layout.

flows between the AP and the STAs through LTG implemented in the upper-level MAC code (see Appendix B) and calculate the throughput as the number of delivered bits of information over a given trial time, using Tx/Rx packet counts at each node.

Specifically, three different scripts have been developed:

1. throughput_traffic.py: This script generates bidirectional symmetric traffic flows of different periodic inter-packet arrival intervals (from long to short) between the AP and each STA with a constant data payload length (i.e., MSDU) of 1400 bytes and a fixed PHY data rate of 54 Mbps. Note that for BidCode only unidirectional data flows from each STA to the AP are configured, since the AP will

automatically generate a CODED packet (with an implicit ACK) for each STA in response to successful data reception when an coding opportunity exists, or simply respond with an ACK.

2. throughput_payload.py: This script varies the MSDU length from 50 to 1500 bytes with a 250-byte interval and considering zero inter-packet arrival interval (i.e., fill up the transmit queues to reach the saturation state) and a fixed PHY data rate of 54 Mbps.

3. throughput_rate.py: This script tunes the PHY data rate from 6 to 54 Mbps with zero inter-packet arrival interval and a constant MSDU length of 1500 bytes.

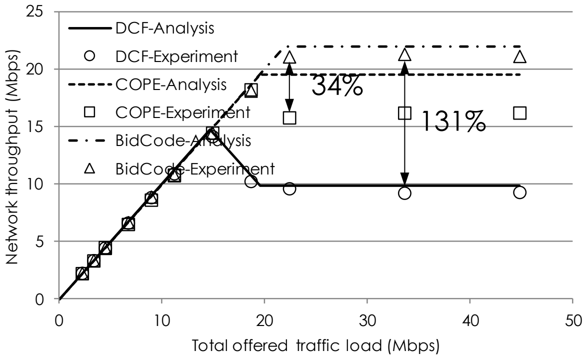
In all these scripts, the trial time for each experiment is set to 30 s and the throughput results are obtained as an average value of 10 repetitions per experiment.

In order to compute the energy efficiency results, the throughput results are divided by the power consumption data of the WARP v3 boards, gathered during the experiments from the Energino meters via custom-design software. Three Energino shields on top of Arduino UNO boards are built following the instructions given in [9] and redesigned in software to achieve sampling rates of 15 kHz. Each Energino shield is connected to the WARP v3 board's power supply and its power charger using the screw terminals. The Arduino UNO board assembled below each Energino shield is connected to a PC using the USB interface. Also, an additional external power source of 9 V is used to supply the Arduino UNO board (see Fig. 3.17 and Appendix C).

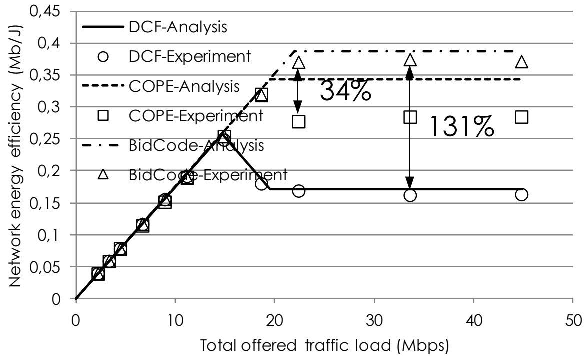
A custom program developed in LabVIEW is executed in each PC to control Energino and acquire samples of voltage, current, and power for each WARP v3 board during a selected period of time. This software allows averaging the samples values, for instance, the average value of power consumption measured in the WARP v3 boards when transmitting (P_t), receiving (P_r), and being idle (P_i) during the experiments is 18.95 W (each board). This value is used in the mathematical expressions derived in previous sections to obtain the theoretical energy efficiency results for the protocols analyzed. Also, note that the Energino meters start sampling 5 s before the beginning of a new experiment in order to gather the power consumption data exactly during the 30 s that each experiment takes.

5.5.3 Analytical and Experimental Results

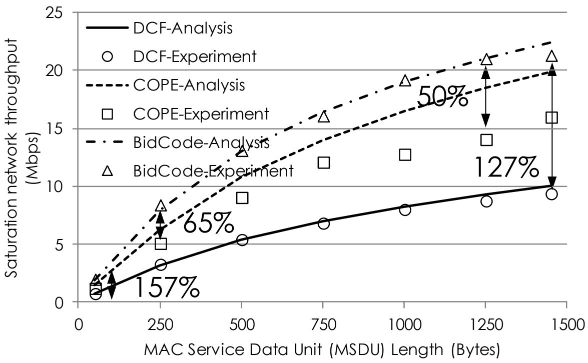
The results of throughput and energy efficiency obtained from the analysis and experiments described in the previous sections for the DCF, COPE, and BidCode protocols are presented and discussed as follows. They are summarized in Fig. 5.18. In general, it can be seen that in all the graphs the experimental results are in line with the analytical results for all the protocols. The differences between analytical and experimental results in DCF are due to channel errors and collisions that may occur during the experiments. On the contrary, in COPE the upper bounds obtained experimentally are significantly lower than those derived analytically. In addition to possible channel errors and collisions, the main reason for this variation is that COPE relies on coding opportunities and thus any asymmetry in the traffic flows received by the AP from STA 1 and STA 2 may have a significant impact on the performance of COPE. In addition, the capacity of the transmit queues also has a significant influence on its performance because coded packets are dropped when the network reaches the saturation state. Similarly, BidCode bounds are slightly lower than expected for similar reasons. However, BidCode is less affected by the network traffic dynamics and packet dropping due to the immediate access method implemented for XOR coded data packets through bidirectional transmissions.



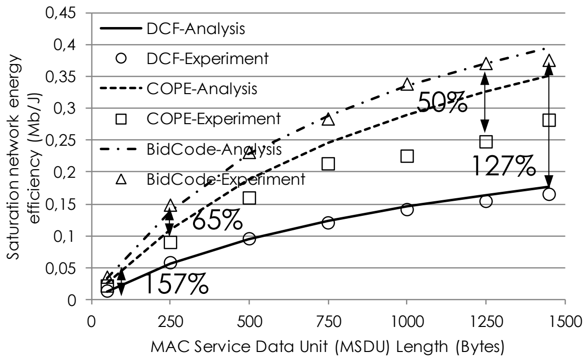
(a) Network throughput vs. Load



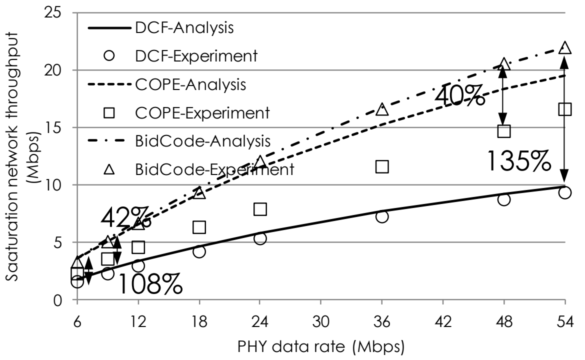
(b) Network energy efficiency vs. Load



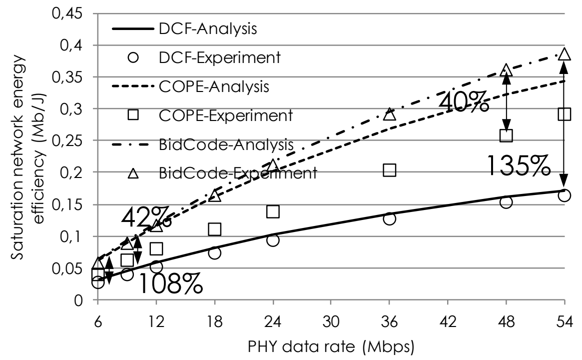
(c) Saturation network throughput vs. MSDU



(d) Saturation network energy efficiency vs. MSDU



(e) Saturation network throughput vs. Rate



(f) Saturation network energy efficiency vs. Rate

Figure 5.18: Experimental throughput and energy efficiency of the NC-aware MAC protocols versus the traffic load, MSDU length and PHY data rate

The network throughputs and energy efficiencies of the protocols versus the traffic load are shown in Figs. 5.18a and 5.18b, respectively. An MSDU length of 1400 bytes and a PHY data rate of 54 Mbps are considered. The performances of the protocols increase linearly as the traffic load increases until saturation. It can be seen that the performance of DCF shows a maximum value for a traffic load of 15 Mbps and then a lower stable under saturation for traffic loads above 20 Mbps. Also, COPE shows a maximum value for a traffic load of 18 Mbps and then decrease down to 22 Mbps under saturation. BidCode achieves the highest performance reaching saturation for traffic loads above 22 Mbps.The experimental maximum gains of BidCode versus DCF and COPE are 131% and 34%, respectively.

