IEEE 802.11 Networks: MAC Protocols for Heterogeneous Multi-Antenna Scenarios and Software-Defined Radio PHY Layer Implementation

Settore scientifico-disciplinare ING-INF/03 TELECOMUNICAZIONI

Ph.D. Candidate
ALJOŠA DORNI

Ph.D. Program Coordinator
Prof. WALTER UKOVICH

Thesis Supervisor
Prof. FULVIO BABICH

Tutor
Dr. MASSIMILIANO COMISSO

Co-Tutor
Dr. GIAMMARCO ZACHEO

ANNO ACCADEMICO 2011/2012
Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Fulvio Babich that has given me the opportunity of working in his group.

I would like to thank my parents, Jožica and Silvan, and my sister, Katja, for their continuous support during my Ph.D. experience. I would also thank all my relatives.

I would like to express my appreciation to my tutor and colleague Massimiliano Comisso, who has assisted me during my Ph.D. studies. My sincere thanks also to my tutor in the Laboratory in Vienna, Giammarco Zacheo, for his support during my period at FTW. Both of you have helped me a lot.

I would like to say: “Thanks, Dudes!” to Alessandro, my Ph.D. colleague, to Marco, Marco, and all the other guys from the Microwave Laboratory. It was a pleasure working with you, and I always had a good time.

Thanks also to the guys from FTW: Mirko (my roommate), Dragan, Giuseppe, Beppe, Pierdomenico, Pasquale, Pierfrancesco, Danilo, Alessandro, Rosa, Alfonso and the other guys of the area N, under the supervision of Prof. Fabio Ricciato. “Thank you!” for the great experience in Vienna.

I am pretty sure that I am going to forget somebody. So, a warm and sincere thanks also to all my friends for the time spent together and to all the people involved in any way into my life: Hvala vsem skupaj!!

And finally, a lovely “Hvala puči!! Smack!!” to the sun of my heart Alenka, who is always by my side.
Summary

The objective of this thesis is to discuss the performance achieved by IEEE 802.11 networks, considering in detail their simulation and experimental analysis, as well as the implementation aspects. The original contribution of this dissertation involves three main research fields within the context of distributed wireless networks: the experimental and theoretical analysis of IEEE 802.11e networks in presence of quality of service mechanisms, the development and the simulation of backward compatible medium access control protocols in presence of smart antenna systems, and, finally, the implementation of the IEEE 802.11ag physical layer on software-defined radio platforms.

The material presented in this dissertation is the result of a three-year study period at the Telecommunication Group of the Department of Engineering and Architecture of the University of Trieste during the Doctorate in Information Engineering. Few months of the Ph.D. studies have been carried out at the Telecommunications Research Center in Vienna (FTW).

The availability of wireless connections is essential not only to cover large distances, but also for providing last-mile and broadband connectivity. Wireless communications are preferably adopted instead of wired connections, since the latter are typically difficult to deploy and/or can be too expensive in some scenarios. Furthermore, wireless networks are used to provide services for mobile devices, in centralized and distributed environments. With the increasing demand for multimedia contents, wireless communications must now guarantee a higher network performance. The research community is focusing a lot of efforts for defining and developing access protocols for wireless networks. Thus, in last decades some of the designed protocols have been officially accepted as standards by the Institute of Electrical and Electronics Engineers (IEEE), the European Telecommunications Standards Institute (ETSI), the International Organization for Standardization (ISO), and other organizations. Additionally, associations between several companies have adopted proprietary protocols in order to cope with particular requirements dictated by the market. Nowadays, the available wireless protocols differ in their application field. A classification may be based on the maximum obtainable throughput, the range extension, the power consumption, and the deployment costs. In fact, all the available wireless technologies are not suitable for being used in all environments. Concerning this point, the Global System for Mobile communications (GSM), the Universal Mobile Telecommunications System (UMTS), and the Long Term Evolution (LTE) standards
are adopted to provide telephony and connectivity to mobile users. By contrast, the IEEE 802.15.4, the ZigBee, and the Wireless Highway Addressable Remote Transducer (WirelessHART), are mostly used in industry and healthcare applications, where low consumption, low throughput and short ranges are the main characteristic aspects. One of the most popular family of wireless standards is the IEEE 802.11x one. This family collects many amendments, where each single extension deals with different capabilities and targets different goals. In particular, the IEEE 802.11abgn standards are at the moment implemented on commercial devices, personal computers, smartphones, and tablets, distributed all around the world for providing a data rate sufficient for allowing Internet connectivity in middle range areas and inside buildings. The IEEE 802.11abg nodes are developed to work typically with single omnidirectional antennas. Omnidirectional antennas are replaced by directional ones, when radiation pattern directivity is required for a specific purpose, i.e. long distance links, area sectorization, etc. With the rise of IEEE 802.11n wireless devices may be equipped with two or more antennas for transmitting and receiving multiple signal replicas in order to increase the connection reliability and/or the network throughput. In fact, the IEEE 802.11n adopts Multiple Input Multiple Output (MIMO) solutions to exploit diversity and spatial multiplexing. However, this multi-antenna technology is usually adopted for the coordinating node in centralized networks, while the wireless clients typically remain equipped with omnidirectional antennas.

In distributed scenarios, IEEE 802.11 nodes equipped with omnidirectional antennas are able to access the channel one at a time. In fact, the IEEE 802.11 protocol performs power sensing to assess whether the channel is idle or not. However, this approach, based on a single wireless communication, may be improved by enabling the network to sustain multiple communications at the same time. As mentioned before, omnidirectional antennas may be replaced by antennas able to provide directional radiation patterns. In this sense, directional radiation patterns enable spatial reuse of the frequency spectrum, thus enabling the coexistence of several simultaneous communications within the same area. Instead of adopting directional antennas, whose radiation pattern is fixed, using an advanced antenna system offers an electrically tunable radiation pattern, whose main lobe may be steered towards the direction of the desired destination. The synthesis of the radiation pattern can be performed by using beamforming algorithms that operate in the signal processing unit of the antenna system itself.

The first topic addressed in this thesis concerns the comparison between the analytical and experimental results obtained for the prioritized access adopted in the IEEE 802.11e amendment. Additionally, the deployed experimental setup is used as a not invasive system for testing the reliability of commercial wireless devices.

The design and the simulation of two novel medium access protocols for nodes
equipped with advanced antenna systems represent the second topic, and the main contribution of this dissertation. The proposed solutions provide a performance increase and are able to guarantee a fair coexistence between the nodes with omnidirectional antennas and those with advanced antenna systems. The two developed protocols are the result of the analysis of the problems that can appear at the Open Systems Interconnection (OSI) layer 2 when directional antennas are used. A design strategy is presented in order to reduce the system drawbacks, such as hidden terminal and deafness, that may lead to a performance decrease. Two simulation tools are used to assess the performance of the proposed protocols, accounting for advanced antenna systems and realistic propagation models. The network environment in which the developed access schemes operate is an asynchronous heterogeneous network. The term asynchronous reflects the fact that the analyzed network is completely distributed. In fact, since the communications are not coordinated by a central unit, the nodes can perform the transmission attempts in different time instants. Besides, the considered network is heterogeneous, since different nodes can be equipped with different antenna systems. The presented work focuses on the development of channel access protocols using advanced antenna systems to increase the network performance, while maintaining backward compatibility with the nodes relying on omnidirectional antennas.

The third topic of this dissertation focuses on the software implementation of a complete wireless transceiver. The implementation is carried out on software-defined radio platforms fulfilling real-time constraints. This latter contribution steps into the software-defined radio concept for developing and testing protocols suitable for future wireless networks.

The thesis is organized in two parts. The first part provides the background material and introduces the topics that will be of interest for the research contribution described in the second part. The second part is dedicated to the original results obtained during the Ph.D. studies. With reference to the first part, the first chapter includes an extended description of the IEEE 802.11 medium access control and physical layers, considering the IEEE 802.11e extension for the support of quality of service mechanisms and the IEEE 802.11ag transmit and receive chains. Besides, the first chapter identifies the propagation environment, describing the fundamentals of smart antenna systems, and providing a literature overview on the proposed 802.11 extensions using adaptive arrays. The second chapter focuses on the software, drivers and applications adopted in this thesis. In particular, this chapter describes the software tools used to simulate the network performance, and those employed to obtain experimental network measurements. Additionally, this chapter provides a description of the software-defined radio concept, discussing an optimization technique adopted for parallelizing the data processing in order to respect real-time con-
The second part presents the original results and consists of four chapters. The third chapter presents a comparison between the measurements obtained by a deployed experimental setup and the results obtained by a developed analytical model for 802.11e networks. In addition, a theoretical analysis, which is performed on the intervals between consequent transmissions, is compared to the experimental investigation of the inter-transmission intervals, in order to develop a method able to discover if the backoff distribution implemented in a given 802.11 wireless card is uniformly distributed or not. A statistical comparison between analysis and measurements is performed, since some manufacturers, differently from what stated by the 802.11 standard, adopt non-uniform backoff distribution in their 802.11 devices. The fourth chapter describes two network simulation platforms, one obtained by integrating MATLAB with Network Simulator 2 (ns2) and the second one developed integrating Octave with ns2. Both platforms are extensions of the open source network simulator ns2, and allow considering in detail the channel-antenna characteristics of each node in order to properly simulate 802.11 heterogeneous networks in the presence of smart antenna systems. The fifth chapter deals with the support of multiple communications in a wireless network, which are enabled by the use of smart antenna systems. In this chapter two backward compatible medium access protocols enabling multi-packet communication are presented and simulated. The sixth chapter describes the implementation of the IEEE 802.11ag transmit and receive chains on a software-defined radio platform, with the aim of achieving a real-time decoding performance. Finally, the seventh chapter summarizes the thesis contributions and the most relevant conclusions.

The purpose of this work is to provide a numerical and experimental assessment of IEEE 802.11 distributed networks, focusing, at medium access control sublayer, on the quality of service and the throughput requirements, and, at physical layer, on the protocols’ implementation.
Contents

Summary v
List of acronyms xvii
List of symbols xxiii

I Background 1

1 Distributed wireless networks: fundamentals 3
   1.1 General aspects of DWNs ........................................... 3
   1.2 IEEE 802.11: basic MAC layer characteristics .................... 5
       1.2.1 Architecture .................................................... 5
       1.2.2 IEEE 802.11 access mechanisms .............................. 5
           1.2.2.1 Basic access ............................................ 6
           1.2.2.2 RTS/CTS access ......................................... 7
       1.2.3 IEEE 802.11e - QoS extension .................................. 8
           1.2.3.1 Enhanced distributed channel access ..................... 8
   1.3 MAC layer extensions in the presence of advanced antennas systems 10
       1.3.1 Propagation environment ....................................... 10
           1.3.1.1 Path-loss attenuation ................................... 11
           1.3.1.2 Fading ................................................... 12
           1.3.1.3 Spatial channel model .................................. 13
       1.3.2 General aspects of multi-antenna systems in low-rank environment 14
           1.3.2.1 Physical antenna system: array geometry ................. 16
               1.3.2.1.1 Linear array ..................................... 16
               1.3.2.1.2 Circular array .................................. 17
               1.3.2.1.3 Square array .................................. 17
               1.3.2.1.4 Concentric ring array ............................ 18
           1.3.2.2 Signal processing unit: beamforming and direction of arrival estimation algorithms 19
               1.3.2.2.1 Temporal reference techniques ..................... 20
               1.3.2.2.2 Spatial reference techniques ..................... 22
       1.3.3 MAC layer issues in presence of advanced antenna systems .......... 24
<table>
<thead>
<tr>
<th>Section</th>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.3.1</td>
<td>Typical problems</td>
<td>24</td>
</tr>
<tr>
<td>1.3.3.2</td>
<td>Literature overview</td>
<td>25</td>
</tr>
<tr>
<td>1.4</td>
<td>The 802.11 OFDM physical layer</td>
<td>27</td>
</tr>
<tr>
<td>1.4.1</td>
<td>PHY frame structure</td>
<td>29</td>
</tr>
<tr>
<td>1.4.2</td>
<td>Transmit chain</td>
<td>29</td>
</tr>
<tr>
<td>1.4.2.1</td>
<td>Cyclic redundancy check calculator</td>
<td>30</td>
</tr>
<tr>
<td>1.4.2.2</td>
<td>SERVICE field, tail bits and pad bits</td>
<td>30</td>
</tr>
<tr>
<td>1.4.2.3</td>
<td>Scrambler</td>
<td>31</td>
</tr>
<tr>
<td>1.4.2.4</td>
<td>Convolutional encoder</td>
<td>31</td>
</tr>
<tr>
<td>1.4.2.5</td>
<td>Puncturer</td>
<td>32</td>
</tr>
<tr>
<td>1.4.2.6</td>
<td>Interleaver</td>
<td>32</td>
</tr>
<tr>
<td>1.4.2.7</td>
<td>Symbol mapper</td>
<td>33</td>
</tr>
<tr>
<td>1.4.2.8</td>
<td>Pilot insertion</td>
<td>34</td>
</tr>
<tr>
<td>1.4.2.9</td>
<td>Inverse FFT</td>
<td>36</td>
</tr>
<tr>
<td>1.4.2.10</td>
<td>Cyclic prefix insertion</td>
<td>36</td>
</tr>
<tr>
<td>1.4.2.11</td>
<td>Preamble and SIGNAL field insertion</td>
<td>37</td>
</tr>
<tr>
<td>1.4.3</td>
<td>Receive chain</td>
<td>38</td>
</tr>
<tr>
<td>1.4.3.1</td>
<td>Frame detector</td>
<td>38</td>
</tr>
<tr>
<td>1.4.3.2</td>
<td>Coarse carrier frequency offset estimator and compensator</td>
<td>39</td>
</tr>
<tr>
<td>1.4.3.3</td>
<td>Symbol timing</td>
<td>40</td>
</tr>
<tr>
<td>1.4.3.4</td>
<td>Fine carrier frequency offset estimator and compensator</td>
<td>40</td>
</tr>
<tr>
<td>1.4.3.5</td>
<td>Cyclic prefix remover, FFT, block equalizer and comb equalizer</td>
<td>40</td>
</tr>
<tr>
<td>1.4.3.6</td>
<td>Symbol demapper</td>
<td>41</td>
</tr>
<tr>
<td>1.4.3.7</td>
<td>Deinterleaver and depuncturer</td>
<td>41</td>
</tr>
<tr>
<td>1.4.3.8</td>
<td>Viterbi decoder</td>
<td>41</td>
</tr>
<tr>
<td>1.4.3.9</td>
<td>Descrambler</td>
<td>42</td>
</tr>
<tr>
<td>1.4.3.10</td>
<td>CRC checker</td>
<td>42</td>
</tr>
<tr>
<td>2</td>
<td>Adopted software packages</td>
<td>45</td>
</tr>
<tr>
<td>2.1</td>
<td>Network simulations</td>
<td>45</td>
</tr>
<tr>
<td>2.2</td>
<td>Network measurements</td>
<td>48</td>
</tr>
<tr>
<td>2.2.1</td>
<td>MADWiFi driver</td>
<td>49</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Iperf</td>
<td>50</td>
</tr>
<tr>
<td>2.2.3</td>
<td>TCPDump</td>
<td>50</td>
</tr>
<tr>
<td>2.3</td>
<td>Software-defined radio</td>
<td>51</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Software-defined radio concept</td>
<td>51</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Ettus Research USRP N210</td>
<td>52</td>
</tr>
<tr>
<td>2.3.3</td>
<td>SIMD instruction set</td>
<td>52</td>
</tr>
</tbody>
</table>
II Original Results

3 Backoff uniformity and throughput measurements for IEEE 802.11e networks
  3.1 Introduction ................................................. 61
  3.2 Experimental setup .......................................... 63
  3.3 MADWiFi driver modification ................................. 64
  3.4 Theoretical analysis for the backoff uniformity ............. 66
  3.5 Throughput analysis for IEEE 802.11e extension in adhoc networks . 68
  3.6 Numerical results ........................................... 73
  3.7 Conclusions ................................................ 78

4 Discrete-time simulation of smart antenna systems in Network Simulator-2 using MATLAB and Octave
  4.1 Introduction ................................................ 81
  4.2 Modeling methodology ...................................... 82
    4.2.1 Smart antenna system .................................. 83
    4.2.2 Multipath ................................................ 85
    4.2.3 Network node MAC/PHY description .................... 85
  4.3 Smart antenna system extension ............................. 87
  4.4 Ns2 modification ........................................... 88
  4.5 Integration of ns2 with MATLAB and Octave ................. 91
    4.5.1 Matlab integration in ns2 .............................. 91
    4.5.2 Octave integration in ns2 .............................. 93
      4.5.2.1 Library OctaveEmbedded .......................... 93
      4.5.2.2 Octave C++ libraries ............................ 94
      4.5.2.3 FIFO pipes ....................................... 94
  4.6 Simulation results .......................................... 95
    4.6.1 Results ............................................... 95
  4.7 Simulation time comparison of the MATLAB/Octave extension ........ 99
  4.8 Conclusions ................................................. 101

5 Multi-packet communication for distributed wireless networks using advanced antenna systems
  5.1 Introduction ................................................ 105
  5.2 Scenario description ....................................... 107
    5.2.1 Managed scenario .................................... 107
# List of acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>Adaptive Antenna.</td>
</tr>
<tr>
<td>AC</td>
<td>Access Category.</td>
</tr>
<tr>
<td>ACK</td>
<td>ACKnowledgement.</td>
</tr>
<tr>
<td>ADC</td>
<td>Analog-to-Digital Converter.</td>
</tr>
<tr>
<td>AGC</td>
<td>Automatic Gain Control.</td>
</tr>
<tr>
<td>AIFS</td>
<td>Arbitration InterFrame Space.</td>
</tr>
<tr>
<td>AIFSN</td>
<td>Arbitration InterFrame Space Number.</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point.</td>
</tr>
<tr>
<td>ASIC</td>
<td>Application Specific Integrated Circuit.</td>
</tr>
<tr>
<td>BE</td>
<td>Best Effort.</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate.</td>
</tr>
<tr>
<td>BE_AC</td>
<td>Best Effort Access Category.</td>
</tr>
<tr>
<td>BK</td>
<td>Background.</td>
</tr>
<tr>
<td>BK_AC</td>
<td>Background Access Category.</td>
</tr>
<tr>
<td>BSS</td>
<td>Basic Service Set.</td>
</tr>
<tr>
<td>BPSK</td>
<td>Binary Phase Shift Keying.</td>
</tr>
<tr>
<td>CC</td>
<td>Common Channel.</td>
</tr>
<tr>
<td>CCA</td>
<td>Clear Channel Assessment.</td>
</tr>
<tr>
<td>CFO</td>
<td>Carrier Frequency Offset.</td>
</tr>
<tr>
<td>CFP</td>
<td>Contention Free Period.</td>
</tr>
<tr>
<td>cLMS</td>
<td>constrained Least Mean Square.</td>
</tr>
<tr>
<td>CP</td>
<td>Contention Period.</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit.</td>
</tr>
<tr>
<td>CRA</td>
<td>Concentric Ring Array.</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check.</td>
</tr>
<tr>
<td>CRC32</td>
<td>32-bits Cyclic Redundancy Check.</td>
</tr>
<tr>
<td>CS</td>
<td>Carrier Sensing.</td>
</tr>
<tr>
<td>CSE</td>
<td>Cumulative Square Error.</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance.</td>
</tr>
<tr>
<td>CSMA/CD</td>
<td>Carrier Sense Multiple Access with Collision Detection.</td>
</tr>
<tr>
<td>CTF</td>
<td>Component Technology File.</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send.</td>
</tr>
<tr>
<td>CW</td>
<td>Contention Window.</td>
</tr>
<tr>
<td>DC</td>
<td>Down-Conversion.</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function.</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DIFS</td>
<td>DCF Interframe Space.</td>
</tr>
<tr>
<td>DoA</td>
<td>Direction of Arrival.</td>
</tr>
<tr>
<td>DS</td>
<td>Distribution System.</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor.</td>
</tr>
<tr>
<td>DWN</td>
<td>Distributed Wireless Network.</td>
</tr>
<tr>
<td>EBSS</td>
<td>Extended BSS.</td>
</tr>
<tr>
<td>EDCA</td>
<td>Enhanced Distributed Channel Access.</td>
</tr>
<tr>
<td>EIFS</td>
<td>Extended InterFrame Space.</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out.</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform.</td>
</tr>
<tr>
<td>flop</td>
<td>Floating point operation.</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array.</td>
</tr>
<tr>
<td>GPP</td>
<td>General Purpose Processor.</td>
</tr>
<tr>
<td>HAL</td>
<td>Hardware Access Layer.</td>
</tr>
<tr>
<td>HCCA</td>
<td>Hybrid contention function Controlled Channel Access.</td>
</tr>
<tr>
<td>HCF</td>
<td>Hybrid Coordination Function.</td>
</tr>
<tr>
<td>IBSS</td>
<td>Independent BSS.</td>
</tr>
<tr>
<td>ICI</td>
<td>Inter-Channels BSS.</td>
</tr>
<tr>
<td>ID</td>
<td>IDentifier.</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronic Engineers.</td>
</tr>
<tr>
<td>IFFT</td>
<td>Inverse Fast Fourier Transform.</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol.</td>
</tr>
<tr>
<td>ISI</td>
<td>Inter-Symbols Interference.</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial, Scientific and Medical.</td>
</tr>
<tr>
<td>LDPC</td>
<td>Low Density Parity Check.</td>
</tr>
<tr>
<td>LLR</td>
<td>Log-Likelihood Ratio.</td>
</tr>
<tr>
<td>LoS</td>
<td>Line of Sight.</td>
</tr>
<tr>
<td>LS</td>
<td>Least Square.</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bit.</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution.</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control.</td>
</tr>
<tr>
<td>MADWiFi</td>
<td>Multiband Atheros Driver for Wireless Fidelity.</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Adhoc NETwork.</td>
</tr>
<tr>
<td>MCC</td>
<td>Multiple Communications Channel.</td>
</tr>
<tr>
<td>MDL</td>
<td>Maximum Description Length.</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple Input Multiple Output.</td>
</tr>
<tr>
<td>MPC</td>
<td>Multi-Packet Communication.</td>
</tr>
<tr>
<td>MPDU</td>
<td>MAC Protocol Data Unit.</td>
</tr>
<tr>
<td>MPR</td>
<td>Multi-Packet Reception.</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error.</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Square Error.</td>
</tr>
<tr>
<td>MUSIC</td>
<td>MUliple Slgnal Classification.</td>
</tr>
<tr>
<td>NAV</td>
<td>Network Allocation Vector.</td>
</tr>
<tr>
<td>ns2</td>
<td>Network Simulator 2.</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing.</td>
</tr>
<tr>
<td>OFDMA</td>
<td>Orthogonal Frequency Division Multiple Access.</td>
</tr>
<tr>
<td>OTcl</td>
<td>Object-oriented Tcl.</td>
</tr>
<tr>
<td>PAS</td>
<td>Power Azimuth Spectrum.</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer.</td>
</tr>
<tr>
<td>PCF</td>
<td>Point Coordination Function.</td>
</tr>
<tr>
<td>PCI</td>
<td>Peripheral Component Interconnect.</td>
</tr>
<tr>
<td>pdf</td>
<td>probability density function.</td>
</tr>
<tr>
<td>PER</td>
<td>Packet Error Rate.</td>
</tr>
<tr>
<td>PHY</td>
<td>PHYSical.</td>
</tr>
<tr>
<td>PID</td>
<td>Process IDentifier.</td>
</tr>
<tr>
<td>PLCP</td>
<td>Physical Layer Convergence Procedure.</td>
</tr>
<tr>
<td>PMF</td>
<td>Probability Mass Function.</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature Amplitude Modulation.</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service.</td>
</tr>
<tr>
<td>QPSK</td>
<td>Quadrature Phase Shift Keying.</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency.</td>
</tr>
<tr>
<td>RLS</td>
<td>Recursive Least Square.</td>
</tr>
<tr>
<td>RTS</td>
<td>Request To Send.</td>
</tr>
<tr>
<td>SAS</td>
<td>Smart Antenna System.</td>
</tr>
<tr>
<td>SAMPC</td>
<td>SIR Access MPC.</td>
</tr>
<tr>
<td>SDMA</td>
<td>Space Division Multiple Access.</td>
</tr>
<tr>
<td>SDR</td>
<td>Software-Defined Radio.</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short InterFrame Space.</td>
</tr>
<tr>
<td>SIMD</td>
<td>Single-Input Multiple-Data instruction.</td>
</tr>
<tr>
<td>SIR</td>
<td>Signal-to-Interference Ratio.</td>
</tr>
<tr>
<td>SORA</td>
<td>SOrtware RAdio.</td>
</tr>
<tr>
<td>SSE</td>
<td>Streaming SIMD Extension.</td>
</tr>
<tr>
<td>S&amp;C</td>
<td>Schmidl &amp; Cox.</td>
</tr>
<tr>
<td>TAMPC</td>
<td>Threshold Access MPC.</td>
</tr>
<tr>
<td>Tcl</td>
<td>Tool Command Language.</td>
</tr>
<tr>
<td>TclCL</td>
<td>Tcl with Classes.</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol.</td>
</tr>
<tr>
<td>ToS</td>
<td>Type of Service.</td>
</tr>
<tr>
<td>UCA</td>
<td>Uniform Circular Array.</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol.</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>UHD</td>
<td>USRP Hardware Driver.</td>
</tr>
<tr>
<td>ULA</td>
<td>Uniform Linear Array.</td>
</tr>
<tr>
<td>uLMS</td>
<td>unconstrained Least Mean Square.</td>
</tr>
<tr>
<td>USA</td>
<td>Uniform Square Array.</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus.</td>
</tr>
<tr>
<td>USRP</td>
<td>Universal Software Radio Peripheral.</td>
</tr>
<tr>
<td>VANET</td>
<td>Vehicular Adhoc Network.</td>
</tr>
<tr>
<td>VBLAST</td>
<td>Vertical Bell Laboratories Layered Space-Time.</td>
</tr>
<tr>
<td>VI</td>
<td>VId eo.</td>
</tr>
<tr>
<td>VI_AC</td>
<td>VId eo Access Category.</td>
</tr>
<tr>
<td>VO</td>
<td>VOice.</td>
</tr>
<tr>
<td>VO_AC</td>
<td>VOice Access Category.</td>
</tr>
<tr>
<td>WARP</td>
<td>Wireless open-Access Research Platform.</td>
</tr>
<tr>
<td>WCDMA</td>
<td>Wideband Code Division Multiple Access.</td>
</tr>
<tr>
<td>WDS</td>
<td>Wireless Distribution System.</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network.</td>
</tr>
<tr>
<td>WMN</td>
<td>Wireless Mesh Networks.</td>
</tr>
</tbody>
</table>
### List of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>Queue index.</td>
</tr>
<tr>
<td>(\text{AIFS}_{a,b})</td>
<td>Arbitration InterFrame Space of the (a)-th queue of the (b)-th node.</td>
</tr>
<tr>
<td>(\text{AIFS})</td>
<td>AIFS transmission time.</td>
</tr>
<tr>
<td>(A_{0+8})</td>
<td>Convolutional encoder output, puncturer input.</td>
</tr>
<tr>
<td>(B_{0+8})</td>
<td>Convolutional encoder output, puncturer input.</td>
</tr>
<tr>
<td>(X_{0+8})</td>
<td>Convolutional encoder input.</td>
</tr>
<tr>
<td>(y_{0+8})</td>
<td>Puncturer output.</td>
</tr>
<tr>
<td>(A)</td>
<td>Matrix of steering vectors.</td>
</tr>
<tr>
<td>(aC_{\text{Wmin}})</td>
<td>Default value for minimum contention windows calculation.</td>
</tr>
<tr>
<td>(aC_{\text{Wmax}})</td>
<td>Default value for maximum contention windows calculation.</td>
</tr>
<tr>
<td>(b_x)</td>
<td>Symbol mapper output.</td>
</tr>
<tr>
<td>(b_{a,i}(t))</td>
<td>Backoff timer.</td>
</tr>
<tr>
<td>(\hat{v}_{a,i}^{(h,k)})</td>
<td>Backoff timer at the (h)-th retransmission attempt and (k)-th backoff timer.</td>
</tr>
<tr>
<td>(b_{\text{quant}})</td>
<td>Number of quantized bits.</td>
</tr>
<tr>
<td>(B_{rx})</td>
<td>Receiver filter bandwidth.</td>
</tr>
<tr>
<td>(C)</td>
<td>Set of the non-legacy nodes currently involved in a communication in the MCC.</td>
</tr>
<tr>
<td>(\mathbb{C})</td>
<td>Complex number space.</td>
</tr>
<tr>
<td>(C_{\text{S}t})</td>
<td>Carrierc sensing threshold.</td>
</tr>
<tr>
<td>(CSE(m_s))</td>
<td>Cumulative square error exponentially weighted.</td>
</tr>
<tr>
<td>(C_{\text{Wmin}})</td>
<td>Minimum contention windows.</td>
</tr>
<tr>
<td>(C_{\text{Wmax}})</td>
<td>Maximum contention windows.</td>
</tr>
<tr>
<td>(d)</td>
<td>Distance between elements in wavelengths.</td>
</tr>
<tr>
<td>(D)</td>
<td>Subset of the active destinations.</td>
</tr>
<tr>
<td>(d(i))</td>
<td>Destination of (i)-th node.</td>
</tr>
<tr>
<td>(E{\cdot})</td>
<td>Expectation.</td>
</tr>
<tr>
<td>(E_{s})</td>
<td>Average slot time duration.</td>
</tr>
<tr>
<td>(E_{\text{ns}_{a,i}})</td>
<td>Average number of slots during which a DATA packet remains in the chain.</td>
</tr>
<tr>
<td>(f^d)</td>
<td>Doppler spread.</td>
</tr>
<tr>
<td>(F(\varphi))</td>
<td>Equivalent radiation pattern.</td>
</tr>
<tr>
<td>(f_{i,a}(t))</td>
<td>.</td>
</tr>
<tr>
<td>(g_n(\varphi))</td>
<td>Radiation pattern of the (n)-th array element.</td>
</tr>
<tr>
<td>(g_{0,1})</td>
<td>Convolutional encoder generator polynomials.</td>
</tr>
<tr>
<td>(G(\varphi))</td>
<td>Power gain pattern.</td>
</tr>
<tr>
<td>(G_{a}^{i})</td>
<td>Average gain of the (i)-th node.</td>
</tr>
<tr>
<td>(G_{n}^{i})</td>
<td>Null gain of the (i)-th node.</td>
</tr>
</tbody>
</table>
\( G_{\text{RX}} \) Receiver gain.
\( G_{\text{TX}} \) Transmitter gain.
\( H \) Channel matrix.
\( H_{\text{ZF}} \) Channel impulsive response.
\( h \) Retransmission attempt index.
\( H \) Number of simultaneous collisions.
\( \tilde{h}_a \) Index that account for antennas heights with respect to floor.
\( \text{Head}_{\text{IP-UDP}} \) IP-UDP header length.
\( i \) Node index.
\( i_{\text{deint}} \) Index of the second deinterleaver permutation.
\( i_{\text{int}} \) Index of the second interleaver permutation.
\( I_{i,q}(t) \) Experienced interference.
\( \mathcal{G}_{i,q} \) Set of currently active interferers.
\( \mathcal{G}_{i,q}^a \) Set of currently active interferers that can not be suppressed.
\( \mathcal{G}_{i,q}^n \) Set of currently active interferers that can be suppressed.
\( \mathcal{G}_{i,q}^n \) Set, estimated by node \( i \), of interferers not suppressed by node \( q \).
\( \mathcal{G}_{i,q}^n \) Set, estimated by node \( i \), of interferers suppressed by node \( q \).
\( j \) Imaginary unit.
\( j_{\text{deint}} \) Index of the first deinterleaver permutation.
\( j_{\text{int}} \) Index after the second interleaver permutation.
\( k, k' \) Value of the backoff counter.
\( K \) Number of discretized samples.
\( K' \) Number of iterations.
\( K_{\text{conv}} \) Convolutional encoder constraint length.
\( k_{\text{deint}} \) Index after the second deinterleaver permutation.
\( k_{\text{int}} \) Index of the first interleaver permutation.
\( K_{\text{MOD}} \) Normalization factor.
\( k_{nl} \) Generic non-legacy node.
\( k_{ss} \) Subcarrier index.
\( K_{ss} \) Maximum number of subcarrier.
\( k_w \) Wavenumber.
\( L \) Number of sustainable communications of a network.
\( l \) Estimated number of active sources.
\( l_i \) Estimated number of active sources by the \( i \)-th node.
\( i_{\text{max}} \) Maximum number of estimated active sources.
\( L_t \) Load threshold.
\( L_{t_i} \) Number of ongoing transmissions sensed by the \( i \)-th node.
\( i_{\text{cur}} \) Current threshold number of ongoing transmissions for node \( i \).
\( L_{-26,26} \) Long training sequence in frequency domain.
\( \text{len}_{a,i} \) Payload length at MAC layer of the \( a \)-th AC of the \( i \)-th node.
\( m \) Maximum number of retransmission.
\( m_{a,i} \) Maximum number of retransmissions of the \( a \)-th AC of the \( i \)-th node.
\( M \) Number of signal sources.
\( m' \) Maximum backoff stage.
\( m'_{a,i} \) Maximum backoff stage of the \( a \)-th AC of the \( i \)-th node.
\( m_s \) Sample index.
\( M_s \) Number of available samples.
\( m_s \) Bit length of the reference signal.
\( N_s \) Number of contending nodes.
\( N \) Number of radiating elements of an antenna system.
\( n \) Array element index.
\( N_j \) Number of radiating elements of the \( i \)-th node.
\( \mathcal{N} \) Set of nodes.
\( N_{\text{BPSC}} \) Number of bits for subcarrier.
\( N_{\text{CBPS}} \) Number of coded bits per OFDM symbol.
\( N_{\text{DBPS}} \) Number of data bits per OFDM symbol.
\( n_l \) Number of legacy nodes.
\( N_{\text{max}} \) Maximum number of antenna elements.
\( \mathcal{N}' \) Subset of legacy nodes.
\( n_{nl} \) Number of non-legacy nodes.
\( \mathcal{N}^{\text{nl}} \) Subset of non-legacy node.
\( N_{\text{ring}} \) Number of rings.
\( N_x \) Number of radiating elements toward the \( x \) axis.
\( N_y \) Number of radiating elements toward the \( y \) axis.
\( \text{NCT}_i \) NCT entry of the \( i \)-th node.
\( p^c \) Probability of no success.
\( P_{\text{rx}} \) Received power.
\( p_{\text{rx}}^{d(i),i} \) Received power: \( i \) transmitting, \( d(i) \) receiving.
\( P^s \) Probability of success transmission of the \( a \)-th AC of the \( i \)-th node.
\( P^s_{a,i} \) Success probability.
\( P^t \) Probability that one transmission occur in a randomly chosen slot time.
\( P_{\text{tx}} \) Transmitted power.
\( \mathcal{P}_{\text{Rayleigh}}(\zeta) \) Rayleigh fading distribution.
\( \mathcal{P}_{\text{exp}}(\zeta) \) Exponential fading distribution.
\( p_0 \cdots p_{126} \) Sequence of the pilot polarity.
\( p_{a,i}(t) \) Conditional collision probability of the \( a \)-th AC of the \( i \)-th node.
\( p(\phi) \) Power azimuth spectrum.
\( P_{\text{MUSIC}} \) Pseudospectrum.
\( \Pr[\cdot] \) Probability function.
\( Q \) Number of queues.
\( R_c \) Circle circumference.
\( R_{bo} \) Minimum residual backoff.
\( R_c \) Code rate.
\( R_i \) Code rate of the \( i \)-th node.
\( r_{\text{dist}} \) Distance from the transmitter and the receiver.
\( r_{d(i),i} \) Distance from the \( i \)-th node to \( d(i) \)-th node.
\( r_n \) Position of the \( n \)-th element in spherical coordinates.
\( R_{\text{s}}^{i,q} \) Sustainable code rate.
\( R_{\text{rates}} \) Set of the selectable code rates .
\( R_{ss} \) Correlation matrix of the desired signal.
\( R_{\text{sig}} \) Source correlation matrix.
\( R_{xx} \) Correlation matrix of the received signal.
\( \mathcal{S} \) Subset of the active sources.
\( s \) Transmitted signal vector.
\( s_i(t) \) Transmitted signal by the \( i \)-th node.
\( s'(i) \) Source of \( i \)-th node.
\( s_{a,i}(t) \) Backoff stage of the \( a \)-th AC of the \( i \)-th node.
\( S_{a,i} \) Throughput at the transport layer of the \( a \)-th AC of the \( i \)-th node.
\( S_{\text{scr}}(x_{\text{scr}}) \) Scrambler generator polynomial.
\( SIR_{d(i)} \) SIR calculated at the destination of \( i \)-th node.
\( SIR_t \) SIR threshold.
\( S_{-26,26} \) Short training sequence in frequency domain.
\( T_{\text{last}} \) Number of last transmissions.
\( T_{\text{ACK}} \) Time required to transmit the ACK packet.
\( T_c \) Time wasted because of collisions.
\( T_{\text{DATA}} \) Time required to transmit the DATA packet.
\( T_{\text{DATA}_{a,i}} \) Time required to transmit the DATA packet.
\( T_{\text{FFT}} \) Transmission time of the FFT samples for one OFDM symbol.
\( T_{\text{flop}} \) Execution time of a flot operation.
\( T_g \) Duration of the cyclic prefix.
\( T_{\text{GI}} \) Duration of the cyclic prefix of the SIGNAL and DATA symbols.
\( T_{\text{GI2}} \) Duration of the cyclic prefix of the long training sequence.
\( T_i \) Instant of beginning of ACK reception at the node \( i \)-th node.
\( T_{d(i)} \) Instant of beginning of the DATA reception at the node \( d(i) \)-th node.
\( T_{\text{SAMPC}} \) SAMPC protocol computational time.
\( T_{\text{TAMPC}} \) TAMPC protocol computational time.
\( T_{\text{OFDM}} \) Transmission time of one OFDM symbol.
\( T_{\text{PHY}} \) Time required to complete the antenna processing operation.
\( T_{\text{PREPAGE}} \) Preamble transmission time.
\( T_{a,i} \) Time required successful DATA/ACK for the \( a \)-th AC of the \( i \)-th node.
\( T_{\text{simb}} \) Symbol time.
\( T_{\text{tx}} \) Packet transmission time.
\( T_{\text{samp}} \) Beginning instant of the sampling.
\( T_{\text{op}} \) Transmission opportunity
\( \mathbf{V}_n \) Noise subspace.
\( \mathbf{V}_s \) Signal subspace.
\( \mathbf{x} \) Received signal vector (by the antenna elements).
\( \mathbf{x}_s \) Desired signal of SAS.
\( \mathbf{x}_u \) Received interference and noise (by the antenna elements).
\( \mathbf{x}_n \) Received signal at the \( n \)-th element.
\( X_{k_{ss}} \) \( k_{ss} \)-th subcarrier of known transmitted signal.
\( y(t) \) Array output.
\( y_d \) Training signal.
\( y_e \) Error signal.
\( Y_{k_{ss}} \) \( k_{ss} \)-th subcarrier of the received signal.
\( \hat{y} \) Antenna system output estimation.
\( \mathbf{w} \) Antenna weights vector.
\( w_{n} \) Antenna weight of the \( n \)-th element.
\( W(0) \) Contention windows size at the first transmission attempt.
\( W_{a,i}^{(0)} \) Contention windows size at the first transmission attempt.
\( W_{a,i}^{(h)} \) Contention windows size at the \( h \)-th retransmission attempt.
\( \alpha \) Path-loss index.
\( \bar{\alpha}(\theta, \varphi) \) Antenna array steering vector.
\( \varrho_0(k) \) Probability of having \( k \) slots between two consecutive successful transmissions.
\( \beta_0(k) \) PMF of an interval not containing collisions.
\( \beta_1(k) \) PMF of an interval containing one collision.
\( \delta \) Kronecker delta.
\( \Delta_s \) Number of slots between two consecutive successful transmissions.
\( e_n(\theta, \varphi) \) \( n \)-th element field (or amplitude) pattern.
\( \eta \) Estimated probability of activity.
\( \gamma_{\text{LMS}} \) Step-size parameter.
\( \gamma_{\text{RLS}} \) Forgetting factor.
\( \lambda \) Probability to exit from the idle state.
\( \lambda' \) Carrier wavelength.
\( \Lambda_{a,i} \) Probability of having a new packet in the buffer of the \( a \)-th AC of the \( i \)-th node.
\( \mu \) Mean packet arrival rate
\( v_c \) Speed of light.
\( \psi_{k'} \) Probability of being in the \( k' \)-th state.
\( \sigma_{\text{slot}} \)  
Slot time.

