Analytical Modeling of Medium Access Control in Finite-Load and Saturated Wireless LANs

by

Samer El Housseini

B.E., American University of Beirut, 2001
M.Sc., North Carolina State University, 2002

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

July 2006

©Samer El Housseini, 2006
Abstract

The IEEE 802.11 standard provides the specifications for a Wireless Local-Area Network (WLAN) technology, commonly known as Wi-Fi. IEEE 802.11 uses an Ethernet-like contention-window mechanism to resolve the multiple-access problem to the wireless channel. Essentially, a wireless station doubles its contention window size after detecting a frame collision. Although this method is effective in reducing collisions on the channel, it increases packet overhead which results in reducing the throughput.

In general, the performance of the WLAN varies widely depending on the number of customers in the network coverage area and the shape of the traffic on the channel. A few analytical models have been proposed over the past few years to understand the behavior of WLANs. Although insightful, most of these models were based on a highly-simplified "saturated" networks model in which all wireless stations behave like traffic source with infinite number of frames to transmit. The saturation assumption is useful in that it leads to simple steady-state models with fixed transition probabilities. However, it is not realistic to assume that wireless stations are always attempting to transmit frames, in real networks.

This thesis is concerned with the development of improved analytical models for non-saturated, or finite-load, WLANs and also propose enhancements to existing saturation
WLAN models. In particular, we have developed two analytical models for WLANs with finite load. One model for the standard IEEE 802.11 WLAN and the second for the quality-of-service enabled WLAN described by the IEEE 802.11e standard.

The proposed models capture the probabilistic nature of wireless networks and the interdependencies among wireless stations. In our analysis, we rely on novel schemes that use coupled station-view and network-view models to compute the overall throughput, collision probability, and delay in a WLAN. When compared to other recent work, our results prove to be the most accurate.

We complement our work on finite-load models by presenting a more accurate model for IEEE 802.11e WLANs operating under saturation, and propose a few adaptive contention-window algorithms for maximizing the frame transmission rate and consequently the saturation throughput. We show through simulation that the proposed algorithms increase the throughput by several multiples.
Table of Contents

Abstract ................................................................. ii
Table of Contents ............................................................ iv
List of Tables ................................................................. vii
List of Figures ................................................................. viii
List of Symbols ............................................................... x
List of Abbreviations ......................................................... xiii
Acknowledgements ............................................................ xv
Dedication ..................................................................... xvii
Co-Authorship Statement ..................................................... xvii

1 Introduction ................................................................. 1
1.1 Motivation ............................................................... 1
1.2 Performance Issues in Wireless Local Area Networks .............. 2
  1.2.1 Standard 802.11 Wireless Networks ......................... 2
  1.2.2 QoS in Wireless Networks ................................. 3
1.3 Contributions ........................................................... 3
1.4 Thesis Outline .......................................................... 7

2 Overview of Wireless Local Area Networks ................................ 8
2.1 Overview the IEEE 802.11 MAC Protocol .......................... 9
2.2 The IEEE 802.11e MAC Protocol Extension ........................ 11
2.3 Mathematical Modeling of Wireless Networks ..................... 14
2.4 Finite-Load Models for IEEE 802.11 and 802.11e WLANs ........ 15
2.5 IEEE 802.11e WLAN Saturation Models and Algorithms ........ 19
2.6 The Simulator ........................................................ 21
2.7 Concluding Remarks .................................................. 23

3 An Enhanced Finite-Load Model for IEEE 802.11 Wireless LANs .... 24
3.1 Introduction ........................................................... 24
3.2 System Model and Assumptions ..................................... 26
3.3 Basic Analysis ......................................................... 28
TABLE OF CONTENTS

3.4 The Tagged Node Queuing Model ........................................... 33
  3.4.1 Service Time Estimation ............................................. 33
  3.4.2 Service Time Distribution (Tagged Node Queuing Model) ...... 37
  3.4.3 Throughput and Delay .............................................. 41
  3.4.4 Bursty Source ..................................................... 42
3.5 Whole Network Queuing Model ............................................. 43
  3.5.1 Estimation of the Network Service Time and Utilization Rate . 43
  3.5.2 Estimation of Pidle and Pbussy ................................... 45
3.6 Multi-Load and Multi-Rate Wireless LANs ............................... 46
  3.6.1 Multi-Load Wireless LANs ......................................... 46
  3.6.2 Multi-Rate Wireless LANs ......................................... 48
3.7 Simulation versus Numerical Results .................................... 49
  3.7.1 Numerical Example of the Service Time .......................... 50
  3.7.2 Results ............................................................ 50
3.8 Conclusion ............................................................... 53

4 A Finite-Load Model for QoS-Enabled IEEE 802.11e WLANs ............ 60
  4.1 Introduction ..................................................................... 60
  4.2 System Model and Assumptions ......................................... 63
  4.3 Basic Analysis ........................................................... 64
    4.3.1 Estimation of the Collision Probability ......................... 64
    4.3.2 Different AIFS Periods and Slot Occupancy Weighting ....... 66
  4.4 Tagged Node Queuing Model ............................................ 69
    4.4.1 Service Time Estimation (Node Queuing Model) ............... 70
    4.4.2 Ratio Selection .................................................... 75
    4.4.3 Service Time Distribution ....................................... 80
  4.5 Whole Network Queuing Model ......................................... 82
  4.6 Extending the Model to Multiple Priorities .......................... 83
  4.7 Numerical Results ....................................................... 84
    4.7.1 Numerical Example for Service Time ........................... 85
    4.7.2 Results ............................................................ 85
  4.8 Conclusion ............................................................... 87

5 An Improved Saturation Model for IEEE 802.11e WLANs ................ 92
  5.1 Introduction ..................................................................... 92
  5.2 Model Development ...................................................... 94
  5.3 Slot Occupancy ........................................................... 96
  5.4 Throughput Analysis ..................................................... 99
  5.5 Simulation Results ....................................................... 101
## TABLE OF CONTENTS

### 6 Adaptive Contention-Window MAC Algorithms for IEEE 802.11e Wireless LANs

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Introduction</td>
<td>105</td>
</tr>
<tr>
<td>6.2 Effect of the CW Parameters on the Saturation Throughput</td>
<td>106</td>
</tr>
<tr>
<td>6.3 MAC Adaptation Algorithms</td>
<td>111</td>
</tr>
<tr>
<td>6.3.1 The Station-Count Algorithm (SCA)</td>
<td>111</td>
</tr>
<tr>
<td>6.3.2 The Throughput Derivative Algorithm (TDA)</td>
<td>113</td>
</tr>
<tr>
<td>6.3.3 The Analytical Algorithm</td>
<td>116</td>
</tr>
<tr>
<td>6.4 Simulation Results</td>
<td>117</td>
</tr>
<tr>
<td>6.5 Conclusion</td>
<td>119</td>
</tr>
</tbody>
</table>

### 7 Conclusions and Future Research

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>126</td>
</tr>
</tbody>
</table>

### Appendices

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A MATLAB Algorithm for Solving Finite-Load Models - The Non-Linear Set of Equations</td>
<td>138</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Numerical Example of the components of the Service Time ............. 50
3.2 Simulation and Mathematical Models Parameters ...................... 51

4.1 Successes and Collisions of the (3,3) pair combinations ............. 78
4.2 Numerical Example of the components of the Service Time ........... 85
4.3 Simulation and Mathematical model Parameters ....................... 86

5.1 Simulation Parameters for Batch 1 ................................ 102
5.2 Simulation Parameters for Batch 2 ................................ 102

