Performance improvement strategies for current and next generation Wi-Fi systems

Master Thesis

Supervisors:
Andrea Fabio Cattoni
Davide Catania

Authors:
Jagjit Singh Ashta
Marta Gatnau Sarret

13th of June 2012
General Thesis Information

Title: Performance improvement strategies for current and next generation Wi-Fi systems

Project Period: September 2011 - June 2012

Number of Copies: 5

Number of Pages: 139

Authors: Jagjit Singh Ashta
         Marta Gatnau Sarret

Supervisors: Andrea Fabio Cattoni
            Davide Catania

External Censor: Peter Krogsgaard Jensen

Abstract:

When we refer to wireless communications, Wi-Fi quickly comes to our minds despite that the first Wi-Fi standard appeared over 15 years ago. In these 15 years, the standard has evolved a lot, bringing high throughput communications, QoS support, and many other features that granted Wi-Fi a wide footprint.

Today, operators and network equipment vendors see Wi-Fi as part of a bigger picture, a powerful and robust Quality of Service (QoS) oriented network.

Unfortunately, despite of the QoS benefits introduced in Wi-Fi by the 802.11e amendment, these improvements are not enough when combined with the aggregation mechanisms brought by the 802.11n High Throughput amendment. This lack of performance is where our thesis’ study is focused and so we try to improve the throughput and delay presented by Wi-Fi QoS-enabled networks by introducing a Smart Aggregation mechanism.

As described in this thesis, our Smart Aggregation mechanism serves not only as an improvement to the Wi-Fi MAC layer but also works as an adaptive mechanism for a set of aggregation features. In our implementation, it is proven that the proposed Smart Aggregation solution improves the MAC layer performance under certain conditions, while it does not degrade the performance of Wi-Fi 802.11e stations in the rest of the situations.

In other words, we have demonstrated that in a large deployment of Wi-Fi networks we may actually improve the Best Effort goodput almost a 250% while barely affecting the goodput for the other access categories when compared to the 802.11e standard compliant stations.

All information presented in this thesis is strictly confidential. Any copy or distribution of a part or the complete work in any physical or digital media is forbidden without prior permission from the authors, the supervisors or the institutions involved in this project.
Preface

This project has been carried out as a collaboration between Nokia Siemens Networks (NSN) and Aalborg Universitet (AAU) by Marta Gatnau Sarret and Jagjit Singh Ashta as Master Students from Universitat Politècnica de Catalunya (UPC).

The authors would specially like to thank the involved institutions and the supervisors of this project, Andrea F. Cattoni and Davide Catania, for their support and guidance throughout our learning process and the development of our Master Thesis.

We would also like to thank Gustavo Wagner for his support during the development of this project, and Lucas Chavarría for helping us on writing this thesis.

Aalborg, Denmark
13th of June 2012

Marta Gatnau Sarret  
Jagjit Singh Ashta
Acknowledgements

I would like to thank my family for supporting me, not only along these ten months but also all my life, and for encouraging me in difficult moments. I also would thank my friends for being near me every day, my tutor from the UPC Sisco to be so helpful and my supervisors of the Thesis, for all their support and patience and for teaching me how to be a good engineer. Finally, I would acknowledge Nokia Siemens Networks Aalborg and Aalborg University for giving me this great experience and make me feel valued.

I want to dedicate this work to my friend Juan, for making me realize that I could do it.

Marta Gatnau Sarret

I want to thank the support of my family on all the decisions that I took to get to this Master Thesis. Also, I want to acknowledge all the professors and friends, specially my supervisors during this Thesis, that invested time and effort in me, for making me into the engineer that I am today. And lastly, I would like to thank the whole team of Nokia Siemens Networks Aalborg and the company itself, for granting me this opportunity and making this Master Thesis a memorable experience.

I dedicate this work to my sister, because if it weren’t for her, I wouldn’t be a Telecommunications Engineer...

Jagjit Singh Ashta
Contents

List of Figures 1
List of Tables 5
List of Abbreviations 7

1. Introduction 13
   1.1 Motivation of the study 13
       1.1.1 Definition of the problem and scope of this thesis 13
       1.1.2 Brief description of the proposed solution 14
   1.2 Learning Methodology 14
       1.2.1 Initial research on Wi-Fi systems 15
       1.2.2 Scrum Agile software development 15
       1.2.3 Implementation procedure and tools 17
       1.2.4 Results post-processing 18
       1.2.5 Thesis writing 18
   1.3 Thesis outline 18

2. Wireless LAN: The IEEE 802.11 standard 19
   2.1 Wireless Local Area Networks 19
       2.1.1 The 802.11 family: a bit of history 19
       2.1.2 Other improvements to the Wi-Fi standard 21
   2.2 Core architecture and functionalities 22
       2.2.1 Network Architecture 22
       2.2.2 Communication layers structure 23
       2.2.3 Medium Access Mechanism 24
       2.2.4 IFS: Interframe Space 25
   2.3 Adding QoS: The IEEE 802.11e amendment 26
       2.3.1 The QoS mapping from a down-top layer approach 27
Performance improvement strategies for current and next generation Wi-Fi systems

2.3.2 AIFS: Arbitration Interframe Space .......................... 29
2.3.3 TXOP: Transmission Opportunity .......................... 30
2.3.4 EDCAF: Enhanced Distributed Channel Access Function . . 31
2.3.5 Reducing the overhead: The Block Ack mechanism ............ 31
2.4 High throughput communications: The IEEE 802.11n amendment .. 32
2.4.1 Backwards compatibility .................................. 32
2.4.2 RIFS: Reduced Interframe Space ................................ 32
2.4.3 Frame aggregation .................................. 33
2.4.4 Improved Block Ack mechanism ............................ 35
2.4.5 The 802.11n PHY improvements ............................ 35
2.4.6 MCS: Modulation and Coding Schemes ..................... 36

3. Baseline Results ................................................................... 37
3.1 The problem: the inefficient use of QoS in Wi-Fi .................. 37
3.2 Wi-Fi Large Scale Deployments .................................. 39
3.3 QoS single network performance .................................. 44
3.4 QoS neighbouring networks performance ......................... 50
3.5 Adding Aggregation to the System .................................. 52

4. Smart Aggregation .................................................................. 55
4.1 State of the art ......................................................... 55
4.2 Smart Aggregation Overview ........................................ 57
4.3 Smart Aggregation Controller ........................................ 58
4.3.1 Decision criteria .............................................. 58
4.3.2 Smart Aggregation Policies .................................... 60
4.4 Smart Aggregated Frames ............................................... 61
4.4.1 SA-MSDU .................................................. 61
4.4.2 SA-MPDU .................................................. 64
4.4.3 Smart Block Ack ............................................... 65
4.5 Smart Aggregation Process ............................................. 67

5. Wi-Fi MAC Modelling ............................................................. 71
5.1 Wi-Fi MAC Layer ...................................................... 71
5.1.1 Core components .............................................. 72
5.2 Medium contention mechanism ..................................... 73
5.2.1 CCB: Channel Contention Block ......................... 74
5.3 Differentiated Service: QoS ............................................ 76
Performance improvement strategies for current and next generation Wi-Fi systems

Aalborg University

5.3.1 QoS Application layer ........................................... 76
5.3.2 EDCAF State Machine ........................................... 76
5.4 Packet Preparation mechanism .................................... 78
  5.4.1 Buffer limitation model ...................................... 79
  5.4.2 Packet Aggregation mechanism .............................. 80
5.5 Smart Aggregation implementation .............................. 81

6. Smart Aggregation Performance ............................ 85
  6.1 Using Smart Aggregation in a Single Network .............. 85
  6.2 Smart Aggregation in Large Scale Deployments .......... 91
  6.3 Unbalanced Scenarios ........................................... 96
  6.4 Summary .......................................................... 101

7. Conclusions ....................................................... 103
  7.1 Future Work ..................................................... 105

Appendices

A. Wi-Fi MAC Model Validation ................................ 109

B. The Wi-Fi MAC layer in detail ............................. 111
  B.1 CSMA/CA .......................................................... 111
    B.1.1 CSMA Access Modes ...................................... 111
    B.1.2 CSMA Limitations ........................................ 112
    B.1.3 Carrier Sensing Functions .............................. 114
  B.2 DCF: Distributed Coordination Function .............. 114
    B.2.1 Error Recovery in DCF .................................. 115
    B.2.2 Backoff mechanism ...................................... 116
    B.2.3 Fragmentation and Reassembly ......................... 117
    B.2.4 RTS/CTS .................................................. 119
  B.3 PCF: Point Coordination Function ........................ 120
  B.4 HCCA: HCF Controlled Channel Access ................. 120
    B.4.1 TXOP in HCCA ............................................ 121
  B.5 Frame Format .................................................. 121
    B.5.1 General Frame Format .................................. 121
    B.5.2 Data ........................................................ 126
    B.5.3 A-MSDU Frame ............................................ 127
    B.5.4 A-MPDU frame ............................................ 129
    B.5.5 Control ...................................................... 129
    B.5.6 Management ................................................. 133
C. Power Saving in Wi-Fi 135

C.1 802.11-2009: Power Save Multi-Poll 135

References 137
List of Figures

2.1 Maximum theoretical throughput achieved by the 802.11 versions [16] 20
2.2 Wi-Fi Network Architecture. From left to right: Ad-Hoc, Infrastructure BSS, and ESS [18] 23
2.3 MAC Coordination Functions [18] 24
2.4 Representation of IFS and Back-off time [3] 25
2.5 Representation of a Block Ack session procedure [3] 31
2.6 Overhead against Payload comparison under DCF operation w/o RTS/CTS [22] 33

3.1 Downlink delay when considering traffic only in DL and a 54 Mbps PHY rate. 38
3.2 Downlink goodput when considering traffic in UL and DL and 54 Mbps of rate. 39
3.3 Simulated scenarios, varying the deployment ratio and the number of users. 40
3.4 Collision rate against the number of users per network. 41
3.5 Downlink goodput against the number of users per network. 42
3.6 Downlink goodput against deployment ratio. 44
3.7 Downlink goodput with realistic type of traffic in both UL and DL. 46
3.8 Uplink goodput with the same type of traffic in both UL and DL. 47
3.9 Downlink delay with with a fixed traffic scenario only in DL and limited buffer. 48
3.10 Uplink delay with a realistic traffic scenario and limited buffer considering both DL and UL traffic. 49
3.11 Number of packets dropped in dowlink with different type of traffics and limited buffer considering both DL and UL traffic. 50
3.12 Downlink goodput with different type of traffics and unlimited buffer considering both DL and UL traffics in a 2 by 2 system. 51
3.13 Uplink delay with different type of traffics and unlimited buffer considering both DL and UL traffics in a 2 by 2 scenario. 51
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.14</td>
<td>Downlink goodput when considering both DL and UL traffic.</td>
<td>53</td>
</tr>
<tr>
<td>3.15</td>
<td>Downlink delay when considering both DL and UL traffic.</td>
<td>53</td>
</tr>
<tr>
<td>4.1</td>
<td>Smart Aggregation Controller</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>Policy decision</td>
<td>60</td>
</tr>
<tr>
<td>4.3</td>
<td>A-MSDU frame format [4]</td>
<td>62</td>
</tr>
<tr>
<td>4.4</td>
<td>MPDU frame format [4]</td>
<td>62</td>
</tr>
<tr>
<td>4.5</td>
<td>SA-MSDU frame format</td>
<td>62</td>
</tr>
<tr>
<td>4.6</td>
<td>Long SA-MSDU subheader frame format</td>
<td>63</td>
</tr>
<tr>
<td>4.7</td>
<td>Compressed SA-MSDU subheader frame format</td>
<td>64</td>
</tr>
<tr>
<td>4.8</td>
<td>A-MPDU frame format [4]</td>
<td>65</td>
</tr>
<tr>
<td>4.9</td>
<td>Comparison between the Standard aggregation mechanism and the Smart one</td>
<td>66</td>
</tr>
<tr>
<td>4.10</td>
<td>Smart Block Ack frame.</td>
<td>67</td>
</tr>
<tr>
<td>4.11</td>
<td>Diagram that shows when to aggregate</td>
<td>68</td>
</tr>
<tr>
<td>4.12</td>
<td>Aggregated packet depending on the access category that has gained the medium</td>
<td>69</td>
</tr>
<tr>
<td>5.1</td>
<td>Wi-Fi MAC Layer modular representation of our implementation</td>
<td>72</td>
</tr>
<tr>
<td>5.2</td>
<td>Wi-Fi MAC State Machine</td>
<td>73</td>
</tr>
<tr>
<td>5.3</td>
<td>EDCF State Machine</td>
<td>74</td>
</tr>
<tr>
<td>5.4</td>
<td>Frame exchange sequence of a Successful Transmission</td>
<td>74</td>
</tr>
<tr>
<td>5.5</td>
<td>Frame exchange sequence of a Medium access procedure when the back-off countdown is interrupted</td>
<td>75</td>
</tr>
<tr>
<td>5.6</td>
<td>Frame exchange sequence of a Retransmission/DCF recovery mechanism</td>
<td>75</td>
</tr>
<tr>
<td>5.7</td>
<td>Beacon State Machine</td>
<td>75</td>
</tr>
<tr>
<td>5.8</td>
<td>Complete Wi-Fi 802.11e MAC Finite State Machine for an Access Point</td>
<td>76</td>
</tr>
<tr>
<td>5.9</td>
<td>Time diagram for an internal collision</td>
<td>77</td>
</tr>
<tr>
<td>5.10</td>
<td>PSDU preparation model interpretation from the standard</td>
<td>78</td>
</tr>
<tr>
<td>5.11</td>
<td>PSDU preparation model implementation in the simulator</td>
<td>79</td>
</tr>
<tr>
<td>6.1</td>
<td>Downlink delay when there is realistic traffic only in downlink.</td>
<td>87</td>
</tr>
<tr>
<td>6.2</td>
<td>Uplink delay comparison when there is realistic traffic in DL and UL.</td>
<td>89</td>
</tr>
<tr>
<td>6.3</td>
<td>Goodput comparison when there is realistic traffic in DL and UL.</td>
<td>89</td>
</tr>
<tr>
<td>6.4</td>
<td>Downlink delay when there is realistic traffic in UL and DL.</td>
<td>91</td>
</tr>
<tr>
<td>6.5</td>
<td>Uplink goodput when there is realistic traffic in UL and DL.</td>
<td>92</td>
</tr>
<tr>
<td>6.6</td>
<td>Downlink delay when there is fixed traffic in DL.</td>
<td>94</td>
</tr>
<tr>
<td>6.7</td>
<td>Downlink goodput when there is fixed traffic in DL.</td>
<td>95</td>
</tr>
</tbody>
</table>
6.8 Downlink delay when there is realistic traffic in DL and \(1/6^{th}\) of Best Effort traffic in UL. .................................................. 98
6.9 Downlink goodput when there is realistic traffic in UL and DL. . . . 100

7.1 Policy decision ................................................................. 105

A.1 Model validation varying the number of contending stations. ....... 110
A.2 Model validation varying the packet size. ............................. 110

B.1 Hidden node problem scenario [18] .................................... 113
B.2 The Exposed Node Problem [34] ....................................... 114
B.3 Fragmented MSDU [3] ....................................................... 117
B.4 Transmission of a fragmented MSDU [18] .............................. 117
B.5 Using the NAV for virtual carrier sensing [18] ......................... 120
B.6 NAV usage when using Fragmentation and RTS/CTS [18] .......... 120
B.7 802.11n General Frame Format [4] .................................... 121
B.8 Frame Control Field [18] .................................................. 122
B.9 HT Control Field format [4] ............................................. 125
B.14 Frame Control field format for Control Frames [3] .................. 130
B.15 RTS frame format [3] ..................................................... 130
B.16 CTS frame format [3] ..................................................... 130
B.17 ACK frame format [3] ...................................................... 131
B.18 BlockACKReq frame format [4] ....................................... 131
B.19 Block Ack frame format [4] .......................... ........................ 131
B.20 Block Ack Control Field format [4] .................................. 131
B.21 Basic Block Ack Information field [4] ................................. 132
B.22 Compressed Block Ack Information field [4] ......................... 132
B.23 Multi-TID Block Ack Information field [4] ........................... 132
B.24 General Management frame format [3] ............................... 133
B.25 Beacon transmission mechanism in Infrastructure (left) and Ad-hoc (right) networks [36] .................................................. 133
B.26 HT Capabilities element format [4] .................................... 133
B.27 HT Capabilities Info subfield [4] ...................................... 134

C.1 PSMP frame exchange sequence ......................................... 135
List of Tables

2.1 IEEE 802.11 a/b/g/n/ac feature comparison .......................... 20
2.2 802.11 versions released as of 2011 ............................. 22
2.3 Upcoming Wi-Fi versions ........................................... 22
2.4 802.11g ERP PHY values ........................................... 26
2.5 TID subfield values of the QoS Control Field [3] ............... 27
2.6 802.1p User Priority to 802.11e Access Category mapping ... 27
2.7 AF PHB possible values ............................................. 28
2.8 Access Categories ................................................. 29
3.1 Parameters used in Large Scale deployment results. ............... 40
3.2 Parameters used to compare 802.11g stations with and without the 802.11e amendment. ........................................ 44
3.3 Realistic traffic scenario specifications. ............................ 45
3.4 Simulation scenario parameters for A-MPDU vs MPDU comparison. ......................................................... 52
4.1 CRC Type ............................................................. 63
4.2 Block Ack frame variant encoding .................................... 66
6.1 Parameters we have used to run the simulations .................. 86
6.2 Description of the characteristics for each AC for realistic traffic scenarios. ......................................................... 86
6.3 Description of the characteristics for each AC for realistic traffic scenarios. ......................................................... 86
6.4 Delay comparison of figure 6.1 for 19 users ....................... 88
6.5 Delay comparison of figure 6.2 for 19 users ....................... 90
6.6 Goodput comparison of figure 6.3 for 9 users .................... 90
6.7 Goodput comparison of figure 6.3 for 19 users .................... 91
6.8 Delay comparison of figure 6.4 for 5 users/network ............... 92
6.9 Goodput comparison of figure 6.5 for 2 users per network .... 93
6.10 Delay comparison of figure 6.6 for 5 users/network ............... 94
6.11 Goodput comparison of figure 6.7 for 2 users per network .... 95
Performance improvement strategies for current and next generation Wi-Fi systems

6.12 Description of the possible traffic flows in uplink traffic. .......................... 96
6.13 Characteristics of the realistic traffic when there is only BE traffic in UL. ..................... 96
6.14 Characteristics of the realistic traffic when there are all type of traffics in UL. .................. 97
6.15 Characteristics of the fixed traffic scenario when there is only BE traffic in UL. .................. 97
6.16 Characteristics of the fixed traffic scenario when there are all types of traffics in UL. .................. 97
6.17 Delay comparison of figure 6.8 for 14 users ........................................... 98
6.18 Delay comparison of figure 6.8 for 20 users ........................................... 99
6.19 Goodput comparison of figure 6.9 for 14 users ........................................... 100
6.20 Goodput comparison of figure 6.9 for 20 users ........................................... 100

B.1 QoS Control field [4] ............................................................... 124
B.2 ACK Policy subfield ............................................................... 125
List of Abbreviations

3GPP 3rd Generation Partnership Project
A-MPDU Aggregate MAC Protocol Data Unit
A-MSDU Aggregate MAC Service Data Unit
AC Access Category
ACK Acknowledgement
AF Assured Forwarding
AIFS Arbitration Interframe Space
AIFSN Arbitration Interframe Space Number
AP Access Point
API Application Programming Interface
APSD Automatic Power Save Delivery
BA Block Acknowledgement
BAR Block Acknowledgement Request
BRQ Block Ack Response Quality
BSS Basic Service Set
BSSID BSS Identifier
CAP Controlled Access Phase
CCA Clear Channel Assessment
CCB Channel Contention Block
CFP Contention Free Period
CP Contention Period
CS Class Selector
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access/Collision Avoidance</td>
</tr>
<tr>
<td>CSMA/CD</td>
<td>Carrier Sense Multiple Access/Collision Detection</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear-To-Send</td>
</tr>
<tr>
<td>CW</td>
<td>Contention Window</td>
</tr>
<tr>
<td>DA</td>
<td>Destination Address</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>DIFS</td>
<td>DCF Interframe Space</td>
</tr>
<tr>
<td>DL</td>
<td>Downlink</td>
</tr>
<tr>
<td>DS</td>
<td>Distribution System</td>
</tr>
<tr>
<td>DSCP</td>
<td>Differentiated Services Code Point</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>EDCA</td>
<td>Enhanced Distributed Channel Access</td>
</tr>
<tr>
<td>EDCAF</td>
<td>Enhanced Distributed Channel Access Function</td>
</tr>
<tr>
<td>EDCF</td>
<td>Enhanced Distributed Coordination Function</td>
</tr>
<tr>
<td>EF</td>
<td>Expedited Forwarding</td>
</tr>
<tr>
<td>EIFS</td>
<td>Extended Interframe Space</td>
</tr>
<tr>
<td>EOSP</td>
<td>End of Service Period</td>
</tr>
<tr>
<td>ERP</td>
<td>Extended Rate PHY</td>
</tr>
<tr>
<td>ESS</td>
<td>Extended Service Set</td>
</tr>
<tr>
<td>FCS</td>
<td>Frame Check Sequence</td>
</tr>
<tr>
<td>FH</td>
<td>Frequency Hopping</td>
</tr>
<tr>
<td>FHSS</td>
<td>Frequency Hopping Spread Spectrum</td>
</tr>
<tr>
<td>FSM</td>
<td>Finite State Machine</td>
</tr>
<tr>
<td>GI</td>
<td>Guard Interval</td>
</tr>
<tr>
<td>GRASS</td>
<td>Generic Radio Access System Simulator</td>
</tr>
<tr>
<td>HC</td>
<td>Hybrid Coordinator</td>
</tr>
<tr>
<td>HCCA</td>
<td>HCF Controlled Channel Access</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>HCF</td>
<td>Hybrid Coordination Function</td>
</tr>
<tr>
<td>HR</td>
<td>High Rate</td>
</tr>
<tr>
<td>HT</td>
<td>High Throughput</td>
</tr>
<tr>
<td>IBSS</td>
<td>Independent Basic Service Set</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IFS</td>
<td>Interframe Space</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial, Scientific and Medical</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LLC</td>
<td>Logical Link Control</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MCS</td>
<td>Modulation and Coding Scheme</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple In Multiple Out</td>
</tr>
<tr>
<td>MLME</td>
<td>MAC Sublayer Management Entity</td>
</tr>
<tr>
<td>MPDU</td>
<td>MAC Protocol Data Unit</td>
</tr>
<tr>
<td>MSDU</td>
<td>MAC Service Data Unit</td>
</tr>
<tr>
<td>MTU</td>
<td>Maximum Transmission Unit</td>
</tr>
<tr>
<td>MU-MIMO</td>
<td>Multi User MIMO</td>
</tr>
<tr>
<td>NAV</td>
<td>Network Allocation Vector</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>PC</td>
<td>Point Coordinator</td>
</tr>
<tr>
<td>PCF</td>
<td>Point Coordination Function</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>PHB</td>
<td>Per-Hop Behaviour</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical Layer</td>
</tr>
<tr>
<td>PIFS</td>
<td>PCF Interframe Space</td>
</tr>
<tr>
<td>PLCP</td>
<td>Physical Layer Convergence Procedure</td>
</tr>
<tr>
<td>PMD</td>
<td>Physical Medium Dependent</td>
</tr>
<tr>
<td>PPDU</td>
<td>PLCP Protocol Data Unit</td>
</tr>
<tr>
<td>PS</td>
<td>Power Saving</td>
</tr>
<tr>
<td>PS-Poll</td>
<td>Power Save Poll</td>
</tr>
<tr>
<td>PSDU</td>
<td>PLCP Service Data Unit</td>
</tr>
<tr>
<td>PSMP</td>
<td>Power Save Multi-Poll</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RAN</td>
<td>Radio Access Network</td>
</tr>
<tr>
<td>RIFS</td>
<td>Reduced Interframe Space</td>
</tr>
<tr>
<td>RTS</td>
<td>Request-To-Send</td>
</tr>
<tr>
<td>S-APSD</td>
<td>Scheduled Automatic Power Save Delivery</td>
</tr>
<tr>
<td>SA</td>
<td>Source Address</td>
</tr>
<tr>
<td>SA-MPDU</td>
<td>Smart Aggregate MAC Protocol Data Unit</td>
</tr>
<tr>
<td>SA-MSDU</td>
<td>Smart Aggregate MAC Service Data Unit</td>
</tr>
<tr>
<td>SAC</td>
<td>Smart Aggregation Controller</td>
</tr>
<tr>
<td>SDU</td>
<td>Service Data Unit</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short Interframe Space</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>SP</td>
<td>Service Period</td>
</tr>
<tr>
<td>STA</td>
<td>Station in Wi-Fi network</td>
</tr>
<tr>
<td>SVN</td>
<td>Subversion</td>
</tr>
<tr>
<td>TBTT</td>
<td>Target Beacon Transmission Time</td>
</tr>
<tr>
<td>TDD</td>
<td>Test Driven Development</td>
</tr>
<tr>
<td>TG</td>
<td>Task Group</td>
</tr>
<tr>
<td><strong>TID</strong></td>
<td>Traffic Identifier</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------</td>
</tr>
<tr>
<td><strong>TOS</strong></td>
<td>Type of Service</td>
</tr>
<tr>
<td><strong>TS</strong></td>
<td>Traffic Stream</td>
</tr>
<tr>
<td><strong>TSF</strong></td>
<td>Timing Synchronization Functions</td>
</tr>
<tr>
<td><strong>TSPEC</strong></td>
<td>Traffic Specification</td>
</tr>
<tr>
<td><strong>TTL</strong></td>
<td>Time To Live</td>
</tr>
<tr>
<td><strong>TXOP</strong></td>
<td>Transmission Opportunity</td>
</tr>
<tr>
<td><strong>U-APSD</strong></td>
<td>Unscheduled Automatic Power Save Delivery</td>
</tr>
<tr>
<td><strong>UL</strong></td>
<td>Uplink</td>
</tr>
<tr>
<td><strong>UP</strong></td>
<td>User Priority</td>
</tr>
<tr>
<td><strong>Wi-Fi</strong></td>
<td>Wireless Fidelity</td>
</tr>
<tr>
<td><strong>WLAN</strong></td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
Chapter 1.