\( \sigma_n^2 \)  
Noise variance.

\( \sigma_{\text{Gaus}} \)  
Standard variation of Gaussian distribution.

\( \bar{\sigma}_\varphi \)  
Angular spread.

\( \bar{\sigma}_\tau \)  
Delay spread.

\( \tau_{\text{max}} \)  
Maximum delay spread of signal replica.

\( \tau_{a,i} \)  
Transmission probability of the \( a \)-th AC of the \( i \)-th node.

\( \theta \)  
Zenith angle.

\( \varphi \)  
Azimuth angle.

\( \varphi_{i,j'} \)  
Azimuth angle between the \( i \)-th and \( j' \)-th nodes.

\( \varphi^3_{\text{dB}} \)  
Main lobe width at half power.

\( \cdot^H \)  
Conjugate transpose operation.

\( \cdot^T \)  
Transpose operation.

\lfloor \cdot \rfloor \)  
Rounding function.

\lceil \cdot \rceil \)  
Ceiling function.

\( \text{mod} \)  
Modulus function.

\( \text{tr}(\cdot) \)  
Trace function.

\( \ast \)  
Convolution operation.
Part I

Background
Distributed wireless networks: fundamentals

This chapter provides an introduction on Distributed Wireless Networks (DWNs) based on the IEEE 802.11 standard, describing both the Medium Access Control (MAC) and the PHYSical (PHY) layers. A description of the scenario and of the network architecture together with a brief description of the IEEE 802.11 access mechanisms are provided, considering also the 802.11e extension for the support of Quality of Service (QoS) features. Furthermore, this chapter describes the propagation environment in which the DWNs operate and describes the fundamental characteristics of smart antenna systems. A literature overview concerning the medium access control protocols that adopt smart antenna systems is subsequently presented. Finally, a detailed description of the IEEE 802.11ag physical layer transceiver concludes this chapter.

1.1 General aspects of DWNs

DWNs represent networks in which the nodes are connected via wireless, hence no wired links are present. The wireless technology is becoming the preferred technology for providing indoor and outdoor connectivity. This success of the DWNs is also due to the flexibility of deployment due to lower installation and maintenance costs, which typically rise when the connections are wired.

The DWNs consist of nodes, which may differ for several characteristics, as for example mobility support, computational power, antenna system, protocols, broadband connectivity, etc. In particular, the mobility and the connectivity support imply the possibility that each node performs routing functionalities to maintain some required multi-hop connections. The computational power of a single node is related both to its capability of supporting a higher traffic load and to the support of processing operations, including those concerning the antenna system. Furthermore, since in the last years the number of wireless technologies adopted by the telecommunication companies has increased with the development of novel standards, vari-
ous nodes can also act as bridges between different protocols, thus extending the network coverage. Considering the wireless connection access, in the DWN the nodes are autonomous, and there is no central authority able to manage the nodes. In fact, every node is aware of the characteristic of the closer node, but its knowledge is, in general, not extended to the whole network.

In the scientific literature the most investigated types of DWNs are the Mobile Ad-hoc NETworks (MANETs), Vehicular Adhoc Networks (VANETs), and Wireless Mesh Networks (WMNs). These families of wireless networks are characterized by the lack of synchronization, since their nodes access asynchronously the wireless medium. In MANETs the nodes are usually characterized by a pedestrian mobility, which may be assumed in urban and rural areas. Conversely, in VANETs the connections are established between the vehicles travelling within close roads, hence the effects due to the mobility have to be taken into account for the performance evaluation. Additionally, the vehicles are also able to connect to anchor nodes placed at the side of the road. The anchor nodes are in some cases used to collect data of the traffic or to provide information for the drivers. The interests of the research community are considerably growing in the specific field of WMNs. WMNs are particular DWNs that are characterized by self-configuration, scalability, self-organization, and interoperability with existing networks. Typically, WMNs are composed by stations that form the backbone of the network and by a certain number of mobile nodes. The backbone nodes are usually stationary, and provide connectivity to the mobile nodes. The mobile nodes, called mesh clients, are less powerful than the backbone nodes, called mesh routers, and also can present less functionalities. Popular examples of mesh client are represented by smartphones, tablets, personal computers, mobile phones, etc. As mentioned before, the functionalities of the mesh clients are restricted, since the most relevant constraint is due to the power supply provided by the battery. Conversely, mesh routers are mostly powered by the power supply system and provide advanced functionalities in order to cope with the requirements of the backbone. The backbone nodes may be equipped with multi-antenna systems in order to provide a higher performance in terms of throughput and connection reliability. Additionally, the mesh routers may also present several network interfaces to communicate with other nodes adopting different communication standards. The WMNs, which have self-organizing capabilities and scalability properties, are adopted in many application fields, such as broadband networking, wireless communities, metropolitan area networking, building automation, security surveillance systems, spontaneous networking for emergency/disasters, etc. For this reason, several efforts are spent for improving their performance by the research community and industry.
1.2 IEEE 802.11: basic MAC layer characteristics

The IEEE 802.11 standard is one of the most adopted standards for DWNs. Since IEEE 802.11 operates in the unlicensed Industrial, Scientific and Medical (ISM) band (900 MHz, 2.4 GHz and 5.8 GHz), it has become very popular in the communication systems all over the world. Currently, it represents also one of the less expensive solutions for indoor and outdoor systems.

The 802.11 legacy version was approved in 1997 [1]. During the last decades several extensions have been standardized by the different 802.11x task groups. Among the large number of presented extensions, the developed amendments aim to increase the available rate [2–5], to introduce quality of service features [6], and to make secure the data exchange [7]. At present, several proposals regarding prioritization of the control frames, robust audio video streaming, communication in high mobility environment, are still in phase of study or are already present in draft format.

The IEEE 802.11 standard defines the specifications for the medium access control sublayer and the physical layer. Details concerning the IEEE 802.11ag physical layer amendment are provided in Section 1.4.

1.2.1 Architecture

The basic component of the IEEE 802.11 architecture is the Basic Service Set (BSS), which consists of a set of wireless stations. A BSS without any coordinating entity is called Independent BSS (IBSS). Instead, a BSS with a coordinating station representing the center of the star topology is called infrastructured BSS. The coordinating station is called Access Point (AP). In an infrastructured BSS all the stations have to communicate with the AP, which coordinates the access to the channel of the whole network. The AP can be connected to a Distribution System (DS), which is used to connect different BSS and/or different type of networks. The interconnections between different type of networks are allowed using portals, which operate as gateways. Several BSSs connected by the same DS take the name of Extended BSS (EBSS).

1.2.2 IEEE 802.11 access mechanisms

The IEEE 802.11 standard considers two fundamental methods to access to the medium: the centralized one and the distributed one. The centralized access is obtained using the Point Coordination Function (PCF), while the distributed one is obtained using the Distributed Coordination Function (DCF). The PCF consists of a Contention Free Period (CFP) and a Contention Period (CP). During the CFP the network coordinator, typically the AP, manages the medium access of the stations, while during the CP the channel access is driven by the DCF. In infrastructured networks
the AP alternates between CFP periods and CP periods. The synchronization in the IEEE 802.11 networks is performed using beacon frames, which are transmitted by the AP in the infrastructured networks or by the stations in DWNs.

The IEEE 802.11 DCF is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. The CSMA/CA represents the modified version for wireless networks of the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) typically used on wired networks. Since on wireless networks the collision detection is unreliable, in the collision avoidance version the access is regulated by a set of operations having the objective to reduce the collision probability between competing nodes.

The IEEE 802.11 CSMA/CA implements the carrier sensing on two layers: at the physical layer and at the MAC layer. The physical layer carrier sensing reveals the channel activity by measuring the instant power present on the medium. Instead, the MAC layer carrier sensing, so-called virtual carrier sensing, is implemented by sharing information regarding the channel occupation. In particular, the transmitter specifies in a particular field of the IEEE 802.11 header the value expressed in $\mu s$ representing the duration of the channel occupation. This field is called duration field and it is included in the MAC header. Hence, the receivers of the transmitted packet become aware of the period in which the channel will be occupied. Each node maintains the information of the channel occupation in its Network Allocation Vector (NAV). Since the NAV time is not expired, the node turns off its radio in order to reduce power consumption. It is worth noticing that the carrier sensing procedures waste a quantity of power of the same order of the transmission procedures, thus the use of the NAV provides considerable advantages in terms of energy saving.

Since this thesis focuses on DWNs, the IEEE 802.11 DCF is described in more detail. More precisely, the DCF provides two access mechanisms: the basic access and the Request To Send/Clear To Send (RTS/CTS) access.

### 1.2.2.1 Basic access

The basic access is mainly based on the monitoring of the activity on the wireless channel, hence the carrier sensing (physical and virtual) plays a fundamental role. Particularly, the carrier sensing considers the channel busy when the instantaneous received power is larger than a defined Carrier Sensing (CS) threshold $C_S$. Each node has a buffer in which the packets coming from the upper layers are put in order to be transmitted on the wireless channel. When a packet is ready to be transmitted, the carrier sensing checks if there is any activity on the wireless channel. In such a case, the node performs a transmission attempt. Otherwise, the node generates a random number, which is multiplied by the slot time $\sigma_{\text{slot}}$. The result, called backoff time, is inserted in a backoff counter. The random number is chosen in an uniformly distributed interval $[0, CW_{\text{min}} - 1]$, where $CW_{\text{min}}$ represents the min-
imum Contention Window (\(CW\)). The backoff counter is decreased when the channel is sensed idle for more than a Distributed InterFrame Space (DIFS). If a channel activity is sensed, the backoff counter is frozen, and the countdown restarts when no activity is sensed for a DIFS. When the backoff counter reaches the value zero and the channel is sensed idle for a DIFS, the node transmits. The random choice of the backoff counter reduces the probability of simultaneous transmission attempts. The correct packet reception of the DATA packet is notified by the receiver to the transmitter with an ACKnowledgement packet (ACK), which is transmitted a Short InterFrame Space (SIFS) after the DATA reception. If the ACK is not received within a timeout, the transmitter considers the packet not properly received. In such a case, the transmitter reschedules the transmission. For every retransmission the minimum contention window is doubled since it reaches its maximum value, called maximum contention window \(CW_{\text{max}} = 2^{m'} CW_{\text{min}}\), where \(m'\) represents the maximum backoff stage. Further retransmissions are attempted until the maximum number of retransmissions \(m(\geq m')\), called retry limit, is reached. Once the retry limit is reached the packet is discarded.

When a collision is detected on the wireless channel (e.g. when the packet is received corrupted) the backoff counter is frozen for an Extended InterFrame Space (EIFS), and, once the EIFS is expired, the counter is resumed after a DIFS.

The IEEE 802.11 standard defines different inter-frame spaces: SIFS, PIFS, DIFS, and EIFS, where the Point InterFrame Space (PIFS) has the same role of the DIFS, but it is used only by the AP. The different durations intrinsically define the priority of the node access. The SIFS is shorter than a PIFS, which, in turn, is shorter than a DIFS, while an EIFS is longer than a DIFS. Hence, the node transmitting after SIFS will gain the channel access before the node transmitting after a PIFS, or DIFS, or an EIFS. Therefore, an ongoing DATA/ACK handshake has a higher priority with respect to any other new communication. Besides, since the PIFS is lower than a DIFS, the AP has always a higher priority with respect to the other nodes. In particular, the PIFS is equal to a SIFS + \(\sigma_{\text{slot}}\), while a DIFS is equal to a SIFS + 2\(\sigma_{\text{slot}}\). The EIFS is equal to the transmission time of an ACK at lowest basic rate plus the SIFS and the DIFS.

### 1.2.2.2 RTS/CTS access

The RTS/CTS access is based on a 4-way handshake, in which the DATA/ACK exchange is preceded by an RTS/CTS exchange. The backoff mechanism performs in the same way as in the basic access. The transmitter sends an RTS packet to the destination, which replies with a CTS packet after a SIFS. Once, the CTS packet is received by the transmitter, the DATA packet is sent after a SIFS. The destination sends the ACK after a SIFS, if the DATA packet is correctly received. The nodes, which are not involved in the communication, recognize a RTS/CTS exchange and read the duration field contained in the RTS/CTS packet and set their NAVs. Hence,
they put themselves in idle state leaving the transmitter and the receiver to conclude
the DATA/ACK transfer. An example of a time line reporting RTS/CTS/DATA/ACK
exchange is shown in Fig. 1.1.

1.2.3 IEEE 802.11e - QoS extension

The legacy version of the IEEE 802.11 standard does not implement QoS mechan-
isms such as traffic prioritization and/or bandwidth reservation, hence an exten-
sion, the IEEE 802.11e amendment, has been introduced to enable the control of the
QoS at MAC layer [6]. In particular, QoS features are considered of critical import-
ance for delay-sensitive application such as video and audio streaming.

The IEEE 802.11e MAC layer extension defines enhancements to the DCF and
PCF by introducing the Hybrid Coordination Function (HCF). Within the HCF two
channel access mechanisms have been developed by defining four traffic categor-
ies. These mechanisms are the Enhanced Distributed Channel Access (EDCA) and
the HCF Controlled Channel Access (HCCA). The first one enables traffic prioritiz-
atation, while the second one enables admission control and bandwidth reservation.
At present, the EDCA currently represents the most diffused access scheme handling
QoS at MAC layer for DWNs.

1.2.3.1 Enhanced distributed channel access

The EDCA introduces four type of traffic priorities corresponding to four Access
Categories (AC): B ackground (BK_AC), B est Effort (BE_AC), V ideo (VI_AC), and V Oice
(VO AC). Every AC is associated to a transmission queue, which is characterized
by a particular channel access priority. The channel access priority in the EDCA is
achieved by the diversification of the channel access parameters of the IEEE 802.11
DCF. In particular, while the DCF specifies only one set of parameters, the EDCA spe-
1.2. IEEE 802.11: basic MAC layer characteristics

<table>
<thead>
<tr>
<th>AC</th>
<th>( CW_{\text{min}} )</th>
<th>( CW_{\text{max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background AC</td>
<td>( aCW_{\text{min}} )</td>
<td>( aCW_{\text{max}} )</td>
</tr>
<tr>
<td>Best effort AC</td>
<td>( aCW_{\text{min}} )</td>
<td>( aCW_{\text{max}} )</td>
</tr>
<tr>
<td>Video AC</td>
<td>( (aCW_{\text{min}}+1)/2-1 )</td>
<td>( aCW_{\text{min}} )</td>
</tr>
<tr>
<td>Voice AC</td>
<td>( (aCW_{\text{min}}+1)/4-1 )</td>
<td>( (aCW_{\text{min}}+1)/2-1 )</td>
</tr>
</tbody>
</table>

Table 1.1: Calculation of contention window boundaries.

<table>
<thead>
<tr>
<th>AC</th>
<th>( CW_{\text{min}} )</th>
<th>( CW_{\text{max}} )</th>
<th>AIFSN</th>
<th>( Tx_{\text{op}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background AC</td>
<td>15</td>
<td>1023</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Best effort AC</td>
<td>15</td>
<td>1023</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Video AC</td>
<td>7</td>
<td>15</td>
<td>2</td>
<td>3.008ms</td>
</tr>
<tr>
<td>Voice AC</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>1.504ms</td>
</tr>
<tr>
<td>Legacy DCF</td>
<td>15</td>
<td>1023</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1.2: Default EDCA parameters for each AC

cifies one set of parameters for each access category. This set of EDCA parameters consists of:

- minimum contention window
- maximum contention window
- Transmission Opportunity (\( Tx_{\text{op}} \))
- Arbitration InterFrame Space Number (AIFSN)

The minimum and maximum contention windows have the same meaning discussed in Subsection 1.2.2 for the DCF. The transmission opportunity represents the time interval during which a QoS enabled station has the right to send multiple MAC Protocol Data Unit (MPDU) without having to re-contend for access after having succeeded in sending the first frame. The Arbitration InterFrame Space (AIFS) is a generalization of the DIFS. The expression for the AIFS is reported below:

\[
AIFS = \text{SIFS} + \text{AIFSN} \cdot \sigma_{\text{slot}}. \tag{1.2.1}
\]

Table 1.1 presents the expressions for the definition of the EDCA parameters of the ACs in the IEEE 802.11e. The default values \( aCW_{\text{min}} = 15 \) and \( aCW_{\text{max}} = 1023 \) are used in the IEEE 802.11abg standards. In relation to these default values the set of default EDCA parameters are reported in Table 1.2. This second table also reports the values of the legacy DCF adopted in the IEEE 802.11 standard. The higher-priority service class can use shorter AIFS, smaller contention windows and can rely on a large \( Tx_{\text{op}} \) to transmit more frames after a successful contention.
1.3 MAC layer extensions in the presence of advanced antennas systems

The use of the telecommunication spectrum is subjected to several laws, which define the frequency bands that can be utilized. With a special remark to the commercial products, a lot of technologies have been developed for communicating in the unlicensed ISM band (900 MHz, 2.4 GHz and 5.8 GHz). The developed technologies adopt their own communication techniques, which may interfere one to each other. A popular example is the microwave ovens and Audio/Video extenders, which work in the same frequency band as Bluetooth, IEEE 802.15.4-based (ZigBee, WirelessHART), and IEEE 802.11bgn devices. All these technologies differ in their specifications in terms of transmission power, receiver sensitivity, transmission technique, modulation, etc. However, all these different technologies interfere when the devices operate close to each other. The interference comes out due to the power radiation that node transmits on the wireless channel. Besides, the interfering power leads to a degradation of the ongoing communication or prevents the nodes from contending for the channel. In this sense, several known MAC problems arise due to the erroneous or partial information the MAC layer is receiving from the physical layer. Most of the problems, which are described below, are present in networks where the nodes are equipped with omnidirectional antennas. Nowadays the nodes can be equipped with directional antennas or with advanced antenna systems, which provide directional and/or adaptive radiation patterns. Directional radiation patterns are exploited to increase wireless channel capacity and reliability.

1.3.1 Propagation environment

This section identifies the propagation environment that is assumed in the wireless scenario analyzed in this thesis. The most simple model, called free-space propagation model, accounts only for the direct path between the transmitter and the receiver. This model can be used for the calculation of the signal strength due to the path-loss attenuation. However, this model is unrealistic because it fails to account for other phenomena that occur during the signal propagation. In fact, during the propagation, the signal is affected not only by the path-loss attenuation, but also by reflections, refractions, diffraction and scattering (Fig. 1.2). This various mechanisms establish alternate propagation paths all differing in phase, amplitude, delay, and Direction of Arrival (DoA), leading to a phenomenon referred to as multipath. Since, in typical multipath scenarios, the distribution of large numbers of reflecting, diffracting, refracting, and scattering objects is statistically modeled, the multipath is studied considering a probabilistic model for estimating the signal and channel behavior. In particular, the multipath signal amplitudes, phases, delays, and DoAs, be-
come random variables. The replicas of the signal arrived at the receiver, incoming from different propagation paths, may add themselves constructively or destructively, thus causing fluctuations of the received signal envelope in the presence of mobility. The resulting fluctuations can be divided into short-term fluctuations and long-term fluctuations. In the first case the effect is commonly referred to as fading, while in the second case it is referred to as shadowing.

With reference to the contribution of this thesis, the following sections will provide additional details on path-loss attenuation, fading, statistics, and distribution of the DoAs.

1.3.1.1 Path-loss attenuation

The power density reduction of an electromagnetic wave propagating in the free-space mainly depends on the distance between the transmitter and the receiver. The path-loss attenuation can be described by a parameterized ground model, in which the power received by a node at distance $r_{\text{dist}}$ from a transmitter can be expressed using a Friis transmission equation as:

$$p_{\text{tx}} = \frac{p_{\text{tx}} \cdot G_{\text{tx}} \cdot G_{\text{rx}} \cdot \tilde{h}_\alpha}{r^\alpha_{\text{dist}}},$$  

(1.3.1)

where $p_{\text{tx}}$ is the transmission power, $\alpha (> 2)$ is the path-loss exponent, $\tilde{h}_\alpha$ accounts for the height of the transmitting and the receiving antennas with respect to the floor, $G_{\text{tx}}$ and $G_{\text{rx}}$ are the transmitting and the receiving antenna gains, respectively. A typical value $\alpha = 2$ is adopted for the free-space propagation, while $\alpha = 4$ is considered adequate for a two ray ground propagation environment. In a dense urban environment the typical range of $\alpha$ varies between 4 and 6.
1.3.1.2 Fading

As mentioned before, fading denotes the fluctuation of the received signal power due to the combination of multipath and mobility. Fading due to multipath is usually called multipath fading and can be classified focusing on the time domain or the frequency domain.

The classification in the time domain refers to delay spread, which represents a measure of the time dispersion of the channel. The delay spread $\hat{\sigma}_\tau$ can be evaluated by considering the received power and the delay of each received replica.

In the frequency domain multipath fading is classified in flat and frequency selective. In flat fading the delay spread $\hat{\sigma}_\tau$ is much lower than the inverse of the receiver filter bandwidth $B_{rx}$ ($\hat{\sigma}_\tau \ll \frac{1}{B_{rx}}$). With flat fading the frequency response is flat relative to the frequency of the transmitted signal, hence the multipath characteristics of the channel have an influence just on the received signal power. Conversely, in frequency selective fading the delay spread is larger or close to $\frac{1}{B_{rx}}$ ($\hat{\sigma}_\tau \geq \frac{1}{B_{rx}}$). In this case, the channel introduces signal amplitude and phase distortions.

In the scenario where also the mobility of the transmitter, receiver and/or the obstacles are involved, the Doppler effect is present. The Doppler effect introduces a frequency shift of the carrier frequency of the replicas. Each replica can experience a different frequency shift, because the angle of scattering with respect to the moving direction is different for each scattering object. The maximum possible frequency shift, so-called Doppler shift, is named Doppler spread $f_d$. The Doppler effect results in Inter-Carrier Interference (ICI). Considering the Doppler effect, multipath fading can be distinguished between fast fading when the Doppler spread is larger or close to the inverse of the symbol duration $T_{simb}$ ($f_d \geq \frac{1}{T_{simb}}$), and slow fading in the opposite case ($f_d \ll \frac{1}{T_{simb}}$).

The most diffused fading statistic is the Rayleigh fading, which is often assumed to model the behavior of radiomobile channels in which a Line of Sight (LoS) component is not present and the DoAs are uniformly distributed around the transmitter. The Rayleigh distribution is defined as:

$$
\mathcal{P}_{Rayleigh}(\zeta) = \begin{cases} 
\frac{2\zeta}{\bar{P}_{rx}} e^{-\frac{\zeta^2}{\bar{P}_{rx}}} & \zeta \geq 0 \\
0 & \text{elsewhere}
\end{cases},
$$

(1.3.2)

where $\zeta$ is the envelope of the received signal, and $\bar{P}_{rx}$ is the mean received power, while the statistic of the squared $\zeta$ is exponential:

$$
\mathcal{P}_{exp}(\zeta) = \begin{cases} 
\frac{1}{\bar{P}_{rx}} e^{-\frac{\zeta}{\bar{P}_{rx}}} & \zeta \geq 0 \\
0 & \text{elsewhere}
\end{cases}.
$$

(1.3.3)

Recent studies have proved that Rayleigh fading can also be present in scenarios in which the replicas are distributed within small angles. In this thesis, a Rayleigh block
fading model is adopted, thus the fading effects are considered unchanged during the reception time.

### 1.3.1.3 Spatial channel model

The channel model in the spatial domain defines the spatial characteristics of the scattering environment and can deeply affect the system performance when advanced antenna systems are adopted. The spatial channel model accounts for the antenna characteristics with respect to the spatial (angular) distribution of the different signal replicas arriving to the receiver. The Power Azimuth Spectrum (PAS) $p(\phi)$ and the angular spread $\hat{\sigma}_\phi$ are used to characterize the spatial properties of the channel in the azimuth domain. The PAS represents the power distribution along all the possible DoAs of the received signal replicas, while the angular spread can be viewed as the square root of the variance of the PAS. In particular, the angular spread (or azimuth spread) provides a measure of the angular dispersion of the channel. In the spatial domain the channels can be divided into groups: low-rank channels and high-rank channels. In the low-rank channel $\hat{\sigma}_\tau \ll \frac{1}{2\tau_{rx}}$, which denotes flat fading, and $\hat{\sigma}_\phi \ll \varphi^{3\text{dB}}$, where $\varphi^{3\text{dB}}$ is the half power beamwidth of the antenna radiation pattern. In the high-rank channel $\hat{\sigma}_\tau \geq \frac{1}{\tau_{rx}}$ or $\hat{\sigma}_\phi \geq \varphi^{3\text{dB}}$. The low-rank channel denotes very high spatial and temporal correlations between the signal replicas. This means that the incoming multipath components can be considered as confined within a small angle. Conversely, the high-rank channel denotes a low spatial correlation among the incoming replicas, which arrive over a wide angular region. The just provided distinction concerning the spatial domain of the channel propagation refers not only to the angular distribution of the direction of arrivals of the signal replicas, but also to the spacing between the antenna array elements, which are responsible for the correct discrimination of the different paths.

The low-rank channel model is based on the distribution of the scatterers around the transmitting source, and on the distance between the transmitting node and the receiving node. The most popular PASs presented in the literature are the truncated Gaussian, the truncated Laplacian, and the ring of scatterers (Fig. 1.3). The truncated Gaussian is one of the first historically considered distribution and has the advantage of providing a direct relationship between the PAS and the angular spread [27]. The truncated Laplacian is obtained from empirical data, and corresponds to a typical urban environment. These data suggest that most of the energy arrives from the region closer to the transmitter, while the contribution of the scatterers far from the transmitter is considered much lower [28]. Finally, the ring of scatterers distribution is calculated from geometrical considerations by assuming the scatterers as uniformly distributed on a circumference surrounding the transmitting node. Usually, the radius of the circumference is considered smaller than the distance between the transmitter and the receiver [29].
Chapter 1. Distributed wireless networks: fundamentals

1.3.2 General aspects of multi-antenna systems in low-rank environment

A multi-antenna system consists of an array of radiating elements positioned accordingly to a defined geometrical configuration and a signal processing unit. By adopting multi-antenna systems, it is possible to generate radiation patterns with the main beam towards the desired direction and nulls towards the directions of the interferers. Besides, the radiation pattern can be electronically controlled by modifying the phase and/or the amplitude of the radiating element current excitations. This operation is commonly known as beamforming.

Adaptive beamforming techniques are exploited in low-rank wireless channels for maximizing the Signal-to-Interference Ratio (SIR) at the antenna system output. This goal is obtained by focusing the energy towards certain desired directions and mitigating or suppressing the energy towards the undesired ones. As described in the previous section, in the low-rank channel the replicas coming from a defined source are highly spatial correlated. In this environment it is expected that the different multipath components coming from the same source provide a statistically dependent behavior, thus making advantageous the adoption of directional patterns with sharp nulls. Directional radiation patterns obtained by adaptive beamforming allow the re-
use of the spatial domain by Space Division Multiple Access (SDMA).

Beamforming may be realized applying the excitations directly to the antenna elements at the Radio Frequency (RF) stage. This method, called analog beamforming, is expensive due to the high quality of the RF components, such as precise phase shifters and selective power dividers. Conversely, Down-Converting (DC) the received RF signals and digitalizing them by Analog-to-Digital Converters (ADCs), one can realize digital beamforming. Considering Fig. 1.4, for \( N \) radiating elements the array output \( y(t) \) can be expressed as:

\[
y(t) = \sum_{n=1}^{N} w_n^* x_n(t) = \mathbf{w}^H \mathbf{x}(t),
\]

where \( \mathbf{x} = [x_1(t), \ldots, x_N(t)]^T \) is the received signal vector, \( \mathbf{w} = [w_1, \ldots, w_N]^T \in \mathbb{C}^{N \times 1} \) is the vector of the antenna weights, \( (\cdot)^H \) denotes the conjugate transpose operation, and \( (\cdot)^T \) denotes the transpose operation. Hence, digital beamforming is performed by the controlling software using a processing algorithm that adjust the excitations of the antenna elements, also called weights.

Depending on the level of sophistication of the adopted processing algorithm, beamforming techniques can be subdivided in two groups:

- Fixed beamforming;
- Adaptive beamforming.

In fixed beamforming the main lobe of the radiation pattern is fixed or precomputed. The interference in fixed beamforming systems is mitigated but not suppressed. The overall system cost can be relatively low. Instead, adaptive beamforming algorithms are able to steer the main lobe towards the desired direction and to mitigate or suppress the undesired sources. Optimal performance can be achieved by using adaptive beamforming, which, however, yields to higher costs and implementation efforts.
Chapter 1. Distributed wireless networks: fundamentals

Multi-antenna systems, which use adaptive beamforming algorithms, are also referred in literature as Adaptive Arrays (AAs) or Smart Antenna Systems (SASs). The following section provides a brief description of the antenna array geometries and of the beamforming algorithms considered in this thesis.

1.3.2.1 Physical antenna system: array geometry

Several today’s applications require particular radiation patterns with multiple main beams, which have to satisfy main lobe beamwidth, side-lobe level and null constraints. The required radiation pattern can be obtained using more radiating elements disposed in proper geometrical configurations. The total field radiated by an antenna array is determined by the vector addition of the individual elements.

The far field properties of an antenna array can be expressed using the antenna array steering vector, whose \( n \)-th component is defined as:

\[
\hat{\alpha}_n(\theta, \varphi) = \epsilon_n(\theta, \varphi)e^{ijkr_n \times \hat{r}},
\]

where \( \epsilon_n(\theta, \varphi) \) is the electric field pattern in amplitude and phase in the direction \((\theta, \varphi)\), with \( \theta \) and \( \varphi \) denoting the zenith and the azimuth angles, of the \( n \)-th element, \( r_n = (r_n \sin \theta \cos \varphi, r_n \sin \theta \sin \varphi, r_n \cos \theta) \) is the position of the \( n \)-th array element in spherical coordinates \((r_n, \theta_n, \varphi_n)\), \( \hat{r} = (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta)^T \) is the unit vector of the incoming wave, and \( k_w \) is the wavenumber defined as:

\[
k_w = \frac{2\pi}{\lambda'},
\]

where \( \lambda' \) is the carrier wavelength. It is worth noticing that the array steering vector includes both the radiation properties of each element and the geometrical characteristics of the antenna array. In this thesis the presented formulas consider that the antenna system and all the sources lie on the \( xy \)-plane, hence the angle \( \theta \) is assumed equal to \( \frac{\pi}{2} \).

1.3.2.1.1 Linear array The antenna array with the radiators deployed linearly represents the most studied case. The linear antenna array consists of \( N \) radiating elements disposed on a straight line. If the \( N \) elements lie at the same distance, the array is called Uniform Linear Array (ULA). The array steering vector of a linear array with identical and equispaced elements is:

\[
\hat{\alpha}(\varphi) = [\epsilon_0(\varphi), \ldots, \epsilon_n(\varphi)e^{j2\pi nd\cos(\varphi)}, \ldots, \epsilon_{N-1}(\varphi)e^{j2\pi(N-1)d\cos(\varphi)}]^T,
\]

where \( d \) is the distance between adjacent radiators expressed as a multiple of the carrier wavelength. It is worth noticing that the provided equation is valid for an array in which the elements lie on the \( x \) axis with the first element in the origin (Fig. 1.5).
1.3. MAC layer extensions in the presence of advanced antennas systems

1.3.2.1.2 Circular array
In the circular array the radiating elements lie on a circumference. When the elements are equal and equidistant the array takes the name of Uniform Circular Array (UCA), (Fig. 1.6). Thanks to their circular symmetry uniform circular arrays do not produce grating lobes. A property that makes their use preferable to linear arrays for many applications. The steering vector of a circular array lying on the $xy$-plane can be expressed as:

$$\tilde{\alpha}(\varphi) = [\epsilon_0(\varphi)e^{j2\pi R_c\cos(\varphi-\varphi_0)}, \ldots, \epsilon_n(\varphi)e^{j2\pi R_c\cos(\varphi-\varphi_n)}, \ldots, \epsilon_{N-1}(\varphi)e^{j2\pi R_c\cos(\varphi-\varphi_{N-1})}]^T,$$

where $\varphi_n$ is the angle between the $x$ axis and the $n$-th element, and $R_c$ is the radius of the circumference.

1.3.2.1.3 Square array
The rectangular array can be obtained placing a certain number of elements on a rectangular grid. In the particular case of the square arrays, the number of elements toward the $x$-direction ($N_x$) is equal to those toward the $y$-
Chapter 1. Distributed wireless networks: fundamentals

Figure 1.7: Uniform square array.

Figure 1.8: Concentric ring array.

direction \((N_y)\). If the radiating elements are equispaced at distance \(d\) (Fig. 1.7), the square array is called Uniform Square Array (USA). The steering vector of a uniform square array with the first element in the origin can be defined using a matrix as:

\[
\tilde{\alpha}(\phi) = \begin{bmatrix}
\epsilon_{0,0}(\phi) e^{j2\pi d (N_y-1) \sin(\phi)} \\
\vdots \\
\epsilon_{N_x-1,0}(\phi) e^{j2\pi d (N_y-1) \cos(\phi)} \\
\vdots \\
\epsilon_{N_x-1,N_y-1}(\phi) e^{j2\pi d ((N_x-1) \cos(\phi) + (N_y-1) \sin(\phi))}
\end{bmatrix}.
\]

\[(1.3.9)\]

1.3.2.1.4 Concentric ring array

Concentric Ring Arrays (CRAs) can be considered as evolutions of the circular array. In a CRA, there are \(N_{\text{ring}}\) rings with the same origin, but a different radius. The specific configuration that will be adopted in this thesis consists of a radiating element placed in the origin, while the number of equidistant elements on each ring is equal to a factor 4 of the ring number, thus 4 elements on the first ring, 8 on the second and so on. The array steering vector can be obtained by superposing the effect of each single ring:

\[
\tilde{\alpha}(\phi) = [\epsilon_0(\phi) e^{j2\pi R_{c0} \cos(\phi-\phi_0)}, \ldots, \epsilon_n(\phi) e^{j2\pi R_{cn} \cos(\phi-\phi_n)}, \ldots, \epsilon_{N-1}(\phi) e^{j2\pi R_{cN-1} \cos(\phi-\phi_{N-1})}]^T,
\]

\[(1.3.10)\]
where \( R_{cn} \) is the radius of the circumference on which the \( n \)-th element lies.

### 1.3.2.2 Signal processing unit: beamforming and direction of arrival estimation algorithms

As described in the Subsection 1.3.2, adaptive beamforming aims to dynamically control the radiation pattern in order to perform electrical beam steering towards a desired direction and null steering to suppress or mitigate the interfering signals. Hence, a smart antenna system is able to maximize the SIR at the receiver. Since digital beamforming is considered, the received signal is assumed sampled at regular intervals, so as to obtain a discrete-time signal sequence. Hence the vector \( x(m_s) \) of the \( m_s \)-th sample of the signal received by the \( N \) antenna elements can be decomposed in two parts according to:

\[
x(m_s) = x_s(m_s) + x_u(m_s),
\]

where \( x_s(m_s) \) represents the \( N \times 1 \) vector of the desired signal and \( x_u(m_s) \) represents the \( N \times 1 \) vector of the sum of interference and noise. The received signals are used to evaluate the corresponding array correlation matrices, which can be defined as:

\[
R_{xx} = E\{x(m_s)x(m_s)^H\} \in \mathbb{C}^{N \times N},
\]

\[
R_{ss} = E\{x_s(m_s)x_s(m_s)^H\} \in \mathbb{C}^{N \times N},
\]

\[
R_{uu} = E\{x_u(m_s)x_u(m_s)^H\} \in \mathbb{C}^{N \times N},
\]

where, more precisely, \( R_{xx} \), \( R_{ss} \), and \( R_{uu} \) represent the correlation matrix of the received signal, of the desired signal, and of the interference signal, respectively. Assuming that the processes \( x(t) \), \( x_s(t) \), and \( x_u(t) \) are ergodic in the correlation matrix, the respective correlation matrices can be approximated by averaging on the available samples:

\[
R_{xx} \approx \hat{R}_{xx} = \frac{1}{M_s} \sum_{m_s=1}^{M_s} x(m_s)x(m_s)^H,
\]

\[
R_{ss} \approx \hat{R}_{ss} = \frac{1}{M_s} \sum_{m_s=1}^{M_s} x_s(m_s)x_s(m_s)^H,
\]

\[
R_{uu} \approx \hat{R}_{uu} = \frac{1}{M_s} \sum_{m_s=1}^{M_s} x_u(m_s)x_u(m_s)^H,
\]

where \( M_s \) is the number of available samples. The signal processing unit digitally elaborates the correlation matrices using proper signal processing algorithms based on the eigenanalysis.
Adaptive beamforming techniques are subdivided in three main groups: temporal reference techniques, spatial reference techniques and blind techniques, which are not considered in this thesis. Techniques belonging to the first group require a reference signal to be transmitted at the beginning of the frame, while the techniques belonging to the second group generate the antenna weights using the spatial information, which is usually obtained using DoA estimation algorithms.