6.1 Simulation parameters ............................................. 108
List of Figures

1.1 Finite Load Model Architecture .......................................................... 5
2.1 OSI 802.11 Layer Model [5] ................................................................. 9
2.2 AIFS Relationships [7] ................................................................. 13
2.3 EDCA TXOP [8] ................................................................. 14
2.4 Exponential Distribution Error ............................................................. 22
3.1 Tagged Node Queuing Model ................................................................. 26
3.2 Network Queuing Model ........................................................................ 27
3.3 Finite Load Model Architecture ............................................................. 29
3.4 Service Time Distribution Approximation for Typical $p$ ......................... 38
3.5 Service Time Distribution Approximation for Large $p$ ......................... 39
3.6 Utilization vs. load for WLAN with 4, 12, and 20 stations .................... 54
3.7 Collision probability, $p$ vs. load for WLAN with 4, 12, and 20 stations ... 55
3.8 Service time vs. load for WLAN with 4, 12, and 20 stations .................. 56
3.9 Waiting time $T_Q$ vs. load for WLAN with 4, 12, and 20 stations .......... 57
3.10 Waiting time $T_Q$ vs. load for WLAN with 4 stations with Batch Arrivals . 57
3.11 Key Quantities vs. load for a Multi-Load WLAN with 4 stations .......... 58
3.12 Waiting time $T_Q$ vs. load for a Multi-Load WLAN with 4 stations .......... 59
4.1 Tagged Node Queuing Model ................................................................. 62
4.2 Network Queuing Model ........................................................................ 62
4.3 Slot Occupancy Markov Chain (taken from [3]) ..................................... 67
4.4 Demonstration of the Class Ratio .......................................................... 76
4.5 Utilization vs. load for WLAN with 4, 12, and 20 stations .................... 88
4.6 Collision probability, $p$ vs. load for WLAN with 4, 12, and 20 stations ... 89
4.7 Service time vs. load for WLAN with 4, 12, and 20 stations .................. 90
4.8 Waiting time, or Delay, $W_Q$ for Priority A vs. load for WLAN with 4, 12, and 20 stations .......................................................... 91
5.1 Modeling Slot Occupancy for different contention zones ......................... 97
5.2 Collision Probability vs. Number of Stations for Batch 1 .................................. 103
5.3 Saturation Throughput vs. Number of Stations for Batch 1 ................................. 103
5.4 Collision Probability vs. Number of Stations for Batch 2 .................................. 104
5.5 Saturation Throughput vs. Number of Stations for Batch 2 ................................. 104

6.1 Effect of CW\textsubscript{MIN} and CW\textsubscript{MAX} on the Saturation Throughput .................................. 109
6.2 Effect of CW\textsubscript{MIN} and CW\textsubscript{MAX} on the Probability of Collision ................. 110
6.3 The TDA Operation Zones ................................................................. 115
6.4 Static vs. Station-Count Algorithm - Throughput .............................................. 121
6.5 Static vs. Station-Count Algorithm - CW\textsubscript{MIN} Change ................................. 121
6.6 Static vs. Throughput Derivative Algorithm - Throughput .................................. 122
6.7 Static vs. Throughput Derivative Algorithm - Low Pass Filtered Throughput ........ 122
6.8 Static vs. Throughput Derivative Algorithm - Derivative .................................... 123
6.9 Static vs. Throughput Derivative Algorithm - CW\textsubscript{MIN} Change .................... 123
6.10 Static vs. Analytical Algorithm for Priority A .................................................. 124
6.11 Static vs. Analytical Algorithm for Priority B .................................................... 124
6.12 Proof of Optimization for the Analytical Algorithm ........................................... 125
List of Symbols

\( p \) Probability of Collision
\( N \) Number of Wireless Stations or Nodes
\( \lambda \) Frame Arrival Rate
\( P[NT] \) Probability of No Transmission
\( \mu \) Tagged Node Service Rate
\( \mu_{net} \) Wireless Medium Service Rate
\( \rho \) Tagged Node Utilization Rate
\( \rho_{net} \) Wireless Medium Utilization Rate
\( \tau \) Probability of Transmission when Wireless Station is not empty
\( P_{\text{backoff}} \) Portion of Frames going through Backoff
\( CW_{\text{min}} \) Minimum Contention Window
\( CW_{\text{max}} \) Maximum Contention Window
\( \bar{W}_p \) Average Backoff Counter Value excluding Direct Transmissions
\( \bar{W} \) Average Backoff Counter Value including Direct Transmissions
\( I_w \) Average Backoff Contention Window Size
\( P_{\text{busy}} \) Probability of Busy Wireless Medium
\( P_{\text{idle}} \) Probability of Idle Wireless Medium
\( CycleCx \) Average Number of Transmission Cycles Ending in Collisions
\( nodeCx \) Average Number of Collision Slots involving the Tagged Node
\( queueCx \) Average Number of Collision Slots involving the Tagged Queue
\( otherCx \) Average Number of Collision Slots from Other Nodes
\( \text{backoff} \) Average Number of Backoff Slots
\( nodeTx \) Average Number of Successful Transmission Slots from the Tagged Node
**LIST OF SYMBOLS**