Introduction

1.1 Motivation of the study

1.1.1 Definition of the problem and scope of this thesis

As the amount of services offered through the internet grows everyday, the demand for higher and higher throughput rates has become the main focus of most of the modern wireless communication systems.

But increasing the overall throughput is not the only key point to tackle, but an improved QoS support in all types of networks has become a must. In many countries, the large operators are seeking everyday, in the modern all IP-based networks such as LTE, better ways to support Voice and Video streaming. In order to do that, operators and manufacturers have started focusing on end-to-end QoS support.

So in this sense, in order to achieve a good QoS for certain types of services, operators need that each part of their entire communication system, from Radio Access Network (RAN) to Internet, supports a robust QoS mechanism that is attached in perfect harmony with the rest of their network.

Of course, the QoS support and scheduling mechanisms are already part of the 3G and 4G standards, but as the demand grows and the resources start to vanish, operators and network equipment vendors are trying to find the technological solution that will allow the operators to offload the higher amount of traffic that static users generate. These offloading solutions are intended to allow users in high mobility to keep their connections alive through the common mobile communications standards, while the high demanding static users keep their connection alive through a secondary network such as Wi-Fi [1].

Operators, network vendors and standardization institutions see Wi-Fi as one of the best offloading solutions not just because of its popularity, but also for its incredibly low-cost deployment. Unlike GSM, 3G and 4G mobile communications, Wi-Fi uses
license-free spectrum and most of the times the RAN equipment, i.e. the Wi-Fi router, is financed by the end-user itself.

Luckily Wi-Fi has been vastly improved since its first versions appeared, and now it seems like a more reliable offloading solution than ever. But the overcrowding of the Industrial, Scientific and Medical (ISM) band and the demand for large scale deployments of Wi-Fi take us to the fact that the 802.11 standard does not offer good enough QoS support especially when Aggregation mechanisms are being used. Thus not making it yet the perfect companion for the future QoS-enabled networks.

As we will later present in our thesis, the traffic differentiation scheme proposed by the 802.11e amendment does accurately give higher priority to Voice packets by achieving lower delay and higher throughput values for this type of traffic when compared to the traffic of other lower priorities.

Unfortunately in some situations, especially in crowded networks, the performance of an 802.11e station may be worse than the performance obtained by the legacy 802.11g stations. We observed this phenomenon in Wi-Fi in which despite of obtaining some clearly differentiated delay and throughput metrics, even the Voice transmissions take longer and get a lower throughput in comparison to the 802.11g stations that have no traffic differentiation.

This means that the use of traffic differentiation in some cases may result pointless. And this is one of the aspects that we want to deal with the Smart Aggregation mechanism.

1.1.2 Brief description of the proposed solution

Throughout this thesis we have studied, described, modelled and analyzed the performance of current Wi-Fi systems, focusing mainly on the QoS support by enhancing the Aggregation mechanisms introduced by the 802.11n amendment.

Our solution, the Smart Aggregation mechanism, is designed as a packet preparation scheme that intelligently adapts the criteria on which packets the MAC layer should aggregate. As shown in the Smart Aggregation Performance chapter, with our aggregation mechanism we have successfully improved the QoS performance in comparison to the one achieved by legacy 802.11e stations by drastically reducing the delay and improving the goodput for Best Effort traffic while keeping the Voice packets as the highest priority flow.

1.2 Learning Methodology

As an almost year-long project, our skills and knowledge needed some management and a proper learning curve in order to achieve the maximum performance out of
two Master students that had not previously been involved in any complex project as this one is.

To do so, with the help of our supervisors and a set of Nokia Siemens Networks’ resources that were made available to us, we learned and studied a set of technologies and working methodologies that are described in this section.

1.2.1 Initial research on Wi-Fi systems

When we started we took a 2 month research on Wi-Fi systems. This research served as basis for the elaboration of Chapter 2 and allowed us to deepen our knowledge in the IEEE 802.11 standard.

During this research, besides multiple internet resources and some analytical papers, we mainly focused on getting a cleaner understanding of the IEEE 802.11a/b [2], the 802.11g and 802.11e (QoS) [3], the 802.11n [4], as well as getting an overview of some of the key features presented by the IEEE 802.11ac [5].

This research was taken on a task basis in two week cycles that allowed us to concentrate on our study as well as deliver a small summary at the end of each cycle which focused on some key parts of the Wi-Fi standard.

Thanks to this initial study, we were able to point out some of the key areas where the Wi-Fi standard can still be further improved such as mobility, QoS, MAC throughput improvements, PHY rate improvements and others. And after presenting a series of ideas, with a group decision, the Smart Aggregation mechanism was chosen as the most promising solution.

1.2.2 Scrum Agile software development

The rest of our work was also based on small tasks, or items, but now it was efficiently managed using the Scrum methodology, mainly oriented towards coding project developments and based on: transparency, inspection and adaptation.

As engineers working in cooperative teams and on projects basis is a must and thanks to the Scrum methodology we learnt not only to work efficiently in a development team, but we were able to quickly solve and rectify any obstacles that were presented to us throughout our thesis.

This is done by having the following members in the team, all of which are self-organized parts of the scrum:

**Product owner:** maximizes the value of the product and the work of the development team. It also manages the product backlog by keeping track of goals, missions, values, as well as giving higher load to those tasks that are more relevant to the next increment.
Performance improvement strategies for current and next generation Wi-Fi systems

Development team: achieve the tasks marked by the product backlog by generating the product increments. These teams are usually self-organized, not too big (max. 9), cross-functional, and all members are equal in responsibilities, without specific titles nor individualities.

Scrum master: is in charge of interacting outside the scrum team. It may also assist the product owner with the backlog communication to the development team.

In Scrum, the product development is based on an iterative approach by delivering incremental results which maximize the feedback. These iterations are divided in various time intervals:

Sprint (<1 month): is the time to deliver an increment. During this time interval, a maximum number of tasks must be "done". To do so, a prior agreement on what "done" means is needed in the scrum team.

Planning meeting (<8 hours/Sprint): every sprint is preceded by a planning where what tasks have to be delivered in the next increment and how these will be achieved by the development team is discussed.

Sprint review (<4 hours/Sprint): held at the end of a sprint to identify the increment, and discuss the product backlog.

Daily Scrum (<15 min/day): intended to synchronize the work of all members in the development team. This meeting focuses on Accomplishments, objectives and obstacles.

All of these periods, except for the daily scrum, will be proportionally smaller depending on how long the Product owner wants the Sprints to be. In our case a Sprint cycle of two weeks was set.

The definition of Done

As demanded by the Scrum project management methodology, the definition of "done" must be prior to the start of this project. In our case, to be able to mark an increment as "done" the following requirements were met:

- Keep the respect in the team and follow the Scrum guidelines
- Make the code available
- Make the tests available (with extensive coverage)
- The code must pass ALL tests.
Performance improvement strategies for current and next generation Wi-Fi systems

The code runs in overall
- Code and tests are committed to the Subversion (SVN)
- Increment’s model documentation is ready and committed
- The code is Doxygen-ready [6] [7]
- The results are reasonable
- The Twiki is updated with the Sprint presentation and burndown chart

1.2.3 Implementation procedure and tools

During all of the sprints’ work, we did a series of implementations in an NSN proprietary simulator known as GRASS. This was done by using the following series of resources and methodologies:

Coding Language - C++: after a deep study of the C++ programming language by using [8] and [9], we were able to work better with a project of quite big dimensions as GRASS is.

Working environment - Eclipse: as an open source and very flexible solution we developed our work inside the Eclipse Integrated Development Environment (IDE) which allowed us to use the SVN tool with more ease by using Subclipse [10].

Collaboration tool - SVN: allowed each development team member to work independently and at the same time commit the updates to a centralized version [11]

Programming methodology: in order to provide not only a working implemented model, but an actually robust implementation in C++, we used:

- Test Driven Development (TDD) [12]: in order to make sure that the code works now, and keeps running when new features are implemented, as well as to reduce the later time to solve problems in the code. This was done by using the UnitTest++ library [13].
- Design patterns [14]: to somehow standardize and make the code more object oriented. Some of the most commonly used by us are: State Pattern, for the MAC State Machine, Abstract Factory Pattern, to generate the nodes, the Decorator Pattern, to transform the Service Data Units (SDUs) into Protocol Data Units (PDUs), and the Observer Pattern, to connect the different blocks.

By using these implementation tools, we have now reached a level of programming skills that allowed us to implement our Wi-Fi MAC model really fast while keeping the highest quality of Object Oriented Programming.
1.2.4 Results post-processing

Once the implementation was done and an adequate testing was completed, we tested the correct behaviour of all components of our Wi-Fi stations by implementing a logging system which can be enabled or disabled. This logging system fetches information such as on what state the MAC layer is, what frame is being sent among others.

To get our results, we outputted the values obtained from the simulator and post-processed them in Matlab by using advanced scripts that allowed us to generate the results included in this Thesis.

1.2.5 Thesis writing

We are finally concluding our 10 month project with this thesis where we have collected most of our work, from the initial study up to the final results obtained with a basic implementation of the Smart Aggregation mechanism.

1.3 Thesis outline

This thesis is divided in 8 different chapters including this introductory chapter. The remainder of this thesis is organized as follows:

Chapter 2 presents a study of the Wi-Fi features that serve as a base for the Smart Aggregation mechanism.

Chapter 3 presents a set of results that serve as an evaluation of the performance of current Wi-Fi systems. At the end of this chapter we frame the problem that our Smart Aggregation mechanism tries to fight.

Chapter 4 gives a detailed and complete description of the proposed Smart Aggregation solution.

Chapter 5 is an overview of the implementation model that was used in order to simulate and compare the current Wi-Fi technologies and the improvement proposed in this thesis.

Chapter 6 shows the performance of our proposed solution and compares it against the current Wi-Fi systems.

Chapter 7 concludes this thesis by pointing out the main results we obtained and also including some proposal improvements to our own Smart Aggregation mechanism to achieve a truly cognitive MAC features management.
Chapter 2.

Wireless LAN: The IEEE 802.11 standard

2.1 Wireless Local Area Networks

2.1.1 The 802.11 family: a bit of history

The first Wireless Local Area Network (WLAN) standard appeared in 1997 [2] and specified a first set of Medium Access Control (MAC) and Physical Layer (PHY) specifications that were based on the FHSS/DSSS and would offer 1 or 2 Mbps of throughput in the 2.4 GHz ISM band. The more commercially known 802.11a and 802.11b appeared later in 1999 as amendments to the original standard. In 2003 the 802.11g was finalized and along the 802.11b is currently the most widely adopted Wi-Fi technology[15].

The constant increase of services offered through the Internet and the fast improvement of the wired Local Area Network (LAN) which had surpassed the Gigabit speed in standard specification in 2003 demanded faster WLAN communications which gave birth to the final version of the 802.11n [4] in 2009. Finally the next big thing to look for among the Wi-Fi technologies is the 802.11ac [5] which goes beyond the Gigabit for wireless communications.

In table 2.1 and figure 2.1 [16] we can see the main aspects that changed throughout time in the different Wi-Fi standards and amendments.

This advancement in Wi-Fi technologies from 54 Mbps to over 1 Gbps of speed in roughly 10 years has been possible mainly thanks to the reduced price of DSPs and the improvements in the Multiple In Multiple Out (MIMO) technology. Also alike other radio access technologies such as Long Term Evolution (LTE), Wi-Fi did also multiply its maximum throughput rates by increasing the number of radio channels used for a single communication, but unlike other technologies, most of 802.11’s
wider bandwidth schemes are based on channel bonding rather than separate carrier aggregation. It is only the 160 MHz bandwidth supported by the 802.11ac which allows for two separated 80 MHz channels.

<table>
<thead>
<tr>
<th>802.11a</th>
<th>802.11b</th>
<th>802.11g</th>
<th>802.11n</th>
<th>802.11ac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Wi-Fi5</td>
<td>Wi-Fi</td>
<td>Wi-Fi</td>
<td>Wi-Fi-n</td>
</tr>
<tr>
<td>Operating Frequency</td>
<td>5 GHz</td>
<td>2.4 GHz</td>
<td>2.4 GHz</td>
<td>2.4/5 GHz</td>
</tr>
<tr>
<td>Maximum Throughput</td>
<td>54 Mbps</td>
<td>11 Mbps</td>
<td>54 Mbps</td>
<td>600 Mbps</td>
</tr>
<tr>
<td>Modulation</td>
<td>OFDM</td>
<td>HR/DSSS</td>
<td>HR/DSSS, CCK, OFDM</td>
<td>HR/DSSS, CCK, OFDM</td>
</tr>
<tr>
<td>Channel Bandwidth</td>
<td>20 MHz</td>
<td>20 MHz</td>
<td>20 MHz</td>
<td>20 or 40 MHz</td>
</tr>
<tr>
<td>MIMO</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Up to 4 streams</td>
</tr>
<tr>
<td>Others</td>
<td>Devices did not arrive until 2000</td>
<td>Compatible with 802.11b</td>
<td>Backwards compatible; Maximum of 64QAM</td>
<td>a and n compatible; Maximum of 256QAM</td>
</tr>
</tbody>
</table>

Table 2.1: IEEE 802.11 a/b/g/n/ac feature comparison

Figure 2.1: Maximum theoretical throughput achieved by the 802.11 versions [16]
2.1.2 Other improvements to the Wi-Fi standard

The letter behind each Institute of Electrical and Electronics Engineers (IEEE) 802.11 release is related to when the Task Group (TG) started to investigate that improvement to the last revision of the standard. And so we find a nice timeline [17] that summarizes all of the current and upcoming 802.11 versions.

Some of these improvements, some of which may not be so commonly known by the end-user, are shown in Table 2.2, where the most commercially known versions have been marked in bold. The next Table 2.3 shows a summary of yet to be released versions of Wi-Fi.

<table>
<thead>
<tr>
<th>Year</th>
<th>Version</th>
<th>Features description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>802.11</td>
<td>Standard definition: MAC and PHY Layer specifications. Based on FH/DSSS. 1 or 2 Mbps @ 2.4 GHz</td>
</tr>
<tr>
<td>1999</td>
<td>802.11a</td>
<td>PHY Layer extension to work in 5 GHz, up to 54 Mbps. Operates using OFDM. Also known as Wi-Fi 5. Offers 12 non overlapped channels.</td>
</tr>
<tr>
<td></td>
<td>802.11b</td>
<td>PHY Layer extension to work in 2.4 GHZ, at 5.5 or 11 Mbps, uses FH/DSSS. Also known as Wi-Fi. Offers 3 non overlapped channels.</td>
</tr>
<tr>
<td>2000</td>
<td>802.11c</td>
<td>MAC Bridge support for 802.11d</td>
</tr>
<tr>
<td>2001</td>
<td>802.11d</td>
<td>Support for international spectrum regulations in 2.4 GHz band.</td>
</tr>
<tr>
<td>2003</td>
<td>802.11f</td>
<td>Recommendation to use the Inter-Access Point Protocol to allow roaming stations to switch between AP seamlessly.</td>
</tr>
<tr>
<td></td>
<td>802.11g</td>
<td>Higher data rates using OFDM, backwards compatible with 802.11b. Up to 54 Mbps.</td>
</tr>
<tr>
<td></td>
<td>802.11h</td>
<td>Better support for European spectrum in 5 GHz band. Adds MIMO and allows for up to 19 non overlapped channels.</td>
</tr>
<tr>
<td>2004</td>
<td>802.11i</td>
<td>MAC Layer security enhancements based on AES for a/b/g technologies</td>
</tr>
<tr>
<td></td>
<td>802.11j</td>
<td>Spectrum extension for operations in Japan at 5 GHz.</td>
</tr>
<tr>
<td>2005</td>
<td>802.11e</td>
<td>MAC QoS enhancement including packet bursting and HCF.</td>
</tr>
<tr>
<td>2007</td>
<td>802.11-2007</td>
<td>New release of the standard including all previous amendments and standards.</td>
</tr>
<tr>
<td>2008</td>
<td>802.11k</td>
<td>Radio resource measurement enhancements</td>
</tr>
<tr>
<td></td>
<td>802.11r</td>
<td>Fast roaming</td>
</tr>
<tr>
<td></td>
<td>802.11y</td>
<td>Spectrum extension for USA in 3650-3700 MHz band.</td>
</tr>
<tr>
<td>2009</td>
<td>802.11n</td>
<td>Higher throughput by using MIMO, channel and frame aggregation, and backwards compatibility.</td>
</tr>
</tbody>
</table>

Table 2.2 – Continued on next page...
Performance improvement strategies for current and next generation Wi-Fi systems

<table>
<thead>
<tr>
<th>Year</th>
<th>Version</th>
<th>Features description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>802.11p</td>
<td>Wireless Access for Vehicular Environment (WAVE)</td>
</tr>
<tr>
<td>2011</td>
<td>802.11z</td>
<td>Extension of the Direct Link Setup</td>
</tr>
<tr>
<td>2011</td>
<td>802.11v</td>
<td>Wireless Network Management</td>
</tr>
<tr>
<td>2011</td>
<td>802.11u</td>
<td>Interworking with non-802 networks, such as the existing cellular networks.</td>
</tr>
</tbody>
</table>

Table 2.2: 802.11 versions released as of 2011

In our thesis and study we have mainly focused on the currently available 802.11g and the 802.11n versions with the QoS improvements that the 802.11e amendment brought to the standard. We have also taken into account the upcoming 802.11ac version’s features and frame formats when deciding how to define the Smart Aggregation mechanism.

<table>
<thead>
<tr>
<th>Version</th>
<th>Features description</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11s</td>
<td>Mesh networking support using ESS</td>
</tr>
<tr>
<td>802.11-2012</td>
<td>Standard redefinition including all previous amendments.</td>
</tr>
<tr>
<td>802.11aa</td>
<td>Robust streaming of Audio/Video streams</td>
</tr>
<tr>
<td><strong>802.11ac</strong></td>
<td><strong>Very High throughput under 6 GHz</strong></td>
</tr>
<tr>
<td>802.11ad</td>
<td>Very High throughput near 60 GHz</td>
</tr>
<tr>
<td>802.11ae</td>
<td>QoS management by prioritization of management frames.</td>
</tr>
<tr>
<td>802.11af</td>
<td>Operation in TV Whitespaces spectrum</td>
</tr>
<tr>
<td>802.11ah</td>
<td>Operation in the sub 1 GHz bands</td>
</tr>
<tr>
<td>802.11ai</td>
<td>Fast initial Link Setup.</td>
</tr>
</tbody>
</table>

Table 2.3: Upcoming Wi-Fi versions

2.2 Core architecture and functionalities

2.2.1 Network Architecture

The WLAN can be classified in very different ways, and one of them is based on their network configuration, according to which they can work in two modes:

**Ad-Hoc**: Direct point-to-point communication. Very limited in number of stations (up to 4). Also known as Independent Basic Service Set (IBSS).
Performance improvement strategies for current and next generation Wi-Fi systems

Aalborg University
RATE Section

**Infrastructure:** All communications take place through one centralized Access Point (AP). This is the type of network we will focus on.

Each infrastructure WLAN network defines a Basic Service Set (BSS) and is assigned to an alphanumeric BSSID. When multiple BSS have been assigned to the same BSSID, this group of networks is then known as Extended Service Set (ESS). Unlike some of the 3rd Generation Partnership Project (3GPP) standards, in Wi-Fi one user or Station (STA) can only be connected to a single BSS but may freely "roam" from one AP to another inside the same ESS.

In the next figure, extracted from [18], we can see an example of the types of network that a Wi-Fi communication defines, inside which an STA’s mobility is defined. This mobility has been numerous times improved with various amendments and is an interesting part of Wi-Fi, but it lies out of the scope of this study.

![Figure 2.2: Wi-Fi Network Architecture. From left to right: Ad-Hoc, Infrastructure BSS, and ESS [18]](image)

In Wi-Fi networks, to create the backhaul network to which the end-users connect to get to the Internet or to other external networks, all APs from that same ESS are connected to a common Distribution System (DS) which usually relies on the wired Ethernet protocol. This DS is then connected to a Router which serves as a gateway to the Internet.

### 2.2.2 Communication layers structure

In comparison with the OSI model of the ISO [19], the Wi-Fi standard specifies the following Layer 2 and Layer 1 communication protocols:

Layer 2 is divided in two sublayers, the IEEE 802.2 Logical Link Control (LLC) and the IEEE 802.11 MAC sublayer.

Layer 1, known as the Wi-Fi PHY, is also divided in two separate sublayers, the Physical Layer Convergence Procedure (PLCP) and the Physical Medium Dependent (PMD).
The commercial 802.11a/b/g versions mainly improved the PHY efficiency, using the exact same MAC layer as the original 802.11 standard [2], whereas other amendments mainly focused on improving the MAC, such as the 802.11e (adding QoS support) or the 802.11i (improved security).

Finally, the more modern 802.11n/ac versions of the standard, despite highly improving the PHY rate, they demanded a new and improved MAC to avoid reaching the maximum saturation throughput due to overhead of the protocol.

2.2.3 Medium Access Mechanism

The Wi-Fi standard is based on Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) which is a CSMA p-persistent based mechanism, further explained in section B.1. On top of which, some medium access coordination functions have been defined throughout the various versions of Wi-Fi:

Distributed Coordination Function (DCF): based on CSMA/CA, which uses a random back-off time to reduce the probability of collisions.

Point Coordination Function (PCF): in which the AP acts as a Point Coordinator (PC) and sends a beacon every Target Beacon Transmission Time (TBTT). Transmissions are granted by the PC by polling all stations and guaranteeing a Contention Free Period (CFP) to the polled station if it has packets in queue. Since the implementation of PCF is not mandatory, usually networks where a PC exists, divide their TBTT in the CFP, and the Contention Period (CP) where DCF is used.

Hybrid Coordination Function (HCF): was defined in the 802.11e to allow traffic differentiation. HCF defines two new access schemes known as HCF Controlled Channel Access (HCCA) which based on polling but allows for the Controlled Access Phase (CAP) to be initiated almost any time during a CP; and an Enhanced Distributed Channel Access Function (EDCAF) which is contention-based.

![Figure 2.3: MAC Coordination Functions](image)
All these coordination functions are further described in the appendix B., except for the EDCAF which is focus of our improvement and is therefore detailed in the section 2.3.4.

2.2.4 IFS: Interframe Space

The 802.11 [2] CSMA/CA mechanism uses up to four different Interframe Space (IFS) periods during which a frame must be hold in queue before a transmission shall occur. These are used for the following functions and characteristics:

- To determine the medium access.
- To create different priority levels for different types of traffic.
- Fixed amount of time, independent of the transmission speed.
- Different PHY layers can specify different interframe space times.

The four type of IFS described in the first standard are:

**Short Interframe Space (SIFS)** Used to send high priority transmissions, typically ACK or CTS frames, or to separate burst transmissions. When a frame is waiting SIFS to be transmitted it is not necessary for the medium to be sensed as free.

**PCF Interframe Space (PIFS)** this frame spacing applies only when PCF is being used in the network and for transmissions within a CFP, to prioritize contention-free traffic against contention-based traffic from other stations.