### 1.3.2.2.1 Temporal reference techniques

Temporal reference algorithms are based on a training signal $y_d(m_s)$, which is known to the transmitter and the receiver. Since at the receiver the output from the antenna system is available, the receiver itself can evaluate the difference:

$$y_e(m_s) = y_d(m_s) - y(m_s),$$

(1.3.18)

where $y_e(m_s)$, $y_d(m_s)$, and $y(m_s)$ are the vectors of the $m_s$-th sample of the error signal, the training sequence, and the antenna system output, respectively. Using (1.3.18) the beamforming algorithm can adapt the antenna weights of the smart antenna system. The excitations of the antenna array are calculated by iteratively minimizing the error between the received signal and the reference one (training sequence). The following subsections present two popular beamforming algorithms based on the temporal reference approach: the unconstrained least mean square algorithm and recursive least square algorithm, which will be implemented and used in the following of the thesis.

**Unconstrained Least Mean Square (uLMS)**

The uLMS algorithm adjusts the vector of the array weights $\mathbf{w}$ by minimizing the Mean Square Error (MSE) between the received signal and the known training se-
1.3. MAC layer extensions in the presence of advanced antennas systems

Using (1.3.4), (1.3.18) can be rewritten as:

\[ y_e(m_s) = y_d(m_s) - w^H(m_s)x(m_s), \]  
(1.3.19)

from which the squared error can be obtained as:

\[ |y_e(m_s)|^2 = |y_d(m_s) - w^H(m_s)x(m_s)|^2, \]  
(1.3.20)

which, in turn, leads to the MSE equation given by:

\[ \text{MSE}_{LMS} = E[|y_e|^2] = E[|y_d|^2] - 2w^H\tilde{y} + w^HR_{xx}w, \]  
(1.3.21)

where \( \tilde{y} = E[x(m_s)y_d(m_s)] \). The minimization of the MSE can be performed using the gradient method, which, however, requires the computation of the inverse of the array correlation matrix \( R_{xx}^{-1} \), and the calculation of the term \( \tilde{y} \). These two operations are computationally intensive and hence alternative, iterative, solutions have been developed. One of the most diffused algorithm for the iterative minimization of (1.3.21), is the LMS algorithm, which reduces the computational burden by updating the weights at each iteration and estimating the gradient of the MSE. The LMS estimation is performed substituting \( R_{xx} \) and \( \tilde{y} \) with their noisy estimations available at each iteration. The weights are updated according to the following relationship:

\[ w(m_s + 1) = w^H(m_s) - \gamma_{LMS}(m_s + 1) y_e^*(m_s), \]  
(1.3.22)

\[ y_e(m_s) = w^H(m_s)x(m_s + 1) - y_d(m_s + 1), \]  
(1.3.23)

where \( \gamma_{LMS} \) is the step-size parameter, which controls the tradeoff between speed of convergence and accuracy of the solution. The convergence of \( w \) to the optimum weight vector and the stability of the algorithm are guaranteed by the following condition:

\[ 0 < \gamma_{LMS} < \frac{1}{\nu_{\text{max}}}, \]  
(1.3.24)

where \( \nu_{\text{max}} \) is the maximum eigenvalue of \( R_{xx} \), even if, for practical purposes, (1.3.24) is usually replaced by:

\[ 0 < \gamma_{LMS} < \frac{1}{\text{tr}(R_{xx})}, \]  
(1.3.25)

where \( \text{tr}(R_{xx}) \) represents the trace of \( R_{xx} \), which is easier to compute than the estimation of the eigenvalues of \( R_{xx} \).

**Recursive Least Squares (RLS)**

A second and widely adopted temporal reference technique is the RLS algorithm, in which the gradient step is replaced by a gain matrix \( \hat{R}_{xx} \) that is estimated at each
iteration. This reduces the dependence of the convergence characteristics of the algorithm on the eigenvalues of $R_{xx}$. The RLS algorithm minimizes the Cumulative Square Error (CSE) exponentially weighted defined as:

$$CSE(m_s) = \sum_{m_s'=0}^{m_s} \gamma_{RLS}^{m_s-m_s'} |y_e(m_s')|^2,$$  \hspace{1cm} (1.3.26)

where $\gamma_{RLS} \in (0, 1)$ is a parameter called forgetting factor. The weights are updated following the rule:

$$w(m_s) = w(m_s - 1) - \hat{R}_{xx}^{-1}(m_s)x(m_s)y_e^*(m_s - 1),$$  \hspace{1cm} (1.3.27)

where the inverse of the gain matrix $\hat{R}_{xx}(m_s)$ at $m_s$-th iteration is obtained as:

$$\hat{R}_{xx}^{-1}(m_s) = \begin{cases} \frac{1}{\gamma_{RLS}} I_N & m_s = 0 \\ \frac{1}{\gamma_{RLS}} [\hat{R}_{xx}^{-1}(m_s - 1) - \frac{\hat{R}_{xx}^{-1}(m_s - 1)x(m_s)x^H(m_s)\hat{R}_{xx}^{-1}(m_s - 1)}{\gamma_{RLS} + x^H(m_s)\hat{R}_{xx}^{-1}(m_s - 1)x(m_s)}] & m_s > 0 \end{cases}$$  \hspace{1cm} (1.3.28)

The RLS algorithm solves the problem related to the influence of the eigenvalues on the convergence of the LMS. In particular, RLS provides a higher speed of convergence, but at the cost of a higher complexity.

### 1.3.2.2 Spatial reference techniques

The beaforming algorithms using spatial reference techniques often require the use of DoA estimation algorithms to obtain the angle of arrivals. The DoA are estimated on the basis of the signal received from the antenna array. Subsequently, the antenna weight are estimated by the beamforming algorithm to obtain the desired radiation pattern (Fig. 1.10).
**MUltiple SIgnal Classification (MUSIC)**

The MUSIC algorithm is a widely adopted technique for DoA estimation that belongs to the family of spectral-based techniques. The other main family of DoA estimators is based on a parametric approach and is not considered in this thesis. Spectral-based methods rely on the evaluation of a function of the azimuth angle $\phi$, called speudospectrum. The maxima of this pseudospectrum give an indication of the position of the DoAs of the sensed sources. The MUSIC algorithm exploits the eigenstructure properties of the antenna array correlation matrix assuming uncorrelated sources and uncorrelation between the sources and the noise. This uncorrelation does not hold in general, hence the MUSIC algorithm is less reliable when the correlation between the sources becomes significant, while it can not be adopted when the sources and the noise are correlated.

Assuming uncorrelation between $M$ incoming sources and the noise, the correlation matrix $R_{xx}$ of the antenna array can be decomposed as:

$$R_{xx} = A R_S A^T + \sigma^2_n I,$$

(1.3.29)

where $A$ is the $N \times M$ matrix of the steering vectors, $R_S$ is the $M \times M$ source correlation matrix, $\sigma^2_n$ is the noise variance, and $I$ is the $N \times N$ identity matrix. If the received signals are uncorrelated the smallest eigenvalues correspond to the noise variance $\sigma^2_n$.

The space spanned by the eigenvectors of $R_{xx}$ can be partitioned in two orthogonal subspaces: the signal subspace $V_s$ and the noise subspace $V_n$. Considering $M$ ($< N$) incoming signals, the number of eigenvectors and associated eigenvalues of the signal subspace is also equal to $M$. Similarly, the number of eigenvectors and associated eigenvalues of the noise subspace is equal to $N - M$. The noise subspace $V_n$, which is spanned by the eigenvectors associated to the smallest eigenvalues, is orthogonal to the antenna array steering vector $\alpha(\phi)$. The MUSIC pseudospectrum can be defined as:

$$P_{\text{MUSIC}}(\phi) = \frac{1}{\alpha^H(\phi) V_n V_n^H \alpha(\phi)},$$

(1.3.30)

whose maxima provides the DoAs. Once the DoAs are obtained, a beamforming algorithm is required to compute the array weights in order to steer the main lobe toward the desired direction and to place the nulls toward the undesired sources. To this purpose, the “constrained” version of the previously described LMS algorithm can be used. In particular, the constrained LMS algorithm performs in the same way as the unconstrained version, but constrains are applied to the estimated weights in order to impose a resulting gain equal to one in the desired direction.

Further mathematical details regarding the operating mode of these algorithms can be found in [60].
1.3.3 MAC layer issues in presence of advanced antenna systems

Directional radiation patterns can considerably increase the throughput of a DWN, but can also introduce new problems related to the scheduling of the communications. Most of the new problems arise from the fact that the node with a directional radiation pattern covers a limited area. This area represents the area in which the node is active and is perceived by the other nodes during their carrier sensing operations.

As mentioned before, the performance decrease is the consequence of power interference, which may degrade the communications or may interfere in the contention of the channel access. Under these conditions the node may starve, because it can not achieve the contention of the channel or it wins the contention but the following transmission fails. When the collisions and other failures occur more frequently than a successful transmission the network collapses.

The following part of the section describes the most common MAC layer problems, with a special remark on the type of the transmission (omnidirectional or directional) in which they appear.

It is worth noticing that in the presented thesis, the term “directional antenna” is used to denotes either proper directional antenna or advanced antenna system, which are able to provide directional pattern. Hence, the term “directional antenna” identifies antenna systems providing directional radiation patterns.

1.3.3.1 Typical problems

This section describes the typical problems, which may affect the nodes operating in wireless networks. This problems are presents when the nodes are equipped either with omnidirectional or with directional antennas. Typically, when directional antennas are adopted, the problems are even more relevant.

1. Hidden terminal

In omnidirectional antenna scenario the hidden terminal problem may occur when two nodes are communicating, but their carrier sensing mechanisms are not experiencing the same environment. In particular, one node is “sensing” the channel as free, while the second node is “sensing” some activity on the channel. Similar situation may occur, when the nodes are equipped with directional antenna, since their carrier sensing mechanism is working only in the range of their antennas. In particular, if in this sensing range there is no activity, the node assumes the medium as idle. The node may hence attempt a transmission towards another node. If this second node is communicating with a third node using a directional antenna, the first node becomes a hidden terminal for the second node.

The problem of the hidden terminal when omnidirectional antennas are used
is partially solved with the RTS/CTS mechanism. However, in the networks where nodes are equipped with advanced antenna systems, transmitting and receiving RTS/CTS packet in directional mode may still lead to hidden terminal situations.

2. Exposed terminal
Considering two pairs of nodes where one node of the first pair lies in the sensing range of the second pair. Since the second pair of nodes are communication with a RTS/CTS access, the node from the first pair performs a virtual carrier sensing. In this sense, its virtual carrier sensing in preventing this node from attempting a transmission toward the second node of the first pair.

3. Deafness
The deafness particularly affects nodes, which are equipped with directional antennas. In fact, the node affected by the deafness is unable to communicate to its intended receiver, because the directional antenna of the receiver is steered towards a direction away from the transmitter, e.g. towards another node which is actually engaged in a communication.

4. Muteness
The node affected by the muteness is unable to communicate to its intended receiver, because the intended receiver is performing virtual carrier sensing due to communications of other nodes. In fact, during the virtual carrier sensing the node turns off his radio and it is not able to receive incoming transmission. Only after the virtual carrier sensing expiration it will continue the listening of the wireless channel.

5. Suicide ACK
The suicide ACK is more common in scenarios where an AP equipped with directional antennas is transmitting simultaneously to several nodes, which conversely are equipped with omnidirectional antennas. If the AP transmits one frame to the first node and a second frame to the second nodes, and the two frames have different lengths, after the conclusion of the shorter frame, the receiver that sends the ACK back to the AP can collide the ongoing communication between the AP and the other node, which is not even concluded. This may introduce unfairness in uplink transmissions.

1.3.3.2 Literature overview
As introduced above, multiple simultaneous communications can be obtained thanks to the reuse of the spatial domain allowed by the use of smart antenna systems. Networks in which there are simultaneous communication between different
node pairs denote a Multi-Packet Communication (MPC) scenario. In this scenario the communications are asynchronous, since there is a lack of transmission synchronization between the different node pairs. Conversely, in the Multi-Packet Reception (MPR) scenario the central node is receiving simultaneously from different nodes, hence the network requires the time synchronization, which is typically provided by the central node. In most of the cases, the central node is the AP, while in the MPC scenario the coordinating node is not required. Furthermore, in the last years the literature proposals are dealing with Multi-Packet Transmission (MPT) scenario, in which the AP is transmitting simultaneously to different nodes. Also in MPT the synchronization between the nodes is mandatory and is adopted in the draft extension IEEE 802.11ac [92].

As mentioned in previous section, the use of smart antenna systems in order to obtain multiple communications, may lead to some drawback that have been widely considered in the research literature. In particular, the hidden terminal problems due to unheard RTS/CTS packets and asymmetry in antenna gains, which are related to the use of smart antenna systems, are deeply investigated in [8]. A multi-hop RTS approach is presented in [9], where the proposed design is inspired by the typical problems related to the adoption of directional antennas in 802.11 networks, including deafness and hidden terminal, and the obtained results reveal the throughput dependence on the topology and the traffic patterns. In several proposed protocols the channel reservation is obtained introducing tones [10–12]. Enabling the use of tones requires a control channel for tone transmission, which in most of the cases has to be transmitted in omnidirectional mode. In any case the tone signal may identify the station and must not be confused with other packets.

Among the two possible mechanisms available using the 802.11 DCF, the large part of the proposed multi-packet protocols for multi-antenna systems adopt the RTS/CTS access. In these cases usually new fields in the RTS/CTS packets are introduced [13, 14]. Besides, the novel MPR/MPC functionalities are introduced by modifying the 802.11 handshake by adding further or novel control frames [9,15–17]. In [17] a longer handshake is introduced to guarantee higher protection against interference, while in [15] a throughput increase is achieved combining a new handshake mechanism with a directional RTS/CTS. In [10] the use of adaptive antennas is extended to the slotted Aloha and to the 802.11b protocols. The authors show the performance increase achieved by the proposed protocols with respect to the omnidirectional case. Improvements on the RTS/CTS access and on the virtual carrier sensing mechanisms of the 802.11 MAC layer are presented in [13] for increasing the channel utilization in the presence of smart antenna systems. In [16] directional antennas are adopted to enable MPC, while circular directional RTS transmission is used to disseminate information concerning the availability of the different antenna beams. The directional antennas are nowadays often replaced by the advanced an-
1.4 The 802.11 OFDM physical layer

Together with the MAC layer, the IEEE 802.11 standard defines the physical layers for the implementation of wireless local area networks [23]. In particular, several modulation schemes are defined for the physical layer, while the upper layers remain unchanged. The standard allows the use of Orthogonal Frequency Division Multi-
plexing (OFDM) on the 2.4 GHz (802.11g), 5 GHz (802.11a) and 5.9 GHz (802.11p) frequency bands. OFDM is a multi-carrier modulation scheme for high-speed wireless systems, in which the transmitted symbols are modulated on subcarriers with zero ICI. This approach guarantees a better usage of the spectrum with respect to classic frequency division systems, achieving higher data rate by multiplexing the transmitted information over multiple carriers at the same time. The OFDM systems have the inherent advantage over single carrier systems in frequency-selective fading channels due to the fact that equalization can be performed directly in the frequency domain. Furthermore, OFDM systems can be designed to be very robust against Inter-Symbol Interference (ISI) caused by multipath propagation, due to its specific characteristics. In addition, OFDM systems present high spectral efficiency and low sensitivity to time synchronization errors, while they are sensitive to Doppler frequency shift and to frequency synchronization errors.

The 802.11agp OFDM physical layer provides bitrates at 6, 9, 12, 18, 24, 36, 48, and 54 Mbit/s with a channel bandwidth of 20 MHz. The standard also provides half-rate and quarter-rate modes, using a bandwidth of 10 and 5 MHz respectively, with maximum bitrates scaled accordingly. The available modulation schemes are Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), 16-Quadrature Amplitude Modulation (QAM) and 64-QAM. The OFDM symbols are transmitted on 52 active subcarriers, that includes 48 data subcarriers and 4 pilot subcarriers, on a total of 64 subcarriers. Each OFDM symbol is 4 $\mu$s long, which equals to 80 samples without oversampling, and can carry from 24 to 216 bits, depending on the chosen modulation scheme. A summary of several rate-dependent parameters is reported in Table 1.3, where $N_{BPSC}$ represents the number of bits in each OFDM subcarrier, $N_{CBPS}$ represents the number of coded bits per OFDM symbol, and $N_{DBPS}$ represents the number of data bits per OFDM symbol.

The baseband processing of the IEEE 802.11 physical layer consists of seven major steps defined in [23]. This seven steps process the data in a stream-like fashion, and define in one way the transmit chain and in the other way the receive chain. The transmit chain accepts the input data bits coming from upper layers and produces complex baseband symbols ready to be transmitted on the wireless channel. On the other hand, complex baseband samples are processed by the receive chain, which in turns sends the decoded bits to the upper layers. Subsection 1.4.2 provides the description of the transmit chain, while Subsection 1.4.3 provides the description of the blocks of the receive chain. Additional details on the transmit and receive chains are available in Section 17 of the IEEE 802.11 standard [23].
### 1.4. The 802.11 OFDM physical layer

<table>
<thead>
<tr>
<th>Data rate (Mbit/s)</th>
<th>Modulation</th>
<th>Coding rate (R)</th>
<th>Coded bits per subcarrier (N_{BPSC})</th>
<th>Coded bits per OFDM symbol (N_{CBPS})</th>
<th>Data bits per OFDM symbol (N_{DBPS})</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>BPSK</td>
<td>1/2</td>
<td>1</td>
<td>48</td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td>BPSK</td>
<td>3/4</td>
<td>1</td>
<td>48</td>
<td>36</td>
</tr>
<tr>
<td>12</td>
<td>QPSK</td>
<td>1/2</td>
<td>2</td>
<td>96</td>
<td>48</td>
</tr>
<tr>
<td>18</td>
<td>QPSK</td>
<td>3/4</td>
<td>2</td>
<td>96</td>
<td>72</td>
</tr>
<tr>
<td>24</td>
<td>16-QAM</td>
<td>1/2</td>
<td>4</td>
<td>192</td>
<td>96</td>
</tr>
<tr>
<td>36</td>
<td>16-QAM</td>
<td>3/4</td>
<td>4</td>
<td>192</td>
<td>144</td>
</tr>
<tr>
<td>48</td>
<td>64-QAM</td>
<td>2/3</td>
<td>6</td>
<td>288</td>
<td>192</td>
</tr>
<tr>
<td>54</td>
<td>64-QAM</td>
<td>3/4</td>
<td>6</td>
<td>288</td>
<td>216</td>
</tr>
</tbody>
</table>

Table 1.3: Rate-dependent parameters.

#### 1.4.1 PHY frame structure

The frame structure of the IEEE 802.11 OFDM physical layer is provided in [23]. The standard defines that every 802.11 OFDM frame is composed by:

- Physical Layer Convergence Procedure (PLCP) Preamble
  - Short training sequence
  - Long training sequence
- PLCP Header (also known as SIGNAL field)
- a variable number of DATA symbols

The standard also defines the duration of each part of the frame structure. The overall duration of the short and long training sequences is equal to 16 µs, while the duration of the SIGNAL field or of every data symbol is equal to 4 µs. In particular, the symbol length of 4µs derives from the transmission of 64 Fast Fourier Transform (FFT) complex samples (3.2 µs) preceded by the transmission of a cyclic prefix (0.8 µs). Fig. 1.11 shows the 802.11 PHY frame structure. More details on the preamble, FFT operations and of the cyclic prefix are provided in Subsection 1.4.2.11, 1.4.2.9 and 1.4.2.10, respectively.

#### 1.4.2 Transmit chain

As depicted in Fig. 1.12, the transmit chain consists of several block, whose description is here provided.
1.4.2.1 Cyclic redundancy check calculator

The 32-bits Cyclic Redundancy Check (CRC) is calculated and appended at the end of a frame received from upper layer to check the integrity of the received data in the receive chain.

1.4.2.2 SERVICE field, tail bits and pad bits

The SERVICE field consists of 16 bits and it is added at the beginning of the data frame received from the upper layer.

The first 7 bits are set to zero in order to synchronize the descrambler in the receiver. More details are provided in Subsections 1.4.2.3 and 1.4.3.9. The Tail bit field consists of 6 bits and it is added after the CRC. The Tail bit field is required to return the convolutional encoder to the “zero state” (Subsection 1.4.2.4). While the number of bits in the data frame at the end of the transmit chain shall be a multiple of $N_{CBPS}$ (the number of coded bits per OFDM symbol) at the output of the convolutional encoder, the length of the data frame at the output of the CRC calculator is extended in order to become a multiple of $N_{DBPS}$ (the number of data bits per OFDM symbol).
1.4.2.3 Scrambler

The IEEE 802.11 scrambler consists of 7 shift registers and 2 XORs as shown in Fig. 1.13. The scrambler uses the generator polynomial $S_{scr}(x_{scr})$, given by:

$$S_{scr}(x_{scr}) = x_{scr}^7 + x_{scr}^4 + 1$$  \hspace{1cm} (1.4.1)

The scrambler generates a periodic 127-bit sequence for each of the possible initial states, which are chosen for each frame in a pseudo-random fashion. Each incoming data bit is XORed with the current bit in the 127-bit sequence. The first 7 bits are the beginning of the SERVICE parameter. Those 7 bits are re-written with the initial state of the scrambler. In fact, the XOR operation between the first 7 bits, which are equal to zero, and the randomly chosen initial state produces the copy of the first 7 bits from the input to the output of the scrambler. Hence, the receiver can reconstruct the initial state chosen by the transmitter just by loading the initial 7 bits of the received sequence.

1.4.2.4 Convolutional encoder

The data bits are encoded with a convolutional encoder with coding rate of 1/2. The convolutional encoder uses the generator polynomials $g_0 = 133$ and $g_1 = 171$, with a constraint length of 7. The constraint length (symbolized by the letter $K_{conv}$) is an integer that specifies the "memory" of the code. The convolutional encoder in Fig. 1.14 consists of 7 shift registers and 3 XORs. Each input bit $X_x$ going into the encoder produces 2 output bits, $A_x$ and $B_x$, respectively. The bit denoted as $A_x$ shall be outputted from the encoder before the bit denoted as $B_x$. For example, considering the convolutional encoder input bits $X_0, \ldots, X_7$, the convolutional encoder output
Chapter 1. Distributed wireless networks: fundamentals

1.4.2.5 Puncturer

The convolutional encoder presented above uses a fixed coding rate of 1/2. The IEEE 802.11 standard also specifies other coding rates (2/3 and 3/4), which are derived from the convolutional encoder by employing puncturing patterns. Puncturing is a procedure for selecting a subset of the encoded bits to be transmitted. In fact, an increase in the coding rate is obtained by reducing the number of transmitted bits. As reported in Fig. 1.15a, the coding rate 3/4 is obtained as follows: considering the output of the convolutional encoder equal to $A_0, B_0, \ldots, A_8, B_8$, the puncturer omits the bits $B_1, A_2, B_4, A_5, B_7$ and $A_8$. This pattern is obtained if for every 6 bit from the encoder 2 of them are discarded. Conversely, as reported in Fig. 1.15b for the coding rate 2/3, the puncturer omits 1 bit every 4 bits produced by the encoder. In particular, considering the $A_0, B_0, \ldots, A_5, B_5$, the bits $B_1, B_3$ and $B_5$ are omitted. The puncturing operations are visually described in Fig. 1.15.

1.4.2.6 Interleaver

The aim of the interleaver is to avoid the effect of deep fading on a set of closer subcarriers. In fact, if one subcarrier faces deep fading, it is very likely that nearby subcarriers will be also affected. Hence, the interleaver is used to spread adjacent coded bits on non-adjacent subcarriers and to alternate them in an optimal way on the most and least significant bits of the subcarrier constellation. In the IEEE 802.11 standard the interleaver operates at OFDM symbol level with a block size of 48, 96,
1.4. The 802.11 OFDM physical layer

192, or 288 bits. To this aim, the bits of each block are reordered in two-step permutation. Let $k_{\text{int}}$ denote the index of the coded bit before interleaving, $i_{\text{int}}$ denote the index after the first permutation, and $j_{\text{int}}$ denote the index after the second permutation. The value $s_{\text{int}} = \max(N_{\text{BPSC}}/2, 1)$. The first index permutation is defined by:

$$i_{\text{int}} = (N_{\text{CBPS}}/16)(k_{\text{int}} \mod 16) + \lfloor k_{\text{int}}/16 \rfloor,$$

where $k_{\text{int}} = 0, 1, ..., N_{\text{CBPS}} - 1$. The second index permutation is defined by:

$$j_{\text{int}} = s_{\text{int}} \cdot \lfloor i_{\text{int}}/s_{\text{int}} \rfloor + (i_{\text{int}} + N_{\text{CBPS}} - 16 \cdot i_{\text{int}}/N_{\text{CBPS}}) \mod s_{\text{int}},$$

where $i_{\text{int}} = 0, 1, ..., N_{\text{CBPS}} - 1$. The $\lfloor \cdot \rfloor$ represents the rounding function, while $\mod$ represents the modulus function. In the first step, adjacent coded bits are reordered to be mapped on non-adjacent subcarriers. In the second step, adjacent coded bits are mapped alternately into less and more significant bits of the subcarrier constellation. These indices can be calculated in advance as a function of the data rate, so that the interleaver basically reorders the input bits based on a predefined permutation pattern.

1.4.2.7 Symbol mapper

The OFDM subcarriers are modulated by using BPSK, QPSK, 16-QAM, or 64-QAM modulation schemes, depending on the selected data rate. The data bits coming from the interleaver are divided into groups of $N_{\text{BPSC}}$ bits, i.e. 1, 2, 4, or 6 bits according to the data rate. Afterwards, the grouped bits are converted into complex symbols representing BPSK, QPSK, 16-QAM, or 64-QAM constellation points. The conversion is performed according to Gray-coded constellation mappings, where the bit $b_0$ is the earliest in the bit stream. The complex output values are multiplied by a normalization factor $K_{\text{MOD}}$, which depends on the modulation scheme, as reported in Table 1.4. The purpose of the normalization factor is to achieve the same average power for all mappings. Considering $b_0, \ldots, b_5$ the bits received from the interleaver the constellation points are obtained using real (I-out) and imaginary (Q-out) values.
Table 1.4: Normalization factor $K_{\text{MOD}}$.

<table>
<thead>
<tr>
<th>Modulation</th>
<th>$K_{\text{MOD}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK</td>
<td>1</td>
</tr>
<tr>
<td>QPSK</td>
<td>$\frac{1}{\sqrt{2}}$</td>
</tr>
<tr>
<td>16-QAM</td>
<td>$\frac{1}{\sqrt{10}}$</td>
</tr>
<tr>
<td>64-QAM</td>
<td>$\frac{1}{\sqrt{142}}$</td>
</tr>
</tbody>
</table>

in Table 1.5 for BPSK modulation, in Table 1.6 for QPSK modulation, in Table 1.7 for 16-QAM modulation, and in the Table 1.8 for 64-QAM modulation. The values of the obtained complex sequence are multiplied for the corresponding $K_{\text{MOD}}$ factor.

### 1.4.2.8 Pilot insertion

In each OFDM symbol, four of the subcarriers are used by pilot signals. The pilot signals are used for a robust coherent detection against frequency offset and phase noise. The four pilot signals are placed in the subcarriers number -21, -7, 7, and 21. The pilots are BPSK modulated (imaginary part is equal to zero) by a pseudo-binary sequence, in order to prevent the generation of spectral lines. The sign of the pilot signal carried on subcarrier 21 is inverted with respect to the other pilot signals. The sign of the pilot subcarriers is controlled by the sequence $p_{0:126}$, which is described below. It is 127 elements long, and it is wrapped when necessary.

$$p_{0:126} = \{1, 1, 1, 1, -1, -1, 1, -1, -1, 1, 1, 1, 1, -1, -1, 1, -1, 1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, -1, 1, -1, -1, 1, -1, 1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, -1, 1, -1, 1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, 1, -1, 1, -1, 1, -1, -1, 1, 1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, -1, 1, -1, -1, 1, 1, -1, 1, 1, -1, -1, 1, 1, -1, 1, 1, -1, 1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, -1, 1, -1, 1, -1, -1, 1, -1, 1, -1, 1, -1, -1, 1, -1, -1, 1\} \quad (1.4.4)$$

This pseudorandom sequence can be generated by the scrambler defined in Fig. 1.13, when the all ones initial state is used. Once the scrambled sequence is obtained all 1’s are replaced with $-1$, while all 0’s are replaced with 1. Each element of the sequence is used for one OFDM symbol. The first element, $p_{0}$, multiplies the pilot subcarriers of the SIGNAL symbol, while the elements from $p_{1}$ on are used for the DATA symbols.
1.4. The 802.11 OFDM physical layer

Table 1.5: BPSK encoding table.

<table>
<thead>
<tr>
<th>Input bit $(b_0)$</th>
<th>I-out</th>
<th>Q-out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1.6: QPSK encoding table.

<table>
<thead>
<tr>
<th>Input bit $(b_0)$</th>
<th>I-out</th>
<th>Input bit $(b_1)$</th>
<th>Q-out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1.7: 16-QAM encoding table.

<table>
<thead>
<tr>
<th>Input bit $(b_0 b_1)$</th>
<th>I-out</th>
<th>Input bit $(b_2 b_3)$</th>
<th>Q-out</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>-3</td>
<td>00</td>
<td>-3</td>
</tr>
<tr>
<td>01</td>
<td>-1</td>
<td>01</td>
<td>-1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1.8: 64-QAM encoding table.

<table>
<thead>
<tr>
<th>Input bit $(b_0 b_1 b_2)$</th>
<th>I-out</th>
<th>Input bit $(b_3 b_4 b_5)$</th>
<th>Q-out</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>-7</td>
<td>000</td>
<td>-7</td>
</tr>
<tr>
<td>001</td>
<td>-5</td>
<td>001</td>
<td>-5</td>
</tr>
<tr>
<td>011</td>
<td>-3</td>
<td>011</td>
<td>-3</td>
</tr>
<tr>
<td>010</td>
<td>-1</td>
<td>010</td>
<td>-1</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
<td>110</td>
<td>1</td>
</tr>
<tr>
<td>111</td>
<td>3</td>
<td>111</td>
<td>3</td>
</tr>
<tr>
<td>101</td>
<td>5</td>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>100</td>
<td>7</td>
<td>100</td>
<td>7</td>
</tr>
</tbody>
</table>
1.4.2.9 Inverse FFT

For every OFDM symbol, 48 modulated subcarrier (complex values) obtained from the symbol mapper, 4 pilot subcarriers (with the imaginary part equal to zero), and 12 null subcarriers are placed together and passed to the 64-point Inverse Fast Fourier Transformation (IFFT) block. The IFFT inputs are numbered from 0 to 63. The pilot sequences, which correspond to the subcarriers -21, -7, 7, and 21, are positioned to IFFT inputs numbered with 43, 57, 6, and 21, respectively. The subcarriers from 1 to 26 are positioned to the IFFT inputs numbered from 1 to 26, while the subcarriers from -26 to -1 are positioned to the IFFT inputs numbered from 38 to 63. The IFFT inputs 0, and in the interval from 27 to 37 have null subcarriers. The IFFT block outputs 64 time domain complex values. The advantage of using IFFT is that the system does not need $K_{ss}$ oscillators to transmit $K_{ss}$ subcarriers, where $K_{ss}$ represents the number subcarriers. Besides, in order to create the OFDM symbol $K_{ss}$ serial data symbols are converted into $K_{ss}$ parallel data symbols using the symbol mapper and the IFFT.

1.4.2.10 Cyclic prefix insertion

The OFDM symbols propagate through the wireless channel typically through different paths and reach the destination at different instants. Hence, the multipath propagation can introduce ISI and, in case of Doppler shift effects, ICI at the receiver side. In OFDM systems, in order to cope with ISI, the last part of the OFDM time domain waveform is replicated to the front in the so-called cyclic prefix, as depicted in Fig. 1.16 illustrates its use. The value $T_{OFDM}$ represents the duration of the whole OFDM symbol, which consists of the time required to transmit the FFT samples $T_{FFT}$ and of the time required to transmit the cyclic prefix $T_g$. The value $\tau_{\text{max}}$ accounts for the worst-case delay spread of the target multipath environment. At the receiver side, $T_{\text{samp}}$ represents a certain position within the cyclic prefix chosen as the sampling
1.4. The 802.11 OFDM physical layer

starting point. At the receiver, if the term \( \tau_{\text{max}} \) satisfies the following criteria:

\[
\tau_{\text{max}} < T_{\text{samp}} < T_g
\]  

(1.4.5)

the previous symbol will only have effect over samples within the interval \([0, \tau_{\text{max}}]\).
Thus, if the overall length of the impulse response is shorter than the cyclic prefix, there will be no overlapping between consecutive OFDM symbols yielding zero intersymbol interference. Moreover, the use of the cyclic prefix also helps in maintaining the orthogonality among subcarriers. A time synchronization error between the transmitter and receiver may lead to a shift of the symbol timing, causing distortion due to the overlapping of consecutive symbols. The cyclic prefix exploits the periodicity of the FFT, so that a timing error in the time domain only results in a phase shift in the frequency domain, which is corrected at the receiver side by the equalization. The cyclic prefix is also useful to smoothen the transition between two consecutive periods of FFT, which may introduce relevant spectral sidelobes of the transmitted waveform. In the whole PLCP subsystem three kinds of \( T_g \) are defined:

- for the short training sequence (0\( \mu \)s)
- for the long training sequence (\( T_{\text{GI2}} = 1.6 \mu \)s)
- for the SIGNAL and DATA OFDM symbols (\( T_{\text{GI}} = 0.8 \mu \)s)

1.4.2.11 Preamble and SIGNAL field insertion

The OFDM PHY layer data frames are preceded by the PLCP preamble and the PLCP header, which are used by the receiver for timing synchronization purposes and for providing information on the length. The PLCP preamble is composed by:

- short training sequence: 10 repetitions of a “short training symbol”
- long training sequence: 2 repetitions of a “long training symbol”

Mainly, the short training sequence is used for Automatic Gain Control (AGC) convergence, for diversity selection, for timing acquisition, and for coarse frequency acquisition in the receiver. The long training sequence, instead, is used for channel estimation and for fine frequency acquisition in the receiver. More details of the preamble use are provided in Section 1.4.3, which describes the physical layer operation related to frame detection and synchronization.

Every OFDM short training symbol uses 12 subcarriers, which are modulated by the elements of the sequence \( S_{-26,26} \), given by:

\[
S_{-26,26} = \frac{\sqrt{13}}{6} \{0, 0, 1 + j, 0, 0, 0, -1 - j, 0, 0, 0, 1 + j, 0, 0, 0, -1 - j, 0, 0, 0, \\
-1 + j, 0, 0, 0, 1 + j, 0, 0, 0, 0, 0, 0, 0, -1 + j, 0, 0, 0, -1 - j, \\
0, 0, 0, 1 + j, 0, 0, 0, 1 + j, 0, 0, 0, 1 + j, 0, 0, 0, 1 + j, 0, 0}\}
\]  

(1.4.6)
Since the resulting OFDM symbol uses 12 out of 52 subcarriers, the multiplication by the factor of $\sqrt{\frac{13}{6}}$ is required to normalize the average power of the symbol. In the sequence $S_{-26,26}$ only spectral lines multiple of 4 have nonzero amplitude. This results in a duration of the short training symbol of $0.8\,\mu s$. The short training symbol is repeated 10 times, so that the whole sequence is $8\,\mu s$ (2 OFDM symbols) long.

The long training symbol consists of 53 subcarriers (including a zero value at direct current), which are modulated by the elements of the sequence $L_{-26,26}$, given by:

$$L_{-26,26} = \{1, 1, -1, -1, 1, 1, -1, 1, 1, 1, 1, -1, -1, -1, 1, 1, -1, 1, 1, -1, 1, 1, -1, 1, 1, -1, 1, 1, -1, 1, 1, -1, -1, -1, 1, 1, -1, 1, 1, -1, 1, 1, -1, 1, 1, -1, -1, -1, 1, 1, -1, 1, 1, -1, -1, 0, 1, -1, -1, -1, 1, -1, -1, 1, -1, 1, 1, 1, 0\}$$

(1.4.7)

The resulting duration of a single long training symbol is equal to $3.2\,\mu s$. Hence, the long training sequence consists of 2 repetitions of the long training symbol and considering also the cyclic prefix ($1.6\,\mu s$), the total duration of the long training sequence is equal to $8\,\mu s$ (two OFDM symbols).

The PLCP preamble is followed by the PLCP header, which is also called SIGNAL field. The SIGNAL field is transmitted at basic bitrate and contains information regarding the bitrate of the DATA symbols (expressed in 4 bits), the length of the data frame (expressed in 12 bit), and other reserved bits, which are typically equal to zero. The SIGNAL field is one OFDM symbol long and the contained data are not scrambled.

### 1.4.3 Receive chain

The reception of the OFDM PHY layer frame consists of two main states. In the first state the receiver tries to synchronize with the transmitted signal; when proper timing is acquired, the receiver tries to decode the PHY layer frame. In particular, when the synchronization with the training sequence is acquired, the preamble is used for estimating the frequency offset between transmitter and receiver and to obtain a first estimate of the channel impulse response. The flow of the receive chain is depicted in Fig. 1.17.

#### 1.4.3.1 Frame detector

The frame detector detects the presence of the preamble in order to identify the start of a PHY layer frame. The Schmidl & Cox (S&C) algorithm is one of the most popular algorithm adopted for the frame detection and carrier frequency offset estimation [24]. The start of the frame is identified performing auto-correlation on the received samples. As previously reported, the preamble begins with the repetition of ten short training symbols, hence the auto-correlation between the signal and
1.4.3.2 Coarse carrier frequency offset estimator and compensator

The OFDM is vulnerable to frequency shifts, which derive from Doppler frequency shift and frequency impairments between the oscillator at the transmitter and receiver. Frequency shifts destroy the orthogonality between the subcarriers and cause inter-subcarriers interferences. Furthermore, this effect introduces intersymbol interferences at the output of the demodulator. The carrier frequency offset (CFO) estimation is based on the S&C approach, described in [24]. The total frequency offset consists of two parts: the integer one and the fractional one. For this reason, the frequency offset estimation algorithm performs in two steps: the coarse and fine CFO. In particular, the two types of frequency offset are separately identifying in two steps. This subsection will provide information on the coarse CFO estimation, while information on the fine CFO estimation are provided in Subsection 1.4.3.4. The coarse frequency estimation is adopted in order to estimate a frequency offset bigger than the subcarrier spacing. The coarse estimation performed cross-correlation on the short training sequence of the frame preamble before the symbol timing.