- \( queueTx \) \quad \text{Average Number of Successful Transmission Slots from the Tagged Queue}
- \( otherTx \) \quad \text{Average Number of Successful Transmission Slots from Other Nodes}
- \( P_x \) \quad \text{Portion of Frames Arriving to an Empty Queue with Busy Medium}
- \( P_x^b \) \quad \text{Portion of Frames Arriving to an Empty Queue \( x \) with Busy Medium}
- \( P_{otherCx} \) \quad \text{Average Number of Collisions from Other Nodes}
- \( P_{otherTx} \) \quad \text{Average Number of Successful Transmissions from Other Nodes}
- \( p_o \) \quad \text{Portion of Transmissions from Other Nodes that results in Collisions}
- \( T_s \) \quad \text{Number of Slots that a Successful Transmission takes to complete}
- \( T_c \) \quad \text{Number of Slots that a Collision takes to complete}
- \( b(t) \) \quad \text{Proposed Service Time Distribution of the Tagged Node}
- \( b_{net}(t) \) \quad \text{Proposed Service Time Distribution of the Wireless Medium}
- \( B(s) \) \quad \text{Laplace Transform of the Proposed Service Time Distribution}
- \( Q(z) \) \quad \text{PGF of the Queue Length Distribution of the Tagged Node}
- \( Q_{net}(z) \) \quad \text{PGF of the Queue Length Distribution of the Wireless Medium}
- \( \pi_k \) \quad \text{Queue Length Probabilities of the Tagged Node}
- \( \pi_{net_k} \) \quad \text{Queue Length Probabilities of the Wireless Medium}
- \( N_q \) \quad \text{Average Queue Length}
- \( V_{\mu} \) \quad \text{Variance of the Average Service Time}
- \( T_q \) \quad \text{Average Waiting Time}
- \( W_q \) \quad \text{Average Delay}
- \( P_j^N \) \quad \text{Probability of Counting \( j \) Slots before the first Backoff-Counter fires}
- \( T_{net}^N \) \quad \text{Expected Value of} \( P_j^N \)
- \( R_s \) \quad \text{Ratio of Idle Slots in a Wireless Medium Transmission Cycle}
- \( P_{r_{Zone_i}} \) \quad \text{Probability of Transmission in Contention Zone \( i \)}
- \( P_{pr_{Zone_i}} \) \quad \text{Probability of Transmission in Contention Zone \( i \)}
- \( P_{c_{Zone_i}} \) \quad \text{Probability of Collision of Priority \( x \) in Contention Zone \( i \)}
- \( P_{c_{pr_{Zone_i}}^x} \) \quad \text{Probability of Collision of Priority \( x \) in Contention Zone \( i \)}
- \( b_i \) \quad \text{Steady State Probabilities of the Slot Occupancy Markov Chain}
- \( b_{i,k} \) \quad \text{Steady State Probabilities of the Backoff Markov Chain}
- \( Ratio_{XY} \) \quad \text{Av. Number of Successful Transmissions from \( Y \) in \( X \)’s Service Time}
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\overline{P}_j$</td>
<td>Average Probability of Collision for Priority $j$</td>
</tr>
<tr>
<td>$E_D$</td>
<td>Expected Length of the Transmission Duration</td>
</tr>
<tr>
<td>$P^S_{z=i}$</td>
<td>Probability of any given Transmission being Successful</td>
</tr>
<tr>
<td>$P^S_{jz=i}$</td>
<td>Probability of Priority $j$’s Transmission being Successful</td>
</tr>
<tr>
<td>$P^c_{z=ol}$</td>
<td>Probability of any given Transmission being a Collision</td>
</tr>
<tr>
<td>$E_D^j$</td>
<td>Expected Length of the Transmission Duration</td>
</tr>
<tr>
<td>$E_p^j$</td>
<td>Expected Payload of Priority $j$</td>
</tr>
<tr>
<td>$T_j$</td>
<td>Average Throughput of Priority $j$</td>
</tr>
</tbody>
</table>
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Access Category</td>
</tr>
<tr>
<td>ACK</td>
<td>Acknowledge Frame</td>
</tr>
<tr>
<td>AIFS</td>
<td>Arbitration Inter-Frame Spacing</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>CFP</td>
<td>Contention-Free Period</td>
</tr>
<tr>
<td>CP</td>
<td>Contention Period</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access/Collision Avoidance</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>CW</td>
<td>Contention Window</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>DIFS</td>
<td>DCF Inter-Frame Spacing</td>
</tr>
<tr>
<td>DVD</td>
<td>Digital Video Disc</td>
</tr>
<tr>
<td>EDCA</td>
<td>Enhanced Distributed Channel Access</td>
</tr>
<tr>
<td>EFL</td>
<td>Enhanced Finite-Load</td>
</tr>
<tr>
<td>EIFS</td>
<td>Extended Inter-Frame Spacing</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>HCCA</td>
<td>HCF Controlled Channel Access</td>
</tr>
<tr>
<td>HCF</td>
<td>Hybrid Coordination Function</td>
</tr>
<tr>
<td>IFS</td>
<td>Inter-Frame Spacing</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>NAV</td>
<td>Network Allocation Vector</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PCF</td>
<td>Point Coordination Function</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Distribution Function</td>
</tr>
<tr>
<td>PGF</td>
<td>Probability Generating Function</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical Layer</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RTS</td>
<td>Request To Send</td>
</tr>
<tr>
<td>RV</td>
<td>Random Variable</td>
</tr>
<tr>
<td>SCA</td>
<td>The Station-Count Algorithm</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short Inter-Frame Spacing</td>
</tr>
<tr>
<td>ST</td>
<td>Saturation Throughput</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TDA</td>
<td>The Throughput Derivative Algorithm</td>
</tr>
<tr>
<td>TGe</td>
<td>IEEE 802.11 Task Group E</td>
</tr>
<tr>
<td>TGn</td>
<td>IEEE 802.11 Task Group N</td>
</tr>
<tr>
<td>TXOP</td>
<td>Transmission Opportunity</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice Over IP</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>IEEE 802.11 Wireless Local Area Network</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WWiSE</td>
<td>World-Wide Spectrum Efficiency</td>
</tr>
</tbody>
</table>
Acknowledgements

I would like to thank my supervisor, Dr. Hussein Alnuweiri, for all the guidance, support, and time that he offered. I firmly believe that without all his expertise and understanding, I wouldn't have completed my Ph.D. And for this, I am eternally grateful. I would like to thank the other members of my committee for the invaluable help they provided during my research journey: Dr. Victor Leung, Dr. Vikram Krishnamurthy, Dr. Vincent Wong, and Dr. Dave Michelson. Being totally honest, my experience with each and every professor was a very pleasant one.

I also received lots of support and guidance from my group members. And any work would be terribly boring without the friendly, sometimes too-much-fun-to-work, atmosphere in our lab. The friendships of these people mean a lot to me: Ayman Kaheel, Junaid Khan, Tamer Khattab, Yasser Pourmouhammadi, Tariq Al-Khasib, Amr Salem, Zaman Mollah, Salman Khan and Mahmood Minhas. I would also like to thank all the other people whose non-technical help made the writing of this thesis possible.
To my parents whose immense support was illogical by all scientific standards ...
Co-Authorship Statement

Chapter 5 has been taken from a paper that was submitted as an IEEE Communication Letter. The co-authors of this paper are Yasser Pourmouhammadi, Hussein Alnuweiri, and myself. As a research group, we have been working on modeling IEEE 802.11 and 802.11e WLANs, and this was a common area where we shared our knowledge and work. The research and model analysis have been done in long group sessions on a round table. Yasser wrote the greater part of the paper, while I implemented the mathematical model in C, generated the simulation data and plots, and finished writing the last part of the paper.
Chapter 1

Introduction

1.1 Motivation

The most common technology that allows wireless communication is a wireless local area network, also known as 802.11 network, WLAN, and wi-fi network. Grand Haven, Michigan has been the first city in the US to deploy a Wi-Fi network with full-city coverage in 2004 [1]. The metropolitan city of Philadelphia is comparing Wi-Fi networks to water and electricity networks, and is planning to cover its 135-square-mile neighborhoods with high speed wireless internet. The city of San Francisco is soon to follow [2].

With such massive and expansive deployments, wireless local area networks will certainly become a legacy technology in the future. But until that day, more research needs to be conducted to identify and harness the true limits of this technology. Academic and industrial research sometimes takes years to fine-tune a certain technology to befit the commercial market, forcing early releases to evolve over time.
In the complex phases of research and development, modeling is the most time-consuming albeit rewarding phase and can even outlast the design phase. Models are often built to foretell the behavior of the system under study, thereby giving designers the freedom to alter the design, without any further cost and time.

For all the reasons mentioned, we found it beneficial to build mathematical and simulation models for wireless local area networks, especially that the literature was lacking accurate models in this area. Also, with the new wireless revolution being multimedia-centric, we decided to extend our models to cover Quality-of-Service enabled WLANs, or 802.11e wireless networks.

1.2 Performance Issues in Wireless Local Area Networks

1.2.1 Standard 802.11 Wireless Networks

A wireless local area network is a network of stations that use radio frequencies, transmitting through the air as a means of communication. With today's wireless network cards, each station has a maximum range of about two hundred meters, and a varying bandwidth that depends on the communication hardware capabilities and the conditions of the wireless medium. Typical bit rates range from 11 Mbps for 802.11b networks, to 54 Mbps for 802.11g networks, and upwards of 100-200 Mbps in 802.11n (up to 540 Mbps for WWiSE and 630 Mbps for TGn).

A WLAN is normally deployed as the last-mile carrier of the traffic, with data traveling in wired networks until the last hop, which is then transmitted wirelessly. A WLAN could
also very well replace the backbone of the network, through a multi-hop wireless network, connecting both ends of the data exchange.