**DCF Interframe Space (DIFS)** this is the minimum medium idle time for a contention-based transmission. Stations will have access to the medium if it has been free for a period longer than the DIFS.

**Extended Interframe Space (EIFS)** shall optionally be used instead of DIFS whenever an erroneous frame is received, as this means that the medium has become error-prone, thus a larger contention time will result in a lower collision rate at the cost of some overhead.

![Figure 2.4: Representation of IFS and Back-off time [3]](image-url)
All of the above time intervals are fixed depending on the PHY that is being used. The values used in 802.11g are shown in table 2.4.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot Time</td>
<td>aRxRFDelay + aRxPLCPDelay + aMACProcessingDelay + aRxTxTurnaroundTime</td>
<td>Long 20 µs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Short 9 µs</td>
</tr>
<tr>
<td>SIFS</td>
<td>aCCATime + aRxTxTurnaroundTime + aAirPropagationTime + aMACProcessingDelay</td>
<td>10 µs</td>
</tr>
<tr>
<td>PIFS</td>
<td>aSIFSTime + aSlotTime</td>
<td>30 µs / 19 µs</td>
</tr>
<tr>
<td>DIFS</td>
<td>aSIFSTime + 2 x aSlotTime</td>
<td>50 µs / 28 µs</td>
</tr>
<tr>
<td>EIFS</td>
<td>aSIFSTime + DIFS + ACKTxTime</td>
<td>Rate dependent</td>
</tr>
</tbody>
</table>

Table 2.4: 802.11g ERP PHY values

Note: the short Slot Time in 802.11g networks shall only be used when no backwards compatibility is required.

2.3 Adding QoS: The IEEE 802.11e amendment

This amendment focused on MAC layer improvements to support QoS for WLAN applications. This is done mainly by adding the new HCF, which is based on two new methods of channel access. 802.11e also adds better power management and new acknowledgement techniques.

When sending a QoS frame, the Wi-Fi standard adds a new field to the MAC header to describe the type of transmission and set the priority levels for that frame. This QoS Control Field is further described in B.5.1.

Applications like Video or audio require synchronized timers shared among different STAs. Thus, an additional requirement for the BSS is the Timing Synchronization Functions (TSF). Which is why the 802.11e defines a MAC service that enables layers above the MAC to do this type of functions, and it can be used for more than one application at a time.
2.3.1 The QoS mapping from a down-top layer approach

Wi-Fi Access Categories and Traffic Identifiers

To create traffic differentiation, the 802.11e introduces the concept of Access Category (AC). Traffic is therefore divided into four possible types depending on the level of priority requested from the above layers: Voice, Video, Best Effort and Background.

Once an MPDU has been assigned an AC, it is assigned a Traffic Identifier (TID) which is a combination of the access mechanism being used and the User Priority (UP) to which that MPDU belongs. Given the 8 types of UP, and the two access mechanisms in HCF (EDCAF and HCCA), a total of 16 different TID values can be assigned, which is a value that is then attached inside the QoS Control Field.

<table>
<thead>
<tr>
<th>Access policy</th>
<th>Usage</th>
<th>Allowed values in bits 0–3 (TID subfield)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDCA</td>
<td>UP for either TC or TS, regardless of whether admission control is required</td>
<td>0–7</td>
</tr>
<tr>
<td>HCCA</td>
<td>TSID</td>
<td>8–15</td>
</tr>
<tr>
<td>HEMM</td>
<td>TSID, regardless of the access mechanism used</td>
<td>8–15</td>
</tr>
</tbody>
</table>

**Table 2.5:** *TID subfield values of the QoS Control Field [3]*

Ethernet User Priorities

<table>
<thead>
<tr>
<th>Priority</th>
<th>802.1p Priority</th>
<th>802.1p Designation</th>
<th>Access Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowest</td>
<td>1</td>
<td>BK</td>
<td>Background (AC_BK)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Spare</td>
<td>Background (AC_BK)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>BE</td>
<td>Best Effort (AC_BE)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>EE</td>
<td>Best Effort (AC_BE)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>CL</td>
<td>Video (AC_VI)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>VI</td>
<td>Video (AC_VI)</td>
</tr>
<tr>
<td>Highest</td>
<td>6</td>
<td>VO</td>
<td>Voice (AC_VO)</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>NC</td>
<td>Voice (AC_VO)</td>
</tr>
</tbody>
</table>

**Table 2.6:** *802.1p User Priority to 802.11e Access Category mapping*

The UP is an 802.1 traffic classification naming, which is why the 802.11 expects to receive traffic prioritization requests through UP designation.

Assigning a UP to a frame can be done directly in the MSDU during the MPDU preparation (see B.5.2), or through the MAC Sublayer Management Entity (MLME) when a Traffic Specification (TSPEC) is assigned to that frame.
The default value for this parameter is 0, which translates into the Best Effort Access Category for all traffic that did not require any specific prioritization. This means that Wi-Fi allows for two higher-than-normal and one lower-than-normal priority traffic set-ups.

**IP Differentiated Services Code Point**

The 6-bit Differentiated Services Code Point (DSCP) field [20], found in the header of all IP packets, is the layer 3 (network) labelling used to request QoS to the lower layers. The DSCP replaces the outdated 8-bit Type of Service (TOS) field.

The DSCP field allows for a wide range of classification types but it defines mainly 4 types of Per-Hop Behaviour (PHB) schemes:

**Default PHB:** in this case the value assigned to the DSCP field is ‘000000’. This is the default value that IP packets get when no traffic classification is requested from upper layers. The UP assigned to this DSCP value is: 0 - Best Effort

**Expedited Forwarding (EF) PHB:** used for low delay, low loss and low jitter traffic. Suitable for Voice, Video and other real time services. The DSCP field is typically set to ‘101110’. The UP assigned to this DSCP value is: 6 - Voice

**Assured Forwarding (AF) PHB:** is a matrix of possible values which goes from Class 1 (low priority) to Class 4 (high priority), and from low drop to high drop.

<table>
<thead>
<tr>
<th>Class Selector (CS) PHB:</th>
<th>Low Drop</th>
<th>Med Drop</th>
<th>High Drop</th>
<th>UP Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>AF11 (DSCP 10)</td>
<td>AF12 (DSCP 12)</td>
<td>AF13 (DSCP 14)</td>
<td>3 - Excellent Effort</td>
</tr>
<tr>
<td>Class 2</td>
<td>AF21 (DSCP 18)</td>
<td>AF22 (DSCP 20)</td>
<td>AF23 (DSCP 22)</td>
<td>4 - Controlled Load</td>
</tr>
<tr>
<td>Class 3</td>
<td>AF31 (DSCP 26)</td>
<td>AF32 (DSCP 28)</td>
<td>AF33 (DSCP 30)</td>
<td>5 - Video</td>
</tr>
<tr>
<td>Class 4</td>
<td>AF41 (DSCP 34)</td>
<td>AF42 (DSCP 36)</td>
<td>AF43 (DSCP 38)</td>
<td>6 - Voice</td>
</tr>
</tbody>
</table>

**Table 2.7: AF PHB possible values**

**Class Selector (CS) PHB:** these set of values for the DSCP field are used only to keep backwards compatibility with the TOS byte. The DSCP value in this case is in the form of ‘xxxx000’, where the ‘xxx’ define the 8 possible Precedence values in TOS. The UP assigned in this case matches the decimal value of the IP Precedence subfield, i.e if the DSCP value is ‘101000’ (CS5) the assigned UP is 5 - Video.

The values for the DSCP to UP mapping were extracted from [21].
The operating system

Typically at the top-most level we find the application layer, which interacts with the lower levels through the Operating System (OS).

In Windows, a set of API for network communication is given that allow an application to request a certain DSCP value in its transmissions. Unfortunately, after some network analysis, we saw that by default most applications such as Skype, YouTube, some VoIP software, and others get the same DSCP value, Best Effort or ‘000000’.

After further research we found out that by default, neither Linux-based nor Windows OSes apply DSCP values. To support DSCP values assignment, in both OSes, manual traffic classification policies must be created based on application, destination or destination port.

However, some special conditions apply where the default DSCP value of 0x0 might still be assigned to the frames regardless of the QoS policies created in the OS, e.g. when in a public network, the network controller may force the default PHB for all transmissions.

2.3.2 AIFS: Arbitration Interframe Space

The EDCAF introduces new contention times for each of the ACs. To do so, a different IFS as well as a different set of Contention Window (CW) values is assigned to each AC in order to prioritize the traffic by allowing it to gain access the medium before other lower traffic contention mechanisms expire. Typical values for 802.11g stations when using 802.11e improvements are shown in Table 2.8.

<table>
<thead>
<tr>
<th>Access Categories</th>
<th>CWmin</th>
<th>CWmax</th>
<th>AIFSN</th>
<th>Max TXOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background (AC_BK)</td>
<td>31</td>
<td>1023</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Best Effort (AC_BE)</td>
<td>31</td>
<td>1023</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Video (AC_VI)</td>
<td>15</td>
<td>31</td>
<td>2</td>
<td>3.008ms</td>
</tr>
<tr>
<td>Voice (AC_VO)</td>
<td>7</td>
<td>15</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 2.8: Access Categories

Despite of these typical values, there are two other aspects that may vary this table:

- The AIFSN[AC] value must be bigger or equal to 2 for QoS STAs and bigger or equal than 1 for QoS APs. This allows for an unbalanced Uplink (UL)-Downlink (DL) scenario.
The AP may update these values any time by sending a Beacon with the new table.

After taking these considerations into account, the contention process is the same as the one used in DCF except that now the backoff may have different minimum and maximum values depending on the AC and instead of DIFS (see figure 2.4), AIFS[AC] is used and calculated as:

\[
AIFS[AC] = AIFSN[AC] \times aSlotTime + aSIFSTime
\]

Regardless of the requested priority, some restrictions apply to certain frames when they are being transmitted:

- Management frames shall be sent using the AC_VO without being restricted by admission control procedures.
- BAR and BA control frames must be sent with the same QoS parameters as the corresponding QoS data frames.
- A PS-Poll control frame will be sent using the AC_BE in order to reduce the likelihood of collision following a Beacon frame.

### 2.3.3 TXOP: Transmission Opportunity

A Transmission Opportunity (TXOP) is a time period during which a station may transmit as many packets as it can in a burst separated by SIFS or RIFS (see 2.4.2). If a single packet is too big to fit in the assigned TXOP, the station may fragment the packet to send it in multiple TXOPs using the Wi-Fi fragmentation mechanism (B.2.3).

There are two ways to get a TXOP:

- When using **EDCAF**, the TXOP works in a similar way as the RTS/CTS mechanism (B.2.4), but in this case no additional frames are necessary to carry the NAV value. If the Wi-Fi MAC considers necessary to reserve the medium for a burst transmission, then it shall add the time during which the burst transmission is expected to last in the first frame that gains access to the medium.
- When using **HCCA**, the TXOP is assigned by the Hybrid Coordinator (HC). See B.4.

In either cases, the maximum time during which the medium may be reserved is given by the AP and may be different for each of the ACs.
2.3.4 EDCAF: Enhanced Distributed Channel Access Function

This access mechanism is based on a traffic prioritization where higher priority traffic has higher chances of being sent than the one with lower priority. This is done by using shorter Arbitration Interframe Space (AIFS) and CW values for higher priority packets. The contention free access in EDCA is provided by using TXOP intervals.

2.3.5 Reducing the overhead: The Block Ack mechanism

The 802.11e also defines a block acknowledgement mechanism which highly reduces the ACK overhead when packets are sent from one station to another. The main idea behind it is to positively or negatively acknowledge a set of received frames in a single transmission instead of sending an ACK for each of the received packets. In order to use this mechanism, an active Block Acknowledgement (BA) session is required which will allow a certain BA mechanism to be used between two stations.

![Representation of a Block Ack session procedure](image)

**Figure 2.5:** Representation of a Block Ack session procedure [3]

The type of BA being used for this session can be set during the establishment frame exchange of the session.

In 802.11e there are two different types of BA sequence defined when the ACK policy (B.2) is set to Block Acknowledgement:

**Immediate Block Ack:** the receiving station responds to the BAR frame SIFS after its reception with a BA frame in the same TXOP.

**Delayed Block Ack:** the receiving station responds to the BAR frame with an ordinary ACK to acknowledge the reception of the request in the same TXOP. Later, in another contention or TXOP, the receiving station sends the BA response frame and awaits an ACK to confirm the reception of the response.
Faster acknowledgement transmission

Besides aggregating multiple ACKs in a single frame, the BA and BA Request are allowed to be transmitted at a rate which is supported by both the transmitter and the receiver. If desired, this allows for the BA frame to be sent at the same rate as the data frames, thus occupying much less time the medium.

Regardless of the supported rates, the BA frame, if sent as a response to a BAR frame, must be sent at the same rate at which the BAR frame was received.

ACK Policies

The 802.11e introduces for the first time in the Wi-Fi standard the knowledge of differentiated ACK mechanisms depending on the frame type (see Table B.2). There are two service classes in order to request the MAC entity a certain ACK policy:

- QoSAck: the MSDU is transmitted through a QoS frame with the QoS Control Field set to Normal Ack or Block Ack.
- QoSNoAck: the MSDU is transmitted through a QoS frame with the QoS Control Field set to No Ack.

2.4 High throughput communications: The IEEE 802.11n amendment

2.4.1 Backwards compatibility

When the 802.11g version was released to share the band with existing 802.11b devices, it provided ways to ensure coexistence between legacy and successor devices. It is a similar case with the 802.11n which extends its protection mechanisms, while mainly utilizing RTS/CTS (see B.2.4), to ensure coexistence with 802.11b/g/a devices as well as to be able to communicate with these legacy devices while keeping some of the improvements brought by the 802.11n.

2.4.2 RIFS: Reduced Interframe Space

The 802.11n defines a new interframe spacing, the Reduced Interframe Space (RIFS), with a duration of 2 $\mu$s. It improves the performance by reducing the amount of dead time required between atomic OFDM transmissions. The use of this new IFS is restricted to HT-Greenfield transmissions, i.e. when communication is taking place between two 802.11n stations that support all greenfield features.
2.4.3 Frame aggregation

In the previous versions of Wi-Fi, every frame transmission has a significant amount of overhead (radio level headers + MAC frame fields + IFS + ACK) which at high PHY data rates results in a consumption of more bandwidth than the data payload of the frame.

As demonstrated in many white papers and studies [22], an increase in channel capacity (PHY throughput) does not translate in a direct increase of MAC throughput. This is mainly due to the overhead and the slow speeds at which the PLCP and MAC headers are transmitted which become the bottleneck no matter how fast we are transmitting the payload.

**Wi-Fi Overhead**

Some of the main reasons behind this overhead are:

- Headers: MAC header, Frame Check Sequence (FCS), PHY Header,
- Inter-Frame Spacing (IFS): SIFS, DIFS, EIFS and AIFS sometimes being longer than the Payload itself
- Back-off time
- ACK mechanisms
- RTS/CTS mechanism

![Overhead against Payload comparison under DCF operation w/o RTS/CTS](/image)

**Figure 2.6:** Overhead against Payload comparison under DCF operation w/o RTS/CTS [22]

The 802.11n amendment defines two types of frame aggregation mechanism to deal with this problem: the Aggregate MAC Service Data Unit (A-MSDU) and the Aggregate MAC Protocol Data Unit (A-MPDU).
A-MSDU: Aggregate MAC Service Data Unit

The A-MSDU mechanism is aimed to deal with very short upper layer packets. Since a common use of TCP/IP is assumed as part of the higher stack of protocols, a typical Maximum Transmission Unit (MTU) size of 1500 bytes needs to be taken into account. Small MSDUs, when the upper layer uses TCP/IP, are present not only due to the MTU, but we must also take into account the TCP ACKs which are sent like any other transmission in Wi-Fi using the normal contention mechanism.

A-MSDU is the exact opposite operation of fragmentation:

- Fragmentation divides a single MSDU into multiple MPDUs for robustness
- A-MSDU allows for multiple MSDUs to fit in a single MPDU for better performance

The new frame format introduced by the A-MSDU mechanism is further detailed in B.5.3.

If Block Ack is being used, the receiving station may deny or approve the transmission of A-MSDU frames during the active session by setting appropriately the BA Parameter Set when initiating the BA session.

A-MPDU: Aggregate MAC Protocol Data Unit

In this case, each upper layer frame is wrapped in a single 802.11n MAC frame, but the transmission of these is sent in a single contiguous transmission requiring the Block Ack mechanism. This may not be as efficient as an A-MSDU as it uses the MAC header, but it is more robust to channels with high error rates as each aggregated unit can be independently acknowledged, allowing smaller retransmission when needed.

This additional aggregation frame format is presented in B.5.4.

Both aggregation schemes can be combined together, in which case the use of fragmentation is deprecated.

The contents of an A-MPDU may vary, but there are some rules according to what information is being carried, according to which the MPDUs must be reordered or sent individually within the A-MPDU. One of these context examples is when the A-MPDU is carrying an ACK/Block Ack frame, where this must be situated at the start of the A-MPDU. Each A-MPDU is then carried in a PSDU.
2.4.4 Improved Block Ack mechanism

In Wi-Fi-n the Block Ack mechanisms introduced by the 802.11e are improved by adding new frame exchange sequences as well as reducing the overhead of some unnecessary properties by introducing new Block Ack frame formats.

The two new frame formats added by 802.11n are the Compressed BA, which disregards the use of fragmentation as it can not acknowledge independent fragments, and the Multi-TID BA which allows for MPDUs of different QoS types to be acknowledged in a single BA. Both frame formats are included in B.5.5.

In [4] is also explicitly stated that Multi-TID BA frames shall only be used inside a PSMP sequence.

The support of the Block Ack mechanism is mandatory in HT devices, while it was set as optional in 802.11e.

HT-Delayed Block Ack

The main improvement that the 802.11n brings to the delayed Block Ack mechanism is the possibility to use any of all four frame exchange sequences: BAR+ACK+BA+ACK, BAR+ACK+BA, BAR+BA+ACK, BAR+BA. With the only difference between the HT-Delayed and HT-immediate being that the BA frame may be sent in another TXOP.

2.4.5 The 802.11n PHY improvements

The 802.11n version of the standard introduces a variety of mechanisms such as MIMO and Channel bonding, besides creating a new set of PPDU frame formats in order to preserve backwards compatibility and provide the maximum data transfer rate when possible.

Another PHY improvement brought by the 802.11n is seen in how the OFDM signal is created. When using a single spatial stream, and a single 20 MHz channel, 802.11n may reach 65 Mbps of PHY rate, whereas 802.11a and 802.11g are limited to 54 Mbps. This is done mainly by the following improvements:

- **More data subcarriers**: 802.11n uses all 52 OFDM subcarriers as data subcarriers, whereas the 802.11g reserves 4 as pilot subcarriers.

- **Shorter Guard Interval (GI)**: in 802.11n a shorter GI, focusing on the possibility of smaller rooms and therefore shorter Delay Spread values, is introduced. In 802.11n if both stations agree, a shorter 400ns GI can be used, while the normal standardized GI is 800ns.

- **Higher encoding rate**: while both, 802.11g and 802.11n use 64-QAM as they’re highest modulation for each subcarrier, 802.11n introduces a higher coding rate of 5/6, while in 802.11g the maximum is 3/4.
2.4.6 MCS: Modulation and Coding Schemes

802.11n APs and stations need to negotiate some capabilities like the number of spatial streams and the channel width. Furthermore, these have to agree in the RF modulation, coding rate and guard time. The combination of these factors determines the PHY data rate, which goes from the minimum value of 6.5 Mbps to the maximum value of the 600 Mbps.

The 802.11n defines 77 possible permutations, known as Modulation and Coding Scheme (MCS), depending on the factors that determine the data rate. However, among all, the most used are the first 32 MCS as the other 35 are designed to improve the received SNR by using spatial diversity. Support for MCS 0 to 15 is mandatory for APs and support for MCS 0 to 7 is mandatory for all stations using 802.11n.
Chapter 3.

Baseline Results

So far we have described the Wi-Fi system. In this chapter we are going to show how these Wi-Fi systems behave in a variety of scenarios. This performance study is based on the results from the simulations of our Wi-Fi MAC model of 802.11g and 802.11e stations described in chapter 5.

Before getting started with the results for realistic scenario simulations, we have validated our Wi-Fi MAC model in the appendix A against a mathematical model to give a higher reliability to our results.

We have divided this chapter in 5 parts. First of all we are going to present the main problem that the Smart Aggregation mechanism tries to solve by comparing the aggregation mechanism with and without QoS traffic differentiation, i.e. for 802.11e vs 802.11g stations.

The rest of this chapter serves as an analysis of the individual Wi-Fi MAC features. In the second section of this chapter we see the effects of the RTS/CTS mechanism in large scale deployments. After that, we are going to present the results of adding QoS in the system, i.e. the improvements added by the 802.11e amendment. These results are presented for single network and for various networks working on the same channel of the 2.4 GHz ISM band.

Finally we are going to show the improvements that the 802.11n aggregation mechanism adds to the system, comparing it to the standard MPDU when traffic differentiation is given through different access categories.

3.1 The problem: the inefficient use of QoS in Wi-Fi

In this first section we want to expose the reason why we have designed a new mechanism, both combining aggregation and multi-QoS. Let us show some graphics
which picture the delay, Figure 3.1, and the goodput, Figure 3.2, in a system when using aggregation with and without access category distinction.

![Comparison between using A-MPDU with QoS and non QoS downlink only traffic](image)

**Figure 3.1:** Downlink delay when considering traffic only in DL and a 54 Mbps PHY rate.

As you can see in these figures, adding QoS to the system affects a lot the delay because only Voice packets, which are the ones with the highest priority, can keep a similar delay as the Legacy packets’ delay, the Video and Best Effort packets have a much bigger delay.

This huge increase in delay, as explained in the upcoming sections, is mainly because the 802.11e amendment, in order to prioritize Voice traffic, it seriously affects and downgrades the performance of the lower priority traffic.

The same behaviour is reflected in the next figure 3.2. The goodput achieved by the QoS packets, except for the Voice ones, is very low in comparison to the non-QoS packets of the 802.11g network.

Overall, as we will show in some of our results in the upcoming sections, the use of the 802.11e amendment may sometimes not only not bring any benefits to an 802.11g or 802.11n station, but it may actually deteriorate the performance of the station due to its much higher collision rate.
As mentioned earlier, this higher collision rate is mainly given due to the Voice AC, which not only contends the medium for a much shorter amount of time, but it actually keeps a really short Back-Off contention window even when multiple retransmissions have occurred, thus increasing even further the collision rate when multiple stations are trying to access the channel.

Additionally, the worse performance of 802.11e in comparison to the simpler 802.11g station is not the only problem. We consider it is unfair that we may end up in situations where the Best Effort traffic gets highly delayed, and sometimes even gets no throughput just to be able to send a few kbps of a higher priority class.

The ambition to fix and deal with these problems motivated us and led us to the creation of our Smart Aggregation mechanism explained in the next chapter. This mechanism, when compared to the graphics presented in this section, has proven to be really useful while keeping the QoS differentiation only in those cases where the system may actually benefit from it.

### 3.2 Wi-Fi Large Scale Deployments

In the Wi-Fi System we can find a lot of problems, like in other systems. One of them is the Hidden Node Problem, explained in the Appendix B.1.2, that is more...
likely to appear when more than one network is present in the scenario and then user stations and access points may not be able to listen to each other’s transmissions. Therefore collisions due to this lack of medium sensing appear.

In order to check this type of deployments, we have analyzed different scenarios such as the ones Figure 3.3 shows:

![Simulated scenarios, varying the deployment ratio and the number of users.](image)

Figure 3.3: Simulated scenarios, varying the deployment ratio and the number of users.

The parameters we have used are shown in Table 3.1. These two packet size values have been chosen because 1500 bytes is the typical ethernet MTU and 2304 bytes is the maximum MSDU size from the Standard [3].

The deployment ratio is a parameter that allows us to randomize the presence or absence of a network in an apartment. This means that if a value of 1 is chosen, all cells will contain a network, whereas if a value of 0.5 is selected, there is a 50% chance of having a network in a particular cell.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value/State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of users</td>
<td>1, 2, 5, 7 and 10.</td>
</tr>
<tr>
<td>Deployment ratio</td>
<td>0.2, 0.4, 0.6, 0.8 and 1.</td>
</tr>
<tr>
<td>RTS/CTS mechanism</td>
<td>Activated or deactivated.</td>
</tr>
<tr>
<td>Packet size</td>
<td>1500 and 2304 bytes.</td>
</tr>
</tbody>
</table>

Table 3.1: Parameters used in Large Scale deployment results.