Figs. 5.18c and 5.18d show the saturation network throughputs and energy efficiencies of the protocols versus the MSDU length. A PHY data rate of 54 Mbps and MSDU lengths from 50 bytes to 1500 bytes with a 250-byte interval are considered. In general, the performances of the protocols increase as the MSDU length increases. The maximum gains of BidCode versus DCF and COPE are achieved for smaller packet lengths ranging from 157% to 127% and from 65% to 50%, respectively, as the packet length increases.

The throughputs and energy efficiencies of the protocols versus the PHY data rate are reported in Figs. 5.18e and 5.18f, respectively. An MSDU length of 1400 bytes and PHY data rates ranging from 6 to 54 Mbps are considered. In general, the performances of the protocols increase as the PHY data rate increases. The maximum gains of BidCode versus DCF are achieved for faster data rates ranging from 108% and 135%. The maximum gains of BidCode versus COPE are roughly stable for all PHY data rate with an average value of 40%.

5.6 Conclusions

BidCode and GreenCode have been presented in this chapter as new NC-aware energy-efficient distributed MAC protocols that have been designed to improve both the throughput and energy efficiency of wireless networks based on the IEEE 802.11 Standard. The basic idea behind BidCode is to allow the receiver of a valid data packet to perform an RD coded data transmission (with an implicit ACK) back to the transmitter without contending for the channel, as it would be the case in the standard DCF and COPE. Then, GreenCode exploits the longer duration of BidCode transmissions, which include both forward and reverse transmissions, to allow overhearing nodes to turn off their radio transceivers in order to save energy, taking into account the on/off radio transitions of nodes.

The closed expressions of the maximum achievable throughputs and energy efficiencies of DCF, COPE, BidMAC, BidCode, GreenBid, and GreenCode have been derived for two well-known scenarios, namely the Alice and Bob topology (i.e., two source nodes and a relay node) and cross scenario (i.e., four source nodes and a relay node). In addition, a generalized scenario for a finite number of source nodes around a relay node has been considered to obtain general formulas of the throughputs and energy efficiencies of the protocols under consideration. Then, a Python simulation environment where the protocol rules have been implemented has been developed for the validation of the proposed analytical model. The performances of the protocols have been evaluated in both Alice and Bob and cross scenarios considering relevant system parameters such as the traffic load, data payload length, data rate, wakeup radio transition coefficient, and awake/sleep radio transitions time. Both analytical and simulation results have shown the high performances of BidCode and GreenCode when compared to those of legacy DCF, reference COPE, BidMAC, and Green-

Bid for all evaluated parameters.

More specifically, the throughput gains vary from 352% to 277% as the packet length increases and from 289% to 244% as the data rate increases. The maximum energy efficiency gains range from 350% to 325% with increasing packet lengths and from 340% to 321% with increasing data rates. Furthermore, the results have shown the importance of taking into account the wakeup radio transitions in the energy efficiency analysis of NC-aware energy-efficient MAC protocols based on low-power states (i.e., GreenCode), since those transitions have a certain influence on the total energy consumption. In this sense, the energy efficiency gains vary between 337% and 278% as the wakeup radio transition coefficient increases. Similarly, the gains are between 361% and 285% as the awake/sleep radio transition time increases. These parameters will vary depending on the radio hardware design and are critical for the proper operation of GreenCode.

Finally, the reference COPE and proposed BidCode protocols have been implemented on WARP v3 platforms using a reference design that implements the DCF MAC and OFDM PHY from IEEE 802.11a/g. A testbed composed of three WARP v3 nodes where one acts as an AP (i.e., a relay node) and two as STAs (i.e., the source nodes) have been set up. To perform the experiments and gather the experimental results, several scripts that generate traffic flows between the AP and the STAs and calculate the throughput at each node have been developed. Also, Energino meters and a program developed in labVIEW to control Energino have been used to measure the energy consumption of the WARP v3 nodes and then calculate the energy efficiency. The experimental throughput and energy efficiency results of DCF, COPE, and BidCode have been shown versus the traffic load, the packet length, and the data rate. The maximum experimental gain of BidCode versus DCF at the network level is up to 157% and up to

63% versus COPE.

Therefore, this chapter has demonstrated through analysis, computer-based simulation, and real-life experimentation that the proposed NC-aware energy-efficient distributed MAC protocols can improve the throughput and energy efficiency of the legacy DCF and the reference COPE (or DCF with NC) in wireless networks based on the IEEE 802.11 Standard.

Chapter 6

Conclusions and Future Work

The main contributions and findings presented in this thesis are summarized in this chapter. More specifically, a summary of the thesis contents along with a brief description of the proposed solutions and a discussion of the most relevant results are presented in Section 6.1. Then, open lines of research related to the topics addressed in this thesis are described in Section 6.2.

6.1 Summary and Conclusions

This thesis has been aimed at contributing to the global goal of green ICT by improving the energy efficiency of wireless networks. The focus has been put on the design and performance analysis and evaluation of new energy-efficient MAC protocols for WLANs and NC-aware energy-efficient MAC protocols for wireless ad hoc networks. The thesis have been divided into a preliminary part and two main parts:

- A preliminary part composed of Chapters 1 and 2.
- A first main part comprised of Chapters 3 and 4.
- A second main part confined to Chapter 5.

Chapter 1 has been devoted to introduce the related topics, to review the state of the art at a high level, and to expose the motivations and main objectives of the thesis. Chapter 2 has been devoted to define the framework of the thesis, to discuss the main research challenges, and to comprehensively review the state of the art.

In the first part of Chapter 2 both the wireless network architectures and the wireless network protocol stack considered in this thesis have been described. Then, the main aspects that need to be considered at the MAC sublayer of the data link layer to achieve energy saving have been discussed and analyzed for the set of widely used MAC protocols of the IEEE 802.11 Standard. After that, the integration of an NC layer into the protocol stack between data link (MAC) and network (IP) layers for further energy savings have been described. In addition, the interactions between the NC layer and the MAC sublayer have been discussed and analyzed for the IEEE 802.11 MAC protocol. Finally, the most relevant energy-efficient MAC protocols and NC protocols available in the literature have been described and analyzed in the last part of the chapter.

The first main part of the thesis has focused on new energy-efficient distributed and centralized MAC protocols for WLANs based on the IEEE 802.11 Standard. In Chapter 3, BidMAC and GreenBid have been presented as new energy-efficient distributed MAC protocols that have been designed to improve both the throughput and energy efficiency of the contention-based distributed channel access method of the IEEE 802.11 Standard (DCF). In addition, Chapter 4 have presented BidPoll and GreenPoll as new energy-efficient polling-based MAC protocols that have been designed to improve both the throughput and energy efficiency of the polling-based centralized channel access method of the IEEE 802.11 Standard (PCF).

The basic idea behind BidMAC is to allow the receiver of a valid data packet to perform an RD (or bidirectional) transmission (with an implicit ACK) back to the transmitter (or to another receiver STA if the RD executor is the AP) without contending for the channel, as it would be the case in the standard DCF. Then, GreenBid exploits the longer duration of BidMAC transmissions, which include both forward and reverse transmissions, to allow overhearing STAs to enter a low-power sleep state where their radio transceivers are turned off to save energy. This operation takes into account the time and power consumption of the on/off radio transitions of STAs, since depending on the available time for sleeping (i.e. the total transmission time) and the radio transitions time it may not be possible for a third STA to go to sleep.

The basic idea behind BidPoll is to split a CFP into two virtual phases. The first phase is reserved for low-overhead uplink and downlink transmissions between the AP and the STAs that requested TXOPs in the previous CFP. The second phase is used for dynamic (possibly low-overhead) data exchanges between the AP and the rest of STAs that did not take part in the first phase. The first phase of BidPoll is deterministic (i.e., the duration is announced through beacons) and thus BidPoll can significantly reduce the overhead of poll and ACK packets introduced by the CFP. Then, GreenPoll, which extends the BidPoll operation, exploits the duration information of the first phase to allow the STAs involved in this phase to enter the sleep state from the time instants at which they receive the ACK packets to their transmitted data packets until the end of the first phase. In addition, those STAs not involved in the first phase can also enter the sleep state until this phase completes.

The closed expressions of the maximum achievable throughputs and energy efficiencies of DCF, BidMAC, and GreenBid and PCF, BidPoll, and GreenPoll have been derived in Chapters 3 and 4, respectively. Also, a

Python simulation environment where the protocol rules have been implemented has been developed for the validation of the proposed analytical models. The performances of the protocols have been evaluated in a WLAN composed of an AP and 20 STAs considering relevant system parameters such as the traffic load, data payload length, data rate, number of STAs in the network, wakeup radio transition coefficient, and awake/sleep radio transitions time.