Wireless LANs functioning without an Access Point are called Ad-Hoc networks. These networks have special routing algorithms to allow the exchange of data frames between two wireless stations. Wireless LANs operating with an Access Point are called to be working in infrastructure mode. The Access Point acts as the center of the network. Since it often has double the signal range, all frames are forwarded to the Access Point, which then sends the frames to their final destination.

1.2.2 QoS in Wireless Networks

Since the approval and standardization of the 802.11e specification in 2005, Quality-of-Service (QoS) in WLANs has become more reality, rather than just an optional extension. Recent trends of the industry have placed more emphasis on voice and video transmission in laptops and hand-held devices, such as cell phones and PDAs. The more illustrious example is the recently manufactured Wireless DVD Player. Towards this purpose, the 802.11e standard defines multiple classes of service in the network traffic in WLANs. These classes are treated with different priorities, such as setting higher priorities to voice and video traffic over data traffic, to ensure its correct and timely transmission and reception of time-critical multimedia services.

1.3 Contributions

The major contributions of our research are:
1. The development of a new accurate mathematical model for 802.11 networks under the finite load condition called Enhanced Finite-Load Model. The proposed model is, in our view, the most accurate in the literature, as will be illustrated by the results presented in Chapter 3. Many papers tried to tackle the finite load problem while focusing on only one aspect of their model, while over-simplifying the rest. In general, the studies followed one of four different trends: a few studies tried to solve it as a queuing problem with over-simplified averages, while others used only average quantities to solve a non-linear set of equations that describe the system. And out of both, some argued for using the wireless station as the system under study, while others insisted on studying the wireless medium. In Figure 1.1 which represents our work, we show that by merging the four perspectives, each represented by its own block, we have a logically more complete model.

The key contribution of this thesis is the development of a highly detailed wireless modeling framework (the top left block in Figure 1.1) that can be applied to queuing models, mainly M/G/1, and can lead to a much more accurate prediction of the wireless network behavior. The modeling framework incorporates a large number of parameters that are either inputs to the system, or Medium Access Control (MAC) transmission protocol specifics. We have also approximated the service time distribution of frame transmissions with a closed-form equation. This can provide basic analysis and close approximations for QoS purposes, such as minimum and maximum delay and buffer bounds.
The improved accuracy of our model can be also attributed to the fact that we incorporated most of the relevant MAC parameters in our analysis. For example, our model employs different probabilities for direct frame transmissions (i.e. without backoff), transmissions after backoff, incomplete transmissions, etc. As a result, our model provides more accurate expressions for the channel busy-period and the average service time.

2. The other key contribution of this thesis is the extension of our finite load model to 802.11e WLANs. The extended model is presented in Chapter 4. The few papers that were published in this area mostly had the same shortcomings done in 802.11 models. However, due to the prioritized nature of the Access Categories, these short-
comings had a huge effect on the final calculations, and the system converged to a wrong set of values. We also discuss in detail the complexities of a probabilistic multi-priority system, such as the transmission interference between traffic classes.

The development of this finite-load model for QoS-enhanced WLANs is important for the emerging research on transporting multimedia traffic over WLANs. Our work helps to identify wireless networks delays and buffer distributions.

3. To complete the modeling of WLANs in all its operation cases, we have improved an existing model for 802.11e networks under saturation. The improved model corrects invalid assumptions published previously for the Backoff Process [3]. The improved model has high accuracy and will be presented in Chapter 5.

4. The development of three Throughput Improvement Algorithms for WLANs operating in saturation mode. Three algorithms are presented in Chapter 6, and in [4]. These adaptive algorithms monitor the wireless network conditions in an Access Point. Based on the measured quantities, the algorithms perform their calculations, taking into account the number of nodes and the type of their traffic, and decide on a set of 802.11e MAC Protocol values that will maximize the Saturation Throughput. These values are then broadcast to the wireless stations in the network to be used immediately.
1.4 Thesis Outline

Chapter 2 gives an overview of 802.11 networks and its underlying technology. It describes the modes of operations, and access protocols. The same is done for 802.11e networks. Chapter 2 also surveys all previous and related major studies, and introduces our models with a comparison to previous models. Chapter 3 presents a new highly accurate mathematical model of 802.11 networks under the finite-load condition. Chapter 4 extends the model in Chapter 3 to 802.11e networks. Chapter 5 introduces the improved mathematical model for 802.11e networks under saturation, and proposes three new algorithms to improve the throughput in 802.11e networks operating in infrastructure-mode. Chapter 6 presents three adaptive algorithms that are used to improve the Saturation Throughput in overloaded WLANs. Chapter 7 concludes this thesis, and presents some open research problems that can be the starting points for other studies.
Chapter 2

Overview of Wireless Local Area Networks

The IEEE 802.11 standard for wireless LANs was approved in 1997. It defined two of the seven layers of the OSI model, which are the Medium Access Control (MAC), and the Physical (PHY) layers, as shown in Figure 2.1. The 802.11 standard was intended to replace the 802.3 Ethernet standard where wiring was not desired or possible, by offering the same services needed for the upper layers. To deal with the variable and less predictable conditions of the wireless medium, the MAC layer function is more complex in the 802.11 standard, and is therefore still the subject of much recent research. In this chapter, we will present an overview of the 802.11 MAC, as well as the 802.11e MAC.
2.1 Overview the IEEE 802.11 MAC Protocol

The basis for the 802.11 MAC is a CSMA/CA mechanism (Carrier Sense Multiple Access with Collision Avoidance). Carrier sensing is done through physical sensing of the Radio Frequency (RF) carrier as well as a virtual carrier sensing in the MAC itself. Virtual carrier sensing is done through maintaining a Network Allocation Vector signal (NAV) that determines for how long the channel remains busy. NAV is set in a duration field of the MAC messages. Collision avoidance in 802.11 MAC is performed by a mechanism called Distributed Coordination Function (DCF). The MAC has two modes of operation, Contention Period (CP) in which any one or more stations may try to access the medium according to DCF rules, and Contention Free Periods (CFP), in which the stations are only allowed to respond to the poll messages sent by the access point operating in PCF mode. DCF is a protocol that describes how stations can contend for the wireless medium in a distributed manner. PCF, on the other hand, is a protocol that specifies a centralized contention-free access to the wireless medium organized by the access point.

In the DCF mode, otherwise known as the contention period (CP), all wireless stations compete for access to the channel by applying a CSMA/CA protocol. If the channel is busy,
the station will freeze its operation and wait for the channel to become available. DCF provides best-effort, asynchronous data transfer and is mandatory in all 802.11 wireless stations. Medium access control in ad-hoc networks is implemented by means of DCF only.

In order to access the channel under DCF, prior to transmitting, a wireless station must first sense the channel to determine if another station is currently transmitting. If no other station is found to be transmitting, the station waits a DCF Inter-Frame Spacing (DIFS) time period and starts the backoff process if the channel is still free after the DIFS period. If a second station is transmitting, or if one starts to transmit during the DIFS period, the first station will wait for the channel to become free for a DIFS period. After the channel has been idle for a DIFS period, a random backoff timer is started. If the channel becomes busy while the timer is running, the timer will pause. The timer will then restart after the channel has been free for a DIFS period. When the timer expires, the station will begin transmission of its data. The timer is selected using a slotted binary exponential backoff technique. The time following the idle DIFS is slotted and stations can only transmit at the beginning of a slot. The backoff time is uniformly chosen in the interval \([0, CW-1]\) where \(CW\) is the Contention Window. At the first transmission attempt, \(CW\) is set to be equal to Contention Window Minimum \(CW_{\text{MIN}}\), then it is doubled at each retransmission attempt up to Contention Window Maximum \(CW_{\text{MAX}}\). Typical values of \(CW_{\text{MIN}}\) and \(CW_{\text{MAX}}\) are 32 and 1024 slots respectively [6].