In the upcoming Figures 3.4 and 3.5 we can see some of the results that emphasize on the comparison of large deployments comparing the collision rate and the goodput when RTS/CTS is being used and when it’s not.
Figure 3.4 shows the collision rate of the system depending on the packet size and when the deployment ratio is set to 1. As you can see, the results between the two graphs with the RTS/CTS mechanism deactivated are almost the same. This behaviour is due to the large number of competing stations and also because the scenario is very big, so you will be able to see a lot of collisions regardless of the packet size. On the other hand, you can see a difference when the RTS/CTS mechanism is activated. In that case, the larger the packet is, the higher the collision rate will be.
The goodput results presented here are obtained for two different deployment ratios, 0.2 and 1. We have chosen these two values because we are analyzing a large scale deployment scenario, so the most significative deployment ratio value is 1, but we have also added some results with the smaller value, 0.2, to check the trend when the channel is not totally overcrowded.

The trend of the curves due to the scenario configuration, i.e., 2x4, 3x4 and 4x4, is an expected behaviour because the more networks that take part in the simulation,
the more stations will take in, therefore the performance you will get will be worse. The same behaviour is caused by incrementing the number of stations.

One interesting observation from figure 3.4 is to notice that the Collision Rate curves for the 4x4 and 3x4 scenarios when RTS/CTS is disabled are almost the same. This behaviour is due to the fact that when the network is already overcrowded, the number of stations that gain the medium tends to stabilize regardless of how many neighbouring networks we add, which means that there is always a set of stations in each network that has not been able to transmit in this simulation, and therefore their collision rate remains 0. Basically you can’t collide if you don’t transmit.

Figure 3.5 shows the downlink goodput depending on the deployment ratio with a fixed packet size of 1500 bytes. With one and two users the goodput is lower using the RTS/CTS mechanism because you have to send the corresponding RTS and CTS frames, so you are wasting part of your medium time sending control frames. But when the number of stations is bigger, with the RTS/CTS mechanism you are preventing collisions, so you receive more data and your goodput increases.

The difference of goodput between both graphics is caused by the different deployment ratios. If it is 0.2 you have less networks, hence less stations. Thereby you can achieve more goodput.

Like in the previous figure, the trend due to the scenario configuration is caused by the same reason as explained before, the more networks you have, the more stations are in the system, so the achieved goodput is lower.

The last result, seen in Figure 3.6, shows the influence of the deployment ratio. The graphic displays, for a 4 by 4 scenario with 5 users, how the goodput goes down varying the deployment ratio.

When the deployment ratio is 1, it means that we have 16 networks with 5 users each one of them, so in total 80 users. So it seems reasonable to have a lower goodput for higher deployment ratios.

Finally, before moving to the next section, it is important to point out that with these graphics we have understood that the measurements of a single metric are not enough to justify the use of a mechanism.

For instance, by analyzing figure 3.4, we can say that the use of RTS/CTS is a must in large scale deployments where the influence of neighbouring networks is likely to generate the hidden node problem (B.1.2). But if we had only analyzed the goodput results from figure 3.5, we would say that the use of RTS/CTS does not bring any impressive benefits to the system.
3.3 QoS single network performance

The next step after validating the 802.11g single and multiple network results was to check the performance of the 802.11e amendment on top of the previously validated networks. We are going to start our analysis with a 1 by 1 scenario to finish it with the analysis of a group of networks operating on the same radio frequency.

In this section we are going to analyze the results of an isolated network where all stations are using traffic differentiation. The set of parameters used for these results are shown in Table 3.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value/State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wi-Fi version</td>
<td>802.11g with 802.11e</td>
</tr>
<tr>
<td>Number of users</td>
<td>1 to 19</td>
</tr>
<tr>
<td>Deployment ratio</td>
<td>1</td>
</tr>
<tr>
<td>RTS/CTS mechanism</td>
<td>Deactivated</td>
</tr>
<tr>
<td>Type of buffer</td>
<td>Unlimited or limited (512 kBytes)</td>
</tr>
<tr>
<td>Traffic balance</td>
<td>Only DL, only UL or DL+UL</td>
</tr>
</tbody>
</table>

Table 3.2: Parameters used to compare 802.11g stations with and without the 802.11e amendment.
We are simulating 802.11e networks based on two different types of traffic:

- **Fixed Traffic Scenario:** where Voice, Video and Best Effort traffic is generated at the application layer and each of the flows are set to generate 1024 kbps of traffic with 1280 bytes packets every 10 ms. Which means that each application flow will be generating a total of 3 Mbps of traffic towards its destination.

- **Realistic QoS Scenario:** where the application flow specifications were taken from [23] and are shown in table 3.3.

<table>
<thead>
<tr>
<th>Access Category</th>
<th>Packet size</th>
<th>Packet interval</th>
<th>Bit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice</td>
<td>160 bytes</td>
<td>20 ms</td>
<td>64 kbps</td>
</tr>
<tr>
<td>Video</td>
<td>1280 bytes</td>
<td>10 ms</td>
<td>1024 kbps</td>
</tr>
<tr>
<td>Best effort</td>
<td>1500 bytes</td>
<td>12.5 ms</td>
<td>960 kbps</td>
</tr>
</tbody>
</table>

**Table 3.3:** *Realistic traffic scenario specifications.*

The different types of traffic balancing options that we have simulated refer to:

- **Only DL:** data will be generated at the application layer of the AP only and there will be as many flows, of the selected traffic scenario, as STAs in that network.

- **Only UL:** each STA will have a traffic flow, based on the type of selected traffic scenario, that has the AP as its destination, but there will be no traffic in DL.

- **DL+UL:** is the case where both of the previous traffic specifications apply. So the UL/DL offered load ratio will be the same, i.e 1/1, on a per network basis and on a per user basis.

To study the performance of the 802.11e amendment, we will focus on the delay, the goodput and the packet drop. We define the delay as the amount of time since the packet arrives to the buffer until it has been acknowledged successfully.

Regarding the packet drop, this a metric that will of course be meaningless when simulating the unlimited buffer case, as no packet drop may occur due to the buffer size.
We are going to start the analysis of figure 3.7 with the 802.11g case, where all type of packets are treated as legacy because you cannot differentiate between ACs.

As you can see, the goodput for each access category starts with the generation bit rate defined in Table 3.3, and it falls down as the number of stations increases. It is reasonable that when the number of contending stations is very big, the goodput is almost 0 because the probability of gaining the medium is low, and we have already seen in previous results that the throughput does not linearly decrease with an increase of the number of contending stations due to an increasing collision rate.

If we have a look at the 802.11e case, we can see that for one station the behaviour is the same as for the 802.11g case. But when the number of stations starts increasing, the graph shows the prioritization of access categories. Voice packets get always a goodput higher than the one obtained from the 802.11g networks.

Video packets’ goodput in 802.11e has the same behaviour as the one for Video packets in 802.11g almost all the time until a network saturation threshold is reached. On the other hand, Best Effort packets always get a lower goodput, compared to the 802.11g Best Effort goodput, as this loss in goodput is precisely the one that allows higher priority traffic to get higher goodput.
Figure 3.8: Uplink goodput with the same type of traffic in both UL and DL.

Figure 3.8 shows the uplink goodput when using the fixed traffic scenario, where 3 Mbps of traffic is generated in DL and UL per STA in the network.

As before, first we are going to discuss the 802.11g case. Now all type of traffics have the same generation rate, so all the curves start at the same point, and because all of them are treated as legacy packets, the three curves are overlapping. Like in the previous analysis, as the number of stations increases, the goodput decreases.

With this graph you can clearly see the packet prioritization for the 802.11e case. The three curves, Voice, Video and Best Effort, are sharply differentiated. This is the first graph were we can see that crowded networks may not always benefit from the use of QoS differentiation. In fact, due to the faster Voice AC contention machines, the number of collisions is way higher in 802.11e networks against the 802.11g networks.

So thanks to this graph we have found the first focus point to fight with our Smart Aggregation mechanism and is exactly the aspect that is improved in section 6.1.

After analyzing the goodput, we now move on our focus to the delay results. It is important to notice that, when analyzing the delay, there is a big difference between the only UL and only DL traffic load options. If there is only DL traffic, the AP is the only one which will transmit and it does not need to compete with anyone, so the high delay values are mainly given by the time a packet stays in the buffer due to the AP bottleneck. On the other hand, if there is only UL traffic, the AP is not transmitting but the stations are, and they have to compete against each other to access the medium. Therefore that delay result is also affected by the amount of time spent in collisions, which are non-existent when there is only DL traffic.
In Figure 3.9 you can see the downlink delay when using the same type of traffic, a limited buffer and when there is only traffic in the DL.

The graph is quite self-explainable. When looking at the results from the 802.11g, you can see the three curves are the same, because you only have one buffer with legacy packets of the same size, so the packets will be accumulated in the buffer always in the same order and all of them will have the same delay. In these curves we can also see a saturation of the delay due to the buffer limit, because once the AP buffer is full, it starts loosing packets, and the delay stabilizes.

Then, when looking at the curves from the 802.11e, you can clearly see the priority staircase. Voice traffic gets the lowest delay of all the ACs and it is also lower than the 802.11g version until the number of stations is big enough to congest the system. For Video traffic, the delay gets bigger than the one achieved from legacy well before the Voice traffic does.

Finally, Best Effort traffic always gets the biggest delay because the 802.11e amendment does Voice and Video prioritization in exchange for the lowering of the quality of Best Effort and background communications.
In Figure 3.10 you can see the uplink delay when using different types of traffic, a limited buffer and when there is traffic in DL and in UL.

This graph is very interesting because it shows a realistic scenario. In this case we have traffic in both ways, DL and UL, and a limited buffer of 512 kBytes. Legacy traffic always gets the same delay, but when you have a system with QoS, you can see Voice traffic is always sent with a very low delay in comparison to the other ACs but also when comparing with the 802.11g delay.

For Video packets there is a threshold from which the delay is bigger than the legacy curve. As seen before, Best Effort is the most affected and it has always a big delay to produce the prioritization of Video and Voice traffic.

So this is a first result, with a very realistic simulation scenario (crowded isolated network, with limited buffer and realistic traffic profiles), where the use of 802.11e actually results in an improved communication quality for the end user. This is specifically seen in the Voice communications with 20 users, where the delay is still under 200 ms, which is close to the maximum delay value suggested by the ITU-T [24] in order to neglect the delay perceived in a Voice call.

The last result we are interested in is the packet drop. Figure 3.11 shows the number of MSDUs dropped when using different type of traffic, a limited buffer of 512 kBytes and when there is traffic in DL and in UL.
Performance improvement strategies for current and next generation Wi-Fi systems

As we can see, since 802.11e offers traffic classification inside the same buffer (i.e. with the same combined capacity as the 802.11g, see 5.4.1) and prioritizes the Voice traffic inside this buffer, the number of Voice packets dropped in our 802.11e stations is almost null, and the rest of the Access categories get most of the times a lower number of packets dropped than in the 802.11g case.

So the packet dropping rate does also show benefits when using the QoS amendment of the Wi-Fi standard on top of 802.11g networks.

But these are results for isolated networks, so let’s move to the study of how the QoS classification performs in presence of multiple networks.

### 3.4 QoS neighbouring networks performance

In this section we are going to analyze a 2 by 2 scenario comparing and simulating both 802.11g and 802.11e stations, to see the improvements of adding QoS when you have multiple networks.

Just like we did before, the generation bit rate of each access category is set according to Table 3.3, and the results are obtained without using the RTS/CTS mechanism, an unlimited buffer and up to 11 stations.

Figure 3.12 shows the downlink goodput when the stations have a realistic QoS type of traffic and there is traffic in both uplink and downlink.
As you can see, this figure follows a similar trend as the one seen in Figure 3.7, but in this case the goodput drop is more drastically affected, and it tends to 0 before, i.e., in the 1x1 system the goodput tends to 0 when there are 14 stations and in the 2x2 case you can see this behaviour when there are 5 stations because there are now 4 networks sharing the same channel. So there are actually number of stations \cdot number of networks in the same channel, which in this case translates to almost 20 stations.

Figure 3.12: Downlink goodput with different type of traffics and unlimited buffer considering both DL and UL traffics in a 2 by 2 system.

Figure 3.13: Uplink delay with different type of traffics and unlimited buffer considering both DL and UL traffics in a 2 by 2 scenario.
Figure 3.13 shows the uplink delay under the same conditions as the previous graphic, where the stations have a realistic QoS traffic load and there is traffic in both uplink and downlink.

When comparing with figure 3.10, we can see that the trend is pretty much the same, except for the fact that, the presence of many more stations and the higher possibility of having the hidden problem, in the $2 \times 2$ scenario the delay dramatically increases even for a small number of users. In either case, you can see that the correct prioritization is done, and the Voice packets always get the smallest delay, also smaller than legacy traffic, the Video ones have less delay than legacy until a certain saturation threshold is met and the Best Effort ones always have the biggest delay.

It is interesting to notice that the Best Effort traffic reaches in highly loaded channels the delay value of 3 seconds. This particular value is given by the total simulation time that we have been using for all our simulations. So if an average delay of 3 seconds is achieved in a 3 seconds simulation for the Best Effort traffic, this is likely to mean that in average all Best Effort packets are held in the buffers. We can quickly notice this effect in the goodput graphs where the Best Effort traffic gets 0 kbps, i.e. no packets of this Access Category are sent.

After these two sections looking at the results from the QoS feature, and seeing that they perform pretty much as expected, we are able to say that we have the 802.11e amendment working in the simulator, so we can continue adding features on top of it.

### 3.5 Adding Aggregation to the System

In this section we want to show the improvements added by the aggregation mechanism defined in the 802.11n amendment [4]. In order to do that properly, we are comparing the A-MPDU versus the normal MPDU, when using traffic differentiation by using different ACs. The parameters used for the aggregation results are shown in Table 3.4 and the traffic characteristics remain the same as the ones we used for the 802.11e stations, shown in Table 3.3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value/State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wi-Fi version</td>
<td>802.11g with 802.11e and A-MPDU</td>
</tr>
<tr>
<td>Number of users</td>
<td>1 to 19</td>
</tr>
<tr>
<td>Deployment ratio</td>
<td>1</td>
</tr>
<tr>
<td>RTS/CTS mechanism</td>
<td>Deactivated</td>
</tr>
<tr>
<td>Type of buffer</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Traffic distribution</td>
<td>Only DL, only UL or DL+UL</td>
</tr>
<tr>
<td>Scenario</td>
<td>$1 \times 1$</td>
</tr>
</tbody>
</table>

Table 3.4: Simulation scenario parameters for A-MPDU vs MPDU comparison.
We are going to start analyzing the goodput presented in Figure 3.14.

At a first glance we can see that the aggregation mechanism is efficient, because it achieves higher goodput when the number of stations is increased. The trend is very similar to the MPDU one but using the aggregation mechanism you can get more goodput with the same number of stations. It is interesting to notice that the Voice
goodput is almost constant when using A-MPDU in a realistic QoS traffic scenario because in this particular case, the aggregation mechanism highly reduces the MAC overhead for this access category with very small packets.

Figure 3.15 shows the downlink delay when there is traffic in both ways, DL and UL. Looking at this graphic we can remark a couple of things. The first one and the most obvious is the reduction of the delay using an A-MPDU instead of transmitting individual MPDUs. All the access categories get a lower delay except the Best Effort traffic when a certain number of stations is reached. Now we have arrived at the second important issue of the graphic. When the number of stations increases, the probability of accessing the medium and therefore to transmit is lower, and if you are the access category which has the lowest priority you will be more influenced with this situation as we explained in the previous section.

After simulating and analyzing all these traffic scenarios, and realizing that there are some cases where the 802.11e brings no benefits, and actually worsens the communication, in the next section we decided to compare the 802.11e A-MPDU against the 802.11g A-MPDU.
Chapter 4.

Smart Aggregation

In this chapter we will describe the Smart Aggregation mechanism, which we have designed to improve mainly the MAC layer efficiency of the Wi-Fi system described in chapter 2 while keeping intact the traffic differentiation feature brought by the 802.11e amendment, as well as to fight the QoS performance problem presented at the end of chapter 3.

4.1 State of the art

Nowadays the improvement of wireless systems is mainly focused on achieving a high throughput and supporting different types of applications such as Voice applications like Skype, Video streaming or online gaming. This can be done both at the PHY level or at the medium access level. In this thesis we are going to focus on the MAC level, mostly concentrating our study and research on aggregation based mechanisms and QoS support improvements.

The research field is widely spread around the world, and a lot of people work hard on it every day, and since the 802.11 standard is quite old, most of the research achieved so far that tries to improve the 802.11n has based its work on exploiting multi-QoS features, adaptive aggregation mechanisms, and multi-user approaches.

The first feature that Smart Aggregation adds is that it allows the aggregation of Multi-QoS packets at the MPDU level, i.e., it is possible to aggregate packets of different access category, and TID, in a single aggregated frame thus resulting in the new SA-MPDU frame.

One of the approaches to increase the MAC throughput, that tries to solve the multi-user bottleneck present in all Wi-Fi networks at the AP, can be seen in [25] where a new power saving scheme is proposed, named Power Save Multi-Poll (PSMP) (see C.1). In this mechanism, the AP is the only one that can start a PSMP period, although both UL and DL traffic are allowed after the AP has sent some scheduling
packets which will determine the offset time after which a certain station may start transmitting. This scheduling and frame exchange is done with a set of specific frame formats to initiate and manage the frame exchange sequence. They are using aggregation to send the data and also to acknowledge it using Block Ack, therefore they are reducing the overall overhead but also adding some due to the transmission of special frames for the mechanism.

This PSMP mechanism was later adopted by the 802.11n and is the only context during which the standard allows the use of Multi-QoS MPDU Aggregation and Multi-TID Block Ack. Compared to our Smart Aggregation, their mechanism is only useful for the access point and can only be used inside a PSMP period which implies more overhead.

Despite of the Multi-QoS Aggregation in the Smart Aggregation mechanism, it still gives the highest priority to Voice packets, and this is what a lot of studies try to preserve. For instance, in [26] they apply an adaptive aggregation to each of the traffic priorities on an A-MSDU level in combination with a packet scheduler in order to improve the throughput and the delay of Voice packets. As a metric, they use their own mechanism which is based on the average number of retransmissions per node to successfully deliver a packet over a link.

Therefore, with their solution you can improve the throughput and the delay for Voice packets, but with our mechanism you are still improving these performance indicators, while also improving the transmission of packets from the rest of access categories.

Another interesting feature Smart Aggregation adds is that, since it uses an improved A-MSDU frame, it allows the possibility to retransmit, and acknowledge, not only single MPDUs but also single MSDUs from inside an SA-MSDU using an improved Block Ack mechanism and frame set, as it is explained later in this chapter.

The Standard [4] does not support retransmission of individual MSDUs from an A-MSDU, but you can still retransmit the whole A-MSDU, and this is the idea applied in [27] and [28].

In [27] they work both with fragmentation and aggregation, and they have designed their own frames which, based on the throughput and two delay metrics, allow you to get the optimal frame size and to retransmit corrupted fragments in a single A-MPDU.

In [28] they also work with fragmentation and aggregation but they try to get the maximum throughput under a certain BER depending on the delay limit. The striking idea of this paper is that it aggregates not only at an A-MPDU level but also at a PHY level, i.e., it aggregates PSDUs, which means that in this way you can assign a different transmission rate to each PSDU and you can also afford a multi-user mechanism.
Performance improvement strategies for current and next generation Wi-Fi systems

The same idea can be seen in [29], where it also proposes a contention-free polling based period in combination with a multi-user PSDU aggregation mechanism. However, we have rejected this idea because in [18], the author explains that, if a multi-user approach is being taken, the required overhead due to delimiters in this aggregation on a physical level to separate each PSDU is likely to be longer than the RIFS separation presented in 802.11n. Also this delimiter would be less robust in front of errors from the channel since a single error in these separation fields would disable the rest of the A-PSDU from being de-aggregated.

Smart aggregation reduces the overhead, not only for the MAC Header but also for the Block Ack mechanism used. To achieve that, new frames are used - see section 4.4 - in combination with a different Block Ack mechanism from the Standard [4] is used - see section 4.4.3.

There are other publications that try to find the optimal aggregated and normal frame size. In [30] an adaptive aggregation scheme is adopted depending on channel conditions, the retry limit and the packet loss rate. From the Smart Aggregation point of view, the aggregation mechanism will be decided by the Smart Aggregation Controller (SAC) - see section 4.3 - based on the defined criteria and according to the Smart Aggregation Policies.

To the best of the authors’ knowledge, ours is the first attempt to create an adaptive PSDU preparation scheme, which does not try to individually and independently adapt a single MAC feature, such as the A-MPDU, but ours takes into account multiple MAC features and sees the set-up of these as a single block, what we have named the Smart Aggregation Policy.

4.2 Smart Aggregation Overview

The demand for high throughput and the reliable use of new applications like VoIP from the Wi-Fi users requires the implementation of new mechanisms to exploit all the possible resources. There are three options to achieve that, to improve the medium access level, the physical level or both at the same time.

Smart Aggregation aims to increase the throughput and reduce the delay. There are several ways to do that, two of which are reducing the overhead caused by the MAC header and aggregating various frames at the same time, only having to contend for the medium once to send a burst of packets.

The set of tools that our mechanism uses are:

- New frames at the MAC protocol level
- New frames at the MAC service level
- Also new frames to acknowledge the packets.
Additionally, it also changes the behaviour of how to treat this type of frames from the Standard [4].

But the reason why the Smart Aggregation is a really powerful mechanism is because by using it you add improvements to the system while even if you are in the situation where the mechanism delivers its worst performance, this performance is close to the performance of the original standard [3].

4.3 Smart Aggregation Controller

In this section we are going to present the Smart Aggregation Controller, shown in Figure 4.1, which will be the core of the mechanism. In order to decide which type of configuration will be applied to the packet being sent, it will toggle what MAC features are active and which aren’t based on the Wi-Fi MAC Measurements block. This last block stores and updates the criteria, which are explained in section 4.3.1.

![Smart Aggregation Controller Diagram](image)

**Figure 4.1: Smart Aggregation Controller**

4.3.1 Decision criteria

The Wi-Fi System has defined some mechanisms to be used, like RTS/CTS, fragmentation or aggregation of MSDUs and MPDUs, and now we are adding a new one, the Smart Aggregation. As mentioned earlier, a Smart Aggregation Controller is provided in order to decide which one or which combination of MAC features of
the Wi-Fi system have to be used. So in order for the SAC to work, the next step is to define the criteria included in the Wi-Fi MAC Measurements block to decide which mechanism or mechanisms are the optimal to be used in every moment.

These criteria are defined depending on what you can get from your specific system, i.e., from the parameters you can get from the simulation. In our case we have a bunch of parameters that can be combined together in order to get interesting metrics. The following criteria are defined per receiving station or Destination Address (DA), which means that each transmitting node has some statistics regarding the link quality for each of its destinations.

**Collision Rate:** this first criterion is based on the following ratio:

\[
\frac{\text{number of retransmissions}}{\text{number of successful transmissions}} \quad (4.1)
\]

The *number of successful transmissions* can be obtained through the number of acknowledgements received. The higher this rate, the shorter packets we must send. Therefore the SAC will use a less aggressive policy by making the transmission more robust.

Alongside this first criterion, which is commonly used to discuss the adaptiveness of the RTS/CTS mechanism, we have defined two more criteria that we found to be very useful when A-MPDU and SA-MPDU frames are being generated. Despite of a deep search throughout our sources, we were unable to find these criteria in any other publication.

**BRQ: Block Ack Response Quality:** we have defined this metric with the following ratio:

\[
\frac{\text{number of positive acknowledgements in a BA frame}}{\text{number of MSDUs being acknowledged in that Block Ack}} \quad (4.2)
\]

We would use this criterion to know how many packets inside a burst transmission or inside an aggregated frame got corrupted. This parameter would be updated on a per user and per transmission basis, being an instantaneous parameter rather than an averaged one. Therefore, if the Block Ack Response Quality (BRQ) decreases we may reduce the maximum aggregation size or even deactivate the aggregation mechanism if necessary.

**BRQ[AC]:** this is another parameter related to the previous one but taking into account the number of acknowledged MSDUs on a per Access Category basis:

\[
\frac{\text{number of positive acknowledgements for that AC in a BA frame}}{\text{number of MSDUs being acknowledged in that Block Ack}} \quad (4.3)
\]
This parameter would allow us to independently add more robustness to each AC, and/or stop aggregating certain types of AC when sending Multi-QoS SA-MPDU frames. We are going to have four of them, one per each AC. Alike the previous criterion, if the Per QoS Link Quality [AC] decreases below a certain threshold, rather than affecting the overall aggregation size, we may choose to disable Multi-QoS Aggregation.