Both analytical and simulation results have shown the high performances of BidMAC and GreenBid and BidPoll and GreenPoll, respectively, when compared to those of the DCF and PCF for all evaluated parameters. In Chapter 3, the throughput gain of GreenBid versus DCF at the network level is up to 60% whereas the maximum energy efficiency gain of GreenBid versus DCF is 360%. Similarly, Chapter 4 shows that the throughput gains of BidPoll versus DCF and PCF are up to 195% and 49%, respectively, whereas the maximum energy efficiency gains of GreenPoll versus DCF and PCF are 338% and 148%, respectively.

Furthermore, the results have shown the importance of taking into account the on/off radio transitions in the energy efficiency analysis of energy-efficient MAC protocols based on low-power states (i.e., GreenBid and GreenPoll). These transitions are particularly critical for GreenBid since they represent the 70% of the total energy consumption of the network. In contrast, for GreenPoll they are only the 10% of the total energy consumption of the network. The reason is that in GreenBid the total available time for sleeping (i.e., forward and reverse transmissions) is not significantly long with respect to the on/off radio transition times. On the contrary, in GreenPoll the available time for sleeping during the first phase of a CFP includes multiple data exchanges between the AP and the STAs, thus being significantly longer than the on/off radio transition times.

In order to evaluate the performances of the proposed MAC protocols in a real environment, BidMAC has been implemented on WARP v3 platforms using a reference design that implements the DCF MAC and OFDM PHY from IEEE 802.11a/g. A testbed composed of three WARP v3 nodes where one acts as an AP and two as STAs have been set up. To perform the experiments and gather the experimental results, several scripts that generate traffic flows between the AP and the STAs and calculate the throughput at each node have been developed. Also, Energino meters and a program developed in LabVIEW to control Energino have been used to measure the energy consumption of the WARP v3 nodes and then calculate the energy efficiency. The maximum experimental gain of BidMAC versus DCF at the network level is above 60% whereas the maximum experimental gain from the AP perspective is around 100% with minimum impact on the average per-STA performance.

Finally, the second part of the thesis has turned the focus to new NC-aware MAC protocols for wireless ad hoc networks based on the IEEE 802 Standard. Therefore, BidCode and GreenCode have been presented in Chapter 5 as new NC-aware energy-efficient distributed MAC protocols that have been designed to improve both the throughput and energy efficiency of the IEEE 802.11 DCF MAC protocol. BidCode is an extension of BidMAC with NC whereas GreenCode is an extension of GreenBid with NC for wireless ad hoc networks. Similar to BidMAC, the basic idea behind BidCode is to allow the receiver of a valid data packet to perform an RD coded data transmission (with an implicit ACK) back to the transmitter without contending for the channel, as it would be the case in the standard DCF and COPE (DCF+NC). Then, GreenCode exploits the longer duration of BidCode transmissions, which include both forward and reverse transmissions, to allow overhearing nodes to turn off their radio transceivers, in a way similar to GreenBid.

The closed expressions of the maximum achievable throughputs and energy efficiencies of DCF, COPE, BidMAC, BidCode, GreenBid, and GreenCode have been derived for two well-known scenarios, namely the Alice and Bob topology (i.e., two source nodes and a relay node) and cross topology (i.e., four source nodes and a relay node). In addition, a generalized scenario for a finite number of source nodes around a relay node has been considered to obtain general formulas of the throughputs and energy efficiencies of the protocols under consideration. Then, a Python simulation environment where the protocol rules have been implemented has been developed for the validation of the proposed analytical models. The performances of the protocols have been evaluated in both Alice and Bob and cross scenarios considering relevant system parameters such as the traffic load, data payload length, data rate, wakeup radio transition coefficient, and awake/sleep radio transitions time.

Both analytical and simulation results have shown the high performances of BidCode and GreenCode when compared to those of legacy DCF, reference COPE, BidMAC, and GreenBid for all evaluated parameters. The maximum gains of BidCode and GreenCode are achieved for the cross scenario. The throughput gains of BidCode versus DCF and COPE are up to 352% and 127%, respectively, whereas the maximum energy efficiency gains of GreenCode versus DCF and COPE are 350% and 126%, respectively.

In order to evaluate the performances of the proposed MAC protocols in a real environment, COPE and BidCode have been implemented on WARP v3 platforms using the IEEE 802.11 reference design, as described earlier for BidMAC. A testbed composed of three WARP v3 nodes where one acts as a rely node and two as source nodes (i.e., the Alice and Bob scenario) have been set up. The experiments have been performed as explained earlier for BidMAC. The maximum experimental gain of BidCode versus

DCF at the network level is up to 157% and up to 63% versus COPE.

To conclude, this thesis has demonstrated through analysis, computer-based simulation, and real-life experimentation that the proposed energy-efficient MAC protocols and the proposed NC-aware energy-efficient MAC protocols can improve the throughput and energy efficiency of the legacy mechanisms in wireless networks based on the IEEE 802.11 Standard. As mentioned at the beginning of this chapter, the wide range of concepts and ideas involved in this thesis leaves interesting open challenges. The most remarkable future lines of research are outlined in the next section.

6.2 Future Work

This thesis has aimed at contributing to the greening evolution of wireless networks not only with the main contributions summarized in the previous section, but also by giving the floor to many open topics that have not been covered in this thesis but they have been identified through the course of the thesis.

Regarding the first part of the thesis, the main open lines of research are:

- The theoretical analyses of BidMAC, BidPoll, GreenBid, and GreenPoll have been developed considering the saturated network state, where all wireless devices always have data to transmit, an ideal channel, no hidden terminals, and Poisson (best-effort) traffic. Therefore, the development of more advanced analytical models considering non-saturated network states, error prone channels, hidden terminals, and other classes of traffic (e.g., voice, video, or machine-to-machine) would provide a better knowledge of the performances of the proposed MAC protocols.

- Related to the previous point, the simulation results of the MAC protocols have been obtained through a Python simulation environment where the MAC protocol rules have been implemented with an ideal PHY layer that provides error-free packets. Therefore, the implementation of the MAC protocols in more sophisticated simulators (such as, OPNET or network simulator version 2/3) would allow comprehensive performance evaluations in more realistic scenarios.

- An experimental implementation of BidMAC has been carried out in a programmable wireless platform called WARP v3 and tested in a proof-of-concept network composed of a WARP-AP and two WARP-STAs. The proposed implementation could be improved and the experimental evaluation could also consider different traffic classes. Similarly, BidPoll, GreenBid, and GreenPoll could be implemented on WARP to evaluate the performances of these MAC protocols in real-life environments.

- The design of the proposed MAC protocols could be optimized to support batch transmissions (i.e., a sequence of data packets in each transmission) and frame aggregation and to integrate QoS based on EDCA and HCCA of the IEEE 802.11e. For example, the current BidMAC and GreenBid designs only allow the exchange of two data packets in each bidirectional transmission involving a pair of sender and receiver. Thus, BidMAC and GreenBid could be extended to support multiple rounds of bidirectional transmissions and bidirectional frame aggregation between source and destination or involving multiple receivers. Note that other possible extensions have been included in the descriptions of these MAC protocols (the reader may refer to the specific chapter to know more details about them).

- GreenBid and GreenPoll have shown outstanding gains for medium to high loads. Hence, another possible line of research would study the performance of these MAC protocols in combination with other power saving mechanisms that work optimally for low loads (e.g., IEEE 802.11 PSM or IEEE 802.11e APSD).

- Despite the high performances of the proposed MAC protocols, the IEEE 802.11 Standard has shown such a market penetration that it is hardly realistic to believe that already deployed wireless equipment can be drastically replaced by a new technology, no matter the higher performance it attains. Therefore, another open line of research would assess the feasibility of the compatibility and coexistence of the new MAC protocols with legacy users.

- Robustness against attackers through the basic idea behind the proposed MAC protocols of sending after receiving could be studied and new mechanisms could be proposed to control the influence of malicious attackers. Basically, if there is a malicious STA continuously sending fake data with a non-standard CW size, the AP can detect this by maintaining a record of received data from each STA in the network (i.e., a fairness indicator). Then, the AP can use one of the received packets from the malicious STA to initiate a controlled access period where it can send data and grant transmission to those STAs that could not transmit due to channel capture of the malicious STA.

- The combination of GreenBid and GreenPoll and evaluation of the combined approach compared to the combination of the DCF and PCF could be another interesting work to be undertaken in order to see how these MAC protocols work together and how much they are able to improve the coexistence of DCF and PCF.

- In line with the previous idea, the application of the hybrid channel access mechanism to wireless ad hoc networks, smart gird networks, or machine-to-machine networks could be other open research topics.

- Finally, a model of the on/off transitions of radio transceivers (preferably based on experimental measurements) would be very useful to understand the impact of these transitions on the energy efficiency of novel MAC protocols based on low-power states (i.e., duty cycling). Based on this, it would be possible to derive a delay and energy consumption model of a generic duty-cycled MAC protocol considering the influence of the on/off radio transitions.