Upon successful reception of a frame, the receiver waits a Short Inter-Frame Spacing (SIFS) interval and then transmits an Acknowledge Frame (ACK) for the received frame.
When the ACK is received, the station must wait a random backoff time before attempting to access the channel again. If the ACK is not received within a timeout period, the station assumes a lost frame and must contend for channel access again and try a retransmission of the frame. If the maximum number of retransmission attempts is exhausted, the station must drop the frame.

2.2 The IEEE 802.11e MAC Protocol Extension

The 802.11e committee or Task Group E (TGe) has been conceived for the purpose of enabling Quality of Service (QoS) support on 802.11. The 802.11e DCF function, now called the HCF function (Hybrid Coordination Function), supports on top of it two other functions: the EDCA and the HCCA. The EDCA (Enhanced Distributed Channel Access) operates as a Contention Period (CP), and the HCCA (HCF Controlled Channel Access) operates as a Contention-Free Period (CFP). In CP, the EDCA functions of all the stations contend for the medium; whereas in HCCA, the Access Point (AP) controls the medium, and polls the stations based on the decisions of a scheduler.

The IEEE 802.11e specification [7] supports 8 priority queues or User Priorities (UPs), divided among 4 Access Categories (ACs) per station. The 4 ACs are voice, video, data, and background traffic in decreasing priority. The queues have different Arbitrary Inter-Frame Spacing (AIFS) periods according to their traffic type. Furthermore, each AC has a different minimum and maximum CW periods (\(C_{\text{Wmin}}\) and \(C_{\text{Wmax}}\)), as well as separate CW variable for each AC. This provides better prioritized access to the wireless medium. Therefore, the service differentiation in EDCA is provided by a combination of the follow-
Chapter 2. Overview of Wireless Local Area Networks

ing three mechanisms [7]:

1. the Arbitrary Inter Frame Space period (AIFS),

2. the Contention Window parameters (CWMIN, CWMAX),

3. and the Transmission Opportunity (TXOP).

In EDCA, each queue or AC operates independently as if 8 contending stations reside in the same physical stations. This mode of operation classifies collisions into internal and external collisions. Internal collisions are handled inside the station, whereby the higher priority queue transmits, and the lower priority one backs off again. External collisions are the normal collisions that happen physically in the wireless transmission medium.

**Arbitrary Inter Frame Space (AIFS)**

In EDCA, each queue will defer for a number of slots that is equal to the AIFS period specific to the queue’s AC, every time the medium turns idle, as shown in Figure 2.2. This deffering mechanism precedes the backoff mechanism, which may not be reached, if the medium turns busy before the AIFS period ends. By using different AIFS period values, higher priority ACs will have a lower AIFS values, therefore getting access to the medium before a lower priority AC does. Another aspect of the differing AIFS periods, the backoff counter of a higher priority AC starts decrementing before lower priority backoff counters.

For this, consider the scenario where two queues belonging to two different ACs choose the same backoff value. In this case the AC with the lower AIFS period will transmit first with no collision.
Immediate access when Medium is free $\geq$ DIFS/AIFS

AIFS

DIFS

Busy Medium

Contention Window

Backoff Slots

Next Frame

Slot time

Defer Access

Select Slot and Decrement Backoff as long as medium is idle

Figure 2.2: AIFS Relationships [7]

**Contestation Window Size**

Each AC has a separate minimum and maximum contention window parameters (CWMIN and CWMAX), in addition to keeping a separate current contention window variable (CW[AC]). This also contributes to the prioritized access to the wireless medium. Furthermore, the backoff counter in EDCA is chosen in the range $[1, CW[AC]+1]$ instead of $[0,CW]$, to avoid the case where an AC with AIFS[AC] = 0 preempts other ACs.

**Transmission Opportunity**

The Transmission Opportunity (TXOP) is a new concept introduced in the 802.11e specification. After contending for and winning the wireless medium, a station is granted a predetermined amount of time equal to the TXOP value of the winning AC. In this TXOP time window, the station can send as many frames as possible from the corresponding queue. However, the station must ensure that all the frames with their corresponding frame exchanges (RTS, CTS, Data-ACK) should not overstep the TXOP time limit (see Figure 2.3). TXOPs introduce the ability to transmit a burst of frames between a station and the AP, and offer a definite advantage for bursty multimedia sources or low-payload per frame.
Chapter 2. Overview of Wireless Local Area Networks

Figure 2.3: EDCA TXOP [8]

sources. The TXOP concept also enhances the throughput, for the station spends more time sending frames, and less time contending for the medium. In mathematic-like descriptions, less contention means lower transmission overhead and higher throughput, and fewer collisions means better service and lower frame delay.

2.3 Mathematical Modeling of Wireless Networks

A mathematical model captures the behavior of the wireless network, and provides effective means to estimate network performance under various loading conditions. One of the crucial parameters of the model is the backoff process of the MAC layer, which governs the activities of a wireless station. The backoff process decrements a randomly generated number with each slot. This time slotted process makes it possible to model the backoff function as a discrete Markov Chain [9][3].

The success and failure of each frame transmission relies on the interaction between different backoff processes that are running independently in each station/queue sharing the same wireless channel. If two contending stations choose the same initial number, then as they count down to zero, their corresponding transmissions will collide. Therefore
a careful scrutiny of MAC-related probabilities is necessary to complete the mathematical model especially since the backoff process chooses its initial counter values from a contention window with a uniform distribution.

Two types of models have been proposed for WLANs. In a finite-load WLAN model, the queues in a wireless network are seldom full. As the rate of frame arrival and the manner with which they arrive change from network to another, the mathematical model normally needs queuing theory to solve the problem in hand.

On the other hand, in a saturated WLAN model, the system is modeled using only averages, and does not use a queueing model. The reason is, since all the queues are backlogged at all times, the server has a utilization of 1, and therefore quantities such as the average queue size and the average waiting time are not valid. Such studies mainly focus on deriving three quantities: the probability of transmission, the probability of collision, and the saturation throughput.

2.4 Finite-Load Models for IEEE 802.11 and 802.11e WLANs

Zeng and Chlamtac [10] constructed a queuing model for WLANs with finite load using the network as the queuing server, and made several over-simplified assumptions regarding the collision probability in the wireless channel without modeling the post-collision backoff process. Although their model produced crude approximations of the system values, it has the advantage of useful simplicity. Ozdemir and McDonald [11] provide a useful model for the backoff process and service-time distribution, but their work models the sat-
urated network case and is similar to the models proposed earlier by Bianchi [9][12] and Robinson and Randhawa [3]. Zaki and Al-Hadidi [13] proposed another useful model, but as in [14][15], they did not consider the case of directly transmitted frames, and did not provide an expression for service time distribution, as well as a simplified average service time expression. Cheng and Wu [16] proposed a finite-load model, but neglected to include successful transmissions from other stations during the transmitting cycles of a particular station. This led to a distorted value of the service-time and other key quantities. The same simplifying assumption appeared in [17], [18], [19], [20] and [21].

Recently, Tickoo and Sikdar [14][15], proposed a simpler, and fairly accurate, finite-load analysis using only a node (user) model, but did not include the network model. Some of our node model derivations were based on their work. Cantieni et al. [20] have also developed a fairly detailed finite-load model for multirate WLANs but using a different approach that does not rely on the dual-model symbiosis.