All of the previous criteria work on a per transmission basis. The idea behind our mechanism is that the reaction to the changes of these criteria shall take effect when the value crosses a certain threshold. This threshold, would be defined separately for each MAC feature based on the results that we obtain.

4.3.2 Smart Aggregation Policies

Once we have defined the Smart Aggregation criteria, we define a Smart Aggregation Policy as the set of Wi-Fi MAC features’ configurations which may be used when a certain set of Smart Aggregation criteria change.

An inside look of the SAC is shown in figure 4.2 where the set of criteria is formed with the 3 criteria defined earlier.

![Figure 4.2: Policy decision](image)

Each mechanism depends on a certain subset of these criteria, i.e, not every mechanism depends on all the defined criteria, e.g. the criterion defined in equation 4.3 will only affect the Multi-QoS mechanism.

This set of criteria will be updated after a short period of time and depending on the new values, the current set of activated mechanisms will or will not continue toggled. The decision to keep activated or deactivate a certain mechanism will depend on a certain threshold, one per each mechanism. At the begging, the threshold will be set to a defaulted value, which means that all three, of the mechanisms shall be used in its maximum performance if possible.
Smart Aggregation Policy selection mechanism

In a similar way as the rest of the mechanisms in the IEEE 802.11 standard [3], we have assumed a pre-negotiation of the possible MAC features supported by both, the transmitting and the receiving node, upon which the Smart Aggregation Controller will be able to use more or less of these to improve the MAC performance. In a worst case scenario, where no MAC feature offered within the Smart Aggregation environment is supported, our improved stations will perform just like any other IEEE 802.11e station.

As an example of how the mechanism adaptively reacts, let’s consider two stations that support SA-MPDU Multi-QoS. In this case, if at the transmitting node, the BRQ[AC] for Best Effort falls below a certain threshold, we shall limit the number of Best Effort packets in our SA-MPDU frames. If this indicator keeps falling, we shall send Best Effort data separately without affecting our Multi-QoS frames. If, after a long period of time, this indicator does not fall any further, the SAC recovery mechanism, shall re-enable this mechanism for this certain type of AC.

Additionally, let’s take the scenario where the Collision Rate, as expressed in equation 4.1, grows above a threshold, we should then shorten the maximum SA-MPDU size, thus avoiding the use of the RTS/CTS mechanism. Once the SA-MPDU size has been changed, the Collision Rate is expected to go down, but in case that does not happen, the SAC shall lower again the SA-MPDU size, and repeat the operation until the Aggregation mechanism gets completely disabled and MPDUs are no more aggregated.

Just like with the BRQ, if the Collision Rate stays below the threshold for a long enough period of time, the SAC may choose to enable again and restore the original SA-MPDU size to see if the channel conditions have improved.

4.4 Smart Aggregated Frames

In this section we are going to analyze and compare the relevant frames from the Standard [4] with the new Smart Frames.

4.4.1 SA-MSDU

The first Smart Frame is the SA-MSDU, present at the service level. Its smartness is given by adding the possibility to retransmit each individual MSDU and also by the possibility to add different types of protection to each of them.

The A-MSDU subframe format given by the IEEE 802.11n amendment [4] can be seen in Figure 4.3. The configuration and size limitations for the A-MSDU frame format are further explained in B.5.3.
Each A-MSDU is carried inside an MPDU’s frame body field. The A-MSDU sub-frame header includes the DA and Source Address (SA) subfields that are already present in the header of the MPDU frame, as shown in Figure 4.4.

Since the DA and the SA fields are then included twice per MSDU, the SA-MSDU changes the SA subfield to take advantage of these bytes to add some functionalities. We have chosen to modify the SA field, as we know that the transmitting node will remain the same for all of the aggregated MSDUs, so it is a field that offers no additional entropy to the transmission.

This modified subfield allows the SA-MSDU Subframe header to be long or compressed for the rest of the aggregated MSDUs, depending on the position of the MSDU inside the SA-MSDU. Then, if the frame is the first one being aggregated it is going to use the long header. In the rest of the cases it is going to use the compressed one if this SA-MSDU subframe header is the same as in the previous MSDU.

The general SA-MSDU subframe format is shown in the Figure 4.5, where the lengths of the fields are expressed in bytes. As we can see, the length of the new subfield varies from 1 to 6 bytes. In this way, only the first aggregated frame will cause the same overhead as the Standard one but the other ones will reduce that.
The Long SA-MSDU Subframe header is shown in Figure 4.6.

<table>
<thead>
<tr>
<th>DA</th>
<th>C</th>
<th>Required ACK</th>
<th>MSDU no</th>
<th>CRC Length</th>
<th>CRC Type</th>
<th>Reserved</th>
<th>CRC per MSDU</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>b: 48</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>32</td>
<td>16</td>
</tr>
</tbody>
</table>

**Figure 4.6: Long SA-MSDU subheader frame format**

Where the different subfields are:

- **C**: is the concatenation bit. If it is set to 1, it indicates that the next aggregated MSDU has the same SA-information as the previous one, i.e., the Required ACK, the CRC Length and the CRC Type, and the same DA as the previous one, and so, a smaller bit pattern should be expected for the next MSDU.

  If it is set to 0, it indicates that that SA-MSDU is the only one with this Long header, and the Receiving node should expect the next MSDU to carry its own Long SA-MSDU subframe header.

- **MSDU no**: this subfield is a 4-bit subfield like the fragment number, to allow the possibility to retransmit only the MSDU that has been received incorrectly.

- **Required ACK**: when this subfield is set to 1 all the MSDUs within the SA-MSDU have to be acknowledged, i.e., the Smart Block Ack mechanism will use the long bitmap. If it is set to 0 it means that the Smart Block Ack Frame may use the short bitmap.

- **CRC Length**: to indicate the size of the CRC per MSDU field. Since we have decided to allow a possible CRC size that can go from 0 to 4 bytes, this field will encode the CRC size less one, so if the CRC size is 2 bytes, this field shall indicate 01b.

- **CRC Type**: this subfield is used to indicate the type of CRC protection that is being applied to this set of MSDUs. This field may indicate one of the following four possibilities:

<table>
<thead>
<tr>
<th>CRC Type</th>
<th>Protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>CRC is disabled</td>
</tr>
<tr>
<td>01</td>
<td>Per MSDU</td>
</tr>
<tr>
<td>10</td>
<td>Reserved / Not used</td>
</tr>
<tr>
<td>11</td>
<td>Per concatenation / Per set of MSDUs using the same Long subheader</td>
</tr>
</tbody>
</table>

**Table 4.1: CRC Type**
Performance improvement strategies for current and next generation Wi-Fi systems

Aalborg University
RATE Section

- **CRC per MSDU**: its useful length is variable, from 0 to 4 bytes, depending on the CRC Type, but the size of the field is always 32 bits, and if the useful length, as indicated by the CRC Length subfield, is smaller than 4 bytes, the rest of the bits shall be set to 1.

Alternatively, as mentioned earlier, if the Concatenation bit of the previous MSDU was set to 1, the compressed SA-MSDU Subframe header is used, see Figure 4.7.

<table>
<thead>
<tr>
<th>C</th>
<th>MSDU no</th>
<th>Reserved</th>
<th>Length</th>
<th>CRC per MSDU</th>
</tr>
</thead>
<tbody>
<tr>
<td>b:</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>16</td>
</tr>
</tbody>
</table>

**Figure 4.7**: Compressed SA-MSDU subheader frame format

The meaning of these subfields is the same as for the Long SA-MSDU Subframe header, except that in this case, there is no DA field and the CRC per MSDU subfield uses the information from the CRC Type subfield of the first frame of its aggregation, i.e., the one with the long header. This means that, when using the Short SA-MSDU subheader, its size depends on the CRC per MSDU subfield, where if no CRC is being used for that MSDU, the total subheader size for that aggregated MSDU is 2 bytes plus the required padding, instead of the original 14 bytes plus padding.

As a comparison, if we take a TCP ACK frame, which is 40 bytes in length [31], this new Short subheader means a total overhead of just 5%, whereas the A-MSDU subframe proposed in the standard, or our own SA-MSDU Long subheader, mean a 35% of overhead.

At this service level, the deaggregation will be different than in legacy, or standard-based, devices. This is because Smart Aggregation changes the A-MSDU frames and it adds new subfields in order to support this extra robustness. However, thanks to all the new and cheaper technology, we expect the processing time to be really small and simple.

### 4.4.2 SA-MPDU

The Smart Aggregation mechanism does not change the frame format of the A-MPDU but improves its behaviour. The smartness in this type of frame, as mentioned earlier, resides in the fact that our mechanism defines this frame as a Multi-QoS frame, allowing for multiple Access Categories to be aggregated all together.

The A-MPDU subframe format given by the IEEE 802.11n amendment [3] can be seen in Figure 4.8 and its limitations are further detailed in B.5.4.
The amendment 802.11n specifies the following contexts in which it is allowed to transmit an A-MPDU and what type of data it may carry:

- **Data Enabled Immediate Response**: this context is used when an A-MPDU is transmitted outside a PSMP sequence and the TXOP holder station which has initiated this sequence or the reverse direction responder require an immediate response.

- **Data Enabled No Immediate Response**: this context is used when an A-MPDU is transmitted outside a PSMP sequence like the previous context but in this case the TXOP holder station which has initiated the PSMP sequence does not require an immediate response.

- **PSMP**: this context is used when an A-MPDU is transmitted inside a PSMP sequence.

- **Control Response**: this last context is used when an A-MPDU is transmitted by neither a TXOP holder station nor the reverse direction responder, but by a station that needs to transmit any type of immediate acknowledgement frame (ACK, BA or a No Ack Management frame).

The context that allows Multi-TID acknowledgement is the third one, but Smart Aggregation does not need to be inside a PSMP sequence. Our mechanism, as mentioned earlier, will set the ACK Policy of all MPDUs to Normal/Implicit Ack in order to use an immediate response. It means that we are using a mix of the first and the third contexts.

On a protocol level, the deaggregation will not be changed from the standard one, because we are not changing the header, so this part of the chain will not be affected, but now the receiving node shall understand that it may receive packets from different Access Categories, and so the reaction to the reception of this SA-MPDU frame should be an optimally adapted Block Ack response frame, i.e. an Smart Block Ack frame.

### 4.4.3 Smart Block Ack

The Block Ack mechanism from the Standard [3] has been described in 2.3.5. In order to add smartness to our acknowledgement mechanism, we have improved the Multi-TID Block Ack frame and we have called it Smart Block Ack.
The Standard indicates that if Multi-TID TXOPs are used, it is mandatory to use the delayed Block Ack mechanism employing the Multi-TID Block Ack Request and the Multi-TID Block Ack Response frames. As Table 4.2 shows, when using the Multi-TID BA, the BA Info Field from the BA frame (as seen in B.5.5) forces the use of the Compressed Bitmap of 8 bytes. This way you are increasing the overhead due to the transmission of both Block Ack frames, request and response, and in addition you are not allowed to acknowledge more than 64 MPDUs.

<table>
<thead>
<tr>
<th>Multi-TID subfield value</th>
<th>Compressed Bitmap subfield value</th>
<th>Block Ack frame variant</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Basic Block Ack</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Compressed Block Ack</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Multi-TID Block Ack</td>
</tr>
</tbody>
</table>

Table 4.2: Block Ack frame variant encoding

The Smart Aggregation mechanism uses a Smart Multi-TID Block Ack Frame and it does not require the Block Ack Request frame, as shown in Figure 4.9, by setting the ACK Policy to Normal Ack in all MPDUs inside the SA-MPDU. Analyzing this figure, the overhead introduced by the mechanism a), defined in equation 4.4, is the minimum overhead if the Standard mechanism is used. Even taking the most optimistic situation, the overhead caused by a) is bigger than the one introduced by b) from our Smart Aggregation frame exchange sequence.

\[
\text{Overhead}_{\text{802.11n Multi-QoS Tx}} \geq 2 \times RIFS + SIFS + TxBAR \quad (4.4)
\]
The configuration of the modified Block Ack frame can be seen in Figure 4.10, allowing additionally the possibility to use the Multi-TID Block Ack frame with the Compressed Bitmap subfield from the BA Control set to 0 or to 1, depending on the configuration of the received SA-MPDU data frame.

<table>
<thead>
<tr>
<th>MACHeader</th>
<th>BA Control</th>
<th>Compressed TIDs</th>
<th>BA Information</th>
<th>FCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>B: 16</td>
<td>2</td>
<td>2</td>
<td>variable</td>
<td>4</td>
</tr>
</tbody>
</table>

Figure 4.10: Smart Block Ack frame.

When the Compressed Bitmap subfield is set to 0 in the Multi-TID BA frame, a new subfield of 2 bytes will appear next to the variable part of the BA response frame, the Compressed TIDs field. This field is used to specify, which of the 16 possible TIDs will be sent with the long and which ones with the compressed bitmap. Thereby if the Multi-TID subfield of the BA Control is set to 1 and the Compressed Bitmap subfield of the BA Control is set to 0, the new field Compressed TIDs will be read before the bitmaps are decoded.

This extra possibility will allow the Smart BA frame to combine long and short bitmaps inside the same BA Information field. We would like to remind the reader that the long bitmap field is used in the Smart Aggregation process to support the acknowledgement of individual MSDUs inside our SA-MSDU frame. This means that inside an SA-MPDU the receiving node may find some MPDUs containing SA-MSDUs that require MSDU acknowledgement, and some other SA-MSDUs that do not require this extra protection.

What this means is that if, in an SA-MPDU, a certain Access Category, does not request MSDU acknowledgement in its payloaded SA-MSDUs, the short BA bitmap for that certain TID can be used. Whereas if a single one of the payloaded SA-MSDUs requires MSDU acknowledgement for an AC, the BA bitmap subfield for that particular TID must be in its longer version of 128 bytes.

4.5 Smart Aggregation Process

In this last section we are going to explain how and when the mechanism aggregates the frames.

The mechanism works depending on a certain threshold, called $a_{AggregationTime}$. All the EDCF machines start counting their specific AIFS and backoff time. The aggregation mechanism will know, at the moment the machines start counting, which one will finish first and when it will finish. At that point, the Aggregation Trigger
Performance improvement strategies for current and next generation Wi-Fi systems

Aalborg University
RATE Section

will be set for the access category that is going to gain the medium. In this way, $a_{Aggregation\ Time}$ time before reaching the end of the counting, the trigger will launch the Smart Aggregation process.

It is important to highlight that we are assuming that the processing time for the mechanism to complete the aggregation process is lower than the $a_{Aggregation\ Time}$ value, such as 1 $\mu$s. We have chosen this $a_{Aggregation\ Time}$ to be equal or lower than 5 $\mu$s based on the $a_{RxTx\ Turnaround\ Time}$, a period of time during which the medium is no more sensed, and the channel is considered to have been contended.

It must be pointed that if the medium becomes busy before the Aggregation Trigger is reached, this trigger will be canceled; and if the medium becomes busy during the aggregation time, the packet will be sent either ways since the Wi-Fi transceiver has already started its turnaround procedure.

The described mechanism is shown in the next figure.

![Figure 4.11: Diagram that shows when to aggregate.](image)

On an MPDU level, the Smart Aggregation mechanism proposed in this thesis adds Multi-QoS aggregation, which might in a first attempt seem to overrule the traffic classification proposed in 802.11e, but it still takes into account the priority of the packets by intelligently ordering the packets in the SA-MPDU frame. This means that it will always try to send higher priority packets from the buffers before aggregating packets from lower ACs to the same frame. A more clear view of this behaviour can be seen in Figure 4.12, where Voice packets clearly get the highest priority.
Performance improvement strategies for current and next generation Wi-Fi systems

Figure 4.12: Aggregated packet depending on the access category that has gained the medium.
Chapter 5.

Wi-Fi MAC Modelling

To honour our initial studies and to follow some strict implementation rules, we designed a Wi-Fi model with some of the main functionalities and features that a Wi-Fi station requires to operate in an Infrastructure Network.

The modularity of our model allowed us to easily add our Smart Aggregation mechanism as well as to add multiple contention machines as explained in section 5.3.

Also this is the model that we have used to generate the baseline results presented in chapter 3 that allowed us to locate an specific part of Wi-Fi that could be improved.

Additionally, the reliability of our Wi-Fi MAC model has been validated by comparing to an existing mathematical model in appendix A.

We end this chapter by explaining how and what parts of the Smart Aggregation mechanism proposed in chapter 4 we have implemented in our simulator.

5.1 Wi-Fi MAC Layer

When we were first introduced to the GRASS simulator, an extensive understanding of the code as well as knowing how things were actually running was crucial in order to be able to minimize the time to implement the new features that our Smart Aggregation solution demanded, such as the QoS management and the aggregation mechanism.

After an initial study of the simulator, and a deep study of the Wi-Fi State Machine presented in [3], we decided that the current architecture was not flexible enough to easily allow upgrades to be added.

Thanks to our initial study on how to program in C++ with the best possible practices, we ended up with the MAC layer structure proposed in figure 5.1. This model allowed us to easily upgrade independent parts of the MAC layer while keeping the basic features working inside the already existing PHY and application layers.
5.1.1 Core components

In our Wi-Fi MAC model, which highly resembles to the MAC state machine proposed in [3], this horizontal layer is divided in three main entities:

Wi-Fi MAC Receiver: this is the block that mainly processes the received packets and tells the other entities how to act upon the reception of a certain type of packet, e.g. to send an ACK frame when a Data frame has been successfully received. It may also discard (send to trash) some packets when necessary. This entity is equivalent to the "Reception" block of the standard.

Wi-Fi MAC Controller: is responsible for the communication between the receiver and the transmitter at the MAC level, i.e. it is not a vertical communication entity. Besides the communication flow, the controller is the owner of the MAC state machine, which takes care of the medium access mechanism, explained in section 5.2. This entity is equivalent to the "Protocol Control" block of the standard.

Wi-Fi MAC Transmitter: which is, along with the controller, the most complex entity as it takes care of the whole transmission chain, which is divided in two main parts:

- *Packet preparation:* which serves as the set of functions and processes that generate the PSDU that is later sent to the PHY. See section 5.4.
This subset is equivalent to the "MPDU Generation" block of the standard.

- Transmitter: once the medium has been contained, the set of functions included in the Transmitter are the ones responsible of fetching a packet from the buffers and sending it to the PHY, as well as updating the statistics and taking care of the appropriate requests to the MAC Controller. This subset is equivalent to the "Transmission" block of the standard.

5.2 Medium contention mechanism

![Wi-Fi MAC State Machine](image)

Although a three-state state machine is proposed in the standard for the medium contention, we have added a 4th state to "contend" the medium for the high priority transmissions. The medium contention mechanism itself is a separate set of state machines, that run only when the MAC Finite State Machine (FSM) is in the Idle/Listening state.

Since, as specified in the standard, the SIFS/RIFS waiting does not require the medium to be free, i.e., all high priority transmissions are sent regardless of the channel status, we decided to add a separate state inside the MAC FSM instead of making it part of the channel contention mechanism.

All the state machines designed by us have been designed following the State Pattern from [14], but the lack of robustness of this pattern took us to improve the design by wrapping each FSM in its own interface, by hiding all functions that are irrelevant or unnecessary to external entities. Therefore, the MAC Controller is the only entity that has the authority to communicate, through a MAC FSM interface, with the MAC FSM, thus avoiding any unexpected behaviour of the state machines.
5.2.1 CCB: Channel Contention Block

While in Idle/Listening, if there is no ACK timeout awaiting, the MAC FSM allows the Channel Contention Block (CCB) to start contending for the medium.

The CCB is actually the interface that wraps around the state machines that contend the medium. Once again, the MAC FSM is the only entity that has access to the CCB. Our flexible design and this intermediate interface, allow us to add as many medium contention machines as we want plus one Beacon FSM (see figure 5.8).

The core of the CCB is what we have named the Enhanced Distributed Coordination Function (EDCF) FSM, which can be seen in figure 5.3, and serves as a very simplified and flexible object simulating the DCF contention mechanism. The flexibility comes from the fact that the IFS, CW_{min} and CW_{max} values are not hardcoded in the design and can be set when constructing an EDCF FSM object.

![Figure 5.3: EDCF State Machine](image)

To design the EDCF FSM, and make it as accurate as possible, we extracted and summarized all the medium access mechanism’s timings from the standard. Some of these timings are shown in figures 5.4, 5.5 and 5.6.

![Figure 5.4: Frame exchange sequence of a Successful Transmission](image)
Besides strictly following the 802.11 standard’s specifications, we have validated our Wi-Fi MAC model in section A.

**Beacon mechanism**

The CCB not only connects to an indefinite number of EDCF FSM, but may also contain a Beacon FSM, which is a much more simplified model of the EDCF FSM as shown in figure 5.7. This modular design allows us to add the beacon mechanism only to those stations acting as APs.

Besides the simplicity of this beacon contention machine, unlike the EDCF FSM, the only parameter in the Beacon FSM, the PIFS waiting time, is hardcoded and may change only if the Slot Time and SIFS values are changed (see Table 2.4) for the current simulation.
In our simulations, the Beacon mechanism responds to a separate chain of calls which conclude in the Transmitter subset to transmit the Beacon frame through the PHY. By default, the TBTT for the AP that our model uses is 100 ms. Also, to avoid collision of Beacons, when running a simulation, each network is activated at different instants of time.

5.3 Differentiated Service: QoS

5.3.1 QoS Application layer

In order to implement the 802.11e, first of all we needed to support a QoS application layer that generates MSDUs, where each MSDU has an associated a request for a certain prioritization.

To do so, we decided to add an intermediate layer, named QoS Traffic Decorator, between the application layer and the Wi-Fi MAC Layer. This naming is given due to the use of the Decorator Pattern from [14].

The main purpose of this "Decorator" is to label each of the Data Units that the Application layer generates with the desired UP value assigned to that flow ID. Once the packet is pushed to the Wi-Fi architecture, the MAC layer should be in charge of the mapping from UPs to ACs as explained in 2.3.1.

5.3.2 EDCAF State Machine

![Complete Wi-Fi 802.11e MAC Finite State Machine for an Access Point](image)

Figure 5.8: Complete Wi-Fi 802.11e MAC Finite State Machine for an Access Point
The second block that we changed was the CCB interface, now renamed to CCB QoS interface, to support 4 EDCF FSMs, one per AC, and a Beacon FSM when needed, as seen in figure 5.8.

Thanks to our very modular implementation, there was no need to change the MAC FSM except for some minor changes to notify which AC had gained the access to the medium, so that the MAC Transmitter would fetch the packet to be sent from the appropriate buffer.

One of the key things to model was also an internal collisions manager, because now one or more EDCF FSM may gain the medium at the same time, but only the one with the highest priority should be granted the permission to transmit, as shown in figure 5.9.

![Figure 5.9: Time diagram for an internal collision](image)

We have tested and compared our 802.11e model against the 802.11g Wi-Fi networks’ performance focusing on delay and throughput metrics in section 3.3.
5.4 Packet Preparation mechanism

![Diagram](image)

**Figure 5.10: PSDU preparation model interpretation from the standard**

Another important part of the MAC Transmitter chain is how the Data is prepared to be sent and where these frames are stored. The model presented in figure 5.10, is a simplified interpretation of the MPDU preparation chain as given by the standard [4] which we have also included in B.5.2. By excluding some minor functionalities and simplifying the model a bit further, we decided to separate the process in two subentities:

- **MPDU Preparation unit**: receives an MSDU, processes the MSDU(s) until an MPDU is obtained, then stores the MPDU inside the Tx Buffers and sends a transmission request to the controller. This subentity is equivalent to the "Prepare MPDU" block of the standard.

- **Tx Buffers**: holds the prepared MPDUs until the medium has been contended in separate sub-buffers depending on what type of frame it is:
Performance improvement strategies for current and next generation Wi-Fi systems

- Data Buffers - up to 4 data buffers are available to store the MPDUs of each AC separately. When not implementing 802.11e stations, a single Best Effort buffer is used.

- High Priority Buffers - all packets that are going to be sent after a SIFS or RIFS waiting time are first copied in these buffers. Once the frame has been successfully transmitted, it gets deleted from its original location. An example of these type of packets are the ACK, CTS, Data after CTS, Fragments, Data in burst transmissions, etc.

- Beacon Buffer - stores up to one single Beacon frame until the next time that the medium is free for a PIFS.

This subentity resembles to the "PM Filter" block of the standard, although in the standard it only stores packets waiting for a CFP, whereas in our case, all frames are first stored in the buffers and then sent.

The resulting block diagram for the transmission chain is shown in figure 5.11. Due to the complexity of our solution proposed in chapter 3, and the lack of time to implement the full system, we decided to give up some of the MPDU preparation features such as the A-MSDU aggregation mechanism, the fragmentation, encryption overhead and other minor settings.