Regarding the second part of the thesis, most of the previous ideas mentioned for the proposed energy-efficient MAC protocols could be applied to the proposed NC-aware energy-efficient MAC protocols. In addition, other possible lines of research are:

- The analyses and performance evaluations of BidCode and GreenCode have considered two simple topologies, Alice and Bob topology (i.e., a simple chain topology) and cross topology. Thus, further analysis and performance evaluations need to be carried out in more complex topologies where there may be more than one relay node performing NC operations and the proposed mechanisms may need some refinements to work well in these scenarios.

- The basic idea behind BidCode is to allow relay nodes to combine several received packets and immediately forward them upon successful reception of data, hence granting an immediate channel access (maximum channel access priority). However, in the presence of several relay nodes performing NC operations (in line with the previous point), different channel access priorities should be assigned to the nodes based on the level of useful data for the network in each node. Thus, one possible line of research would focus on the design of a new mechanism that combines the IEEE 802.11e EDCA in conjunction with NC, where AIFS, TXOP duration, and CW sizes can be adjusted based on NC information at each node.

- Finally, GreenCode has been designed to allow STAs to enter a low-power state when they are overhearing data transmissions. This idea has not be widely investigated in the literature. Therefore, the design of new mechanisms that optimally combine power saving strategies through low-power periods and NC could be a very interesting line of research.

List of Acronyms

3GPP 3rd Generation Partnership Project

AC Access Category

ACK Acknowledgment

AIFS Arbitration Interframe Space

AP Access Point

APSD Automatic Power Save Delivery

ATIM Announcement Traffic Indication Message

BDCF Bidirectional Distributed Coordination Function

BEB Binary Exponential Backoff

BidCode Bidirectional NC-aware MAC protocol

BidMAC Bidirectional MAC protocol

BidPoll Bidirectional Polling MAC protocol

BS Base Station

BSA Basic Service Area

BSS Basic Service Set

CAP Controlled Access Phase

CDMA Code Division Multiple Access

CE CFP End

CFP Contention Free Period

CP Contention Period

CPU Central Processing Unit

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CTS Clear-To-Send

CW Contention Window

DCF Distributed Coordination Function

DIFS DCF Interframe Space

DPCF Distributed Point Coordination Function

DPSM Dynamic Power Saving Mechanism

DTIM Delivery Traffic Indication Map

EC-MAC Energy Conservation MAC protocol

EDA Energy-efficient Distributed Access

EDCA Enhanced Distributed Channel Access

EE-MultiPoll Energy-Efficient Multi-Polling

EIFS Extended Interframe Space

EOSP End of Service Period

ERP Extended Rate Physical

FCS Frame Check Sequence

FDMA Frequency Division Multiple Access

FPGA Field Programmable Gate Array

GreenBid Green Bidirectional MAC protocol

GreenCode Green NC-aware MAC protocol

GreenPoll Green Polling MAC protocol

HCF Hybrid Coordination Function

HCCA HCF Controlled Channel Access

CAP Controlled Access Phase

CPPA Coded Packet Priority Access

ICT Information and Communication Technologies

IEEE Institute of Electrical and Electronics Engineers

IFS Interframe Space

IP Internet Protocol

IPSM Improved Power Save Mode

ISO International Organization for Standardization

LabVIEW Laboratory Virtual Instrumentation Engineering Workbench

LEPOHA Low Energy Priority Oriented Hybrid Access

LLC Logical Link Control

LTE Long Term Evolution

LTG Local Traffic Generator

MAC Medium Access Control

MIMO Multiple-Input Multiple-Output

MD More Data

MORE Multi-path Opportunistic Routing Engine

MPDU MAC Protocol Data Unit

MSDU MAC Service Data Unit

MU-TXOP Multi-User Transmission Opportunity

NA-PSM Neighborhood Aware Power Save Mode

NAV Network Allocation Vector

NC Network Coding

NCAPP Network Coding-Aware MAC level Packet Prioritization

NCAPQ Network Coding-Aware Priority Queuing

NCQAM Network Coding-Aware Queue Management

NDBPS Number of Data Bits Per OFDM Symbol

OFDM Orthogonal Frequency Division Multiplexing

OFDMA Orthogonal Frequency Division Multiplexing Access

OSI Open Systems Interconnection

PAS Popularity Aware Scheduling

PC Personal Computer

PCF Point Coordination Function

PDU Protocol Data Unit

PHY Physical

PIFS PCF Interframe Space

PLCP PHY Layer Convergence Protocol

PPDU PLCP Protocol Data Unit

PS Power Save

PSDU PLCP Service Data Unit

PSM Power Save Mode

PSMP Power Save Multi-Poll

QoS Quality of Service

RD Reverse Direction

RDP Reverse Direction Protocol

RF Radio Frequency

RIFS Reduced Interframe Space

RTS Request-To-Send

S-APSD Scheduled Automatic Power Save Delivery

SDU Service Data Unit

SIFS Short Interframe Space

SISO Single-Input Single-Output

SP Service Period

STA Wireless Station

STPM Self-Tuning Power Management

TC Traffic Category

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

TIM Traffic Indication Map

TPC Transmission Power Control

TS Traffic Stream

TSPEC Traffic Specification

TXOP PSM Transmission Opportunity Power Save Mode

TXOP Transmission Opportunity

U-APSD Unscheduled Automatic Power Save Delivery

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunications System

UPCF Unified Point Coordination Function

USB Universal serial Bus

VoIP Voice over Internet Protocol

WARP Wireless Open-Access Research Platform

Wi-Fi Wireless Fidelity

WLAN Wireless Local Area Network

Bibliography

[1] Smart 2020: Enabling the low carbon economy in the information age, "The Climate Group, Global e-Sustainability Initiative (GeSI)," 2008, http://www.smart2020.org/_assets/files/02_Smart2020Report.pdf.

[2] G. Perrucci, F. Fitzek, and J. Widmer, "Survey on Energy Consumption Entities on the Smartphone Platform," in *IEEE VTC*, 2011, pp. 1–6.

[3] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, 1997.

[4] P. J. M. Havinga and G. J. M. Smit, "Energy-efficient TDMA medium access control protocol scheduling," in *AMOC*, 2000, pp. 1–10.

[5] E.-S. Jung and N. H. Vaidya, "An Energy Efficient MAC Protocol for Wireless LANs," in *IEEE INFOCOM*, vol. 3, 2002, pp. 1756–1764.

[6] WARP Project. http://warpproject.org.

[7] Mango Communications. http://mangocomm.com.

[8] K. M. Gomez, R. Riggio, T. Rasheed, D. Miorandi, and F. Granelli, "Energino: A Hardware and Software Solution for Energy Consumption Monitoring," in *IEEE WiOpt*, 2012, pp. 311–317.

[9] Energino Project. http://energino-project.org.

[10] Climate change: Commission welcomes final adoption of Europe's climate and energy package, "European Commission press release," Dec. 2008, http://europa.eu/rapid/press-release_IP-09-628_en.htm.

[11] B. Gammage et al., "Gartner's Top Predictions for IT Organizations and Users, 2010 and Beyond: A New Balance," *Gartner Report*, 2009, http://www.gartner.com/id=1268513.

[12] Vodafone Group, "Sustainability report: Environmental footprint - Performance data," 2012, http://www.vodafone.com/content/index/about/sustainability/sustainability_report/issue_by_issue/environmental_footprint/performance_data.html.

[13] Wireless Broadband Alliance and Informa Telecoms and Media, "WBA Industry Report 2011: Global Developments in Public Wi-Fi," 2011, http://www.wballiance.com/wba/wp-content/uploads/downloads/2012/07/16_WBA-Industry-Report-2011-_Global-Developments-in-Public-Wi-Fi-1.00.pdf.

[14] Centre for Energy-Efficient Telecommunications (CEET), Bell Labs, and University of Melbourne, "The Power of Wireless Cloud," 2013, http://www.wballiance.com/wba/wp-content/uploads/downloads/2012/07/16_WBA-Industry-Report-2011-_Global-Developments-in-Public-Wi-Fi-1.00.pdf.

[15] Wireless Broadband Alliance and Informa Telecoms and Media, "WBA Industry Report 2011: Global Developments in Public Wi-Fi - Key Findings," 2011, http://www.slideshare.net/CiscoMobilityCommunity/cisco-wbawifireportinfographic.

[16] IEEE, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE 802.11 Std., 2012.

[17] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," in *ACM SIGCOMM*, 2006, pp. 243–254.

[19] S.-L. Tsao and C.-H. Huang, "Review: A Survey of Energy Efficient MAC Protocols for IEEE 802.11 WLAN," *ACM Computer Communications*, vol. 34, no. 1, pp. 54–67, 2011.