A concept based on using two queuing models, one user-centric and another network-centric, has been explored before by Medepalli and Tobagi [22], where they demonstrated how a WLAN can be modeled using both views. The model we develop in the next chapters is also based on using two queuing models. However, their approach differs from ours in that their two queuing models are applied independently, thus leading to less accurate results. We will show that by coupling the two models, our method leads to much more accurate results that match very closely those produced by detailed WLAN simulators.
Chapter 2. Overview of Wireless Local Area Networks

There is relatively little reported work on modeling QoS-enabled finite-load 802.11e WLANs. In [23], Engelstad and Osterbo built a good model that took into account the different AIFS periods for each priority. However, they did not include successful transmissions from other stations in their service time expressions.

Tickoo and Sikdar extended their model in [15] to 802.11e networks. However, since we used their work as a starting point, our approach yielded more accurate and varied quantities, especially when modeling the "presence" of each priority compared to other priorities and its impact on the average service time of the queue.

But the most notable study on finite-load 802.11e WLANs was delivered by Vassis and Kormentzas in [24]. Their expression of the average service time was fairly detailed, and they followed the same procedure to derive quantities such as the average waiting time. However, they made the simplifying assumptions of overlooking the effect of the AIFS on the probability of collision, and the effect of the interference of transmissions between multiple ACs on the average service time.

The improvement accomplished in our finite load model, labeled the EFL model in this thesis, over the available models, was to use a complete and extensively detailed model as was depicted in Figure 1.1. We represented most of the relevant details of the DCF protocol with mathematical expressions, whether it be a probability, a queueing theory average, or just an obvious approximation, to form a complete mathematical picture. The mathematical expressions for the Node Modeling Framework and the Medium Modeling Framework are grouped in a complementary fashion to form a non-linear set of equations. An efficient
algorithm would exploit some special feature in this set to solve it using iterations and generate all the final values when the system converges, as will be demonstrated in Appendix A. The results of each iteration are fed into two M/G/1 queuing models, to evaluate quantities such as the server utilization. The queuing models then feedback these resulting quantities to the frameworks in the next iteration to refine the results. Derived expressions such as the average queue length and the average queue waiting time are evaluated in the last iteration.

Since the station's behavior changes when the wireless medium is busy with a transmission from another station, it was necessary to derive the probability of a busy medium. This expression can only be derived if the medium was treated as the server in a queueing model. The busy probability would then be equal to a fraction of the utilization rate of this wireless medium server. This detail justifies the use of a double queuing model.

The EFL model was extended in Chapter 4 to heed the changes of the 802.11e newly approved standard. This model is one of the first few models available in the literature for the 802.11e WLANs and, to the best of our knowledge, the most accurate yet. The same algorithm in Appendix A was used to solve the non-linear set of equation generated by this model.
Chapter 2. Overview of Wireless Local Area Networks

2.5 IEEE 802.11e WLAN Saturation Models and Algorithms

Numerous 802.11e saturation models have been published, the most notable ones by Xiao in [25], Tsai and Wu in [26], Zhu and Chlamtac in [27], Hui and Devetsikiotis in [28], Tantra, Foh, and Mnaouer in [29], Xu, Wang and Hassanein in [30], and Kong, Tsang, Bensaou, and Gao in [31]. But the most accurate model by far was that published by Robinson and Randhawa in [3] due to their correct handling of the delicate different-AIFS-period issue.

Our improvement was to alter the already existing mathematical model in [3] to fit the latest changes in the 802.11e standard under study by the TGe. Robinson and Randhawa have put together a robust mathematical model for 802.11e WLANs, but the model has included misinterpretations of details in the 802.11e specification, which has skewed the results. To set things aright, we reconstructed the model with the right assumptions.

In [32], Zhao et al. described a distributed scheme to adaptively change the CW Parameters in order to increase the throughput. They used statistics recorded by the wireless network card to update their probability variables. According to some rules, these variables are used to update the CW Parameters following a mapping function that they derived from simulations. Using the mapping function, they get a Target $CWMIN[i]$, which the current $CWMIN[i]$ converges to by using an adaptive step size.
In [33], Cali, Conti and Gregori used a mathematical model that is very similar to Bianchi’s model to derive the optimum CW Parameters for a 802.11 congested WLAN. The scheme is again distributed, and it is used for ad-hoc networks, where each station has no complete knowledge of the network. Stations gather statistics from successful and collided transmissions to have a clearer view of the network, and depending on these statistics, the new optimal CW Parameters are calculated. Instead of differentiating the original equation by the CW Parameters, the authors approximated the model with a simplified one, and then differentiated it to get the near-optimum CW values.

In [34], Romdhani et al. have written a new algorithm to change the CW parameters. \( \text{CW}_{\text{MIN}}[i] \) and \( \text{CW}_{\text{MAX}}[i] \) are left constant, but \( \text{CW}[i] \) is not reset to \( \text{CW}_{\text{MIN}}[i] \) after each successful transmission but is reset adaptively to a different value. Similarly, after each failed transmission, the \( \text{CW}[i] \) is not doubled, but multiplied by a different factor for each priority.

In [35], Dong et al. derived a new scheme to reach the theoretical capacity of the network by creating a Superframe. By adapting the MAC to change the DCF and PCF intervals, the system performs optimally for any kind of traffic demands. Their study is very similar to ours in the sense that the MAC changes the Superframe values to follow the highest of three overlapping curves (Refer to Figures 6 and 8 in [35], and Figure 6.12 in this study).

The proposals in [32], [34], [33], and [35] differ from our algorithms because these schemes were developed for ad-hoc networks. Furthermore, the method reported in [33]
was derived for the IEEE 802.11 standard. The algorithm proposed in [34] keeps the $\text{CWMIN}[i]$ and $\text{CWMAX}[i]$ constant, and changes only the way $\text{CW[i]}$ is reset. The study in [32] is the closest to ours, but instead of using a mathematical model, it has a mapping function derived from heavy simulations to get the Target CW as a function of the probability of collision. The approach in [35] uses different network parameters for optimization which are the DCF and PCF intervals.

Our Throughput Improvement Algorithms are based on the same fundamental concept of using adaptive Contention Window size control for maximizing the throughput in a saturated 802.11e WLAN. The Contention Window Parameters (CW) consist of two parameters: $\text{CWMIN}$ and $\text{CWMAX}$.

Our improvement over other available algorithms was the amount of Saturation Throughput added to the WLAN. Three different algorithms have been developed for handling the optimal Contention Window parameters problem. In all three algorithms, the Access Point (AP) takes the responsibility of collecting measurements, computing average throughputs, calculating Contention Window parameters, and then broadcasting the results back to the stations in the network, using Beacon frames.

2.6 The Simulator

A detailed simulator has been developed for the purpose of modeling wireless LANs. The simulator was originally developed in OPNET, then migrated to C to improve flexibility and reduce long simulation times. A version of the software has been also evolved
into a working MAC stack and ported to run on VxWorks. The simulator used in this thesis inherits most of its functions from the MAC stack, but has a discrete-event simulator representing the PHY layer.