The Coarse CFO compensation is performed by multiplying the samples by a complex exponential value, which represents the reciprocal of the estimated frequency offset.
1.4.3.3 Symbol timing

The symbol timing block is based on a robust cross-correlation algorithm, thoroughly described in [25]. Since the last short training symbol and the cyclic prefix of the first training symbol are known and they are unique, their concatenation can be considered as a training symbol. The symbol timing block aims to uniquely determine the instant of this training sequence with higher reliability due to the cross-correlation peak. The frame position is then chosen within an observation window of 128 samples.

1.4.3.4 Fine carrier frequency offset estimator and compensator

The fine carrier frequency offset estimation algorithm operates in the time domain. As mentioned in Subsection 1.4.3.2 the fine CFO estimator is able to estimate a fractional part of the frequency offset, thus smaller than a half of the subcarrier spacing. The estimation is performed with the auto-correlation on the long training sequence. In particular, the auto-correlation is calculated between the first long training symbol and the second. After this estimation, the fractional part of the frequency offset is removed from the input signal and in this manner the FFT output is almost free of the ICI.

1.4.3.5 Cyclic prefix remover, FFT, block equalizer and comb equalizer

Since the cyclic prefix was added at the transmission side, it has to be removed at the receiver side. Once the cyclic prefix is removed the 64 point FFT is performed on the remaining samples. In frame-based communication system, training sequence is used at the beginning of frame. Since the frame is short, the channel is assumed affected by slow fading over the whole frame. Under this assumption, the channel estimation is divided in two steps. The first step is performed by the block-type channel estimation at the beginning of the frame, while the second step is performed by the comb-type channel estimation in order to satisfy the need for equalizing when the channel changes even from one OFDM block to the subsequent one. The block-type channel estimation methods relies on the long training sequence symbols to obtain a first estimate of the channel estimation, while the comb-type pilot channel estimation uses the pilot tones present in each OFDM symbol to track the variations and to remove the residual phase shift of each symbol. The equalizer performs the ratio between the subcarriers of the known long training sequence and the average on the subcarriers of the two received long training symbols, as described in [26]. In particular, the simplest, zero forcing block-type channel estimation of impulse channel response \( H_{ZF} \) is defined by:

\[
H_{ZF} = [(x_{kkss} / s_{kkss})]^T \quad k_{ss} = 0, \ldots, K_{ss} - 1, \quad (1.4.8)
\]
1.4. The 802.11 OFDM physical layer

where $x_{k_{ss}}$ is $k_{ss}$-th subcarrier of the received signal and $s_{k_{ss}}$ is the $k_{ss}$-th subcarrier of known transmitted signal. Without using any knowledge of the statistics of the channels, the estimators are calculated with very low complexity.

In comb-type pilot based channel estimation for each transmitted symbol a defined number pilot signals are uniformly inserted into subcarriers one apart from each other. The receiver is aware of the pilot locations, the pilot values, and the received signal. The estimates of the channel conditions at the pilot subcarriers are calculated by the use of Least Square (LS) estimator with a first-order linear interpolation. In particular, the first-order linear interpolation is used to estimate the channel at data subcarriers, where the vector containing only the pilot tones is interpolated to the vector of OFDM symbol in the frequency domain, without using additional knowledge of the channel statistics.

1.4.3.6 Symbol demapper

Once the complex values of each subcarriers in the frequency domain has been equalized, the symbol demapper performs a soft estimation of the complex values into sequence of bits. The number of bits corresponding to each symbol depends on the adopted bitrate.

1.4.3.7 Deinterleaver and depuncturer

The deinterleaver performs the inverse role of the interleaver. It has to reorder the bit moved by the interleaver. As the interleaving, also the deinterleaving is performed in two steps. The first index permutation of the deinterleaver is defined by the rule:

$$i_{deint} = s_{deint} \cdot \lfloor j_{deint}/s_{deint} \rfloor + (j_{deint} + \lfloor 16 \cdot j_{deint}/N_{CBPS} \rfloor) \mod s_{deint}, \quad (1.4.9)$$

where $j_{deint} = 0, 1, \ldots, N_{CBPS} - 1$ and $s_{deint} = s_{int}$ defined in Subsection 1.4.2.6. The second index permutation is defined by the rule:

$$k_{deint} = 16 \cdot i_{deint} - (N_{CBPS} - 1) \lfloor 16 \cdot i_{deint}/N_{CBPS} \rfloor, \quad (1.4.10)$$

where $i_{deint} = 0, 1, \ldots, N_{CBPS} - 1$.

The depuncturer performs the reverse operation of the puncturer. In particular, based on the adopted bitrate, the depuncturer has to add some dummy bit to the sequence obtained from the deinterleaved reinstating to the coding rate 1/2. Fig. 1.18 shows an example, with is related to Fig. 1.15 provided in Subsection 1.4.2.5. The bits $y_x$ represent the coded bits ready to be send to the Viterbi decoder.

1.4.3.8 Viterbi decoder

The Viterbi algorithm is a widespread method for decoding convolutional codes. It is commonly used in many digital communication systems due to its simplicity.
Chapter 1. Distributed wireless networks: fundamentals

and high performance. The Viterbi decoder is based on the principles of maximum
likelihood sequence estimation. It typically uses Euclidean (soft decisions) or Ham-
ing (hard decisions) distances as a decision metric. In the presented work the soft
decision metrics is adopted and the constraint length is set to $K_{\text{conv}} = 7$ as adopted
in Subsection 1.4.2.4. The algorithm of the Viterbi decoder is divided into two steps:
the forward-pass and the traceback. The fist step computes the path metrics, while
the second step returns traces back the received sequence along the most likely path.

1.4.3.9 Descrambler

The descrambler basically operates as the scrambler previously described in Sub-
section 1.4.2.3. The descrambler is initialized by the first 7 bit in which the scrambler
coded the scrambling/descrambling sequence at the transmission.

1.4.3.10 CRC checker

The CRC calculation is performed once all the bits of the frame are received. The
CRC checksum is compared to the one written at the end of the receiver frame in
order to prove the validity of the frame.
Adopted software packages

This section introduces the software, applications and drivers used in this thesis. The Network Simulator 2 is firstly described. Subsequently, the set of tools used for network measurements is presented in the second part of the chapter. Finally, the third part describes the software-defined radio concept and introduces the platform and the software tools used for the implementation of the IEEE 802.11ag physical layer.

2.1 Network simulations

This section introduces the network simulation tool that has been adopted and extended in order to assess the performance of the protocols proposed in this thesis. In particular, this section concerns network simulation 2. The network simulator 2 is a very popular open-source event-driven simulator designed for the research in communication networks. Several releases were published, and the research activity of various groups provided a large number of ns2 extensions and modules, including known and novel protocols.

Ns2 provides an executable command that takes as input a Tool Command Language (Tcl) script, and produces a trace file as simulation output. The ns2 language architecture is depicted in Fig. 2.1.

The ns2 application consists of two parts: the frontend and the backend. The frontend is developed in Object-oriented Tcl (OTcl) and is used to set up the simulation environment by assembling and configuring the simulation object. The backend is developed in the C++ language and defines the mechanisms for executing the simulation. The connection between the C++ domain and the OTcl domain is provided by the Tcl with CLasses (TclCL) interface. When an object is created in the OTcl domain, a so-called shadow object is also created in the C++ domain. In fact, the simulation objects are mapped in the OTcl domain, while the functions and procedures are mapped in the C++ domain. Hence, the TclCL connects the objects with the functions. The C++ and OTcl class hierarchies, which have one-to-one correspondence, are referred to as the compiled hierarchy and the interpreted hierarchy, respectively.
Particularly, the class (or member) variable and function in the compiled hierarchy correspond, in the interpreted hierarchy, to instance variables and instance procedures, as depicted in Fig. 2.2.

The simulations in ns2 are performed in two steps: the network configuration and the simulation. In the first step, ns2 reads the Tcl configuration file to create the network components and to configure them according to the simulation design. The values of the interpreted class variables are acquired by the corresponding C++ compiled class. The configuration file is also used for scheduling additional events requested by the simulation scenario, such as the delayed begins and conclusions of node transmissions. In the second step, ns2 performs the simulation maintaining the simulation clock and executing the events chronologically. The simulation stops when the clock reaches the threshold defined in the configuration file as the simulation end. As previously reported, ns2 during the simulation collects the results and presents them in specific trace files. The trace files can collect data at various layers, which are specified in the configuration file.

As mentioned before, ns2 is an event-driven simulator, hence the system evolution is considered when an event occurs. In the time-driven simulators, the time intervals between two events in which the system evolution is evaluated are constant. Conversely, in the event-driven simulators, time interval between two events does not need to be constant. In particular, the simulation load is smaller considering only the instants of evolution instead of repeating the evaluation in the instances in which the system will remain unchanged. An example of the time-driven and event-driven time lines is reported in Fig. 2.3. In ns2 every event, which is declared as a C++ class, defines an event handler, a corresponding reference time, and an unique event ID. The event handler accounts for the node in which the event will occur and the operation that will be executed. The temporal continuity is observed, since each event contains the link to the previous and next event. The sequence of the event can be modified by adding, e.g. scheduling, new events. In the same way, the events which will not occur can be removed from the time line. Once an event is dispatched,
the corresponding handler executes the evolution on the node state. When the event expires or is dispatched, the scheduler inverts the sign of event ID preventing the prevent re-elaboration.

The ns2 simulation tool is developed to analyze networks concerning the entire protocol stack. In fact, ns2 presents an extensive set of protocols, which include layers from the physical to the application ones. In every node the singular layer denotes a C++ class, which reproduces the layer behavior. When the packet is received or transmitted, ideally, it passed through the various layers following the downstream or the upstream. In ns2 the behavior is similar, since the data are passed from one class to another following the protocol stack architecture. In fact, each network node consists of several classes, which are connected together reproducing the whole protocol stack. The classes, which represent the higher layers are referred to as agents, which typically are adopted to generate traffic and collect the traces. It is worth noticing, that during the simulation, for every node several classes are instanced, hence the evolution of each node concerns only the classes belonging to the node itself.

At the physical layer the nodes are connected with a common class, which represents the channel. Several propagation model are considered in ns2, while a lot of channel propagation model extensions have been proposed in literature. One of the most popular is the two-ray ground model, which accounts for the transmission on receiver power, the distance between the transmitter and the receiver, the path-loss factor and the heights of the adopted antennas. At the physical layer ns2 introduces different antenna systems. The most adopted one is the omnidirectional antenna. However, the generic antenna class denotes the antenna gain for each direction (360 values to cover every direction), hence several power gain patterns can be imported in the model.
Chapter 2. Adopted software packages

The IEEE 802.11 MAC protocol implementation is managed as a state machine in which the states follow the evolution of the transmitting and receiving procedures. In particular, the MAC class implements the basic access and the RTS/CTS access concerning also the backoff procedures. The event in which the backoff time reaches the value zero is defined by scheduling the expiration time. While, when the backoff counter is frozen, the scheduler defines the resume event, and the expiration time is postponed by removing and adding a new event. The physical carrier sensing is performed monitoring the state of a channel class flag, which is enabled when at least one node is in the transmission state. On the other hand, the virtual carrier sensing is performed by scheduling events on the base of the duration field contained in the received packets. Furthermore, the collision event is scheduled for all the nodes, when the simulator notices that more than one node is in the transmitting state.

As mentioned before, in ns2 every network object is represented by a C++ class. Hence, every network object is represented by its generic class from which can derive modified classes. The C++ polymorphism concept allows one to modify and/or extend existing classes introducing new features and behaviors. Additional information regarding ns2 simulation tools can be found in [22].

2.2 Network measurements

This section provides a brief description of the software tools adopted to perform measurements on the observed networks. In particular, this section concerns the Multiband Atheros Driver for Wireless Fidelity (MADWiFi), and the Iperf and TCP-Dump application. The MADWiFi driver is used for two reasons. Firstly, the MAD-WiFi driver, being open source, can be modified to enable QoS features in adhoc mode, and, secondly, it allows one to monitor the channel activity using cards based on Atheros chipset. Iperf is used to measure the network throughout when several nodes are contending for the same channel. Since, the traffic generation rate in Iperf can be tuned, it is possible to collect throughput measurements in both saturated
and non-saturated networks. Finally, the packet analyzer TCPDump is adopted in order to access the packet information regarding the physical and MAC layers.

2.2.1 MADWiFi driver

MADWiFi is an open source driver that supports many consumer wireless cards based on Atheros technology. MADWiFi relays on a binary Hardware Access Layer (HAL) that manages many of the device specific operations. The main task of the HAL is to allow the communication between the operating system stack and the card chipset. The HAL contains an 802.11 MAC implementation and chipset specific calls in order to interact with the hardware. Thanks to the internal features supporting prioritization mechanisms, this driver enables the modification of the QoS parameters defined in the 802.11e extension. Besides, in the present MADWiFi implementation, the HAL module verifies that invalid QoS parameters values are not passed to the card hardware.

MADWiFi allows the wireless card to act in different operation modes: adhoc, monitor, station, AP, Wireless Distribution System (WDS) and adhoc demo. In Atheros-based wireless devices several transmission queues are available. In fact, two high-priority queues are reserved to manage control packets, such as beacons and probe requests, while the other queues are used to send data traffic. Each data queue is described by two sets of EDCA parameters: the first one is employed for AP parameter set, while the second one is considered for station parameter set. Four data queue are associated to the 4 AC of the IEEE 802.11e standard. At the user space level, the EDCA parameters of each queue can be set by using the MADWiFi command `iwpriv`:

```
# iwpriv cwmin [queue number] [ap/sta set] [value]
# iwpriv cwmax [queue number] [ap/sta set] [value]
# iwpriv txoplimit [queue number] [ap/sta set] [value]
# iwpriv aifsn [queue number] [ap/sta set] [value]
```

The parameter [queue number] takes the values from 0 to 3 for BE_AC, BK_AC, VI_AC and VO_AC, respectively. The parameter [ap/sta set] takes the values 0 or 1 to define the AP set or the station set, while, [value] represents the value to set. The range of values is different for the four EDCA parameters. For cwmin and cwmax the range goes from 1 to 10 and it is expressed as the exponent with base two of the value. The aifsn value is an integral number from 0 to 10, while the txoplimit is expressed in microseconds, starting from 224 microseconds. More details can be found in [30].

Unfortunately, in the current MADWiFi release, the EDCA QoS settings can be enabled only in AP and station modes, where the AP manages the access and the priority of each node. To guarantee QoS in adhoc mode some adjustments to the
MADWiFi source code are necessary. The performed modification on the driver will be described in the Section 3.3.

### 2.2.2 Iperf

Iperf is a client-server application, which is suitable for throughput, delay, and bandwidth measurements. The client side of the application generates packets and sends them to the server side. The server side receives the packets and collects their statistics. The client side is able to generate packets of different sizes for Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) transport level protocols at different transmission rates. At the client side the user can specify the Type of Service (ToS) field of the generated packet. Hence, the generated packet is transmitted by the physical layer with the corresponding IEEE 802.11e AC. In particular, the hexadecimal values of 0x20, 0x00, 0xa0 and 0x30 correspond to the BK_AC, BE_AC, VI_AC, and VO_AC, respectively. The server side reports the collected statistics at the user's terminal, providing information on the throughput, bandwidth and jitter. Several instances of Iperf client can run on the same or on different machines forwarding the packets to the same Iperf server. Thus, the measurements can include several nodes of the network.

### 2.2.3 TCPDump

TCPDump is a widely diffused packet analyzer that relies on the portable C/C++ libraries libpcap for network traffic capture. This application is useful to store information related to the received packets, which are added by the network interface card. TCPDump provides a lot of details when the network interface is working in promiscuous/monitor mode. In this particular configuration mode wireless chipsets are able to provide additional headers of the received packets. Hence, the received packet and additional physical layer information are forwarded by the kernel to the user space. The physical layer information, added by the Atheros chipsets, concerns the strength of the received power, the instant of reception begin, the noise level, and other transmission characteristic parameters. In linux-based systems the users are able to properly monitor the activity on the sensed channel by setting the system's kernel parameters dev.ath0.rawdev to one and dev.ath0.rxfilter to the hexadecimal value 0x01ff.
2.3 Software-defined radio

2.3.1 Software-defined radio concept

The software-defined radio (SDR) concept was first introduced in the 70s. The idea is to use software routines, running on programmable hardware for the baseband signal processing required in a telecommunication system (Fig. 2.4). The first SDR projects were carried out by the military, as the required equipment was very expensive. However, nowadays the SDR technology is becoming more and more popular due to the availability and affordable price of the required RF and hardware equipment.

In Commercial Off-The-Shelf (COTS) hardware, the baseband digital signal processing is usually carried out by Application Specific Integrated Circuits (ASICs). This hardware radio devices can only work with on predefined modulation schemes and/or coding techniques, and can be hardly customized. On the other hand, there are several available technologies, such as Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA), General Purpose Processors (GPPs), whose reconfigurability and flexibility make them the perfect choice for a complete reconfigurable system.

SDR offers three major advantages, that is, reconfigurability, modularity, and ease of testing. First, the same RF frontend can be reused for different modulation schemes. Second, the developed blocks and components can be easily reused or adapted in different transmission systems. Finally, a dynamic, complex system can be designed from the ground up and reliably tested in real-life environments, allowing rapid prototyping and debugging.

In the last years, two approaches for SDR have become most popular. The first uses development boards based on DSPs and FPGAs, also including sometimes one or more powerful Central Processing Units (CPUs), to perform all the modulation and demodulation processes. The second group uses a streamlined RF frontend for analog-to-digital conversion and basic DSP operation, such as interpolation or...
decimation, I/Q imbalance compensation, direct current offset removal, streaming the baseband sampled data to a host Personal Computer (PC) to perform the most demanding operations.

### 2.3.2 Ettus Research USRP N210

One of the most popular line of SDR platforms available on the market is the Universal Software Radio Peripheral (USRP) by Ettus Research [32]. The first USRP model, also known as USRP1, was the first affordable, high-performance software-defined radio device featuring an Universal Serial Bus (USB) 2.0 connection towards a host PC, interchangeable RF frontends, and support for 2x2 MIMO. Their flagship device, named USRP N210, uses an FPGA Xilinx Spartan-3A DSP 3400 for base DSP processing. It is large enough to also host custom logic for direct on-board processing. The interface analog-to-digital section which controls a dual 100MS/s with 14-bit of resolution analog-to-digital converter and a dual 400 MS/s with 16-bit of resolution digital-to-analog converter. The software running on the host computer controls can be interfaced to the USRP hardware by means of the Universal Hardware Driver (UHD), a cross-platform middleware which abstracts the interface to the device and takes care of the streaming from and to the host PC. The Gigabit Ethernet link can sustain continuous full-duplex streaming to 50 MS/s of complex data. Ettus Research also offers a wide choice of transceiver and receive-only daughterboards, covering RF bands from 0 to 5 GHz.

### 2.3.3 SIMD instruction set

Single-Input Multiple-Data instructions (SIMDs), according to Flynn’s taxonomy, can execute the same operation on multiple data items simultaneously. In particular, in the traditional scalar processing one operation produces one result, while in the SIMD processing one produces multiple results. Since the data are elaborated in blocks rather than in single values, efficient implementation of algorithms that per-
forms the same operation on large sets may be obtained leading to a certain level of parallelization and vectorization. The SIMD approach is used since the early 70s in scientific computing and has found very large application also for gaming computing, audio/video processing, Finite Impulse Response filtering, Discrete Cosine Transform, Fast Fourier Transform, and so on.

The Intel grouped SIMD instructions in an extension for existing instruction sets and named it Streaming SIMD Extension (SSE). The SSE instruction set has been improved during the years to include new instructions tailored to audio and video processing. Its evolution is here briefly reported:

**MMX** (1996): 64-bit instructions using existing floating point registers

**SSE** (1999): 8 or 16 128-bit dedicated registers, limited to single-precision floating point

**SSE2** (2001): added support for integers (8-, 16-, 32-, 64-bit) and double precision

**SSE3/SSSE3** (2004-2006): added more useful instructions (e.g. MAC)

**SSE4** (2006): not supported by all architectures (Atom)

**AVX** (2008): new standard, uses 3-operand instructions on 256-bit wide instructions (only supported by new i3-i5-i7 architectures)

The SIMD instructions can be executed by means of a special prefix in the operational code of the machine language instruction. From the programmer’s perspective, one could use inline assembly to call the specific operation, but newer compiler also feature seamless integration of special instructions in C and C++ code. These wrapping functions are called intrinsics, and allow the compiler to employ optimizations automatically on the generated machine code (differently from inline assembly, which should be optimized by hand). The base size of an MMX/SSE/AVX operand can be 64, 128 or 256 bits, according to the size of the used CPU special-purpose register. As an example, a 128-bit XMM register can load 2 double-precision floating point values, 4 single-precision floating point values, 4 32-bit integers, 8 16-bit integers or 16 8-bit integers, or a single 128-bit bitmask, as reported in Fig. 2.6. In some cases, one limitation is that SIMD instructions can only operate on aligned data, that is, it is necessary to carefully tailor the algorithms and the memory layout, as an unaligned access needs several cycles to complete and may cause a significant performance degradation.

The intrinsics of the corresponding machine code instructions can be grouped in the following groups of operations, which description is here reported:
Figure 2.6: Layout of XMM register. $b =$ integer byte (8 bits), $w =$ integer word (16 bits), $dw =$ integer doubleword (32 bits), $qw =$ integer quadword (64 bits), $ps =$ single-precision floating point (32 bits), $pd =$ double-precision floating point (64 bits).

**Load and store**: to move data between memory and registers;

**Constants**: to set defined values in registers;

**Arithmetic**: to perform arithmetical operations on data in registers;

**Comparison**: to compare registers

**Conversion**: to convert the types of data present in registers

**Shuffles**: to move data inside registers

The translation of the intrinsics is performed by the compiler during the production of the optimized assembly code. Some of this intrinsics have one-to-one relationship with a single assembly statement, while other intrinsics can be seen as macroinstructions producing several statements. The use of intrinsics should obsolete in most cases the use of inline assembly code, since they offer more flexibility and a better performance of the optimized code on a variety of different architectures.

Intel’s programming interface manuals define for each intrinsics a mnemonic name, the data structure on which the intrinsics can be applied, the CPU’s that support it, the latency and the throughput [33]. The latency represents the number of
2.3. Software-defined radio

required clock cycles to complete the execution of the corresponding instruction. On the other hand, the throughput represents the number of clock cycles required to wait before the same instruction could be accepted again. Thanks to the pipelined architecture of the SSE unit, in many cases the throughput of an instruction can be significantly less than its latency. Comparing different operations, several of them requires only one clock cycle to be executed. Only a small subset of complex operation need a noticeable number of cycles to complete, such as square root, division, and similar.

2.3.3.1 SIMD example

This section provides two simple examples of usage of intrinsics. Fig. 2.7 represents the operation of an adding between two 128-bits registers (reg_a and reg_b) with 4-way optimization. Each register contains 4 single-precision floating point values, so that, four additions are executed in a single operation.

The Intel’s manual provides for the instruction _mm_add_ps information about latency and throughput [33]. In particular, on newest CPUs this instruction has a

\[
reg_c = \_mm\_add\_ps(reg_a, reg_b)
\]

Figure 2.7: Arithmetic intrinsics: _mm_add_ps.

\[
reg_c = \_mm\_unpacklo\_ps(reg_a, reg_b)
\]

Figure 2.8: Shuffle intrinsics: _mm_unpacklo_ps.
latency of 3 clock cycles and throughput of 1 clock cycle. Hence, this means that the addition takes 3 clock cycles to be completed, but for consequents additions after one clock cycle the data for the next operation can be putted at the input of the processing unit.

Fig. 2.8 represents a shuffle operation in which the lowest two single-precision floating points of registers reg_a and reg_b are moved and interleaved in the register reg_c. Again, the fastest CPUs can execute _mm_unpacklo_ps with latency and throughput of one cycle. Hence, this means that for every clock cycle a complete _mm_unpacklo_ps operation can be performed.

More detailed descriptions on SSE instructions and compiler intrinsics are available on [33].
Part II
Original Results
3

Backoff uniformity and throughput measurements for IEEE 802.11e networks

This chapter describes an experimental setup for measuring the throughput of an 802.11 network implementing QoS mechanisms, addressing some issues concerning the comparison between the experimental and the theoretical results. The setup is an adhoc network in which the nodes adopt the MADWiFi driver. Two are the main contributions of this study. The first one describes the adjustments to the MADWiFi source code that guarantee the control of the QoS parameters in adhoc mode, which, in the current version of MADWiFi, can be set only when the network operates in centralized mode. The second contribution aims to provide a practical method for verifying if the backoff generated by a given wireless card is uniformly distributed, as dictated by the 802.11 standard. This represents a key point when theoretical and experimental results must be compared, since many vendors adopt non-uniform backoff distributions in their commercial cards. The measurements obtained using the deployed setup are compared to the results obtained using a theoretical model that provides the throughput of each AC of each node in saturated and non-saturated traffic conditions. ¹

3.1 Introduction

The IEEE 802.11 standard is revealing as the prevailing technology for providing broadband wireless access at low cost. In order to improve a differentiation of the offered connectivity, the 802.11e MAC layer amendment has been developed [6]. This extension guarantees traffic prioritization according to the four ACs: BK_AC, BE_AC, VI_AC, and VO_AC whose access to the medium is managed through the EDCA. The

¹The content of this chapter is based on F. Babich, M. Comisso, M. D’Orlando, and A. Dorni, "Deployment of a Reliable 802.11e Experimental Setup for Throughput Measurements", Wiley Wireless Communications and Mobile Computing, volume 12, number 10, pages: 910–923, July 2012.
prioritization concept differentiates the ACs using four QoS parameters: transmission opportunity, arbitration inter-frame space number, minimum and maximum contention window. Moving from the seminal paper of Bianchi [37], which provides a Markov chain model for the DCF in saturated traffic conditions, several theoretical studies have been developed to analyze the throughput of 802.11 networks in presence of adaptive antenna arrays [38], channel errors [39], or considering non-saturated traffic conditions [40, 41]. Further studies have extended the Bianchi's model to analyze the EDCA functionalities in saturated [42–44], and non-saturated conditions [45], thus enabling the theoretical evaluation of the throughput of each AC as a function of the QoS parameters.

Unfortunately, in 802.11e networks, experimental investigations are more difficult to perform, given that many commercial cards adopt a restricted and not modifiable firmware that does not export the internal QoS parameters to the driver and, hence, to the system interface. Thus, most of the available 802.11e wireless cards cannot be used for research purposes, which can require a certain flexibility in the choice of the parameter values. As a matter of fact, the PHY/MAC parameters, as for examples QoS parameters, are specified by the IEEE 802.11 standard and are subject to regulatory domains defined by the States. Hence, the wireless chipset producers are constraint to put on the Market user-friendly products with tuning restrictions.

The open source project MADWiFi overcomes the above limitations, providing to the users the possibility to modify the internal MAC parameters for Atheros-based wireless cards [30,46]. MADWiFi has been already employed in the research literature to experimentally study the behavior of an 802.11e network managed by an access point [47]. Even if MADWiFi enables the user to control the QoS parameters when the network is employed for centralized operations, this feature is not currently available in adhoc mode. Therefore, some adjustments to the source code are required.

Further problems, which appear when the theoretical and the experimental throughput of an adhoc network must be compared, are due to the backoff distribution. In fact, many commercial cards are characterized by non-uniform backoff generation [48], while the analytical models developed in the literature for the DCF and the EDCA assume a uniformly distributed backoff [37–45]. Thus, a method able to provide information regarding the statistic of the backoff may be desirable when the purpose is to obtain a meaningful comparison between theory and experiments. Therefore, when the throughput of an 802.11e network operating in adhoc mode must be measured, two significant limitations are currently present: the impossibility to set the QoS parameters and the uncertainty of the uniformity of the backoff statistic.

The following sections presents a contribution for the both the above issues, describing the adjustments to the MADWiFi source code that guarantee the control of the QoS parameters in adhoc mode, and presenting a practical method for verifying
if the backoff generated by a given wireless card is uniformly distributed. The measured throughput obtained with the deployed experimental setup is compared to the theoretical one, obtained adopting a Markov chain model that provides the throughput of each AC of each node in saturated and non-saturated traffic conditions.

### 3.2 Experimental setup

The experimental setup consists of six nodes communicating in adhoc mode and an additional one acting as a wireless sniffer (Fig. 3.1). All the nodes are desktop computers equipped with IEEE 802.11g PHY layer interfaces. Three source nodes contend for channel access, while the other three nodes act as destinations. The three sources are generating a certain number of traffic flows, corresponding to the number of ACs that have packets ready for transmission. The sources are equipped with Peripheral Component Interconnect (PCI) wireless cards, while the destinations use Mini-PCI cards mounted over a Mini-PCI-to-PCI adapter. All wireless cards are equipped with Atheros-based chipsets, which are fully supported by MADWiFi. The PCI card mounted on each source is a Surecom EP-9321-gp with a dipole antenna, while the Mini-PCI card of each destination is a TP Link TL-WN660G with a patch antenna [49]. The
three sources as well as the three destinations are equipped with identical cards in order to simplify the interpretation of the results. In fact, as described in [48], commercial cards produced by different vendors may experience different performance because of non-uniform backoff generation and/or possible time offsets. This may lead to difficulties in understanding if the measured throughput variations are consequence of the parameter settings or are due to different implementations of the 802.11 functionalities.

The network is deployed to guarantee similar carrier sensing opportunities to all nodes. To reduce as much as possible the packet losses due to the propagation environment, the source-destination distance is kept low (approximately 3 meters), and each node is in the line of sight of the other nodes. Besides, the channel frequency is selected to minimize the interference with the other wireless networks located in proximity of the investigated scenario. The entire setup is deployed in a unique room, in order to have a certain shielding from the other 802.11 networks. The operating system of each node is the Linux Ubuntu 8.10 (Intrepid) [50], and the adopted MADWiFi release is the 0.9.3.3 [51]. The experimental throughput values are evaluated using the client-server Iperf application, which sends the generated traffic to the destination using the UDP protocol [31]. UDP has been preferred to TCP, because the former allows one to infer the MAC throughput by simply rescaling the UDP throughput. The relationship between TCP and MAC throughput, instead, is less immediate because of the ACK traffic of TCP. In the investigated scenario each source acts as an Iperf client and establishes a connection with the Iperf server installed at the destination. The clients send the packets to the server and periodically receive average statistical reports concerning the connection status and the throughput.

The wireless sniffer is employed to capture and store the entire traffic, allowing the off-line analysis of the available trace files. The sniffer node is a laptop Intel Centrino 1.7 GHz with 1 GB RAM that uses a Netgear WPN-511T wireless card [52]. This card operates in monitor mode to enable the TCPDump tool to capture all the successfully transmitted packets [53]. In the above described setup the commercially diffused USB wireless adapters are not employed because of their power fluctuation and because they are not supported by MADWiFi.

3.3 MADWiFi driver modification

As discussed in Section 2.2.1, the current version of the open source driver MADWiFi does not allow the setting of the EDCA QoS parameters in adhoc mode. Thus, some modifications must be introduced in the MADWiFi source code to enable the modification of these parameters. These modifications, together with their reasons, are briefly listed in the sequel of this section.

1. To check the chipset QoS capabilities during the device initialization, the part
of code:

```c
if (vap->iv_caps & IEEE80211_C_WME)
vap->iv_flags |= IEEE80211_F_WME;
```

must be enabled in the function `ieee80211_vap_setup` of the file `ieee80211.c`.

2. To enable the forwarding of the packet in the correct AC in adhoc mode using the variable ` skb->priority`, the part of code:

```c
if (vap->iv_opmode == IEEE80211_M_STA)
```

must be replaced by:

```c
if ((vap->iv_opmode == IEEE80211_M_STA)
   || (vap->iv_opmode == IEEE80211_M_IBSS))
```

in the function `ieee80211_classify` of the file `ieee80211_output.c`. In particular, using this modification the packets arriving from the upper layers are inserted in the proper AC by exploiting the information provided by a specific Internet Protocol (IP) header field, called Type of Service. At MAC layer, this field is interpreted, and the packet is delivered to the corresponding AC according to a fixed hexadecimal value \[ \text{hex}_v \]. More precisely, a packet with a ToS equal to 0x20 is sent to the BK_AC queue, while the ToS values 0xa0, 0x30 and 0x00 are related to the VI_AC, VO_AC and BE_AC queues, respectively.

3. To ensure that all packets are properly classified, the line:

```c
ni->ni_flags |= IEEE80211_NODE_QOS;
```

must be inserted in the function `ieee80211_hardstart` of the file `ieee80211_output.c` before the calculation of the priority.

The above adjustments allow one to completely modify the QoS parameters of three ACs: BK_AC, VI_AC, and VO_AC. The parameters of the BE_AC queue can be modified too, but, in some cases, they are ignored by the HAL. Hence, the actual values of the BE_AC queue must be verified before starting the measurements. Once the MADWiFi source code is modified, the QoS parameters can be set from the Linux shell using the command `iwpriv`.

65
Chapter 3. Backoff uniformity and throughput measurements for IEEE 802.11e networks

3.4 Theoretical analysis for the backoff uniformity

The MADWiFi modifications presented in the previous section improve the flexibility of the experimental setup, enabling the possibility to perform a wide set of measurements when the network operates in adhoc mode. However, since all developed analytical models assume a uniform backoff distribution, a meaningful comparison between the theoretical and the experimental throughput of any 802.11 distributed network requires the availability of wireless cards implementing a uniform backoff. This represents a key point because in several cases the vendors introduce time offsets and adopt non-uniform backoff statistics, leading to backoff distributions that are not adherent to the 802.11 standard [48]. Besides, in [48], the authors show that some wireless cards adopt a certain backoff statistic in absence of contending nodes and another one when other contending nodes are present. Hence, measurements performed when a single source is present and revealing a uniform backoff distribution do not guarantee that the same source will behave similarly in presence of other nodes. Thus, considering that the backoff distribution for the single node case can be easily inferred from the trace files provided by the wireless sniffer, the main problem is to verify if the backoff remains uniform when \( N_s > 1 \) sources are present. This section proposes a method to verify if the backoff is generated according to a uniform distribution in this second, more significant, case. This method does not require intrusive measurements inside the chipset, since it is based on the comparison between the theoretical and the experimental statistics of the time between two successfully transmitted packets.

Consider a scenario where \( N_s \) contending nodes transmit from a unique AC with identical QoS parameters in saturated traffic conditions. Besides, assume that the maximum and the minimum contention windows have an identical \( W^{(0)} \) value and that the transmitted packets have the same length. Define as \( T_{tx} \) the packet transmission time normalized to the slot time and rounded to the nearest integer. More precisely, \( T_{tx} \) can be obtained as:

\[
T_{tx} = \left\lfloor \frac{\text{DIFS} + T_{DATA} + \text{SIFS} + T_{ACK}}{\sigma_{slot}} \right\rfloor, \tag{3.4.1}
\]

where \( T_{DATA} \) is the time required to transmit the DATA packet and \( T_{ACK} \) is the time...
Theoretical analysis for the backoff uniformity

required to transmit the ACK packet. These settings simplify the elaboration of the experimental data and enable the theoretical analysis of the statistics of the time between two successfully transmitted packets. Experimentally, the distribution of the time between the end of a successful transmission and the beginning of another successful transmission can be obtained from the trace files generated by TCP-Dump [53]. The available traces can be parsed by a simple bash script that provides the reception times, the packet type, and the source node. These quantities can be subsequently processed by a Matlab script that evaluates the transmission time instants. Successive elaborations allow the evaluation of the backoff jitter from the uniform distribution and the Probability Mass Function (PMF) of two successful transmissions (not necessarily consecutive). These experimental statistics can then be compared to the theoretical ones, which are derived in the following of this section under the hypothesis of uniform backoff distribution.

According to the assumption of identical maximum and minimum contention window, the single source behavior can be described by the Markov chain in Fig. 3.2, where \( k \) represents the value of the backoff counter calculated as a multiple of the slot time. This chain can be easily analyzed in order to find the probability of being in a state \( k' \) larger or equal to a given \( k \):

\[
\Pr\{k' \geq k\} = 1 - \sum_{k'=0}^{k-1} \psi_{k'} = \frac{[W(0) - k + 1][W(0) - k]}{W(0)[W(0) + 1]}, \tag{3.4.2}
\]

where \( \psi_{k'} = 2[W(0) - k']/[W(0)[W(0) + 1]] \) denotes the probability of being in the \( k' \)-th state. Equation (3.4.2) represents the probability that a node has a residual backoff (measured as a multiple integer of the slot time) larger or equal to \( k \). Hence, the probability that the minimum residual backoff \( R_{bo} \) between all of the \( N_s \) contending sources is equal to \( k \) is:

\[
\Pr\{R_{bo} = k\} = \Pr\{R_{bo} \geq k\} - \Pr\{R_{bo} \geq k + 1\}, \tag{3.4.3}
\]

where:

\[
\Pr\{R_{bo} \geq k\} = \left\{ \frac{[W(0) - k + 1][W(0) - k]}{W(0)[W(0) + 1]} \right\}^{N_s}. \tag{3.4.4}
\]

Therefore, the probability that the number of slots \( \Delta_s \) between two consecutive successful transmissions (i.e. between the end of a successful transmission and the beginning of a subsequent successful transmission) be equal to \( k \) can be calculated as:

\[
\Pr\{\Delta_s = k\} = \frac{\Pr\{R_{bo} = k\} - \Pr\{R_{bo} = k + 1\}}{1 - \Pr\{R_{bo} \geq 1\}}. \tag{3.4.5}
\]

The case \( N_s = 2 \) is sufficient to analyze the card behavior in terms of backoff in presence of other contending nodes. For \( N_s = 2 \), the collision probability becomes equal
to $1/W^{(0)}$ and (3.4.5) can be rewritten as:

$$\Pr \{ \Delta_s = k \} = \frac{1 + 3[W^{(0)} - k - 1][3W^{(0)} - 3k - 2]}{[W^{(0)}]^3}, \quad (3.4.6)$$

which is obtained using (3.4.3) and (3.4.4). It is worth noticing that the considered interval between two successful transmissions may contain collisions. Hence, this interval consists of a sequence of $H \geq 1$ backoff interval times and $H - 1$ collisions. Remembering that the PMF of the sum of two random variables is given by the convolution of the PMF of the two variables, the PMF corresponding to an interval containing $H - 1$ collisions can be obtained by convolving (3.4.5) $H - 1$ times. More precisely, the PMF of an interval not containing collisions is:

$$\beta_0(k) = P^s \cdot \varrho_0(k), \quad (3.4.7)$$

where $P^s$ is the success probability (equal for all sources in this case), $\varrho_0(k) = \Pr \{ \Delta_s = k \}$, and $0 \leq k \leq W^{(0)} - 1$. The PMF of an interval containing one collision is given by:

$$\beta_1(k) = P^s(1 - P^s) \varrho_1(k - T_{tx}), \quad (3.4.8)$$

where $\varrho_1(k) = \varrho_0(k) * \varrho_0(k)$, the symbol $*$ denotes the convolution operation, and $T_{tx} \leq k \leq 2W^{(0)} - 1$. In general, the PMF of an interval containing $H - 1$ collisions can be recursively obtained as:

$$\beta_{H-1}(k) = P^s(1 - P^s)^{H-1} \varrho_{H-1}[k - (H - 1)T_{tx}], \quad (3.4.9)$$

where $(H - 1)T_{tx} \leq k \leq (H + 1)W^{(0)} - H$, and $\varrho_{H-1}(k) = \varrho_{H-2}(k) * \varrho_0(k)$.