[20] E. Charfi, L. Chaari, and L. Kamoun, "PHY/MAC Enhancements and QoS Mechanisms for Very High Throughput WLANs: A Survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 1714–1735, 2013.

[21] J.-P. Ebert, S. Aier, G. Kofahl, A. Becker, B. Burns, and A. Wolisz, "Measurement and simulation of the energy consumption of a WLAN interface," Telecommunication Networks Group, Technische Universitet Berlin, Tech. Rep., 2002.

[22] S.-L. Tsao and E.-C. Cheng, "PIANO: A power saving strategy for cellular/VoWLAN dual-mode mobiles," *ACM Wireless Networks*, vol. 14, no. 5, pp. 683–698, 2008.

[23] Z. S. Narmawala and S. Srivastava, "Survey of Applications of Network Coding in Wired and Wireless Networks," in *JTG NCC*, 2008, pp. 153–157.

[24] T. Matsuda and T. T. Noguchi, "Survey of Network Coding and its Applications," *IEICE Transactions on Communications*, vol. E94-B, no. 3, pp. 698–717, 2011.

[25] Manish Anand and Edmund B. Nightingale and Jason Flinn, "Self-Tuning Wireless Network Power Management," *Wireless Networks*, vol. 11, no. 4, pp. 451–469, 2005.

[26] A. Paramanathan, P. Pahlevani, D. E. Lucani, and F. H. P. Fitzek, "On the Need of Novel Medium Access Control Schemes for Network Coding enabled Wireless Mesh Networks," in *IEEE ICC*, 2013, pp. 1–6.

[27] F. V. Gallego, J. Alonso-Zarante, C. Verikoukis, and L. Alonso, "A Survey on Prototyping Platforms for the Development and Experimental Evaluation of Medium Access Control Protocols," *IEEE Wireless Communications*, vol. 19, no. 1, pp. 74–81, 2012.

[28] C. Hunter, J. Camp, P. Murphy, A. Sabharwal, and C. Dick, "A Flexible Framework for Wireless Medium Access Protocols," in *IEEE ACSSC*, 2006, pp. 2046–2050.

[29] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A Survey of Energy Efficient Network Protocols for Wireless Networks," *Wirel. Netw.*, vol. 7, no. 4, pp. 343–358, 2001.

[30] N. K. Ray and A. K. Turuk, "A review on energy efficient MAC protocols for Wireless LANs," in *IEEE ICIIS*, 2009, pp. 137–142.

[31] Z.-T. Chou, C.-C. Hsu, and S.-N. Hsu, "UPCF: A New Point Coordination Function with QoS and Power Management for Multimedia over Wireless LANs," *IEEE/ACM Trans. on Net.*, vol. 14, no. 4, pp. 807–820, 2006.

[32] C. Fragouli, J. Widmer, and J. le Boudec, "Efficient Broadcasting Using Network Codings," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 450–463, 2008.

[33] J. Goseling, R. Matsumoto, T. Uyematsu, and J. H. Weber, "On the Energy Benefit of Network Coding for Wireless Multiple Unicast," in *IEEE ICIT*, 2009, pp. 2567–2571.

[34] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," in *ACM SIGCOMM*, 2007, pp. 169–180.

[35] A. Ganz and A. Phonphoem, "Robust SuperPoll with Chaining Protocol for IEEE 802.11 Wireless LANs in Support of Multimedia Applications," *Wirel. Netw.*, vol. 7, no. 1, pp. 65–73, 2001.

[36] K.-C. Ting, F.-C. Kuo, B.-J. Hwang, H. Wang, and C.-C. Tseng, "A Power-Saving and Robust Point Coordination Function for the Transmission of VoIP over 802.11," in *IEEE ISPA*, 2010, pp. 283–289.

[37] A. Kanjanavapastit and B. Landfeldt, "An analysis of a modified point coordination function in ieee 802.11," in *IEE PIMRC*, vol. 2, 2003, pp. 1732–1736.

[38] C. Crespo, J.Alonso-Zarate, L. Alonso, and C. Verikoukis, "Distributed Point Coordination Function for Wireless Ad hoc Networks," in *IEEE VTC*, 2009, pp. 1–5.

[39] J. Alonso-Zarate, C. Crespo, C. Skianis, L. Alonso, and C. Verikoukis, "Distributed Point Coordination Function for IEEE 802.11 Wireless Ad Hoc Networks," *Ad Hoc Networks*, vol. 10, no. 3, pp. 536–551, 2012.

[40] F. Talucci, M. Gerla, and L. Fratta, "MACA-BI (MACA By Invitation)- A Receiver Oriented Access Protocol for Wireless Multihop Networks," in *IEEE PIMRC*, vol. 2, 1997, pp. 435–439.

[41] Y. Wang and J. Garcia-Luna-Aceves, "A Hybrid Collision Avoidance Scheme for Ad Hoc Networks," *Wireless Networks*, vol. 10, no. 4, pp. 439–446, 2004.

[42] H. Zhai, J. Wang, and Y. Fang, "Distributed Packet Scheduling for Multihop Flows in Ad Hoc Networks," in *IEEE WCNC*, vol. 2, 2004, pp. 1081–1086.

[43] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement," in *IEEE INFOCOM*, vol. 2, 2002, pp. 599–607.

[44] D.-H. Kwon, W.-J. Kim, and Y.-J. Suh, "A Bidirectional Data Transfer Protocol for Capacity and Throughput Enhancements in Multi-rate Wireless LANs," in *IEEE VTC*, vol. 4, 2004, pp. 3055–3059.

[45] W. Choi, J. Han, B. J. Park, and J. Hong, "BCTMA (Bi-directional Cut-Through Medium Access) Protocol for 802.11-based Multi-hop Wireless Networks," in *ACM ISSADS*, 2005, pp. 377–387.

[46] M. Ozdemir, G. Daqing, A. B. McDonald, and J. Zhang, "Enhancing MAC Performance with a Reverse Direction Protocol for High-Capacity Wireless LANs," in *IEEE VTC*, 2006, pp. 1–5.

[47] N. S. P. Nandiraju, H. Gossain, D. Cavalcanti, K. R. Chowdhury, and D. P. Agrawal, "Achieving Fairness in Wireless LANs by Enhanced IEEE 802.11 DCF," in *IEEE WiMob*, 2006, pp. 132–139.

[48] D. Qiao, S. Choi, A. Jain, and K. G. Shin, "MiSer: An Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h," in *ACM MobiCom*, vol. 2, 2003, pp. 1210–1214.

[49] Y. Z. S. Yan and S. Wu, “An Adaptive RTS Threshold Adjust Algorithm Based on Minimum Energy Consumption in IEEE 802.11 DCF,” in *IEEE ICCT*, vol. 2, 2003, pp. 1210–1214.

[50] S. Pollin, R. Mangharam, B. Bougard, L. V. der Perre, I. Moerman, and F. C. R. Rajkumar, “MEERA: Cross-layer Methodology for Energy Efficient Resource Allocation in Wireless Networks,” *IEEE Transactions on Wireless Communications*, vol. 6, no. 2, pp. 617–628, 2007.

[51] L. Chen and J. Leneutre, “A Game Theoretic Framework of Distributed Power and Rate Control in IEEE 802.11 WLANs,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 7, pp. 1128–1137, 2008.

[52] J.-R. Hsieh, T.-H. Lee, and Y.-W. Kuo, “Energy-Efficient Multi-Polling Scheme for Wireless LANs,” *IEEE Trans. on Wireless Communications*, vol. 8, no. 3, pp. 1532–1541, 2009.

[53] V. Baiamonte and C.-F. Chiasserini, “Saving energy during channel contention in 802.11 WLANs,” *Mobile Networks and Applications*, vol. 11, no. 2, pp. 287–296, 2006.

[54] R. Bruno, M. Conti, and E. Gregori, “Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless LANs,” *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 10–31, 2002.

[55] K. Hong, S. Lee, K. Kim, and Y. Kim, “Channel Condition Based Contention Window Adaptation in IEEE 802.11 WLANs,” *IEEE Transactions on Communications*, vol. 60, no. 2, pp. 469–478, 2012.

[56] J. A. Stine and G. D. Veciana, "Improving Energy Efficiency of Centrally Controlled Wireless Data Networks," *ACM Wireless Networks*, vol. 8, no. 6, pp. 681–700, 2002.

[57] F. Zhu and Z. Niu, "Priority based power saving mode in WLAN," in *ACM MobiCom*, 2008, pp. 1–6.

[58] Y. hua Zhu and V. C. Leung, "Efficient Power Management for Infrastructure IEEE 802.11 WLANs," *IEEE Transactions on Wireless Communications*, vol. 9, no. 7, pp. 2196–2205, 2010.