![Figure 5.11: PSDU preparation model implementation in the simulator](image)

5.4.1 Buffer limitation model

To simulate a more accurate model we introduced a limited buffer scenario to our Wi-Fi stations.
When discussing on how to model this, to more accurately simulate the hardware world, we found that there were many ways to limit the buffer and drop the packets received from upper layers, especially when implementing a limited buffer model to compare the 802.11g and the 802.11e stations.

In our case, we decided to have a parameter that controls in bits how big the memory of that Wi-Fi chipset is. This meant no problems for the 802.11g stations, but for the 802.11e amendment we decided that this memory was shared among all the access categories. This means that for two stations having 512 kiB of buffering capacity, the 802.11g station and the 802.11e station will have exactly the same packets in buffers if the generation period and size is the same.

We rejected the other possible case of using a 512 kiB memory for each of the ACs as this would result in an unreasonable comparison.

### 5.4.2 Packet Aggregation mechanism

As mentioned earlier, we have implemented the A-MPDU mechanism in our Wi-Fi model in order to compare it with our SA-MPDU frame exchange sequence.

Despite that the 802.11n [4] is very restrictive and has very clear guidelines on how an A-MPDU frame should be (see B.5.4), the standard does not specify when the aggregation mechanism should take place nor how the queueing system should be affected by this mechanism.

Whenever a mechanism is not fully specified in the standard, it is understood to be implementers’ specific, thus the vendor may choose how to set-up that certain mechanism to achieve the best results out of it. This is why some publications, before proposing their solution, try to mitigate with the need of an Aggregation mechanism. For instance, in [32] an aggregation algorithm is proposed limited by the RTSThreshold and does not consider TXOP nor fragmentation’s implementation.

Since ours is an event-driven simulator, and therefore processing times are all atomical to the simulation time, we decided to implement the aggregation mechanism once the medium has been contended as we can see in figure 5.11.

Our aggregation algorithm is based on the following steps:

1. The first packet from the AC that gain the medium is fetched.
2. Based on this first packet, ALL packets for the same AC and same user are aggregated to the A-MPDU
3. The aggregation takes place until one of either of the A-MPDU size limitations is met: 65535 Bytes or 64 frames aggregated.
4. This A-MPDU is then forwarded to the MAC Tx which will then send it to the PLCP layer.
This seemed like the most accurate algorithm to implement aggregation. Although there is one extra feature that we see already as an improvement to the standard, as there is no specification of how the aggregation mechanism should behave when a retransmission shall occur. In this case there were two possibilities:

1. Upon retransmission, the exact same A-MPDU frame must be sent, excluding of course, those MPDU frames that have been positively acknowledged by the BA procedure.

2. Upon retransmission, since our aggregation scheme takes place once the medium has been contended, a new A-MPDU frame shall be prepared, containing all those packets that have received a negative acknowledgement in the previous transmission. Additionally, if the A-MPDU size limitations allow it, the algorithm shall aggregate new MPDUs that might have arrived since the first transmission occurred.

We chose to implement the improved 2nd option in our aggregation algorithm. This might seem unreasonable, but the presence of a ”Retry Bit” (see B.5) in each MPDU makes possible the distinction of frames that are being retransmitted from those that are new transmissions inside the same A-MPDU.

### 5.5 Smart Aggregation implementation

Our Smart Aggregation mechanism, as presented in chapter 4, is a combination of MAC features.

The implementation of such mechanism is itself quite complex, for which we decided to implement, for now, a simpler version to prove that the simplest of the improvements brought by our Smart Aggregation mechanism is already an improvement in throughput and delay at a MAC level. This simpler implementation considers the SA-MPDU frame that allows Multi-QoS aggregation as well as the use of the Multi-TID BA response frame immediately after the reception of the SA-MPDU frame.

In this case, the SA-MPDU algorithm is similar to the shown in section 5.4.2 for the A-MPDU frames, and follows the principles described in section 4.5:

1. The first packet from the AC that gain the medium is fetched.

2. The following aggregation subprocess take place until one of either of the A-MPDU size limitations is met: 65535 Bytes or 64 frames aggregated.
   
   (a) Based on the first packet, ALL packets that belong to the same DA from the AC Voice are aggregated.
(b) Based on the first packet, ALL packets that belong to the same DA from the AC Video are aggregated.

(c) Based on the first packet, ALL packets that belong to the same DA from the AC Best Effort are aggregated.

(d) Based on the first packet, ALL packets that belong to the same DA from the AC Background are aggregated.

3. This A-MPDU is then forwarded to the MAC Tx which will then send it to the PLCP layer.

The SA-MPDU frame which is the result of the above described algorithm is shown in figure 4.12.

The retransmission mechanism for SA-MPDU frames works in the exact same way as in our proposed aggregation algorithm, but in this case it means that if more Voice packets arrive which fill up the SA-MPDU frame, the lower priority frames will have to wait for the next contention. This serves as another reason why our mechanism even further enhances the packet prioritization scheme introduced by the 802.11e amendment.

On a programming basis, by using the Decorator pattern from [14] we were able to simply swap the MPDU preparation block with our A-MPDU or by our SA-MPDU mechanisms, so that whenever the MAC Tx would fetch a packet from the buffers, it should get an aggregated MPDU frame instead of a single MPDU.
Performance improvement strategies for current and next generation Wi-Fi systems

Aalborg University
RATE Section
Performance improvement strategies for current and next generation Wi-Fi systems
Chapter 6.

Smart Aggregation Performance

Now that we have introduced you deeply to the Wi-Fi world, we have presented our model and we have seen that it works properly by validating it against a mathematical model, it is time to show that our Smart Aggregation mechanism works and that it improves the Wi-Fi systems.

This chapter is divided in 3 sections. In the first and in the second section we are going to show the results for a single network and for a 3 by 3 system, respectively, with a set of UL and/or DL traffic combinations, and also varying the transmission rate to be 54 Mbps, which is the maximum one allowed by the 802.11-2007 Standard [3], or 144 Mbps, which is the highest bit rate mandatory in 802.11n devices.

Afterwards, we are going to analyze the results in much more realistic simulations by setting up an unbalanced traffic profile for the downlink and for the uplink, i.e., the amount of bits travelling in downlink are different from the amount of bits transmitted in uplink, while using the same rates as in the previous sections.

Finally, in the last section of this chapter we summarize some of the most remarkable results obtained by using the Smart Aggregation process.

Following the same principles as in chapter 3, we are interested in two metrics, the delay and the goodput, and these are the ones that we are going to present next.

6.1 Using Smart Aggregation in a Single Network

Every graphic of this chapter has multiple curves representing the results for the following cases:

- 802.11g stations with A-MPDU, which means that there will be no traffic differentiation because all the traffic is treated as Legacy
- 802.11e stations with A-MPDU, where we will show the different performance between the ACs for Voice, Video and Best Effort

- 802.11e stations with SA-MPDU, that will be used to compare the performance that each AC gets with our Smart Aggregation mechanism against the other curves presented in the same graph.

The parameters that we have used in the simulations for this section are shown in Table 6.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value/State/Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wi-Fi version</td>
<td>802.11g or 802.11n w/ and w/o 802.11e</td>
</tr>
<tr>
<td>Number of users</td>
<td>Variable</td>
</tr>
<tr>
<td>Deployment ratio</td>
<td>1</td>
</tr>
<tr>
<td>RTS/CTS mechanism</td>
<td>Deactivated</td>
</tr>
<tr>
<td>Buffer size</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Traffic balancing</td>
<td>Only DL, only UL or DL+UL</td>
</tr>
<tr>
<td>Scenario</td>
<td>1x1</td>
</tr>
<tr>
<td>Packet type</td>
<td>A-MPDU or SA-MPDU</td>
</tr>
<tr>
<td>Rate</td>
<td>54 Mbps or 144 Mbps</td>
</tr>
<tr>
<td>Type of traffic</td>
<td>Realistic or fixed</td>
</tr>
</tbody>
</table>

Table 6.1: Parameters we have used to run the simulations.

About the type of traffic parameters, when it is fixed it means that all the access categories are using the same bit rate, and when the traffic is realistic the values for each access category are different. These configurations are shown in Tables 6.2 and 6.3.

<table>
<thead>
<tr>
<th>Access Category</th>
<th>Packet size</th>
<th>Packet interval</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice</td>
<td>160 bytes</td>
<td>20 ms</td>
<td>64 kbps</td>
</tr>
<tr>
<td>Video</td>
<td>1280 bytes</td>
<td>10 ms</td>
<td>1024 kbps</td>
</tr>
<tr>
<td>Best effort</td>
<td>1500 bytes</td>
<td>12.5 ms</td>
<td>960 kbps</td>
</tr>
</tbody>
</table>

Table 6.2: Description of the characteristics for each AC for realistic traffic scenarios.

<table>
<thead>
<tr>
<th>Access Category</th>
<th>Packet size</th>
<th>Packet interval</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice</td>
<td>2560 bytes</td>
<td>5 ms</td>
<td>4 Mbps</td>
</tr>
<tr>
<td>Video</td>
<td>2560 bytes</td>
<td>5 ms</td>
<td>4 Mbps</td>
</tr>
<tr>
<td>Best effort</td>
<td>2560 bytes</td>
<td>5 ms</td>
<td>4 Mbps</td>
</tr>
</tbody>
</table>

Table 6.3: Description of the characteristics for each AC for fixed traffic scenarios.
We are going to start by having a look at the delay results, for 54 Mbps, and when there is traffic only in downlink in figure 6.1.

![Figure 6.1: Downlink delay when there is realistic traffic only in downlink.](image)

Having a look at the graphics, we can see that the A-MPDU using an 802.11g device has a smaller delay than the QoS A-MPDU as we already stated in section 3.1.

This behaviour is due to the maximum contention window value of Legacy packets, i.e., the value to be assigned to the backoff will be one within a bigger range of numbers, therefore it is less likely that two stations get the same value and therefore collide. In other words, you will be able to transmit with lower probability of collision but without an AC distinction.

The highest priority flow is Voice, and the delay is lower than the Best Effort because the packets are small and the time this AC requires to contend the medium is also small. The proportional behaviour comes for Video and Best Effort packets.

The delay for Best Effort is very big, not just because these type of packets have a big contention time but also for the bigger packet size, which means that the transmission time is also bigger.

The outcome of this graph is that in a situation where most of the traffic (a 100% in this case) is in downlink, our Smart Aggregation mechanism not only improves the 802.11e delay, but it also manages to keep a smaller delay for the lower ACs.
Table 6.4: Delay comparison of figure 6.1 for 19 users

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice</th>
<th>Video</th>
<th>Best Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11g w/A-MPDU</td>
<td>17.48 ms</td>
<td>17.48 ms</td>
<td>17.48 ms</td>
</tr>
<tr>
<td>802.11e w/A-MPDU</td>
<td>19.3 ms</td>
<td>44.21 ms</td>
<td>147 ms</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>13.49 ms</td>
<td>14.09 ms</td>
<td>14.28 ms</td>
</tr>
</tbody>
</table>

Delay vs .11g w/A-MPDU: -22.8%  -19.4%  -18.3%
Delay vs .11e w/A-MPDU: -30.1%  -68.1%  -90.3%

To conclude the analysis of this figure we are going to focus on the Smart Aggregation curves. Our mechanism is focused on the QoS feature, so its main functionality is to improve the QoS A-MPDU, and it does that by preserving the traffic differentiation.

As we can see in table 6.4 the Smart Aggregation mechanism preserves the traffic differentiation while improving the overall delay even when the medium is highly loaded.

This means that additionally, in situations where the QoS differentiation brings no benefits, the use of the SA-MPDU even improves the 802.11g delay. The difference between both graphics is, obviously, the delay reduction. As you can guess, adding a new feature to the PHY layer in order to improve the system when we have added the aggregation mechanism in the MAC level will give us great results.

To summarize this whole analysis, we can ensure that the Smart Aggregation mechanism improves all types of delay, even if you are not using it with all its features implemented, since just by using the SA-MPDU frame in situations where the use of traffic differentiation does not add any further value, we have reduced the delay around a 20%.

After these first results of the downlink delay, we are going to move on to the analysis of the uplink delay and goodput when we have traffic in both directions.

The first figure 6.2 shows the uplink delay when we have a realistic traffic scenario both in uplink and in downlink, and the second graph 6.3 pictures the goodput achieved per station in uplink for the same traffic scenario.

We put these two graphs together because they are closely related since a great throughput is not meaningful if we have a very long delay due to buffering, also this way we can see clearly how much the Smart Aggregation improves a system.
Performance improvement strategies for current and next generation Wi-Fi systems

Figure 6.2: Uplink delay comparison when there is realistic traffic in DL and UL.

Figure 6.3: Goodput comparison when there is realistic traffic in DL and UL.
Performance improvement strategies for current and next generation Wi-Fi systems

Aalborg University
RATE Section

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice</th>
<th>Video</th>
<th>Best Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11e w/A-MPDU</td>
<td>441.5  ms</td>
<td>977.5  ms</td>
<td>2942 ms</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>469.9  ms</td>
<td>807.8  ms</td>
<td>809.9 ms</td>
</tr>
</tbody>
</table>

Delay vs .11e w/A-MPDU

Table 6.5: Delay comparison of figure 6.2 for 19 users

When looking at the goodput graphs and tables 6.6 and 6.7, we can see that in this case, the results are not as overwhelming as the ones shown in figure 6.1 since we have clearly worsened the Best Effort and Video performance obtained by the 802.11g, but we can clearly see that we have more than satisfyingly improved the 802.11e stations not just by further prioritizing Voice packets but also by increasing the throughput for Video and Best Effort Access Categories.

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice</th>
<th>Video</th>
<th>Best Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11g w/A-MPDU</td>
<td>60 kbps</td>
<td>922 kbps</td>
<td>865 kbps</td>
</tr>
<tr>
<td>802.11e w/A-MPDU</td>
<td>60 kbps</td>
<td>937 kbps</td>
<td>80 kbps</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>60 kbps</td>
<td>937 kbps</td>
<td>764 kbps</td>
</tr>
</tbody>
</table>

Goodput vs .11e w/A-MPDU

Table 6.6: Goodput comparison of figure 6.3 for 9 users

The delay metrics may seem not that rewarding, but just like we mentioned in the baseline results chapter, it is not important to focus on just one metric, but on the whole. It is true that by using the SA-MPDU frame, the Voice delay increases almost a 10% in comparison to the 802.11e with A-MPDU, but we need to see the bigger picture here, where this loss of 10% has meant an improvement of over a 15% in Video delay, and a reduction to almost one fourth the delay of the Best Effort traffic when focusing on the highest number of users.

Specially in the Best Effort case, what we need to see here is not that in some cases it gets a little less delay than Video since its goodput is still lower, but we need to realize that by barely affecting the Voice and Video transmissions, we are actually sending Best Effort traffic.

We would like to remind the reader, that in our graphs, when the delay is close to the upper boundary, it means that in most of the cases no packets of that AC are sent. And this effect can be seen pretty quickly in the goodput graphs where we manage to have a non-zero throughput when there is a high load of users thanks to the Smart Aggregation mechanism.
After this last analysis, we can conclude that the Smart Aggregation mechanism is useful for single networks with a high load of users to preserve traffic differentiation. Now, the next step is to check if our mechanism is also useful for Large Scale Deployments.

### 6.2 Smart Aggregation in Large Scale Deployments

The list of used parameters and the type of traffic are the same as the ones used in the previous section, described in tables 6.1 and 6.2, except for the scenario size which now is a 3 by 3 apartments configuration.

The first graph we want to show is the downlink delay when there is realistic traffic in both ways and the rate is set to 54 Mbps. The results are shown in Figure 6.4.

![Figure 6.4: Downlink delay when there is realistic traffic in UL and DL.](image-url)
Performance improvement strategies for current and next generation Wi-Fi systems

As you can see in this figure, the Voice delay is the only one injured because the Video and the Best Effort delays are improved compared to the QoS A-MPDU stations. This is because with the Smart Aggregation mechanism you aggregate always all the packets you can, so if you have a Smart Aggregation frame with Voice packets that also carries other type of packets, since the other packets are bigger and with lower priority, the transmission time increases and therefore the delay for Voice increases. This behaviour is similar to the one presented in the previous section.

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice</th>
<th>Video</th>
<th>Best Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11e w/A-MPDU</td>
<td>1205 ms</td>
<td>1918 ms</td>
<td>2759 ms</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>1527 ms</td>
<td>1779 ms</td>
<td>2101 ms</td>
</tr>
</tbody>
</table>

Delay vs .11e w/A-MPDU | +26.7% | -7.3% | -23.9%

Table 6.8: Delay comparison of figure 6.4 for 5 users/network

If we extract the numbers from the figure, as shown in table 6.8, we can see that we are increasing the Voice delay in almost a 25% but this performance loss is compensated by a noticeable decrease in the Video and Best Effort delays.

The other important result, which is shown in Figure 6.5, pictures the downlink goodput when there is traffic in uplink and downlink and the rate is also set to 54 Mbps, i.e., the same situation as the delay figure.

Figure 6.5: Uplink goodput when there is realistic traffic in UL and DL.
In this figure we can see that, even if the delay for Voice is a little bit increased, the goodput we get is still the same as the QoS A-MPDU and still better than the one in the normal 802.11g network.

The opposite behaviour occurs to the Video packets, the delay is improved but the goodput is slightly decreased as we can see in table 6.9.

But, looking at the Best Effort traffic, we can see that we are reducing so much the delay, while vastly improving its goodput, that the end user is likely to perceive an overall better application layer experience.

In other words, loosing 5% of Video goodput while keeping the Voice goodput the same, you are getting way more Best Effort goodput when there are 2 users per network.

In a very realistic scenario, this would mean that while keeping my Voice call active, and loosing a little bit of quality on my Video conference, my cloud back up or my file transfer is actually running 2.5 times faster.

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice</th>
<th>Video</th>
<th>Best Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11g w/A-MPDU</td>
<td>34 kbps</td>
<td>546 kbps</td>
<td>514 kbps</td>
</tr>
<tr>
<td>802.11e w/A-MPDU</td>
<td>46 kbps</td>
<td>514 kbps</td>
<td>131 kbps</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>46 kbps</td>
<td>490 kbps</td>
<td>318 kbps</td>
</tr>
<tr>
<td>Goodput vs .11g w/A-MPDU</td>
<td>+35.3%</td>
<td>-10.2%</td>
<td>-38.1%</td>
</tr>
<tr>
<td>Goodput vs .11e w/A-MPDU</td>
<td>0%</td>
<td>-4.7%</td>
<td>+142.7%</td>
</tr>
</tbody>
</table>

*Table 6.9: Goodput comparison of figure 6.5 for 2 users per network*

The next results we want to show are with a fixed traffic flow and using a PHY rate of 144 Mbps. The reason why we chose these fixed values traffic scenario is to quickly saturate the system and to see how it is able to react to this situation.

To put an example of one real situation, we can imagine that we want to watch a film in High Definition from an Internet server which requires high load of Voice, Video and Best Effort traffic.

Figure 6.6 shows the delay and Figure 6.7 shows the goodput when, for both cases, there is only traffic in downlink.
In figure 6.6 we can see that Smart Aggregation works flawlessly in these kind of situations. The idea of adding prioritization is to allow some kind of packets to be transmitted the first ones, with a lower delay than the non-QoS packets, some other kind of packets to be transmitted with the same delay as in non-QoS stations in average and the less important packets with a bigger delay.

In this picture we can see exactly this behaviour. Where two of the Access Categories get the same or better delay values as the ones in the 802.11n stations.

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice</th>
<th>Video</th>
<th>Best Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11n w/A-MPDU</td>
<td>509 ms</td>
<td>509 ms</td>
<td>509 ms</td>
</tr>
<tr>
<td>802.11e w/A-MPDU</td>
<td>420 ms</td>
<td>645 ms</td>
<td>1395 ms</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>360 ms</td>
<td>482 ms</td>
<td>578 ms</td>
</tr>
<tr>
<td>Delay vs .11n w/A-MPDU</td>
<td>-29.3%</td>
<td>-5.3%</td>
<td>+13.6%</td>
</tr>
<tr>
<td>Delay vs .11e w/A-MPDU</td>
<td>-14.3%</td>
<td>-25.3%</td>
<td>-58.6%</td>
</tr>
</tbody>
</table>

Table 6.10: Delay comparison of figure 6.6 for 5 users/network

Analyzing in detail the values from table 6.10, we see that the Voice delay is improved in comparison to both other cases, the Video delay is inline with the non-QoS A-MPDU one, being at the begging smaller and as the number of stations increases it become slightly bigger than the 802.11n delay. And finally, the Best Effort delay is reduced about a 58% in comparison to the QoS A-MPDU delay, getting closer to
Performance improvement strategies for current and next generation Wi-Fi systems

Aalborg University
RATE Section

the Video delay achieved by the standard compliant 802.11n without QoS stations, while keeping the traffic differentiation in our Smart Aggregation mechanism.

Figure 6.7: Downlink goodput when there is fixed traffic in DL.

The goodput result shown in figure 6.7 presents almost the same behaviour as the delay one, i.e., the Voice goodput is bigger than the 802.11n one, the Video goodput is the one that gets worsen than the other two cases and the Best Effort goodput is bigger than the one obtained by the standard 802.11e machines.

We can say that this situation is really beneficial for the Voice traffic because it increases approximately 30% the goodput against the non-QoS aggregation while the Video and Best Effort goodputs highly decrease. For this particular scenario, the main interest is that the 802.11e prioritization scheme is further enhanced by the Smart Aggregation.

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice</th>
<th>Video</th>
<th>Best Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11g w/A-MPDU</td>
<td>2442 kbps</td>
<td>2442 kbps</td>
<td>2441 kbps</td>
</tr>
<tr>
<td>802.11e w/A-MPDU</td>
<td>2832 kbps</td>
<td>2281 kbps</td>
<td>956 kbps</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>3113 kbps</td>
<td>2008 kbps</td>
<td>1227 kbps</td>
</tr>
</tbody>
</table>

| Goodput vs .11g w/A-MPDU | +27.5% | -17.8% | -39.7% |
| Goodput vs .11e w/A-MPDU  | +10.0% | -12.0% | +28.4% |

Table 6.11: Goodput comparison of figure 6.7 for 2 users per network
After analyzing these graphics, we can conclude that Smart Aggregation allows you to reduce the average delay of the system, even if in certain situations some access categories get injured in order to further prioritize Voice traffic.

### 6.3 Unbalanced Scenarios

To finish with the Smart Aggregation results, we launched a campaign of simulations with non-balanced traffic, i.e., an asymmetric UL/DL ratio, which gives us a more everyday situation where users download a lot of information but upload a very little amount of traffic which is mostly TCP ACKs and information requests.

We have analyzed two types of unbalanced scenarios both considering that the offered traffic in uplink is \(1/6\) of the traffic in downlink. So now we have two different types of uplink traffic, because the downlink remains as defined before in other simulations. The simulation parameters in this section are the same as the ones defined in Table 6.1, except now the traffic distribution in uplink may be one of the shown in Table 6.12.

<table>
<thead>
<tr>
<th>Uplink Traffic</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only BE traffic</td>
<td>Realistic and fixed</td>
</tr>
<tr>
<td>VO, VI and BE traffic</td>
<td>Realistic and fixed</td>
</tr>
</tbody>
</table>

**Table 6.12:** Description of the possible traffic flows in uplink traffic.

Now that we have defined the types of traffic that we will simulate, we are going to summarize their characteristics, which are shown in Tables 6.13, 6.14, 6.15 and 6.16 for realistic and only BE uplink traffic, realistic and all type of uplink traffic, fixed and only BE uplink traffic and finally fixed and all type of uplink traffic, respectively.

<table>
<thead>
<tr>
<th>AC</th>
<th>Packet size</th>
<th>Packet interval</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downlink</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voice</td>
<td>160 bytes</td>
<td>20 ms</td>
<td>64 kbps</td>
</tr>
<tr>
<td>Video</td>
<td>1280 bytes</td>
<td>10 ms</td>
<td>1024 kbps</td>
</tr>
<tr>
<td>Best effort</td>
<td>1500 bytes</td>
<td>12.5 ms</td>
<td>960 kbps</td>
</tr>
<tr>
<td>Uplink</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best effort</td>
<td>640 bytes</td>
<td>15 ms</td>
<td>341.3 kbps</td>
</tr>
</tbody>
</table>

**Table 6.13:** Characteristics of the realistic traffic when there is only BE traffic in UL.
Table 6.14: Characteristics of the realistic traffic when there are all type of traffics in UL.