The proposed method can be summarized as follows. The elaboration of the available trace files does not allow the direct reconstruction of the backoff statistics (except for the particular case $N_s = 1$), but allows the derivation of the PMFs between two successful transmissions. These experimental PMFs can be compared to the theoretical ones $\beta_0(k),...,\beta_{H-1}(k)$, which are derived assuming a uniform backoff. Therefore, the confirmation of a uniformly generated backoff results from a good matching between the experimental and the analytical PMFs. It is worth noticing that the above method may be adopted also to verify the adoption of other backoff statistics. In fact, in the case that a non-uniform backoff statistic is known, the PMFs can still be derived using the proposed approach, but considering different transition probabilities for the Markov chain in Fig. 3.2.

### 3.5 Throughput analysis for IEEE 802.11e extension in adhoc networks

This section describes the Markov chain model adopted for analyzing the deployed network in order to compare the experimental throughput to the theoretical
The objective is not to provide a complete model for the 802.11e extension, which has been already presented in other studies [42–45], but simply to analyze from a theoretical point of view the considered experimental setup. The presented analysis is based on the approach presented in [38], which is extended to include the ACs and the QoS parameters in the single omnidirectional antenna case.

The model is developed for a general scenario where \( N_s \) contending sources can operate both in saturated and non-saturated traffic conditions. Each source supports the \( Q = 4 \) ACs of the 802.11e, which are numbered according to \( a = 1 \) (VO_AC), \( a = 2 \) (VI_AC), \( a = 3 \) (BE_AC), and \( a = 4 \) (BK_AC). Hence, a lower \( a \) value denotes a higher priority. The generic \( a \)-th AC of the \( i \)-th node is characterized by the arbitration inter-frame space AIFS\(_{a,i}\), the minimum contention window \( W_{a,i}^{(0)} \), the maximum number
of retransmissions $m_{a,i}$, and the maximum backoff stage $m'_{a,i}$. The transmission opportunity is considered equal for all sources. The backoff mechanism is modeled by the bi-dimensional stochastic process \{$s_{a,i}(t), b_{a,i}(t)$\}, where $s_{a,i}(t)$ and $b_{a,i}(t)$ represent, respectively, the backoff stage and the backoff timer at time $t$ [37]. The Markov chain adopted to model the $a$-th AC of the $i$-th node is depicted in Fig. 3.3, where $p_{a,i}$ is the conditional collision probability, $W_{a,i}^{(h)}$ is the contention window size at the $h$-th retransmission attempt:

$$W_{a,i}^{(h)} = \begin{cases} 1 & h = -1 \\ 2^h W_{a,i}^{(0)} & 0 \leq h \leq m'_{a,i} \\ 2^{m'_{a,i}} W_{a,i}^{(0)} & m'_{a,i} < h \leq m_{a,i} \end{cases}$$

(3.5.1)

being $W_{a,i}^{(-1)} = 1$ defined to account for the idle state ($-1, 0$), corresponding to the case of empty transmission buffer. Once the generic source has successfully transmitted a packet or has discarded it after $m_{a,i} + 1$ transmission attempts, a new packet is awaiting transmission in a single place buffer with probability $\Lambda_{a,i}$, which accounts for the overall time spent by the source for transmitting the previous packet. The source exits from the idle state with probability $\lambda$, which is a function of the average slot time and hence is not dependent on the source and on the particular AC. The transition probabilities of the chain for the $a$-th AC of the $i$-th node are:

$$\begin{align*}
P_{a,i}((h, k)|(h, k + 1)) &= 1 & k \in [0, W_{a,i}^{(h)} - 2] & h \in [0, m_{a,i}]
\quad & & & \quad \\
P_{a,i}((h, k)|(-1, 0)) &= p_{a,i}/W_{a,i}^{(h)} & k \in [0, W_{a,i}^{(h)} - 1] & h \in [1, m_{a,i}]
\quad & & & \quad \\
P_{a,i}((-1, 0)|(h, 0)) &= 1 - \delta_{h,m_{a,i}} & h \in [0, m_{a,i}]
\quad & & & \quad \\
P_{a,i}((0, k)|(h, 0)) &= (1 - p_{a,i}(1 - \delta_{h,m_{a,i}})) W_{a,i}^{(0)} & k \in [0, W_{a,i}^{(0)} - 1] & h \in [0, m_{a,i}]
\quad & & & \quad \\
P_{a,i}((0, k)|(-1, 0)) &= \lambda/W_{a,i}^{(0)} & k \in [0, W_{a,i}^{(0)} - 1] & h \in [0, m_{a,i}]
\quad & & & \quad \\
\end{align*}$$

(3.5.2)

where:

$$\delta_{h,m_{a,i}} = \begin{cases} 1 & h = m_{a,i} \\ 0 & \text{elsewhere} \end{cases}$$

(3.5.3)

is the Kronecker delta. The first equation in (3.5.2) accounts for the decrement of the backoff timer and the second one models the access procedure at the $h$-th retransmission attempt. The third equation models the persistence in the idle state, while the fourth one describes the source behavior when the transmission buffer is empty. The fifth equation models the case of nonempty transmission buffer and, finally, the sixth equation models the packet arrival when the source is in the idle state. The stationary distribution of the chain $b_{a,i}^{(h,k)} = \lim_{t \to \infty} P_{a,i}\{s_{a,i}(t) = h, b_{a,i}(t) = k\}$ can be
evaluated using the identities:

\[ b^{(h,k)}_{a,i} = \left(1 - \frac{k}{W^{(h)}_{a,i}}\right)p^h_{a,i} b^{(0,0)}_{a,i}, \]  

(3.5.4)

for \( k \in [1, W^{(h)}_{a,i} - 1], h \in [0, m_{a,i}], \) and:

\[ b^{(h,0)}_{a,i} = \begin{cases} (1 - \Lambda_{a,i}) b^{(0,0)}_{a,i} / \lambda, & h = -1 \\ p^h_{a,i} b^{(0,0)}_{a,i} & h \in [0, m_{a,i}] \end{cases} \]

(3.5.5)

for \( k = 0. \) Using (3.5.1) - (3.5.5) and imposing the normalization condition:

\[ \sum_{h=-1}^{m_{a,i}} \sum_{k=0}^{W^{(h)}_{a,i} - 1} b^{(h,k)}_{a,i} = 1, \]  

(3.5.6)

the probability \( b^{(0,0)}_{a,i} \) can be evaluated as:

\[ b^{(0,0)}_{a,i} = \left\{ \frac{1 - \Lambda_{a,i}}{\lambda} + \frac{W^{(0)}_{a,i} 2^{m_{a,i}' + 1} m_{a,i}' + 1}{2(1 - p_{a,i})} + \frac{W^{(0)}_{a,i} [1 - (2p_{a,i})^{m_{a,i} + 1}]}{2(1 - 2p_{a,i})} + \right. \]

\[ - \left. \frac{[W^{(0)}_{a,i} 2^{m_{a,i}' + 1} m_{a,i}' + 1]}{2(1 - p_{a,i})} \right\}^{-1}. \]

(3.5.7)

Once the stationary distribution of the chain relative to the \( a \)-th AC of the \( i \)-th source is calculated, the corresponding throughput can be obtained by analyzing the events that can occur during a slot time. The probability that at least one transmission occurs in a randomly chosen slot time is given by:

\[ P_t = 1 - \prod_{i=1}^{N_i} \prod_{a=1}^{Q} (1 - \tau_{a,i}), \]  

(3.5.8)

where \( \tau_{a,i} \) is the transmission probability of the \( a \)-th AC relative to the \( i \)-th source. In absence of channel errors, a successful transmission for the \( a \)-th AC of the \( i \)-th source requires that none of the other sources transmits (from any AC) in the same slot time, and that no packets of an AC with a higher priority have to be transmitted by the \( i \)-th source itself in the same slot time. Therefore, the success probability can be evaluated as:

\[ P_{s}^{a,i} = \frac{\tau_{a,i}}{P_t} \prod_{i'=1}^{N_i} \prod_{a' \neq i}^{Q} (1 - \tau_{a',i'}) \prod_{a'<a}^{Q} (1 - \tau_{a',i}). \]  

(3.5.9)
This equation models the real collisions, due to simultaneous channel access by different nodes, and the internal collisions, which are managed by the source itself by transmitting the packet with the higher priority and generating a further backoff for the packet with the lower priority. The probability that at least two sources collide is given by the probability of no success:

\[ P^c = 1 - \sum_{i=1}^{N_s} \sum_{a=1}^{Q} P_{s,a,i}^s = 1 - \sum_{i=1}^{N_s} \sum_{a=1}^{Q} \frac{T_{a,i}}{p_t} \prod_{i'=1}^{N_s} \prod_{a'=1}^{Q} (1 - \tau_{a',i'}) \prod_{a''=1}^{Q} (1 - \tau_{a'',i}). \quad (3.5.10) \]

From (3.5.8)-(3.5.10), the average slot duration can be evaluated as:

\[ E_s = (1 - p)^{\sigma_{\text{slot}}} + p^i \sum_{i=1}^{N_s} P_{s,a,i}^s T_{s,a,i}^s + p^1 p^c T^c, \quad (3.5.11) \]

where \( T_{s,a,i}^s \) is the time required by the successful DATA/ACK handshake corresponding to the \( a \)-th AC of the \( i \)-th node, and \( T^c = \max_{1 \leq a \leq Q, 1 \leq i \leq N_s} T_{s,a,i}^s \) is the time wasted because of collisions. This value is assumed equal to the largest time required by a couple of nodes to complete their handshake. Observe that, as in [37], the term slot time refers to the constant value \( \sigma_{\text{slot}} \) specified in the 802.11 protocol, while the term average slot duration \( E_s \) in (3.5.11) refers to the variable time interval between two consecutive backoff time counter decrements. Considering the basic access mechanism, the time required by a successful handshake \( T_{s,a,i}^s \) is calculated as:

\[ T_{s,a,i}^s = \text{AIFS}_{a,i} + T_{\text{DATA}_{a,i}} + T_{\text{ACK}} + \text{SIFS}, \quad (3.5.12) \]

where \( \text{AIFS}_{a,i} = \text{SIFS} + \text{AIFS}_{N_s,a,i} \cdot \sigma_{\text{slot}}, \ T_{\text{DATA}_{a,i}} \) and \( T_{\text{ACK}} \), are, respectively, the time required to send the DATA packet relative to the \( a \)-th AC of the \( i \)-th node and the time required to send the ACK packet. To evaluate the network behavior in non-saturated traffic conditions, the adopted model requires the calculation of the average number of slots during which a DATA packet remains in the chain:

\[ E_{\text{NS}_{a,i}} = \sum_{h=0}^{m_{a,i}} \frac{W_{a,i}^{(h)} - 1}{2} p_{a,i}^h. \quad (3.5.13) \]

The non-linear system:

\[
\begin{align*}
\tau_{a,i} &= \sum_{h=0}^{m_{a,i}} b_{a,i}^{(h,0)} = \frac{1 - p_{a,i}}{1 - p_{a,i}} b_{a,i}^{(0,0)}, \quad a = 1, \ldots, Q, \quad i = 1, \ldots, N_s \\
p_{a,i} &= 1 - \prod_{i'=1}^{N_s} \prod_{a'=1}^{Q} (1 - \tau_{a',i'}) \prod_{a''=1}^{Q} (1 - \tau_{a'',i}), \quad a = 1, \ldots, Q, \quad i = 1, \ldots, N_s \\
\Lambda_{a,i} &= 1 - e^{-\mu E_{\text{NS}_{a,i}}} E_s, \quad a = 1, \ldots, Q, \quad i = 1, \ldots, N_s \\
\lambda &= 1 - e^{-\mu E_s}
\end{align*}
\]

72
3.6 Numerical results

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{\text{slot}}$</td>
<td>20 $\mu$s</td>
</tr>
<tr>
<td>SIFS</td>
<td>10 $\mu$s</td>
</tr>
<tr>
<td>$T_{\text{op}}$</td>
<td>224 $\mu$s</td>
</tr>
<tr>
<td>$R$</td>
<td>54 Mbits/s</td>
</tr>
<tr>
<td>$R_c$</td>
<td>24 Mbits/s</td>
</tr>
<tr>
<td>$\text{Head}_{\text{IP-UDP}}$</td>
<td>(20 + 8)$\cdot$8 bits</td>
</tr>
<tr>
<td>$AIFS_{a,i}$</td>
<td>2, for $a = 1, ..., Q$ and $i = 1, ..., N_s$</td>
</tr>
<tr>
<td>$m_{a,i}$</td>
<td>1, for $a = 1, ..., Q$ and $i = 1, ..., N_s$</td>
</tr>
<tr>
<td>$m'_{a,i}$</td>
<td>$\begin{cases} 0 &amp; \text{for backoff analysis} \ 1 &amp; \text{for throughput analysis} \end{cases}$, for $a = 1, ..., Q$ and $i = 1, ..., N_s$</td>
</tr>
<tr>
<td>$\text{len}_{a,i}$</td>
<td>1498$\cdot$8 bits, for $a = 1, ..., Q$ and $i = 1, ..., N_s$</td>
</tr>
<tr>
<td>$T_{\text{DATA}_{a,i}}$</td>
<td>96$\mu$s + $\text{len}_{a,i}$/$R$</td>
</tr>
<tr>
<td>$T_{\text{ACK}}$</td>
<td>14$\cdot$8/$R_c$</td>
</tr>
</tbody>
</table>

Table 3.1: System parameters.

can be used to obtain the transmission probabilities, the conditional collision probabilities, and the probabilities of nonempty transmission buffer assuming the packet arrival as Poisson distributed with mean $\mu$ (equal for all sources). This system is composed by $3N_sQ + 1$ equations in order to derive $\lambda$ and $\tau_{a,i}$, $p_{a,i}$, $\Lambda_{a,i}$ for the $Q$ ACs of the $N_s$ nodes. The first $N_sQ$ equations are obtained considering that the transmission begins when the backoff timer becomes equal to zero. The second $N_sQ$ equations are derived observing that the packet of the $a$-th AC of the $i$-th node collides if at least one of the remaining nodes transmits or if a packet with higher priority has to be transmitted in the same slot by the $i$-th node itself. The third set of equations models the arrival process during the transmission of the previous packet, and the last equation models the packet arrival in the idle state.

Once the $3N_sQ + 1$ unknowns are calculated, the throughput at the transport layer corresponding to the $a$-th AC of the $i$-th node can be evaluated as:

$$S_{a,i} = \frac{p_t p_s}{E_s} \left( \text{len}_{a,i} - \text{Head}_{\text{IP-UDP}} \right),$$

(3.5.15)

where $\text{len}_{a,i}$ is the payload length at MAC layer and $\text{Head}_{\text{IP-UDP}}$ denotes the IP-UDP header length.

3.6 Numerical results

The adopted system parameters are shown in Table 3.1, where $R$ denotes the data rate and $R_c$ represents the control rate. Note that the verification of the uniformity of
Chapter 3. Backoff uniformity and throughput measurements for IEEE 802.11e networks

Figure 3.4: (a) Backoff distribution as a function of the backoff counter for \( N_s = 1 \) and \( W^{(0)} = 8 \);
(b) Backoff jitter as a function of the backoff counter for \( N_s = 1 \) and \( W^{(0)} = 8 \).

the backoff distribution is performed using a maximum backoff stage equal to zero, according to the settings used for the method presented in Section 3.4.

Figs. 3.4a-3.5b report the results concerning the backoff analysis of the adopted wireless cards. The measurements shown in Fig. 3.4a reveal that, when a unique source is active in the network, the backoff behavior closely parallels the uniform distribution. A confirmation of the accuracy of the presented experimental results is given by Fig. 3.4b, also derived for \( N_s = 1 \), which shows the measured jitter. This verification, performed in presence of a unique source, can be carried out by a simple parsing of the trace files provided by the wireless sniffer. However, since a given card may adopt different backoff distributions in presence of other contending nodes, the results in Fig. 3.4a do not guarantee that the card operates using a uniform backoff when other cards are sensed. Thus, a further investigation is performed employing the proposed verification method.

Figs. 3.5a and 3.5b are both obtained for \( N_s = 2 \) nodes, but considering two different \( W^{(0)} \) values. These figures show that the measured PMFs of the time interval between two successful transmissions are very close to the theoretical ones. It is worth to notice that, even if the retry limit is equal to one and only one retransmission can be performed, the PMF \( \beta_2(k) \) (case \( H = 3 \) in (3.4.9)) can still be evaluated because the proposed method does not require that successive collisions involve always the same packet. Once a packet is discarded after the last possible retransmission attempt, the subsequent one may collide too. Thus, a considered time interval may contain more collisions involving different packets. Summarizing, the results in Figs. 3.4a-3.5b confirm that the backoff is uniformly generated when the wire-
Figure 3.5: (a) PMF of the time interval between two successful transmissions as a function of the backoff counter for \( N_s = 2 \) and \( W^{(0)} = 8 \); (b) PMF of the time interval between two successful transmissions as a function of the backoff counter for \( N_s = 2 \) and \( W^{(0)} = 16 \).
Fig. 3.6: Throughput as a function of the mean arrival rate for $N_s = 3$ sources using the first AC with $W_{1,1}^{(0)} = 8$, $W_{1,2}^{(0)} = 16$, $W_{1,3}^{(0)} = 32$.

Fig. 3.6. The curves are obtained considering a scenario in which each of the three nodes uses a unique AC. In particular, all the three nodes are using the first AC, which corresponds to the AC for the VOice, for 8, 16, and 32 values of $W_{1,i}^{(0)}$, respectively. Fig. 3.7 represents the single AC throughput when a unique node is transmitting. In fact, there is a prioritization law between the different ACs, which can be observed in this figure. The good agreement between experimental and theoretical results is confirmed also in a more sophisticated scenario in Fig. 3.8, which represents the throughput of each AC of each node when three sources are transmitting from two ACs. In this case, two contention mechanisms are take into account. The first one, is active between the first and the second ACs of the single node, while the second one is active between the node contending the channel.

Fig. 3.9 reports the saturation throughput as a function of the minimum contention window of the first source $W_{1,1}^{(0)}$ when all nodes transmit from a unique AC. These latter curves show the expected increase of the throughput of the other two sources due to the increase of $W_{1,1}^{(0)}$. The satisfactory matching between theory and measurements, which holds also for the transition zones of the throughput curves, confirms that the adopted non-saturated analysis properly models both the external and the
3.6. Numerical results

Figure 3.7: Throughput as a function of the mean arrival rate in presence of a unique source using four ACs with $W_{a,1}^{(0)} = 8$ ($a = 1, 2, 3, 4$).

--- $S_{1,1}$ (theory) ♦ $S_{1,1}$ (measure)
--- $S_{2,1}$ (theory) ○ $S_{2,1}$ (measure)
--- $S_{3,1}$ (theory) □ $S_{3,1}$ (measure)
--- $S_{4,1}$ (theory) * $S_{4,1}$ (measure)

Internal collisions when one or more ACs are active. The above throughput results shows that a really good matching between experimental and analytical curves can be obtained by using wireless cards enabling the control of the QoS parameters and adopting a uniform backoff statistic. The fulfillment of this second requirement can be achieved using the method presented in Section 3.4, which is able to investigate the backoff distribution implemented in the wireless cards in order to ensure that measurements and theory move from the same backoff assumptions. It is also worth to notice that a reliable experimental setup requires also an accurate deployment of the network in order to minimize channel errors, asymmetries of the physical carrier sensing mechanism, and interference with other networks. In particular, this latter exigency may represent a problem difficult to overcome, considering that nowadays the WiFi networks are widely diffused in workplaces as well as in private and public sites. Thus, the availability of a “clean” electromagnetic environment for measurement purposes in the 2.4 GHz band requires a careful selection of the less interfered frequency channel and, in some cases, the adoption of shielded sites for the deployment of the experimental setup.
Chapter 3. Backoff uniformity and throughput measurements for IEEE 802.11e networks

3.7 Conclusions

A discussion concerning the deployment of a reliable 802.11e experimental setup for throughput measurements has been presented. Two considerable aspects have been addressed: the adjustments required by the MADWiFi source code for enabling the setting of the QoS parameters in adhoc mode, and a practical method for verifying the uniformity of the backoff distribution implemented in the adopted wireless cards. This method does not require invasive investigations inside the chipset and can be used to infer the backoff behavior of the wireless cards from a careful processing of the trace files provided by the wireless sniffer. The measurements obtained using the deployed setup, which has been described in detail to enable the reproduction of the presented results, have been compared to the results obtained using a theoretical model that provides the throughput of each access category of
3.7. Conclusions

Figure 3.9: Saturation throughput as a function of the minimum contention window of the first source $W_{1,1}^{(0)}$ for $N_s = 3$ sources using the first AC with $W_{1,2}^{(0)} = 16$, $W_{1,3}^{(0)} = 32$.

---

Each node in saturated and non-saturated traffic conditions. The results have shown that a careful deployment of the network, a complete control of the parameters, and a verification of the fundamental theoretical hypotheses can guarantee a really good matching between theory and measurements.
Discrete-time simulation of smart antenna systems in Network Simulator-2 using MATLAB and Octave

This chapter presents a MATLAB/Octave extension for the simulation of adaptive antennas in ns2 that accounts in detail for the physical antenna system and the beam-forming algorithm. The ns2 channel model, which currently accounts only for path-loss attenuation and shadowing, is improved by including the possibility to select the power azimuth spectrum and the angular spread of the channel. This study describes the integration techniques adopted for the developed MATLAB/Octave-ns2 platform and compares the MATLAB and Octave performance in terms of simulation time and flexibility by considering a network scenario involving the backbone nodes of an 802.11 DWN using adaptive antenna arrays. In such a scenario the developed simulator is used to evaluate the throughput in presence of multipath and fading accounting for the effect of channel coding.\footnote{The content of this chapter is based on F. Babich, M. Comisso, A. Dorni, F. Barisi, and A. Manià, "Discrete-Time Simulation of Smart Antenna Systems in Network Simulator-2 Using MATLAB and Octave", Simulation: Transactions of the Society for Modeling and Simulation International, volume 87, number 11, pages: 932–946, Nov. 2011 [54].}

4.1 Introduction

Network simulators represent powerful tools for testing the capabilities of devices and protocols that can reduce, or avoid, the cost required by experimental measurement campaigns to evaluate the performance of new technologies. Within the set of the possible numerical approaches, discrete-event simulators present advantages in terms of scalability, and hence of simulation time, because the state of the system changes only when a new event occurs [55]. One of the most diffused open source discrete-event tools is ns2, a network simulator developed in the C++ pro-
gramming language at Berkeley National Laboratory and improved by various research groups [56, 57]. A considerable limit of ns2 lies in the models adopted for the physical layer and the propagation channel, which appear oversimplified when a certain adherence to a real wireless environment is required [58, 59]. These limitations become particularly significant when advanced antenna techniques, such as smart antenna systems, must be included in the simulations. A smart antenna system is constituted by an antenna array, whose radiation pattern can be dynamically controlled to perform the electrical beam steering to a desired direction, and null steering to reject interfering signals [60]. In a scenario where the network nodes are able to adapt their radiation patterns, the actual antenna gain becomes a crucial quantity for evaluating the performance figures of the network, such as throughput, drop probability, and packet delay.

Another aspect to consider is that a discrete-event tool does not represent the unique approach for the simulation of a telecommunication network. In fact, the physical layer characteristics, such as modulation, antenna system, and coding scheme, as well as the behavior of the wireless channel can be implemented in detail using a discrete-time approach. The main drawback of a discrete-time tool is due to the very long simulation time usually required to obtain the results. Hence, an hybrid discrete-event discrete-time approach may represent an effective tradeoff between model accuracy and simulation duration. A detailed PHY layer model can be properly implemented using the proprietary software MATLAB and the open source one Octave, since the two tools adopt substantially the same high level language and hence are able to operate on the same scripts. With respect to the C++ language, MATLAB and Octave represent two immediate development environments for implementing procedures and functions that simulate the PHY layer behavior.

Recently, some MATLAB extensions have been proposed to include smart antenna systems in OPNET and OMNeT++ discrete-event simulators [61, 62]. Even if modeling procedures for directional MAC protocols and resource allocation schemes have been already implemented in ns2 and other discrete-event platforms [9, 17, 63–67], and some antenna simulators have been developed adopting the MATLAB language [20, 68, 69], the approach presented in this thesis considering, together, adaptive antenna arrays and multipath are not currently available for ns2.

4.2 Modeling methodology

This section describes the smart antenna system model and the modeling of the multipath effect, which can be present on the wireless channel. In addition, a PHY/MAC description of the nodes adopted in the simulation scenario is provided. In particular, the description focuses on nodes equipped with smart antenna system at the PHY layer and on the access and reception criteria implemented at the MAC
4.2. Modeling methodology

4.2.1 Smart antenna system

The smart antenna system model is shown in Fig. 4.1. As outlined in Section 1.3.2, this model accounts for the physical antenna system, constituted by an array of $N$ elements, and for the signal processing unit, which calculates the complex array excitations $w_1, ..., w_N$ that are applied to the array elements by the beamforming unit in order to generate the power gain pattern:

$$G(\phi) = \left| \sum_{n=1}^{N} w_n g_n(\phi) e^{j \frac{2\pi}{\lambda} r_n \cdot \hat{r}} \right|^2.$$

(4.2.1)

The physical antenna system, which determines the quantities $g_n(\phi)$ and $r_n$, depends on the array geometry, the number of elements, the inter-element spacing, and the single (isolated) element pattern. All these components can be modified in the presented extension, which currently implements the four geometries described in Section 1.3.2.1: ULAs, UCAs, USAs, and CRAs, whose parameters can be directly set from the ns2 Tcl script. In particular, the user can select all the parameters that characterize each array geometry, such as the inter-element distance of a USA, the radius of a UCA, or the radius and the number of elements of each ring of a CRA.
Chapter 4. Discrete-time simulation of smart antenna systems in Network Simulator-2 using MATLAB and Octave

The other basic factor influencing the power gain pattern \( G(\phi) \) is the signal processing algorithm, which determines the array excitations \( w_1, ..., w_N \). As discussed in Section 1.3.2.2, typically, in antenna array processing, the beamforming techniques are classified into two main families: spatial reference techniques and temporal reference techniques. The description of the MUSIC plus cLMS algorithm, which belongs to the first family, and the description of the uLMS and RLS algorithms, which conversely belongs the second family, can be found in Section 1.3.2.2.

In general, the vector of the signals received by the antenna array and incoming from \( M \) sources can be expressed as [60]:

\[
x = Hs + n,
\]

where \( H \) is the \( N \times M \) channel matrix, \( s = [s_1(t), ..., s_M(t)]^T \) is the \( M \times 1 \) vector of the transmitted signals, and \( n = [n_1(t), ..., n_N(t)]^T \) is the \( N \times 1 \) noise vector generated assuming additive white Gaussian noise. Each signal \( s_i(t) \), produced by the generic source \( i \), is replaced by a sequence of \( M_s \) samples \( s_i(1), ..., s_i(M_s) \). Thus, the vector of continuous functions \( x \) is replaced by an \( N \times M_s \) matrix \( \hat{x} \), from which the \( N \times N \) array correlation matrix can be estimated as \( R_{xx} = E{\hat{x}\hat{x}^H} \). The matrix \( R_{xx} \) is then used by the selected algorithm (uLMS, RLS, or MUSIC and cLMS together) to evaluate the array excitations \( w_1, ..., w_N \) in (4.2.1). In particular, the sequence of \( m_{ts} \) bits is used as the reference signal when temporal reference algorithms are used (uLMS, RLS), or as a sequence of training when spatial reference techniques are adopted (MUSIC, cLMS). This approach enables to account in detail for the possible temporal correlation between the generated signals, for the channel variations, and for the influence of the noise, since the adaptation process is performed considering sequences of bits. Using this discrete-time extension, one obtains a realistic physical layer implementation, which can be really difficult to obtain by maintaining the discrete-event approach of the unmodified ns2 tool for the entire network simulation.

The possibility to account for the single element pattern \( g_n(\varphi) \) in (4.2.1) allows one not only to consider non-omnidirectional radiators, but also to account in detail for mutual coupling between the array elements, which may lead to considerable distortions of the final pattern \( G(\varphi) \). These distortions do not influence the behavior of temporal reference techniques, but can largely degrade the performance of spatial reference algorithms. In the presented extension the coupled single element patterns are evaluated before the network simulation using an electromagnetic software. Hence, if required, the MATLAB/Octave functions implemented in the developed tool can acquire from a file the coupled patterns in order to synthesize the array pattern accounting for the real behavior of each radiator in the presence of the other ones.
4.2. Modeling methodology

4.2.2 Multipath

Multipath effects in the angular domain are modeled by using the concept of equivalent radiation pattern [38]:

\[ F(\varphi) = \int_{0}^{2\pi} G(\varphi') p(\varphi' - \varphi) d\varphi', \] (4.2.3)

where \( p(\varphi) \) is the PAS normalized to the transmission power, which represents the probability density function (pdf) of the DoAs corresponding to an active node. \( F(\varphi) \) takes into account that a signal replica incoming from an undesired LoS direction \( \varphi \) is received with the gain corresponding to the null, but all other replicas of the same signal, incoming from \( \varphi' \neq \varphi \) are received with higher gains, leading to an increase of the received interference. Similarly, the signal incoming from the desired source in the LoS direction is received with the maximum gain, but the replicas are received with lower gains, leading to a decrease of the desired signal power. The pdfs currently implemented in the proposed MATLAB/Octave extension are the truncated Laplacian [28], the truncated Gaussian [27], and the pdf corresponding to a ring of scatterers surrounding the transmitting node [29]. An example of the pdfs regarding the considered scatterers distributions can be found in Section 1.3.1.3. The scenario corresponding to absence of multipath can be obtained by choosing \( p(\varphi) \) as the Dirac function.

The main advantage of this approach for modeling multipath is the reduction of the simulation time. In fact, when a complete topology of scatterers must be included in the simulation, all signal replicas corresponding to each active source must be taken into account, thus leading to a considerable increase of the dimensions of the signal matrices used in MATLAB/Octave. Besides, in some cases, the distribution of the scatterers producing a given pdf can be difficult to derive analytically. Instead, using (4.2.3), the performance degradation due to the azimuth spread can be taken into account once the synthesis of the smart antenna pattern \( G(\varphi) \) is completed. Furthermore, some of the adopted statistics, such as the truncated Laplacian and the truncated Gaussian, have been validated by extensive measurement campaigns, and hence represent realistic distributions that have been already adopted in several standardized channel models [70, 71].

4.2.3 Network node MAC/PHY description

The considered heterogeneous 802.11 scenario consists of \( N_s \) contending nodes in which the generic node is equipped with an antenna system of \( N(\geq 1) \) antenna elements. If \( N = 1 \), just a single omnidirectional radiator is available and the node behaves as a typical 802.11 station. Instead, if \( N > 1 \), the node can be equipped with an array antenna of \( N \) radiating elements managed by a processing unit in order to
produce a power gain pattern \( G(\phi) \). This implies that the generic node is able to suppress interference created by other nodes. During the omnidirectional PHY carrier sensing of the medium, an estimation \( l \) of the number of active transmitters is performed. Since the degrees of freedom of a multiple antenna system with \( N \geq 2 \) elements enable the estimation of up to \( N - 1 \) sources, an 802.11 legacy node can just sense the presence of energy and so is completely unable to derive the number of active nodes, while the maximum number of distinct transmitters that can be estimated by the generic node equipped with \( N \) radiating elements is equal to:

\[
I_{\text{max}} = \max\{1, N - 1\}. \tag{4.2.4}
\]

From the point of view of the generic node, all scenarios in which \( l \geq I_{\text{max}} \) are equivalent to \( l = I_{\text{max}} \). Hence, for the case \( l \geq I_{\text{max}} \) the estimation of the transmitting sources is not reliable.

The scheduling strategy adopted at MAC layer is the basic access mechanism and the simulation scenario assumes the absence of hidden terminals. The basic access is a viable mechanism that can be combined with the use of smart antenna systems by adopting a unique modification, consisting in the introduction of the sequence of \( m_t \) bits at the beginning of the DATA packet. This sequence is used by the receiving node to update its radiation pattern using the MUSIC plus cLMS spatial reference technique. Besides, during the PHY CS of the medium, the transmitting node estimates the number of active sources \( l \) by using a Maximum Description Length (MDL) method implemented in the MATLAB/Octave extension [60]. When \( l \) is lower or equal to a given threshold \( L_t \), the backoff counter is decreased and, if the backoff counter reaches the zero value, the packet is transmitted in omnidirectional operating mode. Thus, the backoff behavior is not determined by the idleness of the medium, but by the presence of a number of ongoing communications whose interference can be tolerated, according to the threshold \( L_t \) that must be properly selected [72, 73].

In the reception chain of the PHY/MAC layer two reception criteria are implemented: the threshold criterion (t-criterion) and the sustainable rate criterion (s-criterion). In the t-criterion a packet is correctly received if its SIR is larger than a given threshold \( \text{SIR}_t \) for the entire reception time. In detail, when the \( i \)-th node is transmitting, the \( \text{SIR}_{d(i)} \) of the received packet at the destination \( d(i) \) is calculated as:

\[
\text{SIR}_{d(i)} = \frac{P_{d(i),i}^{\text{rx}}}{\sum_{j'=1, j' \neq d(i), i}^{N_s} P_{d(i),j'}^{\text{rx}}} \quad i = 1, \ldots, N_s, \tag{4.2.5}
\]

where the term \( P_{j',i}^{\text{rx}} \) is the received power by the generic \( j' \)-th node when the generic node \( i' \) is transmitting. Since the considered scenario in asynchronous, the value of
4.3 Smart antenna system extension

The SIR may change during the packet reception period and it is constant only in the intervals when the set of active transmitting interferers remains equal (affecting the denominator of 4.2.5). The term $p_{j,i}^{TX}$ includes the transmitting power, the distance between two nodes and the gain of the transmitting and receiving antenna. In the proposed model the nodes are equipped with adaptive antenna systems, which are exploited in the reception to maximize the SIR by mitigating and suppressing the effect of the interferers.

The t-criterion approach may be too severe and may be not sufficiently realistic to model the behavior of some coding techniques. A more sophisticated solution may require the implementation of the code, but this may lead to a considerable increase of the simulation time. However, as shown in [74], the implementation can be avoided for efficient channel coding techniques, such as turbo codes since the Bit Error Rate (BER) curves as a function of the SIR have a behavior close to that of the step function. In particular, for the efficient codes, one can adopt the s-criterion, which considers the sequence of SIR values that, according to the modulation, provides a sequence of rates, estimated using the sphere packing bound, whose average value (sustainable rate) is compared to the selected code rate [74]. As previously reported, the sequence of SIR values is constant for the intervals in which the set of active interferers hold steady. The s-criterion, which is really accurate for turbo codes, is also applied to convolutional codes by introducing an offset of 5 dB to the SIR used to estimate the sustainable rate, in order to maintain a low simulation time and to properly account for the lower performance of a convolutional code.

4.3 Smart antenna system extension

The MATLAB/Octave extension is a set of functions and procedures implementing the channel-antenna model described in Section 1.3.2 that can be executed by both tools, since the syntax used by MATLAB and Octave is substantially identical. The MATLAB/Octave language has been chosen because it provides a simple and immediate development environment for implementing signal processing algorithms and functions adopted for the simulation of the PHY layer. In fact, the functions can be developed by writing few lines of code, since a large number of mathematical procedures are already present in the MATLAB and Octave environments. Besides, further extensions of the proposed platform can be easily developed by modifying the sole MATLAB/Octave part of the simulator. In this way also non C++ experts can improve the quality of the physical layer simulation of ns2.

The main component of the MATLAB/Octave extension is the function SAS.m, which provides the equivalent pattern $F(\varphi)$ in (4.2.3), and is defined as:

```matlab
function y = SAS(DoAs, Dist, pow, N, TypeArray,
```
where DoAs, Dist, pow are arrays containing the DoAs, the distances and the powers corresponding to the active nodes; \( N \), TypeArray, Distance define the antenna array characteristics; TypeAlg identifies the signal processing technique; MultiPath, TypeMultipath, AngleMultipath define the multipath statistic; and MutualCoupling is a flag that notifies if mutual coupling must be taken into account or not. The function SAS.m invokes the other implemented MATLAB/Octave functions: ULA.m, USA.m, UCA.m, and CRA.m for the generation of the array geometry; uLMS, RLS, MUSIC, and cLMS for the evaluation of the array excitations; Laplacian, Gaussian, and ring for the generation of the PAS; and, finally, create_pattern for the calculation of \( F(\phi) \). These functions operate using a discrete-time approach because the pattern adaptation is performed by processing sequences of bits generated in MATLAB/Octave that simulate the signals transmitted by the sources, according to the model described in Section 4.2.1.

4.4 Ns2 modification

Considering that a description of the ns2 simulator is given in Section 2.1, this section focuses on the required adjustments without describing the ns2 software architecture in detail. The presented modifications are referred to the version 2.33 of ns2 [56].

The introduction in ns2 of the developed channel-antenna extension requires the implementation of a new class, called SmartAntenna, which is defined in two new files, smart-antenna.cc and smart-antenna.h, that are inserted in the existing directory ns-2.33/mobile/. Besides, the existing method MAC802_11::recv_timer, defined in the file mac-802_11.cc in the directory ns-2.33/mac/, must be modified to evaluate the SIR at the receiving node using the new antenna model.