[59] Q. Pang, S. Liew, and V. Leung, "Performance Improvement of 802.11 Wireless Network with TCP ACK Agent and Auto-Zoom Backoff Algorithm," in *IEEE VTC*, vol. 3, 2005, pp. 2046–2050.

[60] M. C. Rosu, C. M. Olsen, C. Narayanaswami, and L. Luo, "PAWP: A Power Aware Web Proxy for Wireless LAN," in *IEEE WMCSA*, vol. 3, 2004, pp. 206–215.

[61] V. Namboodiri and L. Gao, "Energy-efficient VoIP over wireless LANs," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 566–581, 2010.

[62] K. Sivalingam, "Design and Analysis of Low-Power Access Protocols for Wireless and Mobile ATM Networks," *ACM/Baltzer Journal on Wireless Networks Special issue on Advances in Wireless Systems*, vol. 6, no. 1, pp. 73–87, 2000.

[63] Y.-C. Tseng, C.-S. Hsu, , and T.-Y. Hsieh, "Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks," *Computers Networks: The International Journal of Computer and Telecommunications Networking*, vol. 43, no. 3, pp. 317–337, 2003.

[64] T. Lagkas, G. Papadimitriou, P. Nicopolitidis, and A. Pomportsis, "Novel Medium Control for Wireless Networks: Providing Total QoS and Energy Conservation," in *IEEE MELECON*, 2008, pp. 228–233.

[65] A. Belghith and W. Akkari, "Power Saving Mechanisms for Ad Hoc Networks Based on Handshaking Information Tapping," in *International Workshop on Verification and Evaluation of Computer*, 2008, pp. 1–11.

[66] A. Yahya, "Energy-Aware Architecture for Multi-Rate Ad Hoc Networks," *Egyptian Informatics Journal*, vol. 11, no. 1, pp. 22–38, 2010.

[67] E.-S. Jung and N. H. Vaidya, "Improving IEEE 802.11 Power Saving Mechanism," *Wireless Networks*, vol. 14, no. 3, pp. 375–391, 2008.

[68] D. Umehara, S. Denno, M. Morikura, and T. Sugiyama, "Coded Packet Immediate Access for Contention-based Wireless Relay Networks," in *IEEE ICSPCS*, 2010, pp. 1–9.

[69] F. Zhao and M. Medard, "On Analyzing and Improving COPE Performance," in *IEEE ITA*, 2010, pp. 1–6.

[70] H. Seferoglu and A. Markopoulou, "Network Coding-Aware Queue Management for Unicast Flows over Coded Wireless Networks," in *IEEE NetCod*, 2010, pp. 1–6.

[71] H. Seferoglu, A. Markopoulou, and M. Medard, "NCAPQ: Network Coding-Aware Priority Queueing for UDP Flows over COPE," in *IEEE NetCod*, 2011, pp. 1–8.

[72] F. Xie, L. Du, Y. Bai, and L. Chen, "Popularity Aware Scheduling for Network Coding based Content Distribution in Ad Hoc Networks," in *IEEE PIMRC*, 2007, pp. 1–5.

[73] C.-M. Cheng, H. T. Kung, C.-K. Lin, C.-Y. Su, and D. Vlah, "Rainbow: A Wireless Medium Access Control Using Network Coding for Multi-hop Content Distribution," in *IEEE MILCOM*, 2008, pp. 1–10.

[74] D. Umehara, C. Huang, S. Denno, M. Morikura, and T. Sugiyama, "Enhancement of IEEE 802.11 and Network Coding for Single-Relay Multi-User Wireless Networks," in *IEEE ICSPCS*, 2010, pp. 1–9.

[75] N. De Coppi, J. Ning, G. Papageorgiou, M. Zorzi, S. Krishnamurthy, and T. La Porta, "Network Coding Aware Queue Management in Multi-Rate Wireless Networks," in *IEEE ICCCN*, 2012, pp. 1–7.

[76] R. Chandanala and R. Stoleru, "Network Coding in Duty-Cycled Sensor Networks," in *IEEE INSS*, 2010, pp. 203–210.

[77] Socket Mobile CF Wireless LAN Card, Technical Specifications. http://www.socketmobile.com/pdf/wireless-lan/cf-wlan_ug.pdf.

[78] D. Akhmetov, "802.11n: Performance Results of Reverse Direction Data Flow," in *IEEE PIMRC*, 2006, pp. 1–3.

[79] R. Palacios, F. Granelli, D. Gajic, and A. Foglar, "An Energy-Efficient MAC Protocol for Infrastructure WLAN Based on Modified PCF/DCF Access Schemes Using a Bidirectional Data Packet Exchange," in *IEEE CAMAD*, 2012, pp. 216–220.

[80] R. Palacios, F. Granelli, D. Kliazovich, L. Alonso, and J. Alonso-Zarate, "Energy Efficiency of an Enhanced DCF Access Method Using Bidirectional Communications for Infrastructure-based IEEE 802.11 WLANs," in *IEEE CAMAD*, 2013, pp. 38–42.

[81] ——, "An Energy Efficient Distributed Coordination Function Using Bidirectional Transmissions and Sleep Periods for IEEE 802.11 WLANs," in *IEEE GLOBECOM*, 2013, pp. 1641–1647.

[82] R. Palacios, E. M. B. Larbaa, J. Alonso-Zarate, and F. Granelli, "Performance Analysis of Energy-Efficient MAC Protocols using Bidirectional Transmissions and Sleep Periods in IEEE 802.11 WLANs," in *IEEE GLOBECOM*, 2014, pp. 1269–1275.

[83] R. Palacios, F. Franch, F. Vazquez-Gallego, J. Alonso-Zarate, and F. Granelli, "Experimental Evaluation of Reverse Direction Transmissions in WLAN Using the WARP Platform," in *IEEE ICC*, 2015, submitted - under review.

[84] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[85] G. Bianchi and I. Tinnirello, "Remarks on IEEE 802.11 DCF Performance Analysis," *IEEE Communications Letters*, vol. 9, no. 8, pp. 765–767, 2005.

[86] R. Palacios, F. Granelli, D. Gajic, C. Liß, and D. Kliazovich, "An Energy-Efficient Point Coordination Function Using Bidirectional Transmissions of Fixed Duration for Infrastructure IEEE 802.11 WLANs," in *IEEE ICC*, 2013, pp. 1036–1041.

[87] R. Palacios, G. M. Mekonnen, J. Alonso-Zarate, D. Kliazovich, and F. Granelli, "Analysis of an Energy-Efficient MAC Protocol Based on Polling for IEEE 802.11 WLANs," in *IEEE ICC*, 2015, submitted - under review.

[88] R. Palacios, J. Heide, F. H. P. Fitzek, and F. Granelli, "Design and Performance Evaluation of Underwater Data Dissemination Strategies using Interference Avoidance and Network Coding," in *IEEE ICC*, 2012, pp. 1410–1415.

[89] R. Palacios, F. Granelli, A. Paramanathan, J. Heide, and F. H. P. Fitzek, "Coding-aware MAC: Providing Channel Access Priority for Network Coding with Reverse Direction DCF for IEEE 802.11-based Wireless Networks," in *IEEE ICC*, 2014, pp. 1272–1277.

[90] R. Palacios, H. Haile, J. Alonso-Zarate, and F. Granelli, "Analysis of a Network Coding-Aware MAC Protocol for IEEE 802.11 Wireless Networks with Reverse Direction Transmissions," in *IEEE GLOBECOM*, 2014, pp. 1230–1236.

[91] R. Palacios, D. Kliazovich, and F. Granelli, "Reverse Direction Transmissions and Network Coding for Energy-Efficient Wi-Fi Networks," in *IEEE CAMAD*, 2014, pp. 1–5.

[92] F. Franch, R. Palacios, and F. Granelli, "Performance Evaluation of Energy Efficient and Network Coding-Aware MAC Protocols for IEEE 802.11 Wireless Networking Using the WARP Experimental Platform," Master's thesis, DISI, University of Trento, 2014.

Appendices

Appendix A

Wireless Open-Access Research Platform (WARP)

WARP is a high-performance programmable wireless platform to implement PHY, MAC, and network layer protocols. It was originally developed by Rice University within the WARP Project [6] and is currently manufactured and distributed by Mango Communications [7]. The latest generation of WARP hardware is WARP v3 (see Fig. A.1). This is an FPGA board with the following hardware features.