<table>
<thead>
<tr>
<th>AC</th>
<th>Packet size</th>
<th>Packet interval</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downlink</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voice</td>
<td>160 bytes</td>
<td>20 ms</td>
<td>64 kbps</td>
</tr>
<tr>
<td>Video</td>
<td>1280 bytes</td>
<td>10 ms</td>
<td>1024 kbps</td>
</tr>
<tr>
<td>Best effort</td>
<td>1500 bytes</td>
<td>12.5 ms</td>
<td>960 kbps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AC</th>
<th>Packet size</th>
<th>Packet interval</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uplink</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voice</td>
<td>160 bytes</td>
<td>20 ms</td>
<td>64 kbps</td>
</tr>
<tr>
<td>Video</td>
<td>260 bytes</td>
<td>15 ms</td>
<td>138.6 kbps</td>
</tr>
<tr>
<td>Best effort</td>
<td>260 bytes</td>
<td>15 ms</td>
<td>138.6 kbps</td>
</tr>
</tbody>
</table>

Table 6.15: Characteristics of the fixed traffic scenario when there is only BE traffic in UL.

<table>
<thead>
<tr>
<th>AC</th>
<th>Packet size</th>
<th>Packet interval</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downlink</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voice</td>
<td>2560 bytes</td>
<td>5 ms</td>
<td>4 Mbps</td>
</tr>
<tr>
<td>Video</td>
<td>2560 bytes</td>
<td>5 ms</td>
<td>4 Mbps</td>
</tr>
<tr>
<td>Best effort</td>
<td>2560 bytes</td>
<td>5 ms</td>
<td>4 Mbps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AC</th>
<th>Packet size</th>
<th>Packet interval</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uplink</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voice</td>
<td>1280 bytes</td>
<td>5 ms</td>
<td>2048 kbps</td>
</tr>
<tr>
<td>Video</td>
<td>640 bytes</td>
<td>7.5 ms</td>
<td>682.6 kbps</td>
</tr>
<tr>
<td>Best effort</td>
<td>640 bytes</td>
<td>7.5 ms</td>
<td>682.6 kbps</td>
</tr>
</tbody>
</table>

Table 6.16: Characteristics of the fixed traffic scenario when there are all types of traffics in UL.

To analyze what happens with the delay we have chosen the scenario where you only have Best Effort traffic in uplink by using a rate of 54 Mbps. The results are shown in Figure 6.8.
As you can see, even if we are using a rate of 54 Mbps and the traffic is asymmetric, like in many real life situations, you get really good results using the Smart Aggregation mechanism. As explained earlier in this chapter, the delay for Voice increases a little bit, but this behaviour is compensated by the decrease of Video and the Best Effort delay.

In comparison to other results shown in this chapter, it is also important to notice that the delay reduction in this scenario is also caused by the reduction of the amount of traffic in uplink, because then the transmission time is lower and there is more probability of getting the medium access so the packets in the buffers do not have to wait that much time to be sent.

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice</th>
<th>Video</th>
<th>Best Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11n w/A-MPDU</td>
<td>113 ms</td>
<td>113 ms</td>
<td>113 ms</td>
</tr>
<tr>
<td>802.11e w/A-MPDU</td>
<td>22 ms</td>
<td>48 ms</td>
<td>220 ms</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>15.2 ms</td>
<td>15.9 ms</td>
<td>16.2 ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Delay vs .11n w/A-MPDU</th>
<th>-86.5%</th>
<th>-85.9%</th>
<th>-85.7%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay vs .11e w/A-MPDU</td>
<td>-30.9%</td>
<td>-66.9%</td>
<td>-92.6%</td>
</tr>
</tbody>
</table>

Table 6.17: Delay comparison of figure 6.8 for 14 users

The most important thing to be pointed out in this graph is that we can see the SA-
MPDU frame in full action delivering all the benefits we have mentioned throughout this chapter:

**No traffic differentiation when it is not needed:** as we can see in table 6.17, the SA-MPDU frame deprecates the traffic differentiation where it brings no benefits since the network is not yet overloaded.

When 14 users are present in the network, our SA-MPDU not only widely improves the delay achieved by the 802.11g stations, but even improves the 802.11e delay for all the ACs.

**Traffic differentiation in crowded networks:** When the network is loaded with 20 users, see table 6.18, the SA-MPDU curves clearly lead to a traffic differentiation scenario. And although the Voice delay gets highly increased in comparison to the 802.11e stations, which is affected by the extra payloading of Best Effort packets, this delay is still under the 150 ms of delay that the user can neglect in a Voice call [24].

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice</th>
<th>Video</th>
<th>Best Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11n w/A-MPDU</td>
<td>710 ms</td>
<td>710 ms</td>
<td>710 ms</td>
</tr>
<tr>
<td>802.11e w/A-MPDU</td>
<td>53 ms</td>
<td>121 ms</td>
<td>859 ms</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>121 ms</td>
<td>121 ms</td>
<td>206 ms</td>
</tr>
</tbody>
</table>

| Delay vs .11n w/A-MPDU   | -83.0%| -83.0%| -71.0%      |
| Delay vs .11e w/A-MPDU   | +128.3%| 0%| -76.0%      |

**Table 6.18: Delay comparison of figure 6.8 for 20 users**

Furthermore, if the adaptiveness of our Smart Aggregation mechanism had been implemented, we are quite confident that the results when the network is highly loaded, would have had an even more remarkable differentiation, which would lead to a reduction of the maximum SA-MPDU size thus leaving out more Best Effort packets while keeping the Voice prioritization.

The results of the goodput achieved in an even more realistic scenario are shown in Figure 6.9, where you have Voice, Video and Best Effort traffic in uplink, using a PHY rate of 144 Mbps.
In this picture you can see that the goodput that we get by using our mechanism with a high PHY rate is higher than the one achieved by the QoS A-MPDU.

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice (kbps)</th>
<th>Video (kbps)</th>
<th>Best Effort (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11e w/A-MPDU</td>
<td>63.8</td>
<td>998.3</td>
<td>81.3</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>63.8</td>
<td>1019</td>
<td>956.2</td>
</tr>
<tr>
<td>Goodput vs .11e w/A-MPDU</td>
<td>0%</td>
<td>+2.1%</td>
<td>+1076.1%</td>
</tr>
</tbody>
</table>

Table 6.19: Goodput comparison of figure 6.9 for 14 users

Either ways in this graph we can see that our SA-MPDU can keep up the good performance up to 14 users whereas the standard compliant 802.11e stations already begin to fade with the presence of 8 users.

<table>
<thead>
<tr>
<th>Station</th>
<th>Voice (kbps)</th>
<th>Video (kbps)</th>
<th>Best Effort (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11e w/A-MPDU</td>
<td>59</td>
<td>435</td>
<td>1.6</td>
</tr>
<tr>
<td>802.11e w/SA-MPDU</td>
<td>59</td>
<td>761</td>
<td>70</td>
</tr>
<tr>
<td>Goodput vs .11e w/A-MPDU</td>
<td>0%</td>
<td>+74.9%</td>
<td>+4275.0%</td>
</tr>
</tbody>
</table>

Table 6.20: Goodput comparison of figure 6.9 for 20 users
With this last set of results we can see that even in the most realistic scenarios, our Smart Aggregation mechanism vastly improves the 802.11e, although sometimes losing some negligible decrease in Voice performance, we manage to get up to 10 times higher Best Effort throughput in downlink.

### 6.4 Summary

Throughout this chapter and chapter 3 we have presented all the results from our system. As a last step we want to summarize the most significative results in order to make things clear.

In the simplest case, a single network with 9 stations competing for the medium, the Best Effort goodput is almost 0 under certain simulation scenarios if QoS aggregation is used. Regarding the delay, the QoS aggregation gets around 3 times the delay that the non-QoS aggregation mechanism, but it is much worse if 19 stations are in the network, where the delay of stations using QoS is up to 8 times higher.

The most common situation in Wi-Fi is when there is a substantial amount of downlink throughput compared to the uplink. In this situation, by using the Smart Aggregation mechanism we are able to improve the delay for all the access categories not only against the 802.11e version but also against the 802.11g. Compared to the standard QoS aggregation, we are reducing more than 90% the Best Effort delay, 68% the video delay and 30% the voice delay. In uplink when there is traffic in both ways we can still see some improvements, mainly for the Best Effort traffic where it is around 9 times smaller than the 802.11e aggregation, but also for the other access categories when the network is overcrowded.

For large scale deployments, such as 3 by 3 scenarios, and taking into account the previous scenario (only downlink traffic), the improvements in the delay and in the goodput are still there. Compared to the QoS aggregation we are highly improving the whole system.

Focusing on the most important scenario, where we have defined the uplink traffic as 1/6 of the total downlink traffic, we have got really good results.

Regarding the 802.11g version and the downlink, we are reducing the delay for all the access categories when there are 18 stations in the network an average of 80%. Comparing the Smart Aggregation delay with the 802.11e, we have an increase in Voice delay but this is compensated by having almost the same delay for Video and drastically reducing up to a 75% the Best Effort delay.

Finally, we have also seen that the uplink is also improved in comparison to the standard 802.11e stations, where the performance in goodput when we have 14 users in an isolated network for Voice and Video packets is almost the same. But for Best Effort traffic we can get up to 10 times more throughput than with the 802.11e compliant devices.

With this study we can consider that our mechanism has successfully reached its goal of improving the QoS.
Chapter 7.

Conclusions

The wireless world is very complex, and as a real system that it is, it has its own strengths and weaknesses. Our job has been to investigate and go deeply inside these weaknesses to, somehow, improve them without injuring its strengths.

We have focused on the MAC level. We have seen that adding QoS in the Wi-Fi system is not always a good option because then the performance of the system goes down. The goal of this thesis has been twofold, to find a new mechanism to bring the possibility of using QoS and aggregation at the same time without losing performance, and to improve the actual standard frame aggregation mechanism by adding the Multi-QoS feature. We have named our mechanism Smart Aggregation.

The project has been developed in several steps. Once we learnt how the Wi-Fi system works and what we could expect from it, we designed our own MAC layer model. When we had it implemented, we launched a set of simulations in order to get the results from the system working with our model. Finally, we validated our design through a complex mathematical model (see Appendix A.).

To implement the Smart Aggregation mechanism we needed to add other features like the 802.11e amendment, in order to add QoS to the system, and the aggregation mechanism.

Throughout the project we have been very much Standard compliant in order to design the most accurate MAC layer model. In addition, because the Standard [4] does not specify how the aggregation mechanism works, giving some freedom to develop several algorithms, so we have implemented this mechanism in the most optimal way, to get the maximum possible performance.

Analyzing the results from the simulator, working with the new MAC model and the necessary mechanisms, we found two important problems. The first one is concerning the number of collisions within crowded networks, where the collision rate for the 802.11e is higher than the one obtained for the 802.11g stations. The second one is the unfairness between access categories, mainly for Best Effort traffic’s goodput.
and delay, where sometimes just to keep up the performance of the higher Voice priority packets, we may end up in a situation where low priority traffic is not even transmitted.

Once we had the standard tools working in the simulator, we had to define our mechanism. From that moment onwards we started being creative and also being not as much standard compliant as before. We changed some MAC frames to reduce the overhead caused by the MAC header in order to get more goodput, and we redefined the acknowledgement mechanism and also the frame settings to reduce the delay of the system.

We defined different types of scenarios to find out which were the most advantageous situations to use the Smart Aggregation mechanism. If we think about the Wi-Fi system, we can agree that the most usual scenario is when the access point is transmitting a lot of information all the time while the users have a very little impact on the offered load in uplink. This means that even if we have to improve the maximum number of situations possible, one of the most important situation is the one in which there is more traffic or substantially only traffic in downlink, rather than in uplink.

After taking these very realistic scenarios into account we implemented and tested our Smart Aggregation mechanism, which has proven to improve the delay and goodput performance both, in isolated overcrowded networks, and in scenarios where multiple networks share the same radio channel where the hidden node problem and some other Wi-Fi MAC limitations are a common problem.

As we have shown in our results, in comparison to the standard complaint networks, our Smart Aggregation solution is able to solve both problems that were the focus of this thesis. We managed to reduce the number of collisions by designing a Multi-QoS strategy where an intelligent aggregation process reduces the overall number of channel access to transmit a set of packets. And by respecting the traffic differentiation brought by the 802.11e amendment, in combination with our Smart Aggregation process, we were able to drastically reduce the delay and highly improve the throughput for low priority traffic.

Additionally, we focused on some unbalanced traffic scenario simulations, where we have an UL/DL ratio of 1/6, i.e. the offered load in uplink is much smaller than the traffic generated at the Access Point. This served as the most realistic traffic scenario possible.

In these set of simulations, we realized that with our Smart Aggregation we have also improved the Access Point bottleneck, while keeping the focus on the reduced overhead and on the QoS fairness improvements. This bottleneck problem is quite common in Wi-Fi networks because the Access Point gets the same probability of accessing the medium as the rest of the stations, but the traffic load at the AP is higher.

Overall this study has served us to understand better the Wi-Fi networks’ performance, and present the main flaws in the performance of it, on which we based our proposed solution that has successfully improved the QoS delay and goodput.
7.1 Future Work

The Smart Aggregation mechanism described in this Thesis is already quite a complex set of functionalities. Despite of that, the bigger picture of our solution was a Smart Controller MAC layer unit, that would not only manage the various aggregation processes studied throughout our Thesis, but would also dynamically combine other MAC features such as the Fragmentation and the RTS/CTS mechanism.

![Policy decision](image)

As it is described in the Appendix A of this thesis, the Fragmentation process and the RTS/CTS mechanism are both based on a fixed Threshold upon which the mechanism is activated. So in a similar way to the Aggregation maximum size, which is also a fixed value in the standard, but we propose to dynamically adapt it based on the channel conditions, these other thresholds shall also vary up to a certain point where using Fragmentation or RTS/CTS may be beneficial.

For instance, when using A-MSDU, the standard deprecates the use of Fragmentation, but this improved Smart Controller solution that we would like to study, would disable the MSDU aggregation in favour of activating the Fragmentation mechanism if the channel conditions have worsen.

A study on how these threshold values behave against our proposed criteria is required and may be extracted from many previous publications.

We expect that this improved Smart Controller would result in a highly cognitive MAC layer that adapts itself to the channel conditions in a more efficient way trying to reach always the saturation throughput of the network while not overloading it any further.

This would work in somewhat a similar way to the already present Link Adaptation mechanisms, except that these are PHY adaptations based on PHY criteria, whereas we have proposed a MAC layer adaptation based on MAC layer criteria.
Appendices
Appendix A.

Wi-Fi MAC Model Validation

Each simulator has its own implementation, and this implementation has to be validated taking some reference as a model. In our case we have chosen Bianchi’s model [33] which tries to model the DCF mechanism from [3], based on the CSMA/CA MAC protocol and a binary exponential backoff. In the paper they propose a new backoff decrement model in order to maintain a correlation between slots time, like in the Standard.

In our simulations, we are using the parameters defined in the 802.11-2007 Standard [3] for stations using the 802.11g version:

- Slot time = 20 µs
- Interframe Spacing: SIFS = 10 µs and DIFS = 50 µs.
- Control rate = 6 Mbps.
- Data rate: the maximum allowed, 54 Mbps.
- Frame length: ACK = 112 bits, RTS = 160 bits and CTS = 112 bits.
- Headers: MAC = 256 bits and PHY = 120 bits or 96 µs (the short one).

We have focused on validating the throughput curve in two ways, depending on the number of contending stations - see Figure A.1 - and also depending on the packet size - see Figure A.2.

Bianchi’s paper only checks its model varying the number of stations which contend for the medium, but we have added a new validation level because the throughput is closely related to the packet length.

The results that we are showing are obtained for one network, i.e., one access point and different number of user stations and with the RTS/CTS mechanism deactivated. The packet size used in Figure A.1 is 3000 bytes.
Based on these results we can ensure that our model is close enough to Bianchi's one, and we can continue adding functionalities from newer amendments, like the 802.11e with the QoS performance.
Appendix B.

The Wi-Fi MAC layer in detail

B.1 CSMA/CA

Wi-Fi uses CSMA/CA to divide the medium resources among all users, unlike the Ethernet protocol which uses Carrier Sense Multiple Access/Collision Detection (CSMA/CD). This is due to the lack of ability to listen to the entire medium permanently. This access scheme and the radio communications combined bring some limitations described later in B.1.2.

CSMA is a MAC protocol designed for shared mediums where stations or users must compete in order to use the medium during a certain amount of time.

This technique implies the ability for all communicating entities to listen to the medium. In which case, the stations must wait for the medium to be free if the carrier sensing finds the medium busy. Therefore collisions occur only when, due to the propagation delay, two stations find the medium to be free.

This type of Multiple Access also involves the fact that a certain transmission is received by all other stations using the medium thus requiring for better overall security measures to be applied such as authentication, confidentiality and integrity. Unlike TDMA, FDMA or others techniques which use a slotted access and a man-in-the-middle attack is more unlikely to happen unless someone invades the resource assigned to a certain station.

CSMA supports slotted access but neither in Ethernet (802.3 uses CSMA/CD) nor WLAN (802.11 uses CSMA/CA) use it.

B.1.1 CSMA Access Modes

There are mainly three types of CSMA access of carrier sensing decisions:
1-persistent: This was the type of CSMA used to develop CSMA/CD and is now used in Ethernet to stop the transmission if a collision is detected thus avoiding further collisions. This can only be applied in mediums where all stations can listen to one another, where there are no problems like the hidden node in WLAN.

0-persistent: It is designed for a slotted access where the station waits one time slot when a packet is ready. The main difference with non-persistent is that with non-persistent the station does not monitor the medium and if a collision is detect it waits one time slot before re-transmission occurs.

P-persistent: An adaptation of this carrier sensing technique is used in CSMA/CA which reduces the number of collisions by waiting a random time if the channel is sensed to be free. The procedure this access mode follows to generate successful transmissions is:

1. Wait for a packet to be ready for tx
2. Check if the medium is busy
3. If the medium is busy, then the station retains the packet until the medium becomes available.
4. If the medium is free then the packet is sent with a the packet is sent with a probability p. This is done by generating a random number \( r \epsilon [0,1] \) and comparing it with p.
5. If the packet is decided not to be sent, then the transmitter waits a certain time before retrying.
6. In case of collision, the station waits a random time and goes back to carrier sensing on step 2.
7. If an ACK packet is received then the transmission has been successful.

CSMA/CA is a type of CSMA p-persistent, but unlike the common CSMA, collision avoidance adds a random exponential back-off time if the channel is found occupied making it more efficient by not sensing the channel all the time. This specific type of CSMA is used in wireless mediums where collisions are much harder to detect and transmitting stations cannot sense the medium and send data at the same time.

B.1.2 CSMA Limitations

RF Link Quality

The reception of the frame is not guaranteed in radio links, specially when the frequencies used are unlicensed ISM bands, thus each transmitted frame must be acknowledged to know if the frame was lost or successfully received. Noise and multipath propagation are some of the RF Link problems alike other radio communications.
The hidden node problem

The basic access mechanism in Wi-Fi is not controlled by a central base station or the AP, but instead is based on medium sensing. In an infrastructure network, all stations can listen to the AP, but not all stations see each other’s transmissions, thus generating this hidden node problem.

This problem is based in a scenario with basic access mechanism where node 1 may transmit to node 2, with node 3 not being able to sense this transmission. Therefore, if during that transmission, in a certain moment, node 3 decides to communicate with node 2, packet collision will occur in node 2.

This problem is solved by using RTS/CTS, described in B.2.4, since 1 would send RTS to 2, and node 2 would broadcast CTS to all its surrounding nodes to avoid collision. But this is not the only way of solving this problem, other solutions perhaps costlier, proposed to fight this are:

- Transmit more power to make nodes visible
- Remove obstacles
- Move the nodes
- Use antenna diversity
- Use omnidirectional antennas

The exposed node problem

This problem, as we can see in figure B.2 extracted from [34], is related to self-network or neighbouring networks interference where station S2 may successfully transmit to station R2, but due to proximity of other ongoing transmissions, station S2 sees the medium to be busy.
B.1.3 Carrier Sensing Functions

Carrier-Sensing is used to determine if the medium is available. There are two types of carrier sensing mechanisms:

- **PHY C-S function**: it is provided by the Clear Channel Assessment (CCA) module, that based on a power threshold signals the medium to be busy or free, where this threshold depends on the medium and modulation used.

- **Virtual C-S function**: it is provided by the Network Allocation Vector (NAV), which is a timer that indicates the amount of time the medium will be reserved by another station, and therefore must be sensed as busy. This NAV value is extracted from frames which are sensed by the MAC layer but are not addressed to the station extracting this information. A frame containing a NAV value of 0 may be used to reset the Virtual C-S mechanism. The NAV value is typically carried in the frame headers inside an Request-To-Send (RTS)/Clear-To-Send (CTS) frame exchange.

B.2 DCF: Distributed Coordination Function

The DCF, based on CSMA/CA allows multiple independent stations to interact without central control, and thus may be used in either Ad-hoc or infrastructure networks.

There are two techniques used for packet transmitting in DCF. The default one is a two-way handshaking mechanism, also known as basic access method. A positive MAC aAcknowledgement (ACK) is transmitted by the destination station to signal the successful packet transmission. The other optional one is a four-way handshaking mechanism, which uses RTS/CTS technique to reserve the channel before data transmission (see B.2.4). This technique has been introduced to reduce the performance degradation due to hidden terminal (B.1.2), however, the drawback of RTS/CTS mechanism is the increased overhead for short data frames.
B.2.1 Error Recovery in DCF

In Wi-Fi, errors must be detected by the transmitting station. In Wi-Fi a station detects that a data transmission has been erroneous when the expected Acknowledgement (ACK) for that transmission has not been received. To be able to do that, the Wi-Fi MAC layer implements an ACK Timeout mechanism, which is defined as:

$$ACK_{Timeout} = aSIFSTime + aSlotTime + aPHY-RX-START-Delay$$ [3]

Where $aPHY-RX-START-Delay$ is the amount of time from the moment that an incoming signal is first sensed until the PHY layer indicates the start of a reception of a frame. This is typically the total time length of the PHY header, i.e. Preamble + PLCP Header.

Retry Counter

In a similar way to the Time To Live (TTL) in the Internet Protocol (IP) networks, each Wi-Fi frame or fragment has a single retry counter associated with it, i.e. it is associated to each MPDU.

Stations have two maximum retry counters values though:

**Short retry counter**: counts transmissions of frames that are sent without the RTS/CTS mechanism.
This counter has typically a maximum value of 7.
This counter will be reset to 0 when:
- a CTS frame is received in response to a transmitted RTS,
- a MAC-layer acknowledgement is received after a non fragmented transmission,
- a broadcast/multicast frame is received.

**Long retry counter**: counts frames that are sent using the RTS/CTS mechanism.
This counter has a typical maximum value of 4.
It will be reset to 0 when:
- a MAC-layer acknowledgement is received from a frame longer than the RTS threshold,
- a broadcast/multicast frame is received.

In either of the cases, if the retry limit is reached, the frame is discarded and its loss is reported to higher-layer protocols.
MSDU Lifetime

Additionally, there is a lifetime counter in the MAC. This lifetime timer, which unlike the Retry Counter is associated to each MSDU, indicates the total amount of time that this packet may stay in the MAC Layer without being acknowledged as successfully sent since it is first put in the air.

Once the timer for a certain MSDU being transmitted has reached the value of dot11MaxTransmitMSDULifetime, it shall be discarded. If the MSDU had been fragmented, all of its fragments shall be discarded too.

B.2.2 Backoff mechanism

The total amount of contention time during which the medium must be sensed as free in DCF is the sum of DIFS and a random amount of time known as backoff as seen in figure 2.4. This backoff time is divided into slots and its length depends on the medium. Higher-speed PHY layers use shorter slot times.

For efficiency reasons, DCF employs a discrete backoff time scale. The random back-off interval is calculated as:

$$\text{int}(CW \cdot t_{\text{random}}) \cdot t_{\text{slot}}$$

Where each value refers to the following, though these may vary according to which transmission modulation is being used (FHSS, DSSS, IR, OFDM,...):

- **CW**: Contention Window integer between \([CW_{\text{min}}, CW_{\text{max}}]\). It defines a contention window which increases in powers of two less 1 (e.g. 1, 3, 7, 15, 31, 63, ...) with each collision. When a successful transmission occurs, this is set to the minimum value.
- **\(t_{\text{random}}\)**: is pseudo-random number between 0 and 1.
- **\(t_{\text{slot}}\)**: is equal to \(t_{\text{CCA}} + t_{\text{Rx/TxTurnaround}} + t_{\text{propagation}}\)
- **\(t_{\text{CCA}}\)**: is the CCA time that is required by the station to determine if the medium is idle.
- **\(t_{\text{Rx/TxTurnaround}}\)**: is the time it takes a node to change its state from Rx to Tx.

If there’s been a collision, then the ACK will not be received and a new DIFS + back-off time will be issued, but this time with a bigger CW value, and so on, to reduce the probability of new collisions.