The unmodified ns2 version allows one to select a unique antenna model, implemented in the class OmniAntenna, which inherits the methods of the main class Antenna and returns a unitary gain for all directions \( \phi \). However, the main class Antenna provides a method, getRxGain, which is defined as virtual and thus allows one to modify the receiving gain by the simulation script. This means that, from the beginning, ns2 has been developed enabling the possibility to implement more advanced antenna models. The main component of the new class SmartAntenna is the method GainCalc, which provides the receiving pattern according to the position of the nodes. The method is declared in the file smart-antenna.h as:

\[
\text{double SmartAntenna::GainCalc(int ids, int nid, double rx, double}
\]
4.4. Ns2 modification

where ids and nds are the identifiers of the receiving/transmitting nodes, \((rx, ry, rz)\) and \((sx, sy, sz)\) represent the positions of the receiving/transmitting nodes, \(RxS\) and \(RxE\) are the time instants corresponding to the beginning and the end of the transmission, \(npt\) is the transmitting power, \(L\) is the path-loss exponent, and \(\lambda\) is the wavelength. The method \text{GainCalc} is invoked by the method \text{WirelessPhy::SendUp}, which in ns2 models the transition of a packet from the PHY to the MAC layer. When a receiving node \textbf{A} senses a transmission from another node \textbf{B}, the method \text{GainCalc} stores the position of \textbf{B} in a Neighboring Characteristic Table (NCT) (Fig. 4.2). Besides, \text{GainCalc} performs a further fundamental operation by enabling the complete knowledge of the SIR behavior of each packet as a function of the time. Observe, in fact, that the capabilities of smart antenna systems are properly exploited in terms of network capacity when more than one communication is allowed at the same time. Thus, in an 802.11-based scenario, the knowledge of the SIR, which
varies because of the activity of the other concurrent communications, becomes essential to establish if a given packet has been successfully received or not. This objective is reached by storing the instants of beginning and conclusion of the last $T_{last}$ transmissions of each node in a global table $Last\_txs$. The $T_{last}$ value is selected in order to cover the reception time of the DATA packet. Once NCT and $Last\_txs$ are updated, the node $A$ establishes if $B$ is an interferer or is the desired node (source or destination, as it is the case). In the first case $A$ continues to monitor the channel using an omnidirectional PHY carrier sensing, while, in the second case, $A$ requires the knowledge of the equivalent radiation pattern $F(\phi)$. If this pattern has been already evaluated in a previous communication, with an identical scenario of active sources, $A$ uses this pattern, which has been stored is a proper table. This improvement can be enabled in a micro-mobility scenario, namely when the path-loss, the shadowing, and the reciprocal position between the nodes do not change significantly during the simulation. This possibility considerably reduces the overall simulation time, since useless repetitions of identical link simulations are avoided. Conversely, if the receiving pattern is not present among those stored at the node $A$, the function GainCalc invokes the MATLAB/Octave functions necessary for the calculation of $F(\phi)$ according to the technique adopted by the interface software (dependent on the adoption of MATLAB or Octave). Observe that the equivalent pattern is calculated not only by the destination to receive the desired DATA packet, but also by the source to receive the ACK packet. Then, to evaluate the receiving gain in a given direction $\phi$, the method getRxGain of the main class Antenna is redeclared in the class SmartAntenna as:

```cpp
double SmartAntenna::getRxGain(double dx, double dy, double dz, double lambda)
where dx = rx - sx, dy = ry - sy and dz = rz - sz.
```

Finally, the method MAC-802_11::recv_timer evaluates the SIR for the entire packet duration, accounting for the possible presence of fading, modeled using a Rayleigh distributed block fading model, and error correcting codes, in order to establish the result of the transmission attempt.

Adopting the proposed MATLAB/Octave extension for ns2 described in this section, the use of smart antenna systems in the simulated nodes can be enabled using the command:

```tcl
set val(ant) Antenna/SmartAntenna
```

in the .tcl script, and the parameters can be directly set as:

```tcl
Antenna/SmartAntenna set N_ 16
Antenna/SmartAntenna set TypeArray_ 1
```
4.5 Integration of ns2 with MATLAB and Octave

The default values for the SmartAntenna class are written in the file ns-default.tcl in the directory /tcl/lib/.

The MATLAB/Octave extension and the modifications of ns2 described in the previous section hold independently of the used tool (MATLAB or Octave). Instead, the software interface in Fig. 4.2 for the integration of the developed extension with ns2 is strongly dependent on the tool. This section describes the techniques used to enable the exchange of data between MATLAB and ns2, and between Octave and ns2.

4.5.1 Matlab integration in ns2

Since the MATLAB package contains also a C++ Compiler, the developed MATLAB set of functions can be converted in the C++ language. This solution avoids the use of the extension as a stand alone application, guaranteeing a reduction of the simulation time with respect to an approach in which two distinct processes have to exchange data. In fact, the MATLAB Compiler provides the dynamic libraries that can be directly invoked from ns2. The main components for integrating ns2 with MATLAB are the procedures InitializeMatlab, EndMatlab, and SASC, which are implemented in the new file matlab.cc that is inserted in the new directory ns-2.33/matlab/. The first of these three procedures, which is invoked by the constructor of the ns2 class God, initializes the MATLAB libraries and is executed at the beginning of the simulation. The second procedure, which is invoked by the destructor of the class God, deallocates the memory allocated by InitializeMatlab and is executed at the end of the simulation. To enable the execution of these two
procedures in the class God, the matlab.h header file must be inserted in the file god.cc in the directory ns-2.33/mobile/. The third procedure, SASC, is the fundamental connection between the MATLAB extension and ns2, and is declared as:

```c
void SASC(double DoAs, double Dist, double pow, double N,
    double TypeArray, double Distance, double TypeAlg,
    double MultiPath, double TypeMultipath, double AngleMultipath,
    double MutualCoupling, double Gain)
```

where Gain is the calculated equivalent power gain, namely an array having 360 elements. The procedure SASC, which is invoked by the method GainCalc of the class SmartAntenna and invokes the MATLAB function SAS for the calculation of the equivalent radiation pattern $F(\phi)$, has a further important role because it converts the standard C++ arrays used by ns2 in the non-standard C++ mwArrays used by MATLAB and vice versa. This conversion is a key point to allow the exchange of information between ns2 and MATLAB, considering that the two tools operate on different data type structures.

The conversion of the developed MATLAB package to C++ language can be performed from the MATLAB prompt using the command:

```
mcc -W cpplib:libSAS -T link:lib SAS.m -v
```

which specifies that the MATLAB Compiler has to produce the shared C++ library libSAS from the script SAS.m. This conversion generates four files: libSAS.so, representing the dynamic library and describing the user-defined function; libSAS.ctf, representing a Component Technology File (CTF) archive that includes the MATLAB based content (.m files); libSAS.cpp and libSAS.h that must be included in the source code of ns2. All these files and all the files of the generated directory MATLAB/extern/include have to be copied into the new directory ns-2.33/matlab. Finally, the following lines must be inserted in the Makefile.in of ns2:

- `-I./matlab` in the section INCLUDES
- `@V_LIB@ -L./matlab -lSAS` in the section LIB
- `matlab/matlab.o, mobile/smart-antenna.o` in the section OBJ_CC

in order to specify the additional files that must be considered during the ns2 compilation.
4.5. Integration of ns2 with MATLAB and Octave

4.5.2 Octave integration in ns2

Differently from MATLAB, no C++ compilers are available for Octave and hence different techniques must be adopted to interface Octave with ns2. Three possible approaches have been investigated and implemented.

4.5.2.1 Library OctaveEmbedded

The first approach is based on the use of the open source library OctaveEmbedded, that contains the files embed.h and embed.cc, defining the procedures octave_init and octave_exit, and the function octave_call, which recall some functions, included in the Octave source code, to enable the execution of Octave from another program. The procedure octave_init disables the command prompt to allow the execution of Octave in background and is inserted in the developed procedure InitializeOctave, which is executed at the beginning of the simulation by the constructor of the ns2 class God. The procedure octave_exit is inserted in the developed procedure EndOctave, which is invoked at the end of the simulation by the destructor of the class God. InitializeOctave and EndOctave perform the same operations executed by InitializeMatlab and EndMatlab when MATLAB was used, but InitializeOctave has a further role. When octave_init is executed, Octave leaves the error management at the program with which is integrated (ns2 in this case), but performs a division by zero to test the system. In general, this test can stop the execution of the application because an error is signaled by the operating system, thus the procedure InitializeOctave must ignore divisions by zero at the beginning of the simulation to avoid the interruption of ns2. The function octave_call of the library OctaveEmbedded is the corresponding C++ of the MATLAB function eval, which parses the string received in input executing the content as an Octave command. Therefore, to enable the execution of the MATLAB/Octave extension, the procedure SASC, previously described in the MATLAB context, must be rewritten. This procedure has to convert the declarations of the C++ variables into Octave language (using further functions contained in the new files conversion.h and conversion.cc) and to use octave_call for interpreting each of the obtained declarations and the function SAS as commands. Furthermore, the SASC has to write the output provided by the SAS, namely $F(\psi)$, into a file that will be read by ns2.

To enable the use of this integration method, the following operations are required: the files octaveLib.h, octaveLib.cc, conversion.h, conversion.cc, embed.h, and embed.cc must be inserted in the new directory ns-2.33/octave/; the library octaveLib.h must be included into the files god.cc and smart-antenna.cc in the directory ns-2.33/mobile/; and, finally, all necessary libraries must be listed in the Makefile.in of ns2.

Observe that the above described approach is completely different from that ad-
opted for integrating MATLAB with ns2, where the function SAS was converted in C++ and then integrated in ns2. In this case, instead, the ns2 input data are processed by an Octave process executed in background that provides the results in a file. This method, however, presents the considerable drawback of requiring a large waste of time for the exchange of data, due to the conversion of all declared variables into strings representing commands (from ns2 to Octave) and to the writing/reading of the output files (from Octave to ns2).

4.5.2.2 Octave C++ libraries

The second approach for the integration of Octave with ns2 is based on the use of the C++ libraries of Octave that define its structure and all types of variables and functions it can employ. These elements should be used by a hypothetical Octave C++ compiler, which currently does not exist, to translate in C++ the scripts written in the MATLAB/Octave language. To employ the Octave C++ libraries, Octave must still be executed in embedded mode and all operations described for the previous integration method must be performed. Using the types of variables provided by these libraries, all input (DoAs, Dist, p, N, TypeArray, Distance, TypeAlg, MultiPath, TypeMultipath, AngleMultipath, MutualCoupling) and output (Gain) variables are grouped in a cell variable of type \textit{octave\_value\_list} and are directly initialized from ns2, avoiding the use of files. Besides, the libraries \textit{oct.h}, \textit{octave.h}, \textit{parse.h}, \textit{toplev.h} must be included in the file \textit{octaveLib.cc}. In this second integration method the procedure SASC has to convert the input C++ variables of type double into Octave variables of type \textit{Matrix} and to join them into a variable of type \textit{octave\_value\_list}, which is passed as input argument to the function SAS using the command \texttt{feval}, defined in the C++ library \textit{parse.h}. The output $F(\phi)$ provided by the SAS, contained in the variable results (of type \textit{octave\_value\_list}), is then copied into the C++ variable Gain (of type double).

With respect to the previous integration technique, this second solution does not require the exchange of files and the variables can be directly exchanged by the two tools. However, the time required by this technique remains considerable, thus a further solution involving the First In First Out (FIFO) pipes has been explored.

4.5.2.3 FIFO pipes

All previous integration methods lead substantially to hybrid platforms MATLAB-ns2 or Octave-ns2. An approach based on the use of FIFO pipes, instead, enables to treat Octave and ns2 as two distinct processes that are executed independently and communicate each other to exchange the input/output data.

The FIFO pipes are half-duplex links between two processes, characterized by two pointers denoting the read and write points of the pipe, through which each process
can send and receive the data to and from the other process. In the adopted implementation strategy the constructor of the class God starts Octave (the father process) and an Octave server, implemented in a file serverOctave.cc, and also reads a file containing the names of the functions required by ns2 and the number of input/output data relative to each function. Then, the initialization procedure generates two FIFOs, whose names are known by the Octave server, from which ns2 sends to Octave the information acquired from the file and the Process IDentifier (PID). The initialization procedure also generates two FIFOs for each used Octave function, one in the direction Octave-ns2 (StoC) and another one in the direction Octave-ns2 (CtoS), and all existing FIFOs that are no more necessary are deleted. At this point the Octave father process generates a child process (a copy of Octave itself) for each requested function and links each function with a pair of FIFOs. The Octave father process, which is no more linked to any FIFO, simply awaits the end of the child processes. At the ns2 side, for each requested function, four procedures must be implemented: create_buffer, which inserts all input data of the requested function in a buffer; fill_pipe, which sends the buffer through a StoC FIFO; empty_pipe, which awaits the results from a CtoS FIFO; and call_octave_funct, which invokes the above three procedures when ns2 needs the use of an Octave function and is inserted in the procedure SASC. Once the Octave child process has executed the requested function, the results are send through the specific CtoS FIFO to ns2. When the network simulation is completed, the destructor of the class God of ns2 sends on each FIFO the request to close the process. Thus, all Octave (child and father) processes are closed and all ns2 FIFOs are deleted.

To enable the above described integration method a Makefile must be created and inserted in the directory ns-2.33/octave/ in order to compile the Octave server. With respect to the two previous integration methods, the use of FIFOs has the further advantage of enabling the simultaneous execution of many simulations, since each ns2 process has its group of reserved FIFOs.

4.6 Simulation results

This section compares the two proposed simulation platforms: MATLAB-ns2 and Octave-ns2, by investigating the possibility to employ smart antenna systems in a DWN.

4.6.1 Results

The receiving radiation pattern generated by the smart antenna system is synthesized considering a UCA with \( N = 8 \) elements. The adopted topology, involving \( N_s = 38 \) nodes placed inside a circle of radius \( R_c \) of 35 m, is shown in Fig. 4.3, which
also reports the source-destination pairs and the corresponding traffic flows. The sources are represented by square markers (outer ring) and the destinations by circle markers (inner ring). The network operates in saturated traffic conditions and 20 seconds of network evolution are considered. The MAC layer parameters are reported in Table 4.1.

The saturation throughput as a function of the active sources’ threshold for the adopted scenario is reported in Figs. 4.4a and 4.4b. More precisely, Fig. 4.4a is obtained using the MATLAB-ns2 platform, while Fig. 4.4b is obtained using the Octave-ns2 platform considering the implementation based on the use of the FIFOs. In fact, preliminary tests have revealed that this integration approach represents the fastest Octave-ns2 integration technique among those described in Subsection 4.5.2. Both figures are obtained using the t-criterion with $\text{SIR}_t = 5$ dB to establish the success of a transmission. This choice guarantees a Packet Error Rate (PER) equal to 1% for packets having length equal to 1470 bytes using a data rate equal to 12 Mbits/s, corresponding a QPSK modulation with a coding rate 1/2 in the 802.11g PHY layer extension [75]. The curves in the two figures prove that the two platforms provide sub-
4.6. Simulation results

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFS</td>
<td>$10,\mu s$</td>
</tr>
<tr>
<td>DIFS</td>
<td>$50,\mu s$</td>
</tr>
<tr>
<td>$\sigma_{\text{slot}}$</td>
<td>$20,\mu s$</td>
</tr>
<tr>
<td>Packet size</td>
<td>1470 bytes</td>
</tr>
<tr>
<td>Maximum backoff stage</td>
<td>1</td>
</tr>
<tr>
<td>Retry limit</td>
<td>4</td>
</tr>
<tr>
<td>$R$</td>
<td>12 Mbits/s</td>
</tr>
<tr>
<td>$CW_{\text{min}}$</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 4.1: Simulation parameters.

Substantially identical results, confirming the reliability of the implementation.

The throughput in Figs. 4.4a and 4.4b accounts for multipath and for the effect of the ACK packets. While the performance reduction due to the increase of the angular spread is somewhat predictable, the significant throughput decrease due to the ACK may seem an unexpected degradation, at least for its dimension. This behavior, which is mainly due to the severe reception criterion, can be explained as follows. Usually, in the simulation of the legacy 802.11 networks, namely those considering a node equipped with a unique omnidirectional antenna, the effect of the ACK is neglected or its duration is included in the duration of the DATA packet to simplify the modeling process. This a reasonable assumption since in the omnidirectional antenna scenario a pair of nodes communicates while the others are silent (assuming the absence of hidden terminals). Accordingly, the throughput values for $L_t = 0$ (only one communication allowed) in Figs. 4.4a and 4.4b are substantially identical regardless of the presence of the ACK packets. Instead, in presence of multiple simultaneous communications ($L_t > 0$) a node can access to the medium even if other nodes are exchanging packets. Besides, the destination protects the reception of the desired packet from the interference generated by the other active sources by properly generating its receiving radiation pattern, according to the angular directions sensed as active. Since the DATA packets have usually a duration much larger than the ACK packets, the destination often senses the transmission of the DATA packets and uses this information to synthesize the receiving pattern. Thus, the transmission of the ACK represents an event of brief duration for which the destination is often unprepared. This means that often the ACK is received from an angular direction not suppressed by the receiving pattern, corresponding to a higher gain. Exploiting the capabilities of the developed simulator to provide the fluctuations of the SIR of a packet due to the beginning and the conclusion of the other ongoing transmissions, one can observe the temporal evolution of the SIR of all DATA packets involved in the simulations. A typical one is shown in Fig. 4.5, which confirms that the concurrent transmission of a unique ACK can lead to the failure of the DATA reception. Summarizing,
in a scenario where multiple simultaneous communications are obtained using adaptive arrays, the ACK has a much more significant impact as compared to an omnidirectional scenario. The adopted reception criterion, based on a threshold, largely emphasizes this impact, which may be mitigated by using an interleaver combined with a convolutional code (solution already present in the 802.11abg extensions) or with a more efficient turbo code. The effects of these two solutions are explored in Fig. 4.6, which reports for $L_t = 7$ the throughput in non-saturated traffic conditions assuming a Poisson distributed packet arrival with mean $\mu$. The figure reveals that, if the complete SIR behavior is considered (s-criterion), more realistic results are obtained and the performance largely increases even if only convolutional codes are adopted. Fig. 4.6 also confirms that the adoption of a turbo code can provide further improvements, guaranteeing a more efficient compensation of the interference due to all concurrent (DATA and ACK) transmissions with respect to a convolutional code. This advantage of turbo codes is more evident when the traffic load becomes relevant. Fig. 4.7 shows the results for a channel characterized by Rayleigh fading when the $s$-criterion is used. The figure proves the advantages of turbo codes, which are able to recover more efficiently the bits degraded by fading as compared to convolutional codes. Moreover, remembering that a single-link with a data rate of 12 Mbits/s and a coding rate of 1/2 can reach, in practice, a rate of approximately 5 Mbits/s when the overhead due to headers and control packets is considered, the through-
4.7 Simulation time comparison of the MATLAB/Octave extension

This section presents the performance of MATLAB-ns2 and Octave-ns2 simulation tools. The study is partially reported in [54]. The comparison is performed adopting the scenario depicted in Fig. 4.3 for both simulation platforms.

Figs. 4.8a and 4.8b report the simulation time necessary to obtain each point of the curves in Figs. 4.4a and 4.4b, respectively. The simulation time does not strictly follow the throughput behavior, but continues to increase after the maximum throughput is reached. In fact, the simulation time is related to the number of transmission/reception attempts (successful or not), which increases as the threshold $L_t$ increases and leads to an increase of the calls to MATLAB or Octave.

As expected, the MATLAB-ns2 platform is faster than the Octave-ns2 one. A direct comparison between MATLAB and Octave can be performed running a simple
Chapter 4. Discrete-time simulation of smart antenna systems in Network Simulator-2 using MATLAB and Octave

Figure 4.6: Throughput as a function of the packet arrival rate for different reception criteria and coding techniques evaluated using the MATLAB-ns2 platform.

—♦— Considering ACK: s-criterion (turbo code)
−−−−−−−−− Considering ACK: s-criterion (convolutional code)
—— Ignoring ACK: t-criterion

script involving only the function SAS.m alone (in absence of ns2) with the following input parameters: DoAs=[353,129,177,328,19], Dist = [26,37,44,28,25], pow = [0.1, 0.1, 0.1, 0.1, 0.1], N = 8, TypeArray = 1, Distance = 0.5, TypeAlg = 2, MultiPath = 0, TypeMultipath = 0, AngleMultipath = 0, MutualCoupling = 0. To execute this script MATLAB requires approximately 0.21 seconds, while Octave requires 0.95 seconds. Thus, MATLAB is approximately 4.6 times faster than Octave. However, comparing Fig. 4.8a with Fig. 4.8b and averaging for all $L_t$ values, one can infer that usually the corresponding MATLAB-ns2 platform is only 3.6 times faster than the Octave-ns2 one. This reduction of the gap between the two approaches reveals that the Octave integration method based on the FIFOs does not produce substantial increases of the simulation time, and confirms the usefulness of the improvement inserted in the ns2 part of the simulator, which invokes the MATLAB/Octave extension only when the receiving pattern is not present among those previously calculated and stored (Fig. 4.2). This also provides a further explanation for the monotone behavior of the simulation times in Figs. 4.8a and 4.8b as a function of $L_t$. In fact, as the load threshold increases, the number of possible combinations of active sources increases
4.8 Conclusions

Two integrated platforms adopting an hybrid discrete-time discrete-event approach for the simulation of smart antenna systems in ns2 have been presented. The first platform is based on the proprietary software MATLAB, while the second platform employs the open source tool Octave. The results, identical for both platforms, have revealed that, as expected, the use of MATLAB requires a lower simulation time. However, the adoption of proper programming improvements have lead to a reduction of the differences in terms of speed between the two presented approaches. Thus, the Octave-ns2 platform may be an interesting solution, with longer (but acceptable) simulation times with respect to MATLAB, and in agreement with much more diagrams must be generated, leading to an increase of the number of MATLAB/Octave calls until the database of the stored patterns is sufficiently complete.

Figure 4.7: Throughput as a function of the packet arrival rate for different coding techniques in presence of fading adopting the s-criterion. The curves are obtained using the MATLAB-ns2 platform.

- Turbo code
- Convolutional code
Chapter 4. Discrete-time simulation of smart antenna systems in Network Simulator-2 using MATLAB and Octave

Figure 4.8: Simulation time as a function of the active sources’ threshold for (a) the MATLAB-ns2 platform and for (b) the Octave-ns2 platform.

- Absence of multipath ($\hat{\sigma}_\phi = 0^\circ$) ignoring ACK packets
- Presence of multipath ($\hat{\sigma}_\phi = 20^\circ$) ignoring ACK packets
- Absence of multipath ($\hat{\sigma}_\phi = 0^\circ$) considering ACK packets
- Presence of multipath ($\hat{\sigma}_\phi = 20^\circ$) considering ACK packets

The current versions of the MATLAB-ns2 platform and of the Octave-ns2 one with the implementation based on the FIFO queues can be freely downloaded from [76].
Multi-packet communication for distributed wireless networks using advanced antenna systems

This chapter discusses the design requirements for enabling multiple simultaneous communications in IEEE 802.11 asynchronous networks in the presence of adaptive antenna arrays, and proposes two novel medium access control protocols to realize multi-packet communication maintaining backward compatibility with the 802.11 standard. Both presented solutions rely on local information and are suitable for distributed and heterogeneous networks, where legacy and non-legacy nodes equipped with different antenna systems can coexist. The results obtained from the presented protocols are discussed with respect to the performance predicted by theory and to that provided by the 802.11 super-g and 802.11n physical layer extensions. Finally, a comparison of the adopted C++ simulation platform with the MATLAB-ns2 simulation tool is presented. ¹

5.1 Introduction

The use of high-throughput applications by many users in 802.11 wireless networks represents a challenging task in terms of performance and capacity increase. To deal with this issue, the adoption of advanced antenna systems for increasing the capacity of adhoc networks represents a widely investigated topic even from the approval of the 802.11 legacy standard [79–83]. Accordingly, the Task Group n has developed the 802.11n extension [5], which uses multi-antenna systems for increasing the data rate of the single link, but maintains the access limited to just one user at a time. To deal with this limitation, the benefits of advanced antenna sys-

Chapter 5. Multi-packet communication for distributed wireless networks using advanced antenna systems

tems for supporting multiple simultaneous communications in 802.11-based networks have been explored by theoretical studies [38, 73, 84–90], and numerical investigations [9, 10, 13–18, 20, 21, 91]. In this field the research efforts are currently focused on two main topics: the simultaneous reception of more packets by a unique destination, also known as multi-packet reception, and the concurrent communication between different node pairs, defined as multi-packet communication. In MPR scenarios the objective is to enable a destination to simultaneously receive multiple packets from different sources. Thus, the access must be synchronized, since all allowed transmissions must begin at the same time instant [38, 84]. By contrast, in MPC scenarios, the aim is to enable multiple concurrent communications between different node pairs. Thus, the access can be asynchronous, since each source may start its transmission independently of the time instants chosen by the other sources [73, 87].

MPR is widely considered as a suitable solution for increasing the capacity of centralized 802.11 networks, where the fundamental role of coordination and synchronization is played by the Access Point (AP) [18, 91]. Accordingly, two further 802.11 draft extensions adopting a multi-packet approach have been recently developed: the 802.11ac amendment for the bands below 5 GHz [92], and the 802.11ad one for the 60 GHz band [93]. In general, MPR may be also applied to a distributed 802.11 network, where the synchronization can be reached by properly modifying the RTS/CTS handshake of the DCF [15]. However, the theoretical analysis presented in [73] proves that, in a distributed 802.11 network, an asynchronous access can provide a higher throughput than a synchronous one, thus making MPC more suitable for a completely distributed environment. On the other hand, as shown in [90], the conditions for fully exploiting smart antenna systems cannot be completely realized in a distributed scenario, since the distribution of the active transmitters can change when a source-destination communication is already in progress. Despite that, a centralized control can be sometimes not realizable or disadvantageous and hence several MAC protocols using advanced antenna systems and enabling MPC in 802.11 networks have been presented [10, 17, 20, 21]. Among these proposals, three relevant aspects must be considered. The first one, taken into account in [20, 21], involves the backward compatibility with the 802.11 standard, since even a minor modification of the access rules may inhibit the communications of the legacy nodes [9, 10, 13–17]. The second aspect concerns the heterogeneity of the scenario, since legacy and non-legacy nodes with different interference suppression capabilities should be able to coexist within the same network, avoiding the case where the more advanced nodes acquire all the resources at the expense of the less advanced ones [21]. Finally, according to [90], the third aspect concerns the ability of the access scheme to provide to each node position and traffic information, in order to approach as much as possible the performance of a centralized network.

After discussing the main requirements that an access scheme operating in pre-
5.2. Scenario description

In the presence of adaptive antenna array for enabling multi packet reception in 802.11 DWNs, two novel MAC protocols are designed according to these requirements and to the theory developed in [73]. The first protocol, called Threshold Access MPC (TAMPC), adopts a medium access policy based on a threshold on the sustainable load that depends on the single-node antenna capabilities. The second protocol, called SIR Access MPC (SAMPC), adopts a reliable estimation of the behavior of the channel encoder and a local but accurate estimation of the SIR, which accounts for the position, the traffic pattern, and the duration of each sensed communication. The performance of the two developed schemes is evaluated using a realistic simulation platform and compared to that predicted by the theory. The objective is to provide advanced MAC layer algorithms that guarantee MPC, 802.11 backward compatibility, and that are suited for heterogeneous scenarios, since, to the best of authors' knowledge, solutions that simultaneously satisfy all these three requirements, even if needed, are still not available.

As previously reported, the simulations are performed adopting a novel network simulation tool instead of using MATLAB/Octave-ns2 simulator introduced in Section 4.5. The new simulation tool has been developed in order to assess the adherence of the simulations to the analytical models obtained from theory. At the end of the section the two simulation tools are compared in terms of results reliability and adherence to the results obtained from theory.

5.2 Scenario description

This section describes the scenario considered in the presented study, with a special remark to the physical layer of the nodes equipped with adaptive antenna arrays. The accurate description of the node physical layer points out some relevant aspects that are useful to the comprehension of the design requirements of an access scheme.

5.2.1 Managed scenario

The considered network scenario is characterized by two main characteristics: asynchrony and heterogeneity. Being the 802.11 MAC protocol based on a slotted CSMA algorithm, the asynchrony does not refer to the lack of clock synchronization, which may lead to an unslotted access, but to the possibility that each source starts its transmission, at the beginning of a slot, independently of the slots chosen by the other sources. Thus, according to [9,10,13–18,20,21,38,73,84–87,90], a slotted access is assumed and the asynchrony implies the absence of policies for grouping the contending sources. The heterogeneity stems in first place from the coexistence within the same network of legacy and non-legacy nodes, where the non-legacy nodes can differ for their antenna system characteristics (number of antenna elements, array
geometry, beamforming algorithm). However, the heterogeneity may be also determined by relative asymmetries in the topology or different channel conditions in the space-time domain. In fact, two nodes with identical antenna systems may anyway experience significant differences in their ability to communicate, since the capability of suppressing an interferer depends on the reciprocal position of the nodes. This implies that a network in which each node has a different reciprocal position with respect to the others is in practice heterogeneous, even if it consists of nodes with identical antenna systems.

In a synchronous-homogeneous scenario, a collision represents a situation where the number of active communications is larger than the number of sustainable ones. Assuming packets of equal length, this event leads to the loss of all involved packets, since the transmissions are perfectly superimposed because of the synchronization (all bits are corrupted). In all the other three possible cases (homogeneous-asynchronous, heterogeneous-synchronous, heterogeneous-asynchronous), the collision may determine the loss of only a subset of the involved packets. In fact, different packets, even of the same length, may be differently corrupted because of the asynchrony, and/or different nodes may have different interference mitigation capabilities because of the heterogeneity. This implies that in a heterogeneous-asynchronous network the consequences of a collision must be evaluated locally, since they depend on the individual characteristics of the involved nodes and on the evolution of each SIR.

5.2.2 Antenna parameters of the single-node

Consider a distributed 802.11 network including a set \( \mathcal{N} \) of \( N \) nodes, where the generic node \( i \in \mathcal{N} \) has an antenna system of \( N_i \geq 1 \) elements. The set \( \mathcal{N} \) can be partitioned in two subsets: the subset \( \mathcal{N}^l \), containing the \( n_l \) 802.11 legacy nodes having \( N_i = 1 \) \((i \in \mathcal{N}^l)\), and the subset \( \mathcal{N}^{nl} \), containing the \( n_{nl} = N - n_l \) non-legacy nodes having \( N_i > 1 \) \((i \in \mathcal{N}^{nl})\). During the reception of a packet each non-legacy node equipped with a smart antenna system can generate a radiation pattern \( G(\varphi) \) having the main lobe steered towards the desired direction and a certain number of nulls placed towards the interferers [60]. In particular, \( G(\varphi) \), which depends on the adopted beamforming algorithm, on the geometry and on the number of elements of the antenna array, can be characterized by the average gain \( G_{a,i} \) and the null gain \( G_{n,i}(<< 1) \) [38].

For a homogeneous scenario, the capability of the node \( i \) to correctly receive a packet in an interfered environment may be modeled by considering the number of communications \( L \) that can be sustained by the network [38], and the number of ongoing transmissions \( L_i(< L) \) that allows a contending node to asynchronously transmit its packet without destroying the communications of the other nodes [73]. In a heterogeneous network each node \( i \) has its own value for \( L_i \), and, differently
from the homogeneous case, \( i \) cannot infer the thresholds of the other nodes from the knowledge of its \( L_t^i \). In particular, if \( i \) is an 802.11 legacy station, then \( L_t^i = 0 \), because a node with a unique omnidirectional antenna can correctly receive a packet only if no other transmissions are simultaneously active within its communication range. Instead, if \( i \) is a non-legacy node, then \( L_t^i \) is related to the channel-antenna characteristics and to the network topology [38, 82].

The availability of the above described antenna parameters suggests two possible MAC layer approaches for introducing MPC in an 802.11 DWN: one based on the ‘resumptive’ parameter \( L_t^i \), and another one based on a detailed estimation of the instantaneous SIR. The first approach has the advantage of summarizing the channel-antenna characteristics in a unique quantity that may be calculated offline and included in the set of the fixed 802.11 MAC layer parameters, similarly to the minimum contention window or the maximum backoff stage. In fact, \( L_t^i \) may be approximately evaluated assuming a homogeneous scenario, in which all \( N_s \) nodes have \( N_t \) antennas, and using the theory developed in [38, 73]. This first approach may reduce the computation required at each node, but it implies adopting an averaged quantity. A second approach, instead, may be developed by considering a real-time local estimation of the instantaneous SIR, where the estimation is able to account for the single-node antenna parameters and the relative position of the node within the network. This second solution may imply a higher computational burden, but may guarantee an access behavior that is more adherent to the real traffic evolution.

5.3 MAC protocol requirements and design strategy

This section focuses the main requirements, which may be considered during the development of new MAC protocols exploiting smart antenna systems in order to increase the performance and the reliability, while maintaining backward compatibility with the 802.11 legacy access rules.

5.3.1 Identified requirements

Independently of the adopted approach, a commercially attractive extension for the 802.11 DCF using adaptive arrays should fulfill three main requirements.

First, the adopted policy should preferably be asynchronous, since, in a distributed 802.11 network, MPC can provide a higher throughput than a synchronous approach [73]. For completely exploiting adaptive arrays in this context, a reliable criterion for the channel access should consider two main objectives [90]: the acquisition of the information concerning the active nodes, and the preservation of the conditions present at the beginning of the transmission for the entire duration of the transmission itself (since the generated pattern \( G(\phi) \) cannot be changed dur-
Chapter 5. Multi-packet communication for distributed wireless networks using advanced antenna systems

ing the reception). In a DWN, while the first objective can be reached by monitoring the channel activity and by exploiting the information contained in the control packets, the second objective cannot be completely achieved, because of the absence of a centralized control and the asynchrony of the scenario [90]. Thus, to enable MPC in distributed environment, a MAC protocol should allow a source to acquire the distribution of the interferers at the beginning of its transmission, and should provide some mechanisms to approximate the possible time evolution of that distribution.

The second fundamental requirement to make the possible introduction of MPC in future 802.11 networks realistic is the backward compatibility of the developed MAC layer extensions. In particular, MPC must be obtained in a way that guarantees the possibility of communicating to the single-antenna legacy nodes, thus avoiding modifications of the 802.11 MAC layer that may be perceived as in contrast with the access rules adopted by these stations.

The third fundamental requirement for a MAC protocol enabling MPC in DWNs concerns the heterogeneity of the scenario. Since 802.11 networks are often extended by adding new routers and the continued progresses in antenna miniaturization enable inserting more and more antennas on a device [94, 95], the scenario of the near future may involve devices of different ‘generations’, characterized by increasing antenna capabilities. Thus, heterogeneity may represent a plausible aspect of forthcoming networks, where the MAC protocol should prevent that nodes equipped with more powerful antenna systems acquire all the resources at the expense of the nodes equipped with less powerful antenna systems.

### 5.3.2 Design strategy

The requirements discussed in the previous subsection suggest some preliminary choices in the design strategy for extending the 802.11 DCF to MPC.

First, the RTS/CTS access seems preferable to the basic access, since the former guarantees the possibility to disseminate a larger quantity of information relative to each starting communication, including the antenna parameters, the duration of the transmission, and the rate adopted by the channel encoder. The basic access mechanism may be even modified to enable the DCF to support multiple simultaneous communications [10], but the achievable throughput performance is considerably lower than that achievable using the RTS/CTS access [90]. Accordingly, almost all MAC protocols proposed for supporting MPC in asynchronous networks rely on the four-way handshake mechanism of the 802.11 [9, 13–18, 20, 21].

The second choice consists in adopting two non-overlapping frequency channels, where the first channel can be used by both the legacy and the non-legacy nodes and represents the Common Channel (CC), while the second one can be used only by the non-legacy nodes and is defined as the Multiple Communications Channel (MCC). Just one communication at a time is allowed on the CC, while multiple sim-
ultaneous communications are allowed on the MCC. A legacy node can monitor only the CC, while a non-legacy node can perform an omnidirectional sensing of both channels. The main reason for using two separate channels is that it allows adding novel fields to the control packets, where the new fields are necessary for providing to all sensing nodes the information required for exploiting their antenna systems. Non standard frames cannot be used in a channel shared with the legacy nodes, which are unable to understand the novel formats. Besides, in the standard RTS/CTS frames there are no bits available for including novel information, since all fields are used for standard operations [18].

The third choice concerns the adoption of omnidirectional transmissions on both channels, which implies that the non-legacy nodes use their interference mitigation capabilities only during the reception of the packet. This choice, which may seem to lead to a performance reduction as a consequence of a lower spatial reuse, leads instead, firstly, to an increased robustness against hidden terminal and deafness problems [9], and, secondly, to a simplification of the sensing of the active transmissions [17]. In particular, this second aspect is relevant when a node, being involved in a communication in the MCC, has lost some RTS/CTS exchanges and, once its communication is completed, tries to update its knowledge of the medium occupation.

In summary, the adoption of RTS/CTS access, omnidirectional sensing, and omnidirectional transmission may represent a suitable combination for an exhaustive dissemination and acquisition of the information concerning the medium occupation, thus matching the first requirement for completely exploiting adaptive arrays in a distributed scenario [90]. Moreover, the use of two separate channels is advantageous in terms of backward compatibility, since a communication involving a legacy node is performed in the CC, and in terms of exploiting the antenna system capabilities, since MPC is allowed in the MCC, where non-standard frames can be used.

5.3.3 Recognition

Since legacy and non-legacy nodes coexists in the CC, the first novel feature that must be introduced in an MPC extension for the 802.11 MAC layer is a mechanism of reciprocal recognition of the non-legacy nodes, but using a method that must not modify the legacy communication rules. To this purpose, one may properly use the More Data field of the RTS/CTS control frames. The 802.11 standard states that this field, consisting of a unique bit, must be set equal to zero and can be ignored by the legacy nodes during the contention period, since More Data has meaning only for centralized operations [1, p. 37]. In the proposed approach for the reciprocal recognition of the non-legacy nodes, instead, this field can be set equal to one and read even during the contention period. In particular, when a generic non-legacy source $i$ has a packet for a destination $d(i)$ of unknown antenna characteristics, $i$ can communicate to $d(i)$ its multi-antenna capabilities by setting to one the More Data field.
of the transmitted RTS packet and can also infer the (possible) multi-antenna capabilities of d(i) by reading the More Data field of the received CTS. Thus, if i and d(i) are multi-antenna nodes, they are aware of being involved in a non-legacy-non-legacy communication. Instead, if i and/or d(i) are legacy nodes, they will not read the More Data field and the communication will be completed following the usual standard operations. Observe that this recognition process is completely transparent for the legacy nodes, being the format of the RTS/CTS packets in agreement with the 802.11 standard.

The adopted mechanism of recognition is used both in the TAMPC protocol, presented in Subsection 5.4.1, and in the SAMPC protocol, presented in Subsection 5.4.2. For both schemes, the CC, in which only one communication is allowed at a time, is used for all communications involving at least a legacy node and for the first exchange (recognition) between a non-legacy source and its non-legacy destination. Instead, the MCC is used only by the non-legacy nodes and represents the MPC channel. In the CC the access rules relative to the decrease of the backoff counter will be maintained identical to the legacy ones, while in the MCC they will be modified.