- Xilinx Virtex-6 FGPA with an embedded PowerPC processor
- 2 programmable RF interfaces each with:
  - 2.4/5GHz transceiver (40MHz RF bandwidth)
  - 12-bit 170MSps DACs, 12-bit 100MSps ADCs
  - Dual-band (20 dBm Tx power)
  - Share clocking for MIMO applications
- FMC HPC expansion slot
- 2 gigabit Ethernet interfaces
- DDR3 SO-DIMM slot

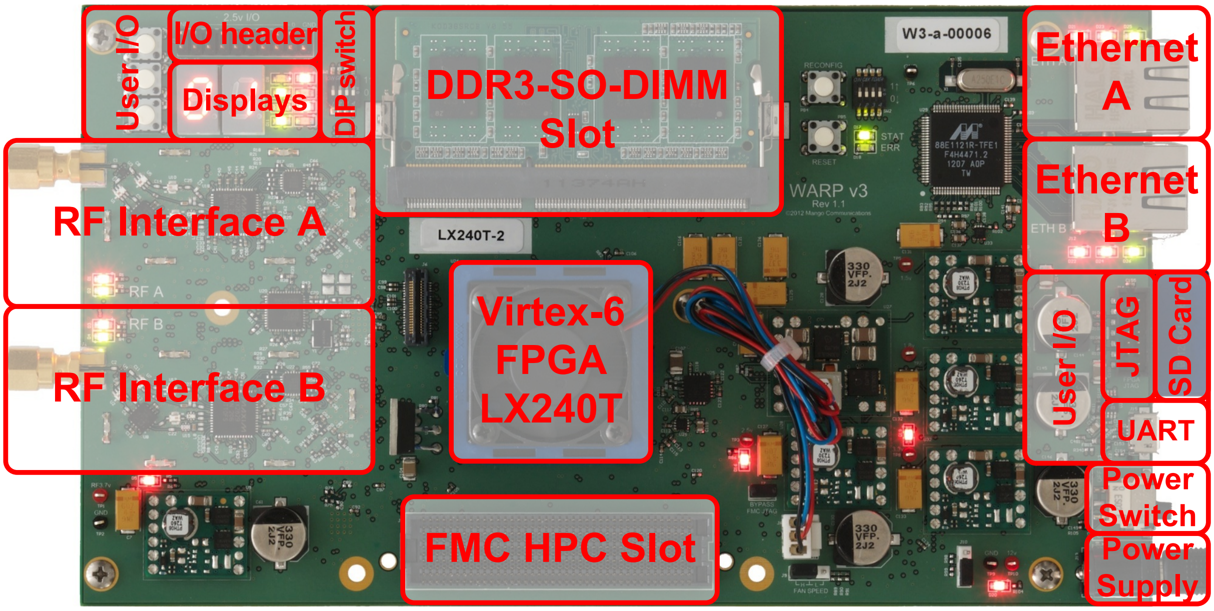


Figure A.1: WARP v3 and its hardware features [6, 7].

- FPGA config via JTAG, SD card or flash
- User I/O:
  - USB-UART
  - 12 LEDs
  - 2 seven-segment displays
  - 4 push buttons
  - 4-bit DIP switch
  - 16-bit 2.5v I/O header

Further information about WARP v3 can be found in [6, 7].

Appendix B

Mango 802.11 Reference Design for WARP v3 Hardware

The WARP Project provides an open-source repository of C-coded reference designs and support materials for the WARP hardware. In particular, the Mango 802.11 reference design is a real-time FPGA implementation of the DCF MAC and OFDM PHY from IEEE 802.11a/g for the WARP v3 hardware, which can operate as an AP or an STA. In the design, PHY processing is performed by the PHY Tx/Rx cores, or CPUs whereas MAC functions are mainly implemented in software running in two MicroBlaze CPUs with an intermediate core interfacing to the PHY Tx/Rx cores and a support core to achieve accurate inter-packet timing.

The overall architecture of this reference design is illustrated in Fig. B.1. The following FPGA cores can be found.

- **CPU High** executes the top-level MAC code (AP/STA implementations) and other high-level functions, such as construction of all non-control packets for transmission and for performing the association handshakes. It also integrates wired and wireless communications by implementing encapsulation and de-encapsulation of Ethernet packets.

- **CPU Low** executes the low-level code for the DCF MAC, which deals with all MAC/PHY interactions and low-level MAC functions, such as transmission of ACK packets (RTS/CTS is not implemented), scheduling of backoffs, maintaining the CW size, and initiating retransmissions.

- **MAC DCF** is an FPGA core that interfaces between the MAC software design and the Tx/Rx PHY cores. It implements the timers required for the DCF (timeout, backoff, DIFS, SIFS, etc.) and the various carrier sensing mechanisms (NAV reset timeout is not implemented).

- **PHY Tx/Rx** includes peripheral cores that implement the OFDM PHY layer transceiver.

- **Hardware Support** includes support cores for WARP v3 that allow control of the various peripheral interfaces on WARP v3 from the code in CPU Low.

Focusing on the MAC layer of the design, the MAC software implementation is split into two pieces: the upper-level MAC and the lower-level MAC, which communicate with each other via inter-processor mailbox. The upper-level MAC code contains the AP/STA implementations (wlan_mac_ap.c and wlan_mac_sta.c) and a collection of their shared inter-packet behaviors that are not time critical, referred to as MAC High Framework. The interactions between the different upper-level MAC implementations and this framework may involve notification of wired/wireless reception and command of wired/wireless transmission. Also, the framework provides an LTG module to generate data packets of arbitrary length up to 1500 bytes (LTG Payload) at periodic or uniform random intervals (LTG Schedule). Note that LTG data packets include an LLC header to avoid that non-WARP devices, e.g. laptops and smartphones, can process them.

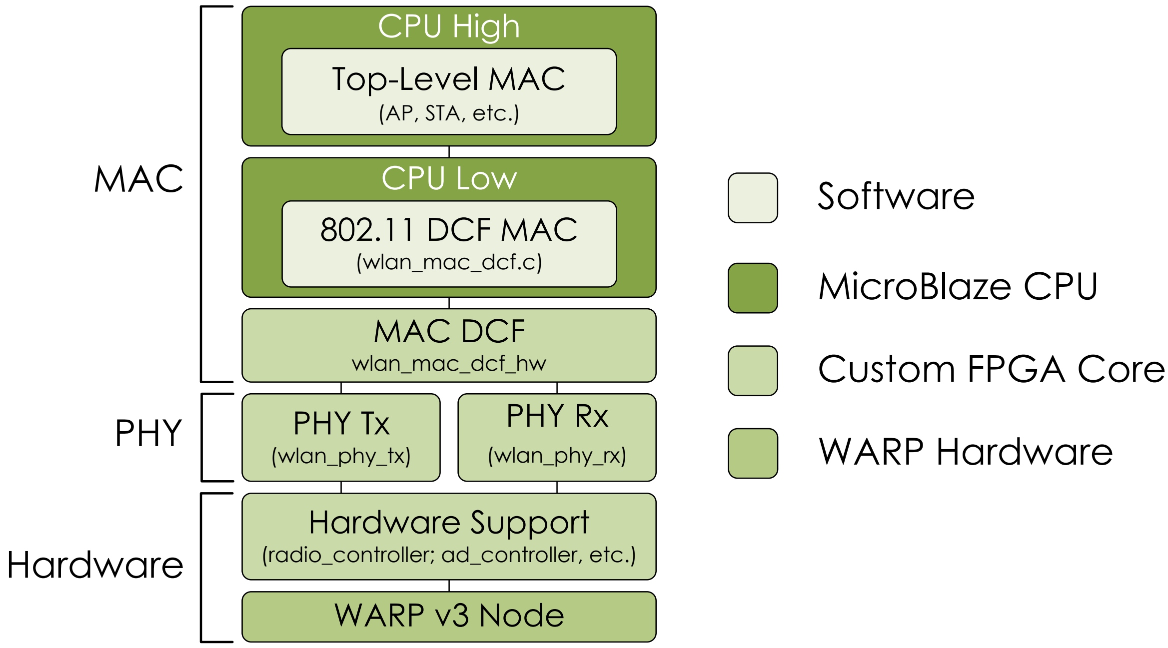


Figure B.1: Architecture of the IEEE 802.11 reference design [6, 7].

On the other hand, the lower-level MAC code (wlan_mac_dcf.c) handles intra-packet states that are time critical for the DCF via the MAC DCF core (wlan_mac_dcf_hw) in order to perform wireless transmission and reception. This core directly connects to the Tx/Rx PHY control and status signals and implements the timers and state machines required to meet the IEEE 802.11 channel access timing requirements. For instance, in this core a small state machine, called Auto Tx or auto-responder, that initiates a PHY transmission in response to a valid PHY reception is integrated to enable transmission of ACK packets immediately after a SIFS.

Further information about the Mango 802.11 reference design of WARP v3 is available in [6, 7].