On the other hand, it was worth noting that generating a Random Variable (R.V.) from an Exponential Distribution introduced an error to our simulations. The rand() function of the C language was used to generate an R.V. with a Uniform Distribution. This R.V. was then passed to another function that produced an Exponential Distribution out of a Uniform one. The inter-arrival times are sampled from this distribution, with an integer number of slots. The operation of discretization, along with the imperfection of the rand() function, introduced an error that is plainly visible in Figure 2.4. The dotted-line plot is the exponential distribution generated by the simulator which starts at probability 0.032, the continuous-line plot is an ideal exponential distribution that starts at 0.03. The $\lambda$ parameter
in both distributions is equal to 0.03. Such a small difference of 0.002 will cause inter-
arrival times to be closer than usual, or lumped together. Because of this, in the finite-load
models, the simulation will have slightly higher values for the mean service times, the
probability of collision, and the utilization probability, than the mathematical models, and
that is especially visible when the load approaches the saturation threshold, i.e. the network
approaches the saturation case.

\section{Concluding Remarks}

In this chapter, we introduced the technology and infrastructure that WLANs rely on
to operate properly. We also surveyed the finite-load models, saturation throughput mod-
els, and adaptive algorithms for throughput enhancement. We also introduced our own
improvements to these models and algorithms. Our aim for creating these models was to
improve accuracy, and as the results will show in the end of each chapter, we met that goal.
Chapter 3

An Enhanced Finite-Load Model for

IEEE 802.11 Wireless LANs

3.1 Introduction

Today's world of technology has seen a proliferation of wireless networks. The access protocol upon which these wireless networks rely to transmit and receive is called CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). This protocol has been created to make use of the medium that all wireless transceivers share. The IEEE 802.11 standard [6] defines two modes of medium access for WLANs: Point Coordination Function, PCF, and Distributed Coordination Function, DCF. An Access Point uses PCF to schedule transmissions of real-time traffic and important data in a collision-free environment. DCF is used when all stations are contending for the medium using their backoff processes, which is the case in this study. The backoff process uses a backoff counter that is chosen uniformly from a contention window, then decremented with every idle slot, and
frozen when the medium gets busy. When the timer reaches zero, and if the medium is still idle, the station begins its transmission. Good modeling of this backoff process is key to constructing an accurate model.

In this chapter, we derive an accurate finite-load analytical model for IEEE 802.11 WLANs and substantiate the validity of the model by comparing it to results obtained from a very realistic WLAN simulator. Our approach is based on combining two queuing models. The first model represents each station (or node) in the network as an M/G/1 queue (single station queue), while the second models the wireless network as an M/G/1 queue (wireless medium queue). The wireless medium queue has a total rate of $N\lambda$, where $N$ is the number of stations and $\lambda$ is the incoming traffic rate per station. In the second model, the network is considered to be the queuing system, with the frames from all the stations as its input.

We will show that the combined model provides a powerful analytical tool for estimating the performance of the individual stations and the entire WLAN (wireless channel). For example, the model can be used to determine what traffic load levels from the stations will cause the network to saturate. Also, when the stations generate different loads (i.e. they have different $\lambda$'s), an individual node can not saturate before the wireless network saturates. Our methodology leads to closed-form equations for the service-time distribution and other related steady state probabilities.

The reader can refer to Section 2.1 for an overview of the underlying technology, and to Section 2.4 to know more about related work on finite-load models.
3.2 System Model and Assumptions

We have based our finite-load analysis on an approach that requires the interaction between two distinct queuing models, one representing the tagged node (or user) view and the other representing the whole network (or WLAN medium) view. In the former case, the node is a special wireless station which is mathematically separated from the network by modeling other nodes' interactions, such as collisions and frame transmissions, as a cumulative delay in the service time of the frame ready for transmission in the tagged node. The tagged node server essentially models the processing done at the IEEE 802.11 MAC and PHY layers (see Figure 3.1).

The tagged node queuing model is logically divided into two parts: the Wireless Node Modeling Framework which supplies averages such as the average service time and the probability of collision, and the Node Queuing Model which uses these averages to output final quantities such as average waiting time and average queue length. This is illustrated in Figure 3.3 by the upper two blocks.
By contrast, the whole network model has the shared wireless LAN medium as the server (see Figure 3.2). In a manner that is similar to the tagged node queuing model, the wireless medium queuing model is logically divided into two parts: the Wireless Medium Modeling Framework which supplies key averages, and the Wireless Medium Queuing Model which uses these averages. This is illustrated in Figure 3.3 by the lower two blocks.

Figure 3.3 also shows the interaction between the two queuing models, and represents the general architecture of our Enhanced Finite Load (EFL) model. As a result of this logical partitioning, our analysis leads to a set of non-linear equations (Wireless Node Modeling Framework and Wireless Medium Modeling Framework) that we solve to derive the steady-state probabilities which are then applied to M/G/1 queuing models for the tagged node and the whole network (Node Queuing Model and Wireless Medium Queuing Model).
In congruence with similar analysis in the literature, we will make the following simplifying assumptions. We assume all stations, or nodes, are within each other's reach and there are no hidden terminal problems. We also ignore the effects of bit-errors due to noise. Therefore, transmitted frames are lost only as a result of collisions caused by other simultaneous transmissions. This also implies that each node has an infinite buffer, which, with today's buffer sizes, is almost realistic. We assume that the packets/frames arrive to the buffer according to a Poisson process (unless stated otherwise) with rate $\lambda$, and are queued in a FIFO manner. In addition, after transmission and successful reception, the frames are destroyed on the receiver's side and do not enter any queue again. We also assume constant collision probability $p$ for a given set of input parameters.

### 3.3 Basic Analysis

The conditional probability of collision $p$ in a finite-load (i.e. non-saturated) wireless network of $N$ nodes has already been established in [9], as follows:

$$p = 1 - P[NT]^{N-1}$$  \(3.1\)

where $P[NT]$ is the probability that a node is not transmitting in a given idle slot. The intuition behind calculating $p$ as above is that $P[NT]^{N-1}$ is the probability that none of the other $N-1$ stations are transmitting during a time slot, and $1 - P[NT]^{N-1}$ is the probability that one or more of the non-tagged nodes are transmitting in the current time slot.
Chapter 3. An Enhanced Finite-Load Model for IEEE 802.11 Wireless LANs

Intermediate Values: Average Service Time, Probability of Collision, etc.

Wireless Node Modeling Framework
(Supplies the Node Queuing Model with accurate means and distributions)

Probability of Busy Medium

Node Queuing Model

Wireless Medium Modeling Framework

Wireless Medium Queuing Model

Inputs: Load, DCF Protocol Parameters, etc...

Outputs: Waiting Time, Queue Length, etc...

Figure 3.3: Finite Load Model Architecture
The probability $P[NT]$ can be calculated as follows:


$$= 1 - (1 - \rho) + (1 - \tau P_{\text{backoff}})\rho$$

$$= 1 - \rho \tau P_{\text{backoff}}$$

(3.2)

where $P[QE]$ denotes the probability that the tagged node queue is empty, $P[QNE]$ is the probability that the node’s queue is non-empty, and $\rho = \lambda/\mu$. Note that $P[NT|QNE]$ is expressed as the complement of the product $(1 - P_{\text{backoff}})$ where $\tau$ is the probability of transmission during a given time slot and $P_{\text{backoff}}$ is the probability that a frame arrives from higher layers then waits through the backoff process in the node’s queue. Therefore, $\tau P_{\text{backoff}}$ represents the probability of transmission given that the frame goes through a backoff process. On the other hand, if a frame finds the system empty and the medium idle, then it gets transmitted directly without going through the backoff process.