5.4 Proposed protocols

This section presents two PHY/MAC protocols, which adhere to the requirements presented in Section 5.3, and are the result of the conducted study in the field of the PHY/MAC protocol design.

5.4.1 Threshold access multi-packet communication protocol

The TAMPC protocol is based on the use of the threshold $L_{t_i}$, introduced in Subsection 5.2.2, which summarizes the channel-antenna characteristics of a node i [38, 73]. The behavior of the proposed scheme is described considering the communication between a generic non-legacy source i and its non-legacy destination d(i).

5.4.1.1 Operations in the CC: single communication

Once the RTS/CTS exchange and the subsequent recognition process are completed, if i-d(i) is a non-legacy pair, i transmits the DATA packet by adding its threshold $L_{t_i}$ and a preamble. This preamble is used by the smart antenna system of d(i) to estimate the direction of arrival of the currently active node and to synthesize the receiving radiation pattern using the adopted beamforming algorithm [60]. Similarly, once the DATA reception is completed, the destination d(i) transmits the ACK packet by adding its threshold $L_{t_{d(i)}}$ and the preamble. After this first RTS/CTS/DATA/ACK
handshake the nodes $i$ and $d(i)$ have acquired the reciprocal characteristics. These characteristics are inserted in a table, called neighboring characteristic table, having one entry for each neighbor with which a communication has been already performed. More precisely, for the node $i$, the generic entry $j'$ of the NCT$_i$ contains the IDentifier (ID) of $j'$ (ID$_{j'}$), the corresponding threshold $L_{t_{j'}}$, the estimated DoA $\phi_{i,j'}$, and the NAV relative to $j'$ (NAV$_{j'}$). Observe that the modifications of the DATA and ACK packets (threshold and preamble) are not perceived by the legacy nodes because, once the RTS/CTS exchange is completed, the legacy nodes set their NAVs, thus turning off their radios for the time specified in the duration field of the sensed RTS (or CTS) packet.

5.4.1.2 Operations in the MCC: multiple communications

Consider the case where a non-legacy source $i$ has a packet for an already recognized non-legacy destination $d(i)$, whose ID is hence already present in the NCT$_i$. In this case the non-legacy pairs are allowed switching to the MCC. As in the 802.11 DCF, $i$ generates the backoff and monitors the medium. To this aim, $i$ uses its antenna array to estimate the DoAs $\phi_{i,1},...,\phi_{i,j'},...,\phi_{i,l_i}$ corresponding to the $l_i$ currently active transmitters. This estimation is performed every time a power detector reveals a variation of the power level in the MCC for a duration longer than a SIFS. This variation means that a new transmitter has started to send a packet or an old transmitter has completed its transmission. Since the scenario is asynchronous, the estimation $l_i$, representing the number of nodes that are physically transmitting a packet, can include RTS, CTS, DATA, and ACK transmissions. To properly exploit this information in a heterogeneous scenario, the source $i$ has to take into account that, since an array with $N_i \geq 2$ elements may allow estimating up to $N_i$ active transmissions, this estimation can be considered reliable only for $l_i \leq N_i - 1$. In fact, the estimation $l_i = N_i$ cannot be considered reliable, since it is provided in all cases where the real value $l$ of the transmissions that are active in the MCC is larger or equal to $N_i$. Thus, from the point of view of $i$, all cases $l \geq N_i$ are equivalent to $l_i = N_i$ and are beyond the antenna capabilities of $i$. Therefore, if $l_i = N_i$, the MCC is considered busy and $i$ freezes its backoff counter. Instead, if $l_i < N_i$, $i$ examines the NCT$_i$ for each $j' = 1,...,l_i$ to verify if the entry relative to $\phi_{i,j'}$ is complete or lacks in the corresponding ID$_{j'}$, $L_{t_{j'}}$, or has a NAV$_{j'}$ equal to zero. If each entry is complete, $i$ continues to monitor the channel, while, if an entry $j'$ is not complete, the packet incoming from $\phi_{i,j'}$ is received. To this purpose, $i$ runs its beamforming algorithm by setting $\phi_{i,j'}$ as the desired direction and all the other estimated DoAs as the undesired directions. The received packet is then delivered from the PHY to the MAC layer. If this packet is relative to a communication between a source $k_{nl}$ and a destination $d(k_{nl})$, the contained information is used by $i$ to update the NCT$_i$. In particular, the control packets sent in the MCC (RTS$_{MCC}$ and CTS$_{MCC}$) are modified versions of the standard
RTS/CTS packets, from which $i$ derives the IDs, the thresholds, and the NAVs relative to $k_{nl}$ and $d(k_{nl})$. The novel control frames also contain the preamble, which enables the antenna processing operations. Observe that, since in the MCC the legacy nodes are not present, the structure of the frames can be modified. If the sensed packet has $i$ as the destination ($d(k_{nl}) = i$) or involves the destination of $i$ ($k_{nl} = d(i)$ or $d(k_{nl}) = d(i)$), $i$ freezes its backoff counter. If not, $i$ updates the current threshold as:

$$L_{\text{cur}}^{i} = \min_{j' \in C \cup \{i, d(i)\}} L_{j'}, \quad (5.4.1)$$

where, using the notation introduced in Subsection 5.2.2, $C = \{p \in \mathcal{N}^{nl} | \text{NAV}_p > 0\}$ is the set of the non-legacy nodes currently involved in a communication in the MCC (having a NAV different from zero in the NCT$_i$). If $l_i \leq L_{\text{cur}}^{i}$, all nodes can sustain the current traffic load and hence the backoff counter can be decreased, since the potential access of $i$ and $d(i)$ would not destroy the current receptions and the communication between $i$ and $d(i)$ can be sustained. Otherwise, the backoff is frozen.

This monitoring operation continues until the backoff counter reaches the zero value and the source $i$ transmits the $\text{RTS}_{\text{MCC}}$ packet. When the destination $d(i)$ receives the $\text{RTS}_{\text{MCC}}$, it replies with the $\text{CTS}_{\text{MCC}}$ after a SIFS, and the rest of the handshake (DATA/ACK) is completed by using the receiving patterns synthesized using the preambles contained in the control packets.

The TAMPC protocol has the advantage of enabling MPC by relying on a unique parameter that can be analytically evaluated [38, 73], hence limiting the real-time calculations to DoA estimation and beamforming. Besides, since the access is based on an average quantity and is allowed only if the potential future communication does not destroy none of the currently active ones, the TAMPC protocol may be mainly oriented towards a conservative approach.

### 5.4.2 SIR access multi-packet communication protocol

With respect to the previous solution, the SAMPC protocol enables MPC by adopting the instantaneous SIR as an indicator. The instantaneous SIR is more adherent to the real network behavior and is able to account in detail and in real-time for the antenna pattern, the topology, and the traffic information. The novelty of the SAMPC protocol lies, on the one hand, on a local but accurate SIR estimation, and, on the other hand, on the introduction of Low Density Parity Check (LDPC) codes, already allowed in the 802.11n extension. In general, the use of a channel encoder is advantageous in an asynchronous multi-user system, where the maintenance of the SIR below a given threshold represents a rather difficult task, unless one accepts a really conservative approach and the resulting throughput reduction. In fact, the channel encoder, and the relative interleaver, enable to adopt a more aggressive approach by allowing one to accept occasionally collided slots within the correction
5.4. Proposed protocols

limits of the used code. In particular, the choice of LDPC codes has two main advantages. First, they are more efficient than the convolutional ones adopted in the 802.11 extensions. Second, the performance of LDPC codes can be reliably modeled adopting a threshold approach, since the block error probability of efficient codes, such as turbo and LDPC codes, is characterized by an ON/OFF behavior [74], which enables a fast and reliable estimation of the success/failure of a transmission attempt.

5.4.2.1 Operations in the CC: single communication

Once i and d(i) have been reciprocally recognized, i sends the DATA packet, which contains the antenna parameters $N_i$, $G_{a_i}$ (average gain), $G_{n_i}$ (average gain in a null), and a preamble in order to enable d(i) to estimate the relative DoA $\varphi_{d(i),i}$, and, subsequently, to synthesize the receiving pattern $G_{d(i)}(\varphi)$ using its beamforming algorithm [60]. Remembering that the transmission gain is equal to one (omnidirectional transmission), the power received by d(i) from i can be expressed using the Friis transmission equation as [38]:

$$P_{rx_{d(i),i}} = \frac{P_{tx} \cdot 1 \cdot G_{d(i)}[\varphi_{d(i),i}] \tilde{h}_\alpha}{r_{d(i),i}^a}, \quad (5.4.2)$$

where $P_{tx} = 100 \text{ mW}$ is the legacy transmission power, and $r_{d(i),i}$ is the distance between d(i) and i. Thus, the distance $r_{d(i),i}$ can be calculated by d(i) by inverting (5.4.2), obtaining:

$$r_{d(i),i} = \sqrt{\frac{P_{tx} G_{d(i)}[\varphi_{d(i),i}] \tilde{h}_\alpha}{P_{rx_{d(i),i}}}}. \quad (5.4.3)$$

Similarly, the ACK of d(i) includes $N_{d(i)}$, $G_{a_{d(i)}}$, $G_{n_{d(i)}}$, and the preamble, hence enabling i to estimate $\varphi_{i,d(i)}$ and $r_{i,d(i)}$. Observe that, since a unique omnidirectional transmission is allowed in the CC, also a non-legacy node $k_{nl} \neq i,d(i)$, sensing the DATA/ACK exchange between i and d(i), is able to acquire the parameters $N_i$, $G_{a_i}$, $G_{n_i}$, $N_{d(i)}$, $G_{a_{d(i)}}$, $G_{n_{d(i)}}$, and to estimate $\varphi_{k_{nl},i}$, $\varphi_{k_{nl},d(i)}$, $r_{k_{nl},i}$, and $r_{k_{nl},d(i)}$. Furthermore, since the positions of i, d(i), and $k_{nl}$ form a triangle, $k_{nl}$ can use these estimations to evaluate on its own the relative distance between i and d(i) as:

$$r_{i,d(i)} = \sqrt{r_{k_{nl},i}^2 + r_{k_{nl},d(i)}^2 - 2 r_{k_{nl},i} r_{k_{nl},d(i)} \cos[\varphi_{k_{nl},i} - \varphi_{k_{nl},d(i)}]}. \quad (5.4.4)$$

Besides, $k_{nl}$ stores the NAVs of i and d(i), NAV_i and NAV_{d(i)}. All these data are inserted by $k_{nl}$ into the NCT_{k_{nl}}, having, as in the TAMPC protocol, one entry for each sensed node $j'$, but now containing $N_{j'}$, $G_{a_{j'}}$, $G_{n_{j'}}$, NAV_{j'}, $\varphi_{i,j'}$, and all $r_{j',p'}$ values estimated using (5.4.4) for the nodes $p'$ that are already present in the NCT_{k_{nl}}.
5.4.2.2 Operations in the MCC: multiple communications

Similarly to the TAMPC protocol, when a transmission in the MCC is sensed, the generic node \( i \) estimates the current DoAs, controlling in the NCT, if one DoA corresponds to a NAV equal to zero. Such condition implies that a new communication between a source \( j' \) and a destination \( d(j') \) has started. Thus, \( i \) runs its beamforming algorithm to receive the RTS\(_{\text{MCC}}\)/CTS\(_{\text{MCC}}\) packets, which, differently from the TAMPC protocol, now contain \( N_j/d(j'), G^a_{j/d(j')}, G^b_{j/d(j')}, \) and the code rates \( R_{d(j')} \) adopted for the corresponding DATA and ACK transmissions, respectively. The antenna parameters are reinserted in the RTS\(_{\text{MCC}}\)/CTS\(_{\text{MCC}}\) to re-inform a node that, being involved in a communication in the MCC, has lost the first DATA/ACK exchange in the CC between \( j' \) and \( d(j') \). After the RTS\(_{\text{MCC}}\)/CTS\(_{\text{MCC}}\) reception, \( i \) first derives the instant of beginning of the corresponding DATA reception \( t_q \) and that of the relative ACK reception \( t'_q = t_d(j') + T_{\text{DATA}} + \text{SIFS} \). Secondly, for each of the two nodes \( q \in \{ j', d(j') \}, i \) stores the ordered set:

\[
\mathcal{F}^0_{i,q} = \{ p' \in \mathcal{N}^{|q|}, |\text{NAV}_{p'}| > 0 ; p' \neq q, d(q) ; P_{q,p'}^{rx} \leq \ldots \leq P_{q_1,p'}^{rx} \}
\]  \( (5.4.5) \)

of the estimated \( l_i \) currently active interferers of \( q \) in the \( q \)-th entry of the NCT, \( i \). The set \( \mathcal{F}^0_{i,q} \) in (5.4.5) is generated by \( i \) by taking the position of \( q \) as reference, since each \( P_{q,p'}^{rx} \) can be calculated from the \( r_{q,p'}^{rx} \) value obtained when sensing the CC. This set accounts for the fact that in an asynchronous scenario the interference configuration cannot be preserved \( [90] \), thus, if \( i \) intends to perform a reliable SIR estimation for a node \( q \), the initial interference configuration experienced by \( q \) must be stored. Since adaptive arrays aim to maximize the SIR \( [60] \), \( i \) can reasonably assume that, for a node \( q \), the \( N_q - 2 \) available nulls are used to suppress the strongest interferers in \( \mathcal{F}^0_{i,q} \) while the other interferers are received with the average gain \( G^a_q \) \( [38] \). Thus, \( \mathcal{F}^0_{i,q} \) can be partitioned in two subsets: \( \mathcal{F}^0_{i,q} \) containing the \( N_q - 2 \) suppressed interferers, and \( \mathcal{F}^0_{i,q} \) containing the not suppressed ones.

The stored information can be used by a non-legacy source \( i \), having a packet for a recognized non-legacy destination \( d(i) \), to establish the decrease or not of the backoff counter according to the SIR estimated not only for the pair \( i - d(i) \), but also for all pairs active in the MCC. To this aim, \( i \) considers the set \( \mathcal{C} = \{ p' \in \mathcal{N}^{|q|}, |\text{NAV}_{p'}| > 0 \} \) containing the nodes currently involved in a communication in the MCC. For each \( q \in \mathcal{C}, i \) evaluates the set \( \mathcal{F}^n_{i,q} = \mathcal{C} \cap \mathcal{F}^0_{i,q} \) containing the still active interferers suppressed by \( q \), and the set \( \mathcal{F}^a_{i,q} = (\mathcal{C} - \mathcal{F}^0_{i,q}) \cup \{ i, d(i) \} \), containing the still present not suppressed interferers of \( q \) together with \( i \) and \( d(i) \). For \( q \in \{ i, d(i) \}, i \) just partitions \( \mathcal{C} \) in the subsets \( \mathcal{F}^n_{i,q} \), containing the \( N_q - 2 \) suppressible interferers, and \( \mathcal{F}^a_{i,q} \) containing the not suppressible ones. Besides, for each \( q \in \mathcal{C} \cup \{ i, d(i) \}, i \) defines the function:
5.4. Proposed protocols

\[ f_{i,q}(t) = \begin{cases} 1 & t_q \leq t \leq t_q + T_{ACK}, q \in \mathcal{S} \cup \{i\} \text{or} t_q \leq t \leq t_q + T_{DATA}, q \in \mathcal{D} \cup \{d(i)\} \\ 0 & \text{elsewhere} \end{cases}, \]

(5.4.6)

where \( \mathcal{S} \subset \mathcal{C} \) is the subset of the active sources, and \( \mathcal{D} \subset \mathcal{C} \) is the subset of the active destinations. The function in (5.4.6) describes the activity of the node \( q \) as a function of the time \( t \). For \( q \in \mathcal{C} \), \( t_q \) is stored in the \( q \)-th entry of the NCT \( \mathcal{N}_i \), while, for \( q \in \{i, d(i)\} \), \( t_q \) is the current time. Moreover, by setting a counter for each \( q \in \mathcal{N}_{nl} \), \( i \) can estimate the probability of activity \( \eta_{i,q} \) as the ratio between the number of packets sent by \( q \) and the overall packets sensed in the MCC. Now, \( i \) can estimate the time evolution of the interference experienced by each node \( q \in \mathcal{C} \cup \{i, d(i)\} \) as:

\[ I_{i,q}(t) = G_q^a \sum_{p' \in \mathcal{F}_{i,q}^p} f_{i,p'}(t) P_{q,p'}^{RX} + G_q^a \sum_{p' \in \mathcal{F}_{i,q}^a} f_{i,p'}(t) P_{q,p'}^{RX} + G_q^a \sum_{p' \in \mathcal{N}_{nl} - \{q\}} \eta_{i,p'} P_{q,p'}^{RX}. \]

(5.4.7)

The first two terms in (5.4.7) are deterministic and account for the time evolution of the surely active interferers, while, the third term estimates the probability that each non-legacy node becomes an interferer. Observe that the set \( \mathcal{N}_{nl} - \{q\} \) in the third term includes also the currently active nodes, because a node that has completed its transmission may begin a further transmission. From (5.4.7), the SIR for the node \( q \) can be directly obtained as:

\[ \text{SIR}_{i,q}(t) = \frac{1}{I_{i,q}(t)} \begin{cases} P_{q,d(q)}^{RX} & q \in \mathcal{S} \cup \{i\} \\ P_{s'(q),q}^{RX} & q \in \mathcal{D} \cup \{d(i)\} \end{cases}, \]

(5.4.8)

where \( s'(q) \) denotes the source of the packet having \( q \) as its destination.

Once all SIRs are available, a reliable estimation for the result of the corresponding reception can be performed using the sustainable rate [74], which is suitable to model the behavior of the adopted LDPC codes. This criterion considers the instantaneous SIR and, according to the modulation, provides a rate function \( R_{s,i,q}(t) \), estimated using the sphere packing bound, whose average value is compared to the selected code rate \( R_q \) [54, 73, 74]. For example, considering the QPSK modulation adopted in the 802.11 PHY layer extensions [1, 5], a conservative value of \( R_{s,i,q}(t) \) can be obtained as [73]:

\[ R_{s,i,q}^s(t) = 1 - \log_2 \left[ 1 + e^{-\text{SIR}_{i,q}(t)/2} \right], \]

(5.4.9)

from which one can obtain the sustainable rate as:

\[ \mathcal{R}_{s,i,q}^s = \begin{cases} \frac{1}{T_{ACK}} \int_0^{T_{ACK}} R_{s,i,q}^s(t) dt & q \in \mathcal{S} \cup \{i\} \\ \frac{1}{T_{DATA}} \int_0^{T_{DATA}} R_{s,i,q}^s(t) dt & q \in \mathcal{D} \cup \{d(i)\} \end{cases}. \]

(5.4.10)
The reception of a packet may be assumed successful if the adopted code rate $R_q$ satisfies $R_q \leq \bar{R}_{i,q}$ \cite{74}. Thus, defining as $\mathcal{R}_{rates}$ the set of the selectable code rates, the condition:

$$\exists R_i, R_{d(i)} \in \mathcal{R}_{rates} \mid R_i \leq \bar{R}_{i,i}^{s}, R_{d(i)} \leq \bar{R}_{i,d(i)}^{s}, \text{ and } R_q \leq \bar{R}_{i,q}^{s}, q \in C, \quad (5.4.11)$$

may be used by $i$ as the criterion for decreasing its backoff counter. This condition requires that, if the potential communication between $i$ and $d(i)$ is feasible and becomes active, this communication and the currently active ones are all successful. Observe that the use of (5.4.10) and (5.4.11) implies that the SAMPC protocol considers the potential effect of the concurrent communications not only on the DATA receptions, but also on the ACK receptions, which are often neglected but can have a considerable impact on the result of the transmission attempt.

Summarizing, the SAMPC protocol adopts an advanced collision avoidance policy based on an accurate estimation of the instantaneous SIR and of the code behavior, in order to enable each node to evaluate the effect of the interference on its potential communication and on all the active ones, taking into account both the DATA and the ACK receptions. Similarly to the TAMPC scheme, the SAMPC algorithm acquires the information on the current medium occupation, thus matching the first condition stated in \cite{90} for exploiting adaptive arrays. Additionally, the SAMPC protocol estimates the evolution of the interference, in order to compensate for the impossibility of preserving the interference configuration in a distributed scenario \cite{90}.

Observe that both proposed protocols do not involve centralized mechanisms and, differently from the 802.11n extension \cite{5}, which provides just a single high data rate communication at a time, the TAMPC and SAMPC protocols enable simultaneous communications between multiple pairs. Moreover, in both cases, the backward compatibility with the 802.11 standard is maintained, since the RTS/CTS packets in the CC are not modified, the More Data field is not read by the legacy nodes during non centralized operations, the DATA/ACK packets containing novel fields in the CC are not received by the legacy nodes that turn off their radios for the NAV duration, and all modifications in the MCC are transparent to the legacy nodes, whose activity is limited to the CC.

### 5.5 IT++-based discrete-time simulator

This section presents the novel network simulation tool used to evaluate the performance of the proposed protocols. The simulator has been written in the C++ programming language using IT++ libraries to implement smart antenna systems at the PHY layer. The adopted smart antenna system model is based on the description given in Section 4.3. The new simulator is developed in order to increase the time resolution of the simulations. In fact, while the ns2 simulation is a discrete-time event-
driven simulation tool, the novel simulator has a time resolution equal to a single slot. For this reason, the evolution of the state is updated simultaneously for every network node at the beginning of the slot. Such behavior avoids updating the state during the slot duration and leads to an increased level of adherence to the mathematical model adopted from the theoretical study in [73]. Conversely, in event-driven simulation tools, such as ns2, the system evolves at the time in which an event occurs. In particular, in the ns2 simulator the backoff counter concerns slots, however, not all the events coincides with a multiple of the slot time. This behavior leads to a not proper system evolution, because two successive events, which theoretically belong to the same slot, are in some cases evaluated separately. Hence, a dependency between events is introduced, which may lead to incorrect results.

The novel simulation tool is basically a state machine with the minimal time unit equal to the slot time. The evolution of the network under investigation is performed at the beginning of the slot and, after that, the network state remains unchanged. The slot time is taken as reference, because there is a direct relationship with the exponential backoff counter of the CSMA/CA. At every slot time the backoff counter of each node is decreased. When it reaches the zero value, the transmission starts at the beginning of the next slot. The SIFS period typically has a duration which is less than a slot time, hence it is concerned into the transmission time.

Since, as presented in Section 4.3, the scripts which implement the PHY layer of the nodes are written in MATLAB, they were completely rewritten in the C++ programming language to be included into the novel simulation platform. The complex matrix calculations required by the spatial and temporal reference algorithms for the direction of arrival estimation and the beamforming operations are performed using IT++ libraries. In particular, the power gain pattern and the equivalent pattern are calculated at the beginning of the packet reception and they hold for the entire packet reception time.

The simulator parses a configuration file to acquire information on the network topology, on the characteristics of the nodes (PHY/MAC layers) and on the simulated channel. The network topology is defined specifying the coordinates of every node. Furthermore, for every node the configuration file specifies the number of elements of the antenna array, the beamforming algorithm, and additional MAC layer options. The simulator is able to account for the wireless channel, which can be affected by multipath and fading, activating appropriate options in the configuration file. The multipath statistics is directly considered in the implementation of the smart antenna system, while the block fading is introduced multiplying the value of received powers with a random number between 0 and 1. The block fading assumption means that the fading factor remains constant for the entire packet reception period. The values of the received power is derived by the use of the transmitted power, of the gain of the receiving antenna, of the relative positions between the nodes and of the
path-loss coefficient. Mainly, the two novel protocols proposed in Section 5.4 are based on the SIR and sustainable code rate estimations. The SIR is estimated considering the available information on the ongoing transmission. The sustainable code rate derived from the sphere packing bound theory is estimated associating the sequence of estimated SIRs to a sequence of equivalent rates. This sequence weighted on the whole packet reception time provides the sustainable code rate.

Summarizing, the developed simulation platform adheres to the theoretical assumptions used to develop analysis of the 802.11 MPC scenario proposed in [73], which will be used as a term of comparison of the presented results. In addition, the objective oriented structure of the developed IT++ simulation platform is suitable for the implementation of further improvements.

5.6 Computational analysis

Since the proposed protocols involve more sophisticated MAC/PHY procedures than the conventional single-packet 802.11, an estimation of the number of operations required by the TAMPC and the SAMPC schemes can be useful for clarifying some implementation aspects. To simplify the analysis, assume that just one floating point operation (flop), namely an addition, a multiplication, or a comparison, can be performed in a clock cycle. Thus, since the CPU mounted on a current wireless router has a clock frequency typically ranging from 200 MHz up to more than 1 GHz [97], one can estimate, pessimistically, that 1 flop requires

\[ T_{\text{flop}} = \frac{1}{(200 \cdot 10^6)} = 5 \text{ns}. \]

In terms of computational burden, the antenna processing algorithms represent the first element that must be taken into account. Two widely used methods are the MUSIC algorithm, for the estimation of the DoAs, and the cLMS algorithm, for the synthesis of the array radiation pattern (refer to Section 1.3.2.2.2). One of the main reasons for the adoption of these techniques lies in the low number of operations with respect to other existing solutions [60]. For such reasons, the MUSIC and the LMS algorithms are adopted for the antenna processing operations of both the proposed protocols. In particular, considering an antenna system of \( N_i \) elements and using \( K \) samples to discretize the azimuth domain, the number of flops required by the MUSIC algorithm to estimate \( l_i \) DoAs is \( l_i N_i^2 + (N_i - l_i)(N_i + 1)K \) [98], while the number of flops required by the LMS algorithm to evaluate the array excitations in \( K' \) iterations can be reduced to \( 2(N_i + 1)K' \) [99]. Therefore, the time required to complete the antenna processing operations can be estimated as:

\[ t_{\text{PHY}} = \left[ l_i N_i^2 + (N_i + 1)(N_i - l_i)K + 2K' \right] \cdot T_{\text{flop}}. \] (5.6.1)

While the previous estimation, involving just the PHY layer, is identical for both the TAMPC and SAMPC schemes, at the MAC layer the two access methods are characterized by different computational burdens. In particular, the TAMPC protocol re-
5.7. Results

quires just the evaluation of (5.4.1), which involves \( l_i + 2 \) comparisons, leading to a computational time equal to:

\[
t_{\text{MAC}}^{\text{TAMPC}} = (l_i + 2) \cdot T_{\text{flop}}. 
\] (5.6.2)

The SAMPC protocol, instead, requires more calculations. For a fixed node \( q \) and a fixed slot time, the first two terms in (5.4.7) require \( 2(l_i + 1) \) multiplications and \( l_i + 1 \) summations, while the third term implies just a single summation, since, being this term not dependent on the time, it does not involve real-time estimations and can be derived offline. The calculation of each sample of the SIR in (5.4.8) involves a division, which typically requires 4 flops. Thus, the evaluation of the SIR in a slot requires \( 2(l_i + 1) + (l_i + 1) + 1 + 4 = 3l_i + 8 \) flops. To reduce the computational complexity due to the evaluation of the rate function in (5.4.9) from the SIR, both the rate and the SIR can be quantized using \( b_{\text{quant}} \) bits. Hence, adopting a bisection method to associate the calculated SIR to the corresponding rate, the number of operations becomes \( 3l_i + 8 + b_{\text{quant}} \) for each node \( q \) in a slot. Since this estimation must be performed for all slots covered by a packet, the number of flops required to estimate the rate function of each node \( q \) is \( (3l_i + 8 + b_{\text{quant}}) \cdot T_q \), where:

\[
T_q = \begin{cases} 
T_{\text{ACK}}/\sigma_{\text{slot}} & \text{if } q \text{ is a source} \\
T_{\text{DATA}}/\sigma_{\text{slot}} & \text{if } q \text{ is a destination}
\end{cases} 
\] (5.6.3)

with \([\cdot]\) representing the ceiling function. The evaluation in (5.4.10) of the mean value of the rate function requires \( T_q \) summations and a division, while the comparison in (5.4.11) requires one flop. This leads to \( (3l_i + 8 + b_{\text{quant}}) \cdot T_q + (T_q + 4) + 1 = (3l_i + 9 + b_{\text{quant}}) \cdot T_q + 5 \) flops for estimating and comparing the sustainable rate of each node \( q \) with the selected coding rate. Considering the \( l_i \) active sources, the \( l_i \) active destinations, the sensing source, and the destination of the sensing source, one can infer that \( 2l_i + 2 \) nodes are involved \( (l_i + 1 \) concerning a DATA packet and \( l_i + 1 \) concerning an ACK packet). Thus, using (5.6.3), the computational time required by the SAMPC protocol for estimating the result of all \( 2l_i + 2 \) receptions can be evaluated as:

\[
t_{\text{MAC}}^{\text{TAMPC}} = (l_i + 1) \left( (3l_i + 9 + b_{\text{quant}}) \cdot \left( \left\lfloor \frac{T_{\text{DATA}}}{\sigma_{\text{slot}}} \right\rfloor + \left\lfloor \frac{T_{\text{ACK}}}{\sigma_{\text{slot}}} \right\rfloor \right) + 5 \cdot 2 \right) \cdot T_{\text{flop}}. 
\] (5.6.4)

This expression, together with (5.6.1) and (5.6.2), will be used in next section to quantify the computational time necessary to manage the backoff decrease process.

5.7 Results

As described in Section 5.5, the proposed MAC protocols have been implemented in a C++ simulation platform developed adopting the IT++ libraries for signal
processing and communications [100]. The selected algorithms and parameters are reported in Table 5.1, where the payload is assumed geometrically distributed with an average length equal to 870 bytes (6960 bits) [21]. For each simulation, 60 seconds of network activity are monitored. To simplify the comparison between the two presented schemes, a fixed code rate $R_i (i \in \mathcal{N})$ equal to 2/3 is assumed for the ACK packets in all the simulations, while the value of the code rate $R_{d(i)} (i \in \mathcal{N})$ for the DATA packets, which is still maintained fixed, will be specified for each examined scenario. The LDPC codes are adopted also for the TAMPC protocol, where, however, they simply replace the usually implemented convolutional codes, but are not involved in the backoff counter process as in the SAMPC protocol. Besides, the threshold adopted by the generic non-legacy node $i$ for the TAMPC scheme is assumed to be equal to the number of degrees of freedom of its antenna system, thus $L_{ti} = N_i - 1$ for $i \in \mathcal{N}^{nl}$ [21]. The adopted performance figures are the throughput and the Jain’s fairness index. In particular, the throughput is expressed in terms of the average number of packets correctly received in a slot time at the net of the adopted code rate. Using this definition, the sum of the throughput of all network nodes directly provides the number of simultaneous communications that can be hosted in a slot.

The presented results are subdivided in two parts. The first part reports the per-

<table>
<thead>
<tr>
<th>Common parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{slot}$</td>
</tr>
<tr>
<td>DIFS</td>
</tr>
<tr>
<td>SIFS</td>
</tr>
<tr>
<td>Maximum backoff stage</td>
</tr>
<tr>
<td>Retry limit</td>
</tr>
<tr>
<td>RTS</td>
</tr>
<tr>
<td>CTS</td>
</tr>
<tr>
<td>DATA header</td>
</tr>
<tr>
<td>Average payload</td>
</tr>
<tr>
<td>ACK</td>
</tr>
<tr>
<td>$\mathcal{R}_{c}$</td>
</tr>
<tr>
<td>Modulation</td>
</tr>
<tr>
<td>Preamble</td>
</tr>
<tr>
<td>DoA estimation algorithm</td>
</tr>
<tr>
<td>Beamforming algorithm</td>
</tr>
<tr>
<td>Path-loss exponent $\alpha$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TAMPC parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_{t(i)}$</td>
</tr>
<tr>
<td>RTSMCC</td>
</tr>
<tr>
<td>CTSMCC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SAMPC parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{i/d(i)}$</td>
</tr>
<tr>
<td>$G_{i/d(i)}$</td>
</tr>
<tr>
<td>$\mathcal{R}_{d(i)}$</td>
</tr>
<tr>
<td>RTSMCC</td>
</tr>
<tr>
<td>CTSMCC</td>
</tr>
</tbody>
</table>

Table 5.1: Adopted parameters and algorithms.
5.7. Results

The first set of simulations is obtained considering a minimum contention window equal to 32 and a code rate $R_d(i) = 8/9 \ (i \in \mathcal{N})$ in presence of a Poisson packet arrival process of mean $\mu$. The adopted topologies with the corresponding $N_i$ values are shown in Figs. 5.1a and 5.1b. In particular, the ring topology in Fig. 5.1a consists of two legacy pairs and three non-legacy pairs, while the random one in Fig. 5.1b consists of four legacy pairs and six non-legacy pairs. The number of antennas of the pair 9-10 in the ring topology and of the pair 15-16 in the random topology are not specified, since they will be used to put into evidence the different performance of the two developed protocols.

Figs. 5.2 and 5.3 report the aggregate throughput, given by the sum of the throughput in the CC and in the MCC. In particular, Fig. 5.2 refers to the topology in Fig. 5.1a considering the cases $N_{9,10} = 3$ and $N_{9,10} = 2$, while Fig. 5.3 refers to the topology in Fig. 5.1b considering the cases $N_{15,16} = 6$ and $N_{15,16} = 2$. To provide a reference, the figures also report the throughput obtained using the 802.11n super-g extension [49], where the adoption of two frequency channels at 12 Mbits/s is assumed, thus en-
Enabling a fair comparison with the proposed schemes. The curves show that both the TAMPC and SAMPC protocols provide a higher performance with respect to the super-g extension, allowing the coexistence of a number of simultaneous communications ranging from two to three for all the considered scenarios. With reference to Fig. 5.2, a direct comparison between the two developed schemes reveals that, when $N_{9,10} = 3$, their performance is substantially identical, while, when $N_{9,10} = 2$, the SAMPC scheme provides a higher throughput with respect to the TAMPC protocol. The behavior of the two solutions can be explained in detail considering the single-node saturation throughput reported in Table 5.2.

In the case $N_{9,10} = 3$, the minimum threshold among all nodes operating in the MCC is $L_i = N_i - 1 = 3 - 1 = 2(i \in \mathcal{N}^{nl})$, and, as described in Subsection 5.4.1.2, using the TAMPC protocol the backoff counter is decreased when two conditions are satisfied: the number of estimated active sources $l_i$ is lower or equal to $L_t = L_t = 2$ and $l_i \leq N_i - 1 = 2$ for $i \in \mathcal{N}^{nl}$. This implies that when two communications are active, a further one can be established and hence three transmissions may be simultaneously performed in the MCC. Considering also the CC, a total of four concurrent transmissions may be sustained by the network. Observe that the adoption of a large minimum contention window, which has been selected equal to 32, and the presence
of the channel encoder, determine throughput values lower than four, as it can be shown in Fig. 5.2 for \( N_{9,10} = 3 \). However, independently of these aspects, in the case \( N_{9,10} = 3 \) the proposed protocols guarantee the same throughput and a fair access to the network nodes.

Instead, in the case \( N_{9,10} = 2 \), the SAMPC protocol provides a higher throughput, while the TAMPC scheme maintains a better fairness. This result can be explained analyzing the access of the nodes operating in the MCC, namely the pairs 5-6, 7-8, and 9-10, for the two developed schemes. For \( N_{9,10} = 2 \), the thresholds become \( L_{t_{5-8}} = 2 \) and \( L_{t_{9-10}} = 1 \). Adopting both the TAMPC and SAMPC protocols, when the pairs 5-6 and 7-8 (having three antennas) are active, the source 10 (having just two antennas) freezes its backoff counter because it cannot estimate more than one active transmitter. A different behavior appears for the two schemes in another configuration of transmitters. Adopting the TAMPC protocol, when one of the two pairs with three antennas (5-6 or 7-8) and the unique pair with two antennas (9-10) are active, the source of the other pair with three antennas (8 or 6) freezes its backoff counter to protect the communication of the pair 9-10, thus satisfying the criterion in (5.4.1) concerning the minimum threshold. This behavior implies a fair reduction of the single-node throughput. Conversely, adopting the SAMPC protocol in the same con-
Chapter 5. Multi-packet communication for distributed wireless networks using advanced antenna systems

Table 5.2: Single-node saturation throughput (in packets/slot) for the topologies in Fig. 5.1.

<table>
<thead>
<tr>
<th>Pair</th>
<th>TAMPC</th>
<th>SAMPC</th>
<th>TAMPC</th>
<th>SAMPC</th>
<th>TAMPC</th>
<th>SAMPC</th>
<th>TAMPC</th>
<th>SAMPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>0.38</td>
<td>0.38</td>
<td>0.38</td>
<td>0.38</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
</tr>
<tr>
<td>3-4</td>
<td>0.38</td>
<td>0.38</td>
<td>0.38</td>
<td>0.38</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>5-6</td>
<td>0.62</td>
<td>0.62</td>
<td>0.30</td>
<td>0.62</td>
<td>0.19</td>
<td>0.19</td>
<td>0.19</td>
<td>0.19</td>
</tr>
<tr>
<td>7-8</td>
<td>0.62</td>
<td>0.62</td>
<td>0.49</td>
<td>0.62</td>
<td>0.21</td>
<td>0.21</td>
<td>0.21</td>
<td>0.21</td>
</tr>
<tr>
<td>9-10</td>
<td>0.60</td>
<td>0.61</td>
<td>0.47</td>
<td>0.39</td>
<td>0.60</td>
<td>0.54</td>
<td>0.36</td>
<td>0.58</td>
</tr>
<tr>
<td>11-12</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
<td>0.06</td>
<td>0.02</td>
<td>0.05</td>
</tr>
<tr>
<td>13-14</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
<td>0.14</td>
<td>0.37</td>
<td>0.40</td>
</tr>
<tr>
<td>15-16</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.60</td>
<td>0.46</td>
<td>0.09</td>
<td>0.05</td>
</tr>
<tr>
<td>17-18</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.15</td>
<td>0.34</td>
<td>0.07</td>
<td>0.35</td>
</tr>
<tr>
<td>19-20</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.23</td>
<td>0.24</td>
<td>0.42</td>
<td>0.27</td>
</tr>
<tr>
<td>Fairness (CC)</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Fairness (MCC)</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.96</td>
<td>0.56</td>
<td>0.76</td>
<td>0.60</td>
<td>0.69</td>
</tr>
</tbody>
</table>

figuration of transmitters, the source 8 or 6 does not freeze its backoff counter, since it estimates that the communication involving the pair 9-10 can remain successful in presence of the other two communications. This implies that the pairs with three antennas acquire more resources, leading to a higher throughput at the expense of a slight fairness reduction.