Appendix C

Energino: A Hardware and Software Energy Consumption Monitoring Solution

Energino [8] is an Arduino-based energy consumption monitoring platform, designed and developed by the iNSPIRE group at CREATE-NET within the Energino Project [9], that provides real-time precise energy consumption statistics for any DC appliance. The main features of Energino are

- Arduino-based, a flexible platform with a very active community
- High sampling rate, up to 10000 voltage or current samples per second
- High resolution, configurable from 26mA down to 1mA

The main building blocks of Energino are

- A voltage sensor implemented using a voltage divider
- A current sensor based on the Hall effect
- A management module implemented using a mechanical relay

The specific features of Energino are

- Supported voltage (for the DCF load): 0 - 60 V
- Supported current (for the DCF load): 0 - 3 A
- Sampling rate: Max. 10000 samples per second
- Voltage sampling resolution: 60 mV
- Current sampling resolution: From 26 mA down to 0.5 mA

In order to assemble an Energino shield the following electronic components are needed (see Fig. C.1):

- Energino PCB from Fritzing
- ACS712 Low Current Sensor Breakout
- Resistors 10K x2
- Resistor 1K
- Resistor 100K
- Diode 1N4001
- Transistor NPN 2N3904
- Screw terminals (2 Pin) x2
- Relay Omron G6E-134PL-ST-US
- Arduino Stackable Header - 8 Pin x3
- Arduino Stackable Header - 6 Pin x1

Three Energino shields on top of Arduino UNO boards are built following the instructions given in [9] and redesigned in software to achieve sampling rates of 15 kHz.

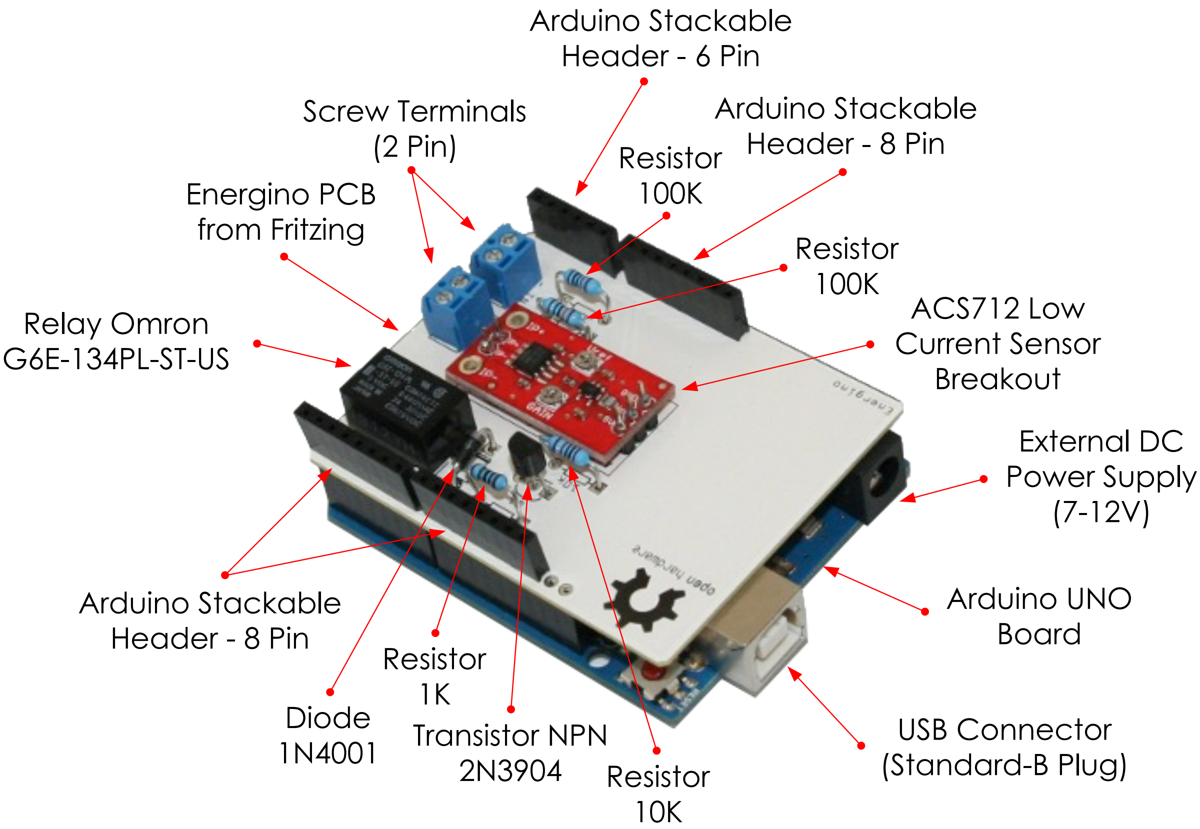


Figure C.1: Energino shield on top of the Arduino UNO board [8, 9].

A custom program has been developed in LabVIEW to control Energino and acquire samples of voltage (V), current (I), and power (P) for each WARP v3 board during a selected period of time. This software provides an easy-to-use visual interface (see Fig. C.2) and also allows averaging the samples values and calibrating the voltage and current sensors of Energino. More details about how to use the custom LabVIEW program are provided in [92].

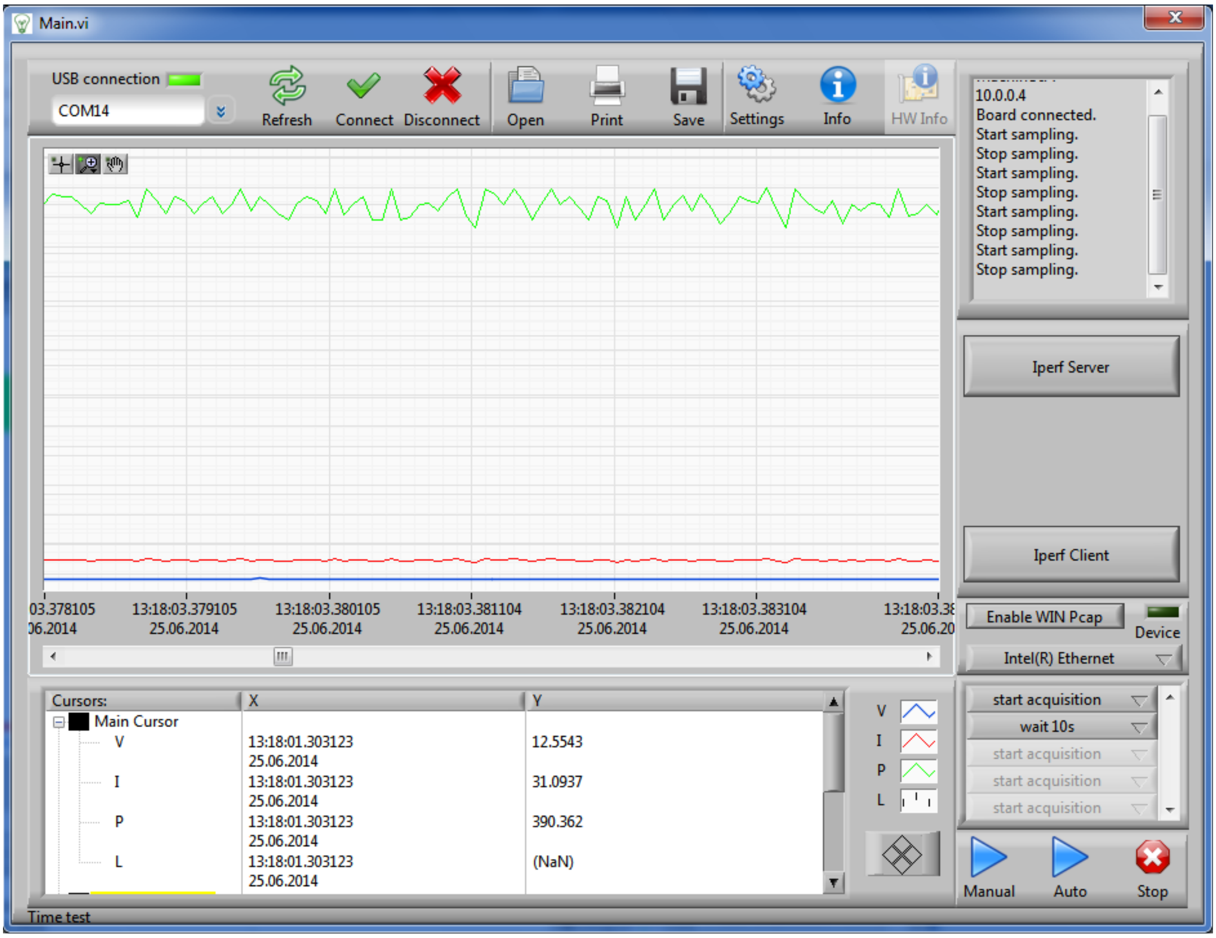


Figure C.2: Visual interface of the custom-design software in LabVIEW to control Energino.