The backoff time counter is decremented as long as the channel is sensed idle; this timer is frozen when a transmission is detected on the channel; and continued from where it was left when the channel is sensed idle again for more than a DIFS. The station transmits when the backoff timer reaches zero.
B.2.3 Fragmentation and Reassembly

The MAC layer may fragment an MSDU into more than one MAC Protocol Data Unit (MPDU) if the MSDU size exceeds the maximum data unit size allowed in the network or when it exceeds the fragmentation threshold. This way, only the first fragment will have to compete with other stations, but rest of the fragments will be sent using a waiting time of SIFS.

All fragments have the same frame sequence number but have ascending fragment numbers. It is used to improve the reliability in high-level packets and large managements packets, i.e. to improve the throughput by reducing the amount of data that can be corrupted by interference.

Fragmentation may also be used to fit an MSDU into multiple TXOP intervals that are not long enough to send that MSDU in a single period. Overall, some extra conditions and criteria are defined in the standard which must be met when using fragmentation:

- Frames that are never fragmented even if their length exceeds the threshold are:
  - Broadcast/multicast frames

![Fragmented MSDU](image1)

**Figure B.3:** Fragmented MSDU [3]

![Transmission of a fragmented MSDU](image2)

**Figure B.4:** Transmission of a fragmented MSDU [18]
− A-MSDU frames
− Frames which are using the HT-Immediate or HT-Delayed Block Ack

- Fragmented frames must not be larger than the fragmentation threshold, unless encryption is also being used, nor it shall exceed the TXOP limit in HCF mode.
- All fragments shall be equal in size except the last one which may be smaller.
- All fragments except for the last one must have an even number of octets.
- Each fragment is independently Acknowledged, allowing for only fragment retransmission to occur instead of resending the entire MSDU.
- Fragments are sent as a burst using a single medium access procedure separated by SIFS/RIFS.
- If a transmission fails, the station must wait back-off time and then send the erroneous MPDU or fragment again.
- The receiving station is responsible for discarding duplicate fragments.
- The sequence number remains the same for all fragments of a single MSDU.
- The fragments shall be transmitted in order from the lowest to highest number.

**Fragmentation Threshold**

The default value for this threshold is the maximum MSDU size allowed for that STA, i.e. fragmentation is disabled. But to find an optimal threshold, at least the following considerations need to be taken into account:

- Small values ⇒ small MSDU ⇒ large overhead
- High values ⇒ bigger MSDU ⇒ higher error rate ⇒ bandwidth waste
- It shall not be greater than the Max MPDU Length, e.g. 2346 Bytes for 802.11 MPDU.
- Can be limited by the TXOP limit assigned by the HC.

If a $L$ bytes long MSDU, where the header has $H$ bytes, is divided in $j$ fragments of $L_{opt}$ length, then the overhead presented by the fragmentation scheme is [35]:

$\text{Overhead} = (j - 1)(H + 2\text{SIFS} + \text{ACK})$. Therefore choosing an optimal Fragmentation threshold is a must. The effects of an inappropriate fragmentation threshold have been studied in multiple papers, but they are out of the scope of this study.
Receive MSDU Lifetime

Besides the MSDU Transmit Lifetime, explained in B.2.1, the use of fragmentation adds a new Lifetime Timer to each MSDU. This is the aMaxReceiveMSDULifetime. This timer is located in the receiving node and it starts counting when an MSDU is being received. For un-fragmented MSDUs it should have no effect, but when multiple fragments are being received, if the timer expires, the following actions shall be taken:

1. Discard all previously received fragments belonging to this MSDU
2. Upon reception of further fragments of the same MSDU, the receiving node must:
   ■ Acknowledge the fragment and
   ■ Discard the fragment without trying to reassemble the MSDU.

B.2.4 RTS/CTS

The RTS/CTS mechanism is used to protect a transmission from possible collisions, such as where a hidden node problem may exist. When using the RTS/CTS mechanism a station that wants to transmit a packet, waits until the channel is sensed idle for a DIFS, follows the backoff rules explained in B.2.2, and then, instead of the packet, preliminarily transmits a special short frame called request to send (RTS).

■ The transmitting station sends a RTS frame to the receiving station
■ After a SIFS interval, the receiver broadcasts a CTS frame
■ All other stations use the received CTS to setup their NAV (Network Allocation Vector) to the length of the transmission packet. NAV - Network Allocation Vector containing the information of the period of time in which the channel will remain busy.
■ After receiving the CTS and waiting an interval of SIFS, the transmitter sends the data and waits for an ACK to be received.

RTS/CTS mechanism is designed to be a better approach to CSMA/CD used in Ethernet, where transmission is cut when a collision occurs. The main problem in the basic access mechanism is that for large MPDUs if packets collide then the stations will keep transmitting and will not acknowledge a collision until their ACK has timed out. This translates not only in a time during which the channel bandwidth is wasted, but also in a higher collision rate due to the presence of longer packets colliding.

Therefore a threshold is decided above which all packets must be sent using the
RTS/CTS access mechanism, which in exchange will translate into less wasted bandwidth for large packets above an optimal threshold after exchanging some short RTS/CTS frames.

![Image of RTS/CTS access mechanism]

**Figure B.5:** Using the NAV for virtual carrier sensing [18]

When fragmentation must be used in combination with RTS/CTS, the NAV value is carried and updated in each frame sent (data and ACK) until the frame exchange is completed. The last ACK sent by the receiving station carries a NAV value of 0.

![Image of NAV usage when using Fragmentation and RTS/CTS]

**Figure B.6:** NAV usage when using Fragmentation and RTS/CTS [18]

### B.3 PCF: Point Coordination Function

The PCF in the 802.11 is a polling-based protocol, which was designed to support collision free and real time services. The implementation of PCF is optional. We have not extended further the information regarding PCF as it is out of the scope of this study.

### B.4 HCCA: HCF Controlled Channel Access

This access scheme is very similar to the previous PCF but unlike it, the HCCA allows for a CFP to be initiated anytime during a CP.
During a CP all stations work in EDCA mode, but whenever the HC, which is usually located in the AP, wants, it can initiate a CAP. A CAP is the same as a CFP and so the decision to initiate one is taken by sending CF-Poll frame to each station, i.e. HCCA is also based on a polling mechanism, but now the AP is allowed to initiate one during the CFP.

Also, unlike PCF, HCCA supports per-session service thanks to TSPEC and Traffic Stream (TS). Additionally, the HC may assign a TXOP to the station which will allow it to transmit a burst of packets separated by SIFS/RIFS. HCCA is one of the most complex and advanced coordination functions, and thus allows for extremely precise QoS applications, although its implementation is not mandatory for 802.11e APs.

### B.4.1 TXOP in HCCA

The TXOP interval is defined by the HC by designating a starting time and the maximum duration of this interval. This limit is usually expressed in a 11-bit field in the unit of 32s, the maximum being 65536s. The maximum TXOP size must not exceed the TBTT which is usually 100ms.

### B.5 Frame Format

#### B.5.1 General Frame Format

A common frame generated by the MAC layer is as follows:

![Figure B.7: 802.11n General Frame Format](image)

The minimum MAC frame format has the following structure: [Frame Control, Duration/ID, Address1, FCS], rest of the fields being optional or specific to certain frame exchanges.

The standard does also specify the maximum MSDU size at 2304 bytes, and for 802.11n that allows frame aggregation, two maximum A-MSDU values of 3839 or 7935 bytes depending on the stations capability.
Frame Control

This 2 byte long field is formed by:

**Protocol version:** it indicates which version of the 802.11 MAC is contained in the rest of the frame.

**Type and subtype fields:** a number of managements functions are incorporated in the 802.11 MAC.

**ToDS and FromDS:** DS indicates Distribution System. All frames on infrastructure networks will have one of the DSs bits set.

**More fragment bit:** if the data or management transmission used fragmentation, then this bit is set to 1 in all fragments except the last one.

**Retry bit:** if this bit is set 1 is to aid the receiving station in eliminating duplicate frames.

**Power Management bit:** it is used to conserve battery life. If it is set to 0 it is active and if it is set to 1 it is in a save mode.

**More data bit:** APs put this bit to 1 to indicate that at least, one frame is available and is addressed to a dozing station.

**WEP bit:** Wired Equivalent Privacy. If a frame needs to be processed by WEP, it will be set to 1.

**Order bit:** when the strict ordering delivery is employed, this bit is set to 1 in all non-QoS data frames.

**Duration/ID Field**

This field may be used for one of three purposes:

- To send the NAV value during which this transmission would like to reserve the medium.
For frames transmitted during a CFP, therefore its value is read as a NAV Value.

To send the address ID within a PS-Poll frame when using the Power Saving (PS) mode.

Address Fields

You can have maximum four addresses, and these are numbered because different fields are used for different purposes depending on the frame type. The number of address fields used also depends on the type of frame.

In general, address one is for the receiver, address two is for the transmitter and address three is for filtering by receiver. Each of these addresses has 48-bits which correspond to the MAC address to which they correspond.

The first bit of these addresses can be:

- 0: the address represents a single station (unicast).
- 1: the address represents a group of physical stations (multicast).

If all bits are set to 1, it indicates a broadcast frame.

Sequence Control Field

It is a 16-bits field and it is used both for defragmentation and for discarding duplicate frames. We can decompose these 16-bits into:

- 4-bits fragment field: it is different for all the fragments, and limits to 16 the number of fragments an MSDU may be cut into.
- 12-bits sequence number field: it operates like a counter of frames transmitted. It is the same for all the fragments.

The fragment number field should be set to 0 when fragmentation is not being used for the current transmission.
QoS Control Field

As seen in the general frame format in B.7, the 802.11e adds a new field to the MPDU frame. The QoS control field format is a 2 byte long field and the usage of its bits are described in the table below:

<table>
<thead>
<tr>
<th>Applicable frame (sub) types</th>
<th>Bits 0–3</th>
<th>Bit 4</th>
<th>Bit 5–6</th>
<th>Bit 7</th>
<th>Bits 8–15</th>
</tr>
</thead>
<tbody>
<tr>
<td>QoS CF-Poll and QoS CF-Ack+CF-Poll frames sent by HC</td>
<td>TID</td>
<td>EOSP</td>
<td>Ack Policy</td>
<td>Reserved</td>
<td>TXOP Limit</td>
</tr>
<tr>
<td>QoS Data+CF-Poll and QoS Data+CF-Ack+CF-Poll frames sent by HC</td>
<td>TID</td>
<td>EOSP</td>
<td>Ack Policy</td>
<td>A-MSDU Present</td>
<td>TXOP Limit</td>
</tr>
<tr>
<td>QoS Data and QoS Data+CF-Ack frames sent by HC</td>
<td>TID</td>
<td>EOSP</td>
<td>Ack Policy</td>
<td>A-MSDU Present</td>
<td>AP PS Buffer State</td>
</tr>
<tr>
<td>QoS Null frames sent by HC</td>
<td>TID</td>
<td>EOSP</td>
<td>Ack Policy</td>
<td>Reserved</td>
<td>AP PS Buffer State</td>
</tr>
<tr>
<td>QoS Data and QoS Data+CF-Ack frames sent by non-AP STAs</td>
<td>TID</td>
<td>0</td>
<td>Ack Policy</td>
<td>A-MSDU Present</td>
<td>TXOP Duration Requested</td>
</tr>
<tr>
<td>TID</td>
<td>1</td>
<td>Ack Policy</td>
<td>A-MSDU Present</td>
<td>Queue Size</td>
<td></td>
</tr>
<tr>
<td>QoS Null frames sent by non-AP STAs</td>
<td>TID</td>
<td>0</td>
<td>Ack Policy</td>
<td>Reserved</td>
<td>TXOP Duration Requested</td>
</tr>
<tr>
<td>TID</td>
<td>1</td>
<td>Ack Policy</td>
<td>Reserved</td>
<td>Queue Size</td>
<td></td>
</tr>
</tbody>
</table>

**Table B.1:** *QoS Control field [4]*

Bits 0-3: The TID subfield in the QoS Control field can indicate:

- For QoS Data + CF-Poll: it indicates the TID of the data.
- For QoS + CF-Poll: it indicates the TID for which de poll is intended.

Bits 5-6: identify the ACK policy that is followed upon the delivery of this MPDU. The four possible options are:
Performance improvement strategies for current and next generation Wi-Fi systems

Aalborg University
RATE Section

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| 00    | Normal Ack or Implicit Block Ack Request.  
In a frame that is a non-A-MPDU frame:  
The addressed recipient returns an ACK or QoS + CF-ACK frame after a SIFS period. For QoS Null (no data) frames, this is the only permissible value for the ACK Policy subfield.  
In a frame that is part of an A-MPDU:  
The addressed recipient returns a Block Ack MPDU, either individually or as part of an A-MPDU starting a SIFS after the PPDU carrying the frame. |
| 10    | No Ack.  
The addressed recipient takes no action upon receipt of the frame.  
The ACK Policy subfield is set to this value in all directed frames in which the sender does not require acknowledgment.  
This combination is also used for group-addressed frames that use the QoS frame format. This combination is not used for QoS data frames with a TID for which a Block Ack agreement exists. |
| 10    | No Explicit Acknowledgment or PSMP Ack.  
Mainly used in polling based access schemes or when PSMP is in use. |
| 11    | Block Ack.  
The addressed recipient takes no action upon the receipt of the frame except for recording the state.  
The recipient can expect a BlockAckReq frame in the future to which it responds with a BA Response or ACK+SIFS+BA Response if Delayed BA is being used. |

Table B.2: ACK Policy subfield

Bit 7: must be set to 1 when using A-MSDU (see 2.4.3) under certain QoS conditions. On the contrary if this field is set to 0, the MAC layer will recognize the frame as a single MSDU or fragment of an MSDU. Rest of the subfields are not further detailed as they lack interest for this thesis.

HT Control Field

![Figure B.9: HT Control Field format [4]](image)

This field is used in the Control Wrapper frame, as well as QoS data and management frames for High Throughput Control in 802.11n
Frame Body

This is the only variable field found in a Wi-Fi packet. This field carries the payload or information for that type of frame. If it is a Beacon frame, for instance, it will carry the various features supported by that network; whereas if it is a data frame, its size depends on the size of the MSDU or A-MSDU that is being carried as the payload.

The maximum payload size is delimited by the maximum MSDU or A-MSDU size defined in each of the versions of Wi-Fi.

FCS: Frame Check Sequence

It is often referred to as the cyclic redundancy check (CRC) because of the underlying mathematical operations. It allows all the stations to check the integrity of received frames. When frames are sent to the wireless interface, the FCS is calculated before those frame are sent over the RF or IR link. Receivers can then calculate the FCS. If the calculated value and the received FCS value are the same, the frame has a high probability to be correct. Correct frames must be acknowledged or they will be retransmitted due to expiration of the ACK timeout.

Broadcast and Multicast Data or Management Frames transmission

These are the simplest of all transmissions because neither of these require an acknowledgement, thus once transmitted, the transmission is immediately noted as successful. These frames cannot be fragmented and their NAV value is 0.

B.5.2 Data

Data frames follow the same format as the general frame format, except for the usage of the Address fields that may vary its content depending on who it is addressed to. The QoS and HT control fields may or may not be present also depending on to whom this frame is being sent.

MPDU Preparation chain

When a Data transmission occurs, the frame being sent needs to go through a preparation flow first. This preparation flow diagram is shown in the following figure included in the standard:
B.5.3 A-MSDU Frame

An A-MSDU frame is made out of a sequence of A-MSDU subframes to increase the throughput in newer versions of the standard. Each of these subframes contain an MSDU, whose maximum supported size is 2304 bytes, and all of them, except for the last one, will be padded to match a multiple of 4 octets length.
Since an A-MSDU is treated like any other MSDU at lower levels of the MPDU preparation chain, a single MAC Header is added to the whole A-MSDU frame, which means that it carries a single Sequence number for all of the MSDUs carried inside the A-MSDU.

According to the standard, the A-MSDU frame must meet the following criteria:

- All MSDUs inside it must have the same DA and SA
- All MSDUs inside it must use the same ACK policy
- The lifetime timer of an A-MSDU expires when all MSDU lifetime timers inside it have expired, i.e. only the longest of the lifetime timers is taken into account.

**A-MSDU sizes**

The maximum MSDU size supported in 802.11n is 2304 Bytes, as seen in the A-MSDU subframe. Each A-MSDU is carried in a single MPDU, which typically has a maximum frame body of 7955 Bytes. Therefore up to two different A-MSDU maximum lengths are supported by setting the HT Capabilities elements Maximum MSDU Length subfield appropriately:

- Bit=0: Max A-MSDU is set to 3839 Bytes
- Bit=1: Max A-MSDU is set to 7935 Bytes

If A-MPDU is used then the max MPDU size is limited by the 12-bit A-MPDU Length field, i.e. Max A-MPDU is 4095 Bytes. Therefore if an A-MSDU is being carried inside an A-MPDU, its size must not exceed 4065 Bytes, that is 4095 minus the QoS MPDU overhead, since the A-MSDU cannot be fragmented.
B.5.4 A-MPDU frame

![A-MPDU frame format](image)

Each A-MPDU subframe, except the last one, is padded with octets to make each one multiple of 4 octets in length. Its maximum length is limited by the Length field in the PHY Header, which is 65535 bytes for 802.11n and 1048575 bytes in the 802.11ac draft.

![A-MPDU frame format](image)

The maximum MPDU length is 4095 Bytes contained by each A-MPDU subframe, which is limited by the MPDU Length field.

The maximum number of MPDUs contained in a single A-MPDU is 64, limited by the Block Ack response frame, which allows only up to 64 MPDUs to be acknowledged at once.

The Duration/ID field in all MPDUs carried by a single A-MPDU is assigned to the same value.

The Delimiter signature is used to locate the MPDUs within the A-MPDU and its value is fixed to the ascii value of the character N, i.e. 0x4E.

Also all MPDUs contained in the A-MPDU must be addressed by and to the same Source and Destination Addresses, as well as the ACK policy in all of them has to be the same. In this last case, the Normal Ack policy of the MPDUs is translated to an Implicit Block Ack Request, which will translate in a Block Ack Response from the recipient sent a SIFS interval after receiving the PPDU carrying the A-MPDU.

B.5.5 Control

Control frames are such those that are preceded by a SIFS interval and present the following Frame Control field configuration:
Performance improvement strategies for current and next generation Wi-Fi systems

In this document we will concentrate on some of the most commonly used Control frames. To find more information about the rest of them we can refer to the complete standard revision of 2007.

**RTS frame**

In this frame the duration value is the time in s required to transmit the management frame, a CTS frame, an ACK and the three SIFS in between them. This duration is then used by the VCS to determine the NAV.

![RTS Frame Format](image)

**CTS frame**

This is the shorter frame that responds to a RTS frame, where the duration is taken from the previous RTS frame and the RA is now the same as the TA in the RTS frame.

![CTS Frame Format](image)

**ACK frame**

This frame is used to answer with a positive acknowledgment to a previous data, management or even to some of the control frame types. In this frame, the RA field is filled with the information of Address 2 of the previous frame.
Performance improvement strategies for current and next generation Wi-Fi systems

Aalborg University
RATE Section

Figure B.17: ACK frame format [3]

BlockACKReq

This type of frame is used in Block Ack sessions as defined in 802.11e (see 2.3.5). Where the Block Ack Starting Sequence control refers to the first MSDU sequence number for which this Block Ack is being requested.

Figure B.18: BlockACKReq frame format [4]

Block Ack

This block is sent in response to a previous BlockACKReq:

Figure B.19: Block Ack frame format [4]

Where the Block Ack control field corresponds to the following bit pattern:

Figure B.20: Block Ack Control Field format [4]
In newer standards, such as the 802.11n [4], the Block Ack information field is variable depending on which variant of the Block Ack frame is being used.

Two main variations of this BA Information field are used in 802.11n:

**Basic Block Ack:** the bitmap contained allows to acknowledge up to 64 MSDUs that may be fragmented in up to 16 fragments each. So it uses $64 \times 16 = 1024$ bits = 128 bytes.

![Basic Block Ack Information field](image)

**Compressed Block Ack:** The higher throughput achieved in 802.11n is mostly achieved by aggregation, thus defeating the purpose of using fragmentation and creating an excessive overhead in the Block Ack response. This smaller information field carries only $64 \times 1 = 64$ bytes.

![Compressed Block Ack Information field](image)

**Multi-TID Block Ack:** For this particular frame format, the BA Control field sets its TID field to the number of TIDs being acknowledged less one. Also, in this case, the BA Information Field may carry as many Multi-TID Info Fields as the number of TIDs being acknowledged, ordered in increasing values of TID. The Per TID Info subfield carries the actual TID value being acknowledged in that information field. As the BA Bitmap included in this information field is 8 byte long, the use of fragmentation within Multi-TID Block Ack is deprecated.

![Multi-TID Block Ack Information field](image)

In either cases the Block Ack starting sequence Control subfield carries the sequence number of the first MSDU or A-MSDU being acknowledged.
B.5.6 Management

The numerous types of management frames are out of the study of this work. Most of the management frames are used to do certain handshake procedures such as authentication or association in the network. These frames are transmitted the same way as data frames, but not always forwarded to upper layers as the information they carry does only concern to the 802.11 network architecture and is irrelevant to upper layers.

Beacon

One of the most common management frames is the Beacon frame, usually sent by the AP in a infrastructure network and by each station in an Ad-Hoc network. Some of the information carried by this type of frame is related to the physical layer interaction in the network, such as: Timestamp, Beacon interval, Capability, SSID, Supported throughput rates,...

HT Capabilities Element

<table>
<thead>
<tr>
<th>Element ID</th>
<th>Length</th>
<th>HT Capabilities Info</th>
<th>A-MPDU Parameters</th>
<th>Supported MCS Set</th>
<th>HT Extended Capabilities</th>
<th>Transmit Beamforming Capabilities</th>
<th>ASEL Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Octets:1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>16</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure B.24: General Management frame format [3]

Figure B.25: Beacon transmission mechanism in Infrastructure (left) and Ad-hoc (right) networks [36]

Figure B.26: HT Capabilities element format [4]
This element is one of the possible Management frame body components. Inside this element, there are two main subfields which are of our interest:

- **The HT Capabilities info subfield:** where the A-MSDU Length bit allows us to choose (in 802.11n stations) whether the maximum A-MSDU shall be 3839 or 7935 bytes long.

  \[
  \begin{array}{cccccccccc}
  & B0 & B1 & B2 & B3 & B4 & B5 & B6 & B7 & B8 & B9 \\
  \hline
  LDPC Coding Capability & Supported Channel Width Set & SM Power Save & HT-Greenfield & Short GI for 20 MHz & Short GI for 40 MHz & Tx STBC & Rx STBC \\
  \end{array}
  \]

  \[
  \begin{array}{cccccccc}
  & B10 & B11 & B12 & B13 & B14 & B15 \\
  \hline
  HT-Delayed Block Ack & Maximum A-MSDU Length & DSSS/CCK Mode in 40 MHz & Reserved & Forty MHz Intolerant & L-SIG TXOP Protection Support \\
  \end{array}
  \]

  \textbf{Figure B.27: HT Capabilities Info subfield [4]} \[4\]

- **The A-MPDU Parameters subfield:** which ensures A-MPDU compatibility between receiving and transmitting stations.

  \[
  \begin{array}{cccc}
  & B0 & B1 & B2 & B4 & B5 & B7 \\
  \hline
  Maximum A-MPDU Length Exponent & Minimum MPDU Start Spacing & Reserved \\
  \end{array}
  \]

  \textbf{Figure B.28: HT Capabilities A-MPDU Parameters subfield [4]} \[4\]
Appendix C.

Power Saving in Wi-Fi

C.1 802.11-2009: Power Save Multi-Poll

This polling mechanism is inherited from the previous Unscheduled Automatic Power Save Delivery (U-APSD) and Scheduled Automatic Power Save Delivery (S-APSD) that was introduced by the 802.11e. PSMP frames are transmitted to stations that are awake at that moment. Therefore, the main difference with the 802.11e amendment is that in this case multiple stations may operate on a group schedule basis rather than individually controlling each station. Some of the benefits that PSMP presents are [25]:

- Retransmission of erroneous packets in the same SP
- Multiple data packets for different applications can be aggregated in a single packet for DL/UL
- Support for burst traffic using multiple PSMP frames in the same SP.

Typically, a Service Period (SP) will terminate once the scheduled UL and DL transfers have been completed, but the AP may terminate a SP for each station by setting the More PSMP field to 0, or by sending a CF-End frame.

![PSMP frame exchange sequence](image)

Figure C.1: PSMP frame exchange sequence
References


Performance improvement strategies for current and next generation Wi-Fi systems


Performance improvement strategies for current and next generation Wi-Fi systems