The curves in Fig. 5.3 confirm the higher performance of the proposed protocols with respect to the 802.11 super-g extension for the random topology in Fig. 5.1b. Besides, the results in Table 5.2 show that in this case the SAMPC scheme outperforms the TAMPC one for the two considered scenarios ($N_{15,16} = 6$ and $N_{15,16} = 2$) both in terms of throughput and fairness. In both proposed protocols each node tries to protect not only its own communication but also those of the other active nodes. The TAMPC protocol reaches this objective substantially accounting for the load threshold sustainable by the active node with the less powerful antenna system, without considering the characteristics of the topology. The SAMPC protocol still adopts a protection mechanism, but, being based on a SIR estimation, this mechanism is able to account for a larger number of network elements, including not only the number of antennas, but also the antenna gains and the relative positions between the nodes. The values in Table 5.2 confirm that the adoption of this more reliable process of estimation of the network conditions provides considerable benefits to the global performance of the network itself. This result suggests the usefulness of the SAMPC protocol for random topologies, which represent a more realistic scenario for the commonly deployed wireless networks.

To further investigate this issue, a third set of simulations is carried out by considering 50 random topologies, each having $N_s = 20$ non-legacy nodes, namely 10 pairs operating in the MCC once the process of recognition is completed. All nodes are equipped with an antenna array having $N_i = 5$ ($i \in \mathcal{A}^{nl} \equiv \mathcal{A}$) elements. Observe that,
while the previously simulated scenarios in Figs. 5.1a and 5.1b were heterogeneous both in terms on nodes and topologies, in this case the heterogeneity is just due to the asymmetry of the spatial distribution of the nodes, since all nodes have identical antenna systems. Figs. 5.4a and 5.4b report the average network throughput and fairness, respectively, in saturated traffic conditions as a function of the minimum contention window for different values of the code rate $R_{d(i)}$. Both figures confirm the higher throughput-fairness performance of the SAMPC protocol with respect to the TAMPC one for a given code rate. In particular, for the TAMPC protocol, the rate 3/4 leads to a higher throughput and a higher fairness, while, for the SAMPC protocol, the rate 8/9 leads to a higher throughput, but not to a higher fairness, which instead is reached using the rate 3/4. However, a direct comparison between the uncoded cases ($R_{d(i)} = 1$) and the coded ones reveals, on the one hand, that a certain channel coding is necessary to maintain an acceptable performance, but, on the other hand, that the required redundancy can be kept low, since even a high code rate, such as 8/9, can provide a satisfactory performance that is close to the one achievable using the lower rate of 3/4. In general, the results presented for heterogeneous scenarios show that the capability of an antenna system alone does not represent an absolute element for the single-node performance, which is also influenced by its position within the network. Therefore, the adoption of a rigid threshold on the number of communications for establishing the success or failure of the transmission attempts in an MPC (or MPR) scenario might sometimes result in a partial under utilization of the network resources.
As a final result concerning the protocols’ operations in heterogeneous networks, this subsection provides an estimation of the computational burden of each node for the three previously investigated sets of scenarios: the topology in Fig. 5.1a, the topology in Fig. 5.1b, and the 50 random topologies with $N_s = 20$ nodes. The estimation is performed using (5.6.1)-(5.6.4) that were derived in Section 5.6. In particular, this calculation can be performed remembering the conservative choice adopted in Section 5.6 for the time required by 1 flop ($T_{\text{flop}} = 5\text{ns}$), and considering for each scenario the node with the highest number of antennas, $N_{\text{max}}$, in presence of the maximum number of active sources that it can detect in the MCC: $l_{\text{max}} = N_{\text{max}} - 1$ [21].

The time required by the antenna processing operations at PHY layer can be evaluated assuming the adoption of $K = 360$ samples in the azimuth domain for estimating the DoAs and $K' = 512$ iterations for synthesizing the radiation pattern by the constrained LMS algorithm. Thus, from (5.6.1), one obtains $t_{\text{PHY}} \approx 34.8\mu\text{s}$ for the scenarios in Fig. 5.1a where $N_{\text{max}} = 4$, $t_{\text{PHY}} \approx 49.3\mu\text{s}$ for the scenarios in Fig. 5.1b where $N_{\text{max}} = 6$, and $t_{\text{PHY}} \approx 42.0\mu\text{s}$ for the 50 random topologies where $N_{\text{max}} = 5$. Since the preamble is transmitted at the control rate, $T_{\text{preamble}} = 128/(2 \cdot 10^6) \approx 64.0\mu\text{s}$ (Table 5.1), thus in both cases $t_{\text{PHY}} < T_{\text{preamble}}$ and hence the capabilities of the CPUs currently installed on the wireless routers are sufficient to sustain the computational requirements of the adopted antenna processing algorithms. Using the same data, the time required by the MAC layer operations when the TAMPC protocol is used can be directly calculated from (5.6.2), which provides $t_{\text{TAMPC MAC}} \approx 25.0\text{ns}$ for the scenarios in Fig. 5.1a, $t_{\text{TAMPC MAC}} \approx 35.0\text{ns}$ for the scenarios in Fig. 5.1b, and $t_{\text{TAMPC MAC}} \approx 30.0\text{ns}$ for the 50 random topologies. Instead, assuming a quantization with $b_{\text{quant}} = 8$ bit for the SIR and using the values in Table 5.1, the average computational time required by the SAMPC protocol can be evaluated from (5.6.4), thus obtaining $t_{\text{SAMPC MAC}} \approx 13.0\mu\text{s}$ for the scenarios in Fig. 5.1a, $t_{\text{SAMPC MAC}} \approx 26.7\mu\text{s}$ for the scenarios in Fig. 5.1b, and $t_{\text{SAMPC MAC}} \approx 19.3\mu\text{s}$ for the 50 random topologies. As expected, the SAMPC protocol requires a higher computational time with respect to the TAMPC scheme, due to the higher number of performed operations. However, the CPU time required by the SAMPC scheme can be considered not prohibitive since it remains in the order of tens of microseconds. To this purpose, it is also worth noticing that the presented estimation of the computational burden has been carried out adopting pessimistic assumptions for the hardware capabilities, since, usually, a CPU can perform more than 1 flop during a clock cycle and, furthermore, many wireless routers are equipped with CPUs having clock frequencies higher than 200 MHz [97]. This latter aspect can be considered even more true as the number of antennas of a router increases, since one can expect that more advanced devices are equipped with more powerful hardware.
5.7. Results

5.7.2 Comparisons

This second part of the results aims to discuss the protocols’ performance with respect to the analysis presented in [73] and, subsequently, to compare the achievable throughput with that of the 802.11n extension in a multipath environment. The simulations are carried out assuming saturated traffic conditions and maintaining the settings reported in Table 5.1. Since the model developed in [73] holds for a homogeneous network, the comparison between theoretical and numerical results is performed considering a symmetric topology in which \( N_s = 20 \) identical non-legacy nodes are placed on two concentric rings, where the sources lie in the outer ring, having a radius equal to 20 m, and the destinations lie in the inner ring, having a radius equal to 10 m (Fig. 5.5). Each node has \( N_i = 5 \) antennas and, for each DATA packet, a fixed code rate \( R_{d(i)} = \frac{8}{9} \) is used. Since only non-legacy nodes are involved, the CC is used just for the process of recognition and the rest of the traffic is delivered using the MCC.

Fig. 5.6 shows the network throughput as a function of the minimum contention window obtained using the TAMPC and the SAMPC protocols, the 802.11 super-g extension, and the analysis proposed in [73]. The fairness results are not reported since they are substantially equal to one in all cases, thus making not possible to distinguish between the different curves. This behavior reveals that when the proposed
Chapter 5. Multi-packet communication for distributed wireless networks using advanced antenna systems

![Network throughput in saturated conditions as a function of minimum contention window for the topology in Fig. 5.5.](image)

**Figure 5.6:** Network throughput in saturated conditions as a function of minimum contention window for the topology in Fig. 5.5.

---

schemes are applied to a symmetric scenario, even if not ideally homogeneous, they are able to guarantee fair access opportunities. Since the theory in [73] considers an access based on a threshold on the number of allowed communications, the comparison with the theory is more meaningful when it involves the TAMPC protocol. In particular, the analysis in [73] and the TAMPC scheme are compared for two values of the threshold: $L_t^i = L_t, L_t = 3(i \in \mathcal{N})$ and $L_t^i = L_t, L_t = 5(i \in \mathcal{N})$. One can immediately observe that the analytical and the numerical throughput of the TAMPC protocol are very close in a large part of the considered range of contention windows. More precisely, the theoretical and simulated curves become coincident for large values of the contention window, while significant differences appear for low values of the contention window, mainly for the case $L_t = 5$. These differences are due to the fact that the studied scenario, even if symmetric and composed by identical nodes, is not perfectly homogeneous. In fact, in the simulation, the result of a reception in presence of a certain number of simultaneous transmissions does not depend only on the number of transmitters itself, but also on which transmitters are active. Thus, different configurations of the same number of active sources can be present, some of which
can lead to unsuccessful receptions, while others can lead to successful receptions. The analysis, instead, assuming a homogeneous scenario, considers all possible configurations of a given number of transmitters as identical and hence leading to the same result.

A direct comparison between Fig. 5.4a and Fig. 5.6 shows that the performance improvement achievable by the proposed protocols with respect to the 802.11 super-g extension considerably increases in the symmetric scenario. The same comparison reveals that the concentric topology provides, for the TAMPC and the SAMPC protocols, a value of the average throughput that is almost doubled compared to the case where the nodes are located at random positions (observe the different scales of the ordinate axis of the two figures). This strong impact of the topology on the achievable performance becomes more evident noticing that the results in Fig. 5.4a and Fig. 5.6 refer only to different topologies, since both networks are characterized by the same number of identical nodes, the same single-node antenna characteristics, and the same code rate. These similarities imply also that the computational time required by each node in Fig. 5.5 to complete its access operations coincides with that derived for the random topologies with $N_s = 20$ nodes.

As a final result, the performance of the TAMPC and SAMPC schemes is compared to that of the 802.11n amendment using a Vertical Bell Laboratories Layered Space-Time (VBLAST) aided PHY layer for achieving an increased data rate through spatial multiplexing [5]. To this purpose, it is worth noticing that the presented protocols and the 802.11n extension differ in two fundamental aspects. First, the 802.11n amendment allows a unique communication at a time, while the proposed protocols adopt an MPC approach. Secondly, the 802.11n extension and the presented schemes provide their respective maximum performance in two complementary propagation conditions. More precisely, the TAMPC and SAMPC schemes are designed to operate and achieve their best performance in a low-rank environment, where the angular spread of each signal can be considered small, in order to exploit the interference suppression capabilities of smart antenna systems. Conversely, the 802.11n extension is designed to operate in a high-rank environment, where the different paths of the signal can be considered independent, in order to exploit the MIMO channel by spatial multiplexing [101]. This implies that the angular spread $\hat{\sigma}_\varphi$ of the channel has a fundamental role in determining the performance of the different schemes, since it affects the statistic of the DoA for the TAMPC and SAMPC schemes and, being related to the spatial correlation between the signal replicas, it also affects the performance of the VBLAST algorithm. In particular, for the TAMPC and SAMPC protocols, $\hat{\sigma}_\varphi$ is simulated using a Laplacian probability density function for the statistic of the DoA [101]. For the VBLAST aided 802.11n scheme, the model of [102], which provides a relationship between angular spread and spatial correlation, is used to correlate the signals arriving from the transmitting antennas.
at a particular receiving antenna. Signals that arrive at different receive antennas are assumed to be uncorrelated, hence the performance of the simulated 802.11n scheme is optimistic. Furthermore, since the TAMPC and SAMPC schemes operate on two channels (CC and MCC) and use LDPC codes, the 802.11n nodes are fairly allowed adopting the dual-band option at 40 MHz and using LDPC codes [5]. The performance of this 802.11n PHY layer is simulated by transmitting a packet, operating a soft decoding of the received symbols using the Minimum Mean Square Error (MMSE)-based successive interference cancellation algorithm, and then measuring the mutual information of the sequence of Log-Likelihood Ratios (LLRs). The packet is considered to be successfully received if the mutual information of the sequence of LLRs is higher than the adopted coding rate [103].

Fig. 5.7 reports the result of the comparison between the three schemes for the topology in Fig. 5.5 using a minimum contention window equal to 32. The presented
curves are referred to three significant values of the angular spread (the case $\hat{\sigma}_\phi = 10^\circ$ is not shown to improve the readability of the figure). Observe that, differently from the previous figures, where the throughput was reported in packets/slot in order to immediately identify the number of simultaneously active links, in this case the throughput is reported in Mbits/s in order to reliably compare the 802.11n protocol, characterized by a single communication at a high data rate, with the proposed MPC schemes, characterized by many communications at a lower data rate. Fig. 5.5 shows that, in a low-rank environment, the TAMPC and SAMPC protocols maintain an acceptable performance even in presence of high values of the angular spread, such as $\hat{\sigma}_\phi = 20^\circ$ and $\hat{\sigma}_\phi = 30^\circ$, which can be considered realistic for many outdoor scenarios [28]. As expected, the throughput of the VBLAST aided 802.11n scheme becomes significant when the angular spread is sufficiently large to guarantee a reduction of the spatial correlation between the signal replicas. Thus, the TAMPC and SAMPC schemes are able to operate in propagation environments complementary to those usually necessary for MIMO operations (spatial multiplexing and diversity), and, furthermore, the two proposed solutions provide an acceptable throughput when the spatial channel conditions move towards high rank characteristics.

### 5.7.3 Results obtained using MATLAB-ns2 simulator

This section reports an alternative set of results, which has been obtained using the MATLAB-ns2 simulator presented in Chapter 4. The results obtained with the MATLAB-ns2 simulation tools are compared to those obtained by the C++/IT++ simulator. The aim is to investigate if the two simulation platforms lead to comparable results. The comparison is performed on the topology in Fig. 5.5, which consists of $N_s = 20$ non-legacy nodes. The nodes are equipped with identical antenna systems with $N_i = 5$ radiating elements. The nodes adopt a fixed code rate $R_d(i) = 8/9$. The throughput is obtained in a channel affected by multipath having a Laplacian distribution. Three different values of the angular spread are considered, i.e. $\hat{\sigma}_\phi = 1^\circ$, $20^\circ$ and $30^\circ$. The MATLAB-ns2 simulator presents a higher throughput for all the considered angular spreads, i.e. approximately 5 Mbits/s for all the cases. MATLAB-ns2 simulation tool seems to be less restrictive, allowing a higher individual throughput and/or allowing approximately one simultaneous communication more than the SAMPC protocol implemented on the C++/IT++ simulator. To this purpose it is worth noticing that the ns2 state machine, which is responsible for the MAC/PHY layers evolution during the simulation, can be affected by the presence of possible mechanisms that may cause a not completely reliable behavior. These mechanisms may be due to possible absence of simultaneity of the events occurring in the same slot, such as the collisions, leading to an overestimation of the success probability. However, one can also observe that two sets of curves, namely those obtained from ns2 and those derived using IT++, have a similar slope, since region of transition of
the throughput is the same for both simulators.

5.8 Conclusions

The design requirements for enabling multi-packet communication in an IEEE 802.11 network have been discussed, proposing two novel MAC protocols, called TAMPC and SAMPC, which exploit advanced antenna systems. The two presented solutions are suitable for asynchronous operations in distributed and heterogeneous scenarios, in which legacy and non-legacy nodes equipped with different antenna systems can coexist.

The results have proved that the developed schemes outperform the legacy 802.11 MAC layer using the super-g PHY layer extension. In particular, the SIR-based access adopted by the SAMPC scheme can guarantee higher throughput and fairness with respect to the threshold-based access adopted by the TAMPC protocol, at the cost of an increased, but anyway acceptable, computational burden. Besides, a com-
parison between the throughput of the TAMPC and that obtained from a theoretical analysis has shown that the selection of the threshold of the number of allowed communications has a significant impact on the final performance. Furthermore, the simulations have put into evidence that the adoption of a channel encoder with a coding rate not far from one can be sufficient to maintain a satisfactory performance in different MPC scenarios. In general, both presented protocols, which have the advantage of being backward compatible with the 802.11 standard, provide significant throughput values in a low-rank environment, where the MIMO approach adopted in the 802.11n extension can suffer from a performance downgrade due to the large spatial correlation between the signal replicas. The here discussed MPC concept may be also useful to provide a point of view complementary to the MPR approach adopted in the 802.11ac/ad multi-packet extensions that are currently under development.
This section presents a complete, real-time, software-defined radio implementation of an OFDM transceiver, compliant with the IEEE 802.11 physical layer specifications. All the baseband components in both the transmitter and the receiver are implemented with fast software functions, running on general purpose CPUs. A real-time operation is achieved on a modern CPU by means of extensive code optimization, mostly using the SIMD instruction set, which is widely available on almost every modern CPU.¹

6.1 Introduction and motivation

Wireless devices used in commercial off-the-shelf hardware feature ASIC tailored to the specific transmission system to execute baseband digital signal processing. This kind of integrated circuits provides sufficient computational resources and a low power consumption. On the other hand, the functionality implemented on this hardware is fixed and cannot be modified or extended in any way. In fact, it is hard, if not impossible, to gain access to the low-level primitives controlling the lower layers (PHY, MAC). Furthermore, simulation tools have sometimes limited features and do not properly adhere to a real-life environment.

In recent years the software-defined radio concept, which introduces fully programmable platforms, aims to exchange the ASIC approach [105, 106]. The goal of SDR is to implement most radio functions in software allowing radios to become

¹This chapter is based on the research studies conducted at the Telecommunications Research Center Vienna (FTW) in Vienna. The results of the research effort have been presented at the 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2012), Krakow, Poland, Sept. 17-21, 2012 [104].
more flexible. This seems especially interesting as nowadays many devices already support an abundance of wireless standards, but these standards are still implemented using dedicated hardware. Furthermore, new standards can be implemented by applying software updates to the radio hardware.

SDR uses general computational architectures such as FPGAs, DSPs, general purpose CPUs, or a blend of those, to perform baseband operations. Available SDR platforms can be roughly divided into two groups: the first consists of low-power stand-alone boards mainly based on DSPs and FPGAs (as used in [107–109]). However, nowadays the technology offers development kits comprising FPGAs and ARM processors with programmable logic onboard, whose performance is considerable [110]. The second group is based on simple radio front-end boards attached to commodity personal computers, therefore performing the actual computing on general purpose CPUs [111–113]. The use of programmable architectures for SDR purpose can be convenient mainly for several reasons. The FPGAs can be reconfigured updating algorithms or adding functionalities, while DSPs and general purpose CPUs can furthermore dynamically assign computational resources between receive and transmit algorithms or between multiple supported wireless standards that do not run concurrently. Since the computational resources available on modern CPUs and FPGAs have increased noticeably during the last decade, it is possible to use all-software implementations to decode high bandwidth protocols, such as Wideband Code Division Multiple Access (WCDMA), OFDM, and Orthogonal Frequency Division Multiple Access (OFDMA).

Recently, some academic development kits have been developed including Rice University's Wireless open-Access Research Platform (WARP) [112], and Microsoft Research's SOftware RAdio (SORA) [113]. The Microsoft SORA is a complete SDR system, including a PCI Express hardware interface and a complete software development kit. As a test example, a complete 802.11 transceiver including physical and MAC layers is provided [113]. The system uses a general-purpose multi-core machine to perform both transmission and reception tasks, using one core for the transmitter and up to two cores for the receiver. Real-time operation is achieved by extensive use of software optimizations and SIMD streaming instructions, while latency constraints are met by running the physical layer routines in kernel threads, which communicate with each other by means of custom inter-process communication primitives. While the performance of the overall system is quite impressive, since it can be used for a very wide range of scenarios requiring high bandwidth, there is no access to the source code of the physical layer library, which is provided as precompiled binary code.

A real-time implementation of an 802.11a receiver is presented in [114]. The authors focus on some of the relevant components in the receive chain. A thorough description and performance analysis are provided for all the implemented software
blocks. However, the proposed architecture does not present a complete receiver, as it only targets per-symbol processing and the related components, and does not cover frame synchronization, carrier offset compensation and channel estimation.

The GNURadio framework [111] provides a real-time, low bandwidth implementation of an OFDM transceiver. The system is highly configurable, as several parameters can be changed, such as the signal bandwidth and the number of active carriers. However, it cannot deal with the high computational requirements of the 802.11 standard.

The implementation of a 802.11 compliant transmitter using GNURadio is described in [115]. The system is built using a set of components written in Python and C++ but without significant performance optimizations, yet enough to guarantee a smooth transmission on reasonably powerful host machines. The goal of this chapter is to prove that an optimized implementation of the physical layer of the IEEE 802.11 standard on SDR solutions can achieve real-time performance. Furthermore, the optimized software will became part of open, accessible, high-performance library of software-defined radio for transmission and reception compliant with the 802.11 OFDM standard that can be interfaced to different RF frontends. This library of open-source components can be used for design and testing of new algorithms and protocols on-air. The implemented solution can reach real-time operation for all modulation schemes and bitrates provided by the standard on most modern Intel CPUs.
6.2 Implementation of the transceiver chain

6.2.1 Software optimizations

All the code presented in this work has been written in the C language. Mainly, two basic approaches have been used to design optimized signal processing code, namely SIMD parallelism and Look-Up Tables (LUTs). SIMD parallelism is a programming technique largely used in video and audio processing. As in digital signal processing it often happens that the same operation must be executed on different sets of data, a relevant increase in performance can be achieved by using parallel computational units. Most modern CPUs have special registers and instruction sets to deal with this kind of operations, such as SSE instructions on Intel and AMD processors, Altivec on PowerPC architectures, and NEON on ARM processors. SIMD parallelism is provided in the form of short vector instructions called streaming SIMD extensions (SSE) on x86 architectures, which are included in SSE libraries. More information on SIMD instructions is provided in Section 2.3.3.

The intrinsics interface enables the programmer to integrate instructions in C and C++ without the use of inline assembly code. The challenge is to minimize loads and stores from and to the memory, to limit the use of in-register vector shuffle operations and to avoid conditional statements. The used approach is almost equivalent to the use of inline assembly, which is normally used to write highly optimized code for a specific architecture to overcome the limitation of the compiler optimizations. The main advantage resides in the fact that the compiler can optimize the execution order of the functions according to the latency and throughput values of each instruction on different target architectures, while inline assembly must be carefully optimized by hand. For instance, most operations take only one clock cycle to execute on the newest i5/i7 processors, while this may not be true for less powerful architectures such as the Atom. Knowing the exact values for latency and throughput, the compiler can choose to reorder the instruction flow in order to minimize latencies and stalls on the target architecture, ensuring almost optimal performance. Currently, the developed software supports Intel and AMD processors featuring the SSE 4.2 instruction set, that is, all the 64-bit processors manufactured by Intel and AMD from 2007 can run the code without modifications. Look-up tables are used when it is more efficient to pre-calculate the values of a certain function rather than calculating it on-the-fly. As long as the data can fit into the higher layers of the CPU cache without accessing the main memory, the high bandwidth of the cache memory can be used for achieving a significant performance gain. When using this approach, the memory footprint should be kept as small as possible in order to avoid any cache miss during execution.
6.2. Implementation of the transceiver chain

16-bit integer interleaved samples
32-bit floating-point interleaved samples
8-bit symbol estimates
8-bit output data

Figure 6.2: The implemented blocks of the 802.11 receive chain.

6.2.2 Transmit chain

This section describes the implementation of the transmit chain of the proposed transceiver, giving the relevant implementation details for each block. The transmit chain is depicted in Fig. 6.1. The first stage after the transmit chain accepts 16-bit complex (I/Q) interleaved baseband sampled data at a sampling rate of 20 Msamples per second. All the details about each processing block can be found in Section 17 of the 802.11 Standard [23].

32-bit CRC Calculator. The 32-bits CRC (CRC32) is calculated and appended at the end of a frame to check the integrity of the received data. It is implemented with a LUT of 32-bit 256 elements in order to reduce the amount of required logical operations.

Bit Swapper and Scrambler. The Bit Swapper reverses the bit order which is inverted on Intel machines with respect to the transmit order. It is easily implemented with SSE logical operations with a 16-way parallelism, hence the bit are swapped on 16 octet in parallel. The swapped bitstream is fed to the Scrambler, which is implemented with LUTs and 128-way parallelism. For each possible initial state of the Scrambler, a complete set of output is precomputed, taking into account that the sequence has a period of 127 bits. A complete sequence of 192 bytes is precomputed for each state, so that the overall size of the LUT is 128 \times 192 bytes. The scrambling operation is then executed on a 128-bit burst with a single SSE exclusive-or operation. The performance penalty introduced by memory accesses is then balanced by the high level of parallelism.

Convolutional Encoder. The Convolutional Encoder is implemented using LUTs. For each input bit, it outputs two bits, which are then interleaved in the output stream. As the internal state and thus the output values of the Convolutional Encoder only depend on the input bit sequence, it is possible to precompute all the
possible outputs for each initial state of the encoder. The input data are processed in 8-bit portions. Then, the last 6 bits of the input are used as the initial state for the next burst of data. The overall size of the look-up table is $64 \times 256$ bytes.

**Puncturer.** The Puncturer uses 128-bit SSE words as shift registers, as the periodicity of the puncturing patterns can be used to reduce the number of logical operations.

**Interleaver.** The interleaving is performed in two steps to leverage the regular and periodic behavior of the permutation patterns more adequately. As in the Puncturer, logical and bitwise operations are used on SSE registers to improve performance. The input bits are then grouped in 16-bit words that can be easily converted into constellation mappings.

**Symbol Mapper.** The Symbol Mapper converts the interleaved bits into constellation mappings using mathematical and logical operations on the input bits with an 8-way parallelism. It also takes care of the pilot insertion and proper rescaling of the symbol, converting the data into single-precision floating point for the IFFT block.

**Inverse FFT and Cyclic Prefix Insertion.** All the FFT code used in the developed software is generated by the Spiral automatic code generator, as described in [116]. It is one of the fastest implementations available for the Intel machines. The code has been modified to include the cyclic prefix at the beginning of the symbol.

**Preamble Insertion.** The preamble is written at the beginning of the transmit buffer at startup without any performance impact at run-time. To avoid memory-to-memory copy operations the cyclic prefix is stored directly from the SSE registers.

### 6.2.3 Receive chain

The receive chain is much more critical than the transmit chain for real-time operation, as it consists of more blocks, some of which are very demanding from the computational resources point of view.

The received baseband stream has a bit resolution of 16 bits per real or imaginary samples. In order to process data in an efficient way the maximum level of parallelism that the CPU can offer is used. The incoming samples are processed at the original resolution and they are switched to the floating point format only when strictly necessary. The flow of the receive chain is depicted in Fig. 6.2. The last two blocks of the chain are identical to the first two of the transmit chain.

**Frame Detector.** The Frame Detector block implements a modified version of the Schmidl and Cox algorithm, which uses the short training sequence of the frame preamble to detect the reception of a frame [24]. The input is scaled to 12 bits in order to achieve full 8-way parallelism without arithmetical overflow. The auto-correlation over the short training sequence is normalized to the power and integrated over its duration. The integration reduces the uncertainty in the estimation of the frame position. In the implementation the interval when the integrated value of the autocorrel-
6.2. Implementation of the transceiver chain

The search of a local maximum values, taking into account also the width of the interval to reduce the probability of a false positive.

**Frequency Offset Estimator and Compensator.** The Frequency Offset Estimator is based on the Schmidl and Cox approach, described in [24], performing the autocorrelation on the long training sequence. It is a required step to evaluate the clock impairment between clocks at transmitter and receiver. A coarse frequency offset estimation is performed on the frame preamble before the symbol timing to improve performance, as the cross-correlation algorithms are sensitive to frequency errors. The Compensator uses an 8-way parallelism on 16-bit samples and outputs 32-bit floating point data.

**Symbol Timing.** The Symbol Timing block is based on a robust cross-correlation algorithm, thoroughly described in [25]. Since the cross-correlation is much more demanding in terms of computational complexity than the auto-correlation (48 complex multiplications per sample instead of just one), this does not impact performance dramatically, as the routine is called only when the previous stage detects a frame. As in the case of the Frame Detection block, rescaling is required in order to avoid overflow during the processing. The block uses a 48-tap matched filter with an 8-way parallelism, which requires the taps to be rescaled to a resolution of 9 bits. The frame position is then chosen within an observation window of 128 samples.

**FFT, Block Equalizer and Comb Equalizer.** This block removes the cyclic prefix at the beginning of the symbol and performs a 64-point FFT. As mentioned before, the FFT code used in this software is generated by the Spiral automatic code generator. The Zero Forcing Block Equalizer is merged in the FFT code, using a Least Squares estimation, as described in [26]. The estimate is obtained on the average of the two symbols transmitted in the long training sequence of the frame preamble. The Comb Equalizer is implemented with a first-order linear interpolation over the pilot carriers, which is a good tradeoff between performance and computational requirements. All code uses a 4-way parallelism, since it elaborates 4 floating point values (32 bit for each).

**Demapper.** The Demapper is implemented as a classical Log-Likelihood Ratio algorithm, with saturation. It is optimized with a 4-way parallelism, and outputs 8-bit soft estimates of the received symbols. A single SIMD optimized demapping function is used for all modulation schemes by leveraging the symmetries of the bit mappings.

**Deinterleaver.** The Deinterleaver is implemented with automatically-generated code. The indexes of the permutations are precomputed, and the memory-to-memory operations are performed by automatically-generated code for each modulation scheme, thus avoiding conditional statements in the code flow.

**Viterbi Decoder.** The Viterbi Decoder has been generated by the online Spiral Code Generator, described in [117]. The code is designed for 16-way parallelism,
Chapter 6. A software-defined radio implementation of an 802.11 OFDM physical layer transceiver

Figure 6.3: Per-symbol performance graphs for the transmit chain.

and it rescales the 6-bit path metrics every fourth Add-Compare-Select step to avoid overflows. The traceback unit is currently not optimized for vector operation.

### 6.3 Performance evaluation

The evaluation of the computational resources required to run the implemented transceiver chain is presented in this section. All tests were performed on a desktop machine featuring an Intel i5/660 CPU and running Linux 3.2.0. The code was compiled with GCC version 4.5.3. Assuming a bandwidth of 20 MHz and a sampling frequency of 20 Msamples per second, the sampling period for the baseband signal is 50 ns. The overall computational time required to decode an OFDM symbol in real-time must not exceed the length of the symbol itself, which is 4 μs for a 20 MHz channel. The execution times have been measured with the rdtsc machine instruction featured by the Intel CPUs. With this instruction it is possible to measure how many clock cycles have elapsed since the machine has been powered on by accessing a 64-bit special register called Time Stamp Counter. Synchronization barriers have been used to avoid timing measurements errors due to dynamic instruction rescheduling at both compiler and CPU levels. The clock of the machine has been set to 3.47 GHz and the execution time has been inferred from the number of elapsed cycles by taking the clock rate into account. The performance indicators have been averaged over a set of one million frames for both chains. The software has been run in a single-threaded userspace application with a normal execution priority.
6.3. Performance evaluation

6.3.1 Transmitter performance evaluation

The transmitter works on a strict per-symbol basis, therefore it is reasonable to assess the performance as such. In order to reduce latencies due to memory transfers, some blocks have been merged together, in order to operate directly on the data contained in the SSE registers. The Interleaver, Symbol Mapper and Pilots Insertion have been implemented within a single function, which outputs complex floating point data already split into real and imaginary parts, which is the optimal order for the IFFT block. The performance results are shown in Fig. 6.3. The proposed solution can reach real-time operation in the transmit chain with low overhead in all cases.

As shown in Fig. 6.3, the computational time per symbol for the 48 Mbit/s bitrate is slightly higher than the one required for the 54 Mbit/s bitrate. This is mainly due to the level of implementation achieved, which can be different for the different blocks of the chains and for the different bitrates. It is worth noticing that different bitrates imply different \( N_{\text{BPSC}} \), \( N_{\text{CBPS}} \), and \( N_{\text{DBPS}} \) as reported in the Table 1.3. In fact, the interleaving plus mapping operations are slightly longer for the 48 Mbit/s, compared to the 54 Mbit/s rate. Additionally, the inverse FFT plus cyclic prefix insertion is faster in the case of the 36 and 54 Mbit/s rate, and slower for other rates. Unfortunately, the last statement is not easy to investigate, since the inverse FFT is implemented using machine-generated software.
Chapter 6. A software-defined radio implementation of an 802.11 OFDM physical layer transceiver

6.3.2 Receiver performance evaluation

The Frame Detector must process the input datastream in a timely fashion in order not to cause buffer overflows. Assuming 16-bit I/Q samples and 20 MHz sampling frequency, the bandwidth of the input signal is 80 MBytes/s. The proposed implementation of the Frame Detector supports 650 MBytes/sec, so that it can be assumed that the real-time constraints are always met. The Symbol Timing execution is triggered only sporadically, when a frame start is signaled by the Frame Detector. Nonetheless, as a worst-case analysis, its performance has been measured in the same stream fashion as the previous block. Observing that it can sustain a maximum throughput of 135 MBytes/sec, one concludes that it easily meets real-time requirements.

The results in Fig. 6.4 show the performance of the blocks that process data on a per-symbol basis. With the presented software, real-time requirements can be met for all bitrates. In the 54 Mbit/s case, which is the most demanding from the computational point of view, a symbol can be decoded in 3.95 µs even without multithreading optimizations. The most demanding block from the computational point of view is the Viterbi Decoder, whose execution time is highly dependent on the bitrate, as higher modulation schemes carry more bits in each symbol. Other blocks, like the Frequency Offset Compensator and the Equalizer have a fixed overhead, as they operate on samples regardless of the information carried in the symbol.

6.4 Work in progress results

The final step of the implementation is still work in progress. The current implementation can produce a baseband representation of an IEEE 802.11 compliant data frame, which is ready to be fed to the up-conversion chain of an SDR hardware device. The native format is complex 16-bit interleaved integers, which could be easily adapted to floating point if required. In the same fashion, the input of the receive chain accepts input as complex 16-bit interleaved samples at native sampling rate, coming from the down-conversion chain.

One example use case would be to use the USRP N210 platform to capture and transmit RF signals, using a host computer to run the transmission and reception software (Section 2.3.2). To this aim, the next step would be to integrate the current implementation with the UHD middleware driver interface for communicating with USRP platforms.

Nevertheless, in order to assess the validity of the implemented solution, several tests involving COTS hardware have been carried out. The frames produced by the transmitter have been transmitted by a USRP using an intermediate file as storage and successfully received by a wireless adapter in monitor mode. All the available
modulation schemes and puncturing patterns, that is, all the available bitrates have been tested for frame sizes spanning from the minimum to the maximum size allowed by the standard. At the receiver side, the algorithms have been tested by using baseband traces produced by the transmitter, filtered by impulse responses generated by a channel modeler and with added additive white Gaussian noise. Moreover, real traces have been recorded on a 802.11 channel with the USRP on a datafile and processed by the receive chain, in order to test the behavior in presence of impairments of real hardware or in presence of interfering transmissions coming from other sources.

The current results will be validated by a more thorough set of measurements and performance assessments.

6.5 Conclusions and future work

In this work, the implementation of a complete software-defined radio based transceiver compliant with the IEEE 802.11 OFDM physical layer specifications has been presented. The transceiver can deal with real-time processing requirements on reasonably powerful machines. Future work will investigate the performance of the receiver on real-time streaming signals integrating a complete RF frontend. Further optimizations can be adopted in the future in order to introduce multi-threading management of the resources. For example in the reception process multi-threading can be used in order to split the chain in two main parts. The Frame detection, which is a very demanding operation from the computational resources point of view and it has to run permanently, can be performed by one thread. Hence, the remaining part of the chain can be performed on the second thread once the frame is detected. Furthermore, with the launch of new platforms which include on the same board CPUs and FPGAs, very relevant performance increases can be obtained. In fact, FPGAs are able to perform complex algebra computations faster than the GPP, achieving higher throughput.
The thesis has addressed the simulation, experimental analysis and implementation issues involved in IEEE 802.11 networks. In particular, three topics have been considered: the theoretical and experimental investigation of the QoS features in 802.11e networks, the development and the simulation of backward compatible MAC protocols that use smart antenna systems, and the implementation of a reliable IEEE 802.11ag transceiver using software-defined radio platforms.

Regarding the first topic, the research activity has focused on the deployment of an experimental setup for testing the QoS features in the IEEE 802.11e amendment. Particularly, the MADWiFi drivers have been modified for enabling QoS in ad-hoc mode and throughput measurements. The obtained results have shown a close match between the experimental values and the developed analytical model. Furthermore, a non invasive experimental setup has been deployed to assess the uniformity of the backoff distribution in Atheros-based wireless network cards. Further investigations may be performed to obtain a reliable setup for the backoff analysis of wireless cards of different manufacturers.

The second considered topic has concerned the development of backward compatible MAC protocols that use smart antenna systems. The obtained results have shown the performance increase achievable by enabling multi-packet communication in distributed wireless networks. Particularly, the two novel MPC protocols have presented an enhanced network performance, while maintaining backward compatibility with the 802.11 legacy standard. The first protocol adopts a medium access policy based on a threshold on the sustainable load that depends on the single node antenna capabilities, while the second protocol adopts a medium access policy based on a reliable estimation of the SIR. The obtained results have shown that the MAC protocol based on the SIR estimation outperforms the threshold-based one. The presented studies have also proposed some design guidelines for the development of access protocols that rely on smart antenna systems. These guidelines lead to a cross-layer architecture that enables cooperation between the physical and the MAC layers. Future investigations may be performed to allow the presented access
schemes to operate in high-rank environments adopting at the physical layer MIMO and/or diversity techniques. In order to evaluate the performance of the proposed schemes two realistic simulation platforms have been developed. In particular, the first developed tool has been designed to provide a smart antenna system extension for the ns2 open source simulation platform. The interface between the SAS extension and ns2 has been developed for both MATLAB and Octave. The second platform has been entirely developed in the C++ language using IT++ libraries for implementing the SAS physical layer. The C++ platform, which represents a time-based simulator, has been compared to the ns2 one, which conversely represents an event-based simulator.

The third topic has concerned the implementation of a complete transceiver able to meet the required real-time constrains dictated by the IEEE 802.11ag amendments. The real-time operation constraints have been fulfilled by using several software optimization techniques. This software implementation techniques are suitable to be employed for developing novel physical layers. Furthermore, by adopting computationally powerful equipment, a “smart” multi-protocol node may be developed, also including the implementation of higher layers.

Concluding, this dissertation has presented an experimental and numerical contribution in the field of 802.11 distributed wireless networks, with particular reference to the quality of service features, a MAC layer extension in presence of adaptive arrays, and a software implementation of the physical layer of IEEE 802.11 networks. The three addressed topics are expected to have a growing importance in forthcoming generations of 802.11 networks, which will have to be able to operate in a cognitive and reconfigurable scenario in the presence of sophisticated multimedia services, requiring stringent QoS constraints and ultra high throughput capabilities.
List of publications

Journal Papers


Conference Proceedings


National Workshops


Bibliography


160


[51] “MADWiFi release 0.9.3.3.” [Online]. Available: http://madwifi-project.org/browser/madwifi/branches/madwifi-0.9.3


165