Let $\bar{W}_\rho$ be the average backoff counter value used by the backoff process for every transmission, or retransmission. The backoff process will virtually draw a number between 1 and $2\bar{W}_\rho$ after a successful transmission (or a collision) occurs. If the node has a non-empty queue, it should transmit during every idle slot with a probability $\tau$. This forms a Geometric Distribution, where the probability of success is $\tau$ and the probability of failure is $1 - \tau$. The first moment of this distribution represents the period within which the node transmits, which is $\bar{W}_\rho$. The expression of the mean of this Geometric distribution is:

$$\bar{W}_\rho = \frac{1 - \tau}{\tau}$$

(3.3)
which results in the following expression for $\tau$

$$\tau = \frac{1}{1 + \overline{W}_p} \quad (3.4)$$

Let $I_w$ be the average backoff window size when a successful transmission occurs given a collision probability $p$. Then,

$$I_w = CW_{\text{min}}(1 - p) + 2p(1 - p)CW_{\text{min}}$$
$$+ 4p^2(1 - p)CW_{\text{min}} + \ldots + (2p)^m CW_{\text{min}} \quad (3.5)$$

$$= CW_{\text{min}} \left[ (1 - p) \sum_{k=0}^{m-2} (2p)^k + (2p)^m \right]$$

and,

$$\overline{W}_p = I_w/2 \quad (3.6)$$

The variable $m$ is the maximum number of retransmissions. We assume that $2^m CW_{\text{min}}$ is equal or less than $CW_{\text{max}}$, otherwise the previous equation can be easily changed to accommodate this. On the other hand, from the network model point of view, it is more accurate to calculate an average backoff counter value, we call it $\overline{W}$, that includes direct transmissions in addition to transmission following a backoff period, such that

$$\overline{W} = P_{\text{backoff}} \cdot I_w/2 = P_{\text{backoff}} \cdot \overline{W}_p \quad (3.7)$$

Note that for the purpose of computing the probability of collision $p$, the expression derived for $\overline{W}_p$ excludes direct frame transmissions (i.e. without backoff). In general, a very small proportion of direct transmissions will collide, with a rate much lower than $p$. 

For this reason, the expression of the collision probability $p$ would prove non-accurate if we used $\bar{W}$ instead of $\bar{W}_p$.

The backoff probability, $P_{\text{backoff}}$, can be computed from the probabilities for the busy and idle periods. Let $P_{\text{busy}}$ be the probability that the medium is busy in a given time slot and let $P_{\text{idle}}$ be the complement of $P_{\text{busy}}$. Then the portion of the transmitted frames that will go through a backoff process is given by:

$$P_{\text{backoff}} = (P_{\text{busy}} + \rho P_{\text{idle}})$$

(3.8)

In other words, the probability $P_{\text{backoff}}$ represents the portion of the frames that will go through a backoff period because either the medium was busy (with probability $P_{\text{busy}}$), or because the node had some frames buffered ahead of the incoming frames while the medium is idle ($\rho P_{\text{idle}}$). The complement of $P_{\text{backoff}}$ is the portion of the frames that will be transmitted directly without going through a backoff process, because the node buffers are empty and the medium is idle.

Note that $\bar{W} \leq \bar{W}_p$, because $\bar{W}$ is the average backoff window when direct transmission are also included. From the network model point of view, $P_{\text{backoff}}$ contributes to the accuracy of calculating the time slots for the average network service time ($1/\mu_{net}$), hence its inclusion in computing $\bar{W}$ but not $\bar{W}_p$. 
3.4 The Tagged Node Queuing Model

The first queuing model we will present is that of the Tagged Node. Because the service time distribution for this model is not Markovian, we use an M/G/1 queuing model.

This section’s model considers the node as the server. All other messages, transmissions, and interference from other nodes sharing the wireless medium will be modeled as delays in the service time.

3.4.1 Service Time Estimation

In this section, we will derive a compound equation for the expected service-time \((1/\mu)\) for the Tagged Node server, using the basic queuing relation \(\rho = \lambda/\mu\). Essentially, the server represents the processing of the MAC and physical layers of an 802.11 station, and the server queue contains frames received from these higher layers. The service time is defined by the frame length distribution, the backoff process, the number of stations (transmissions and collisions of other nodes), and the collision probability \(p\).

We start by defining the transmission cycle as the time it takes a node to transmit a frame, whether this attempt results in a collision or in a successful attempt. The total time it takes a node to transmit a frame successfully may consist of several cycles each ending in a collision except for the last one. The expected number of cycles a node takes to transmit a frame successfully is \((1 - p)\left(1 + 2p + 3p^2 + \ldots\right) + mp^{m-1} = 1 + p + p^2 + \ldots + p^{m-1} = \sum_{k=0}^{m-1} p^k\), with \(m\) being the maximum number of retransmissions. Therefore, the expected number of transmission cycles that end in collisions is...
In each transmission cycle, the node counts down backoff slots, but may be interrupted by either collisions or successful transmissions from other nodes, and finally ends the cycle either with a collision or a successful transmission of its own.

The service time can be expressed in terms of its component delays as follows:

\[ \frac{1}{\mu} = \text{nodeCx} + \text{otherCx} + \text{backoff} + \text{nodeTx} + \text{otherTx} \]  

Thus, the service time is comprised of:

- the accumulated backoff through the expected number of cycles (say C cycles),
- the accumulated successful transmissions from other stations (otherTx) in C cycles,
- the accumulated collisions (otherCx) from other stations in C cycles,
- the accumulated collisions made by the tagged node (nodeCx) throughout C cycles,
- the successful transmission (nodeTx) or discarding of the frame by the tagged node when the maximum number of retransmissions is exceeded.

Now, let the probability \( P_x \) denote the portion of frames that arrive to an empty queue when the medium is busy. These frames will experience only a portion of a transmission from other stations, an average period of \( T_s/2 \) in case of a successful transmission, and \( T_c/2 \) in case of a collision. Here, \( T_s \) is the time it takes to complete a successful transmission,
including the PHY headers overhead, RTS, CTS, and ACK frames, and finally the frame payload. \( T_c \) is the time it takes to detect a collision, including PHY headers, RTS frame, and then a deference period. Then,

\[
P_x = (1 - \rho)P_{\text{busy}} \left[ \rho + \left(1 - \rho\right)^{N-1} \right]
\]  

(3.11)

The factor \( \rho + (1 - \rho)^{N-1}/N \) is empirically included to adjust the impact of \( P_{\text{busy}} \) from the point of view of the tagged node. When the node utilization \( \rho \) is small and the Tagged Node is empty, then the real contributors to \( P_{\text{busy}} \) are the other \( N-1 \) nodes. But as \( \rho \) gets larger, this equation becomes non-linear because of the inter-dependence between \( \rho \) and \( P_{\text{busy}} \).

When the queue utilization \( \rho \) is small and the Tagged Queue is empty, then the real contributors to \( P_{\text{busy}} \) are the other \( N-1 \) queue. But as \( \rho \) gets larger, this equation becomes non-linear because of the inter-dependence between \( \rho \) and \( P_{\text{busy}} \).

Given the fairness of the backoff process over a long time period, each station will have a chance to transmit given that its buffers are not empty. In this case, the number of successful transmissions \( (P_{\text{otherTx}}) \) and collisions \( (P_{\text{otherCx}}) \) arising from other nodes during the service time of a single frame are given by the following probabilities,

\[
P_{\text{otherTx}} = \rho(N - 1)
\]  

(3.12)

\[
P_{\text{otherCx}} = \rho(N - 1)\frac{P_o}{1 - P_o}
\]  

(3.13)
Chapter 3. An Enhanced Finite-Load Model for IEEE 802.11 Wireless LANs

The probability $p_o$ represents the portion of transmissions from other nodes that results in collisions. Equation 3.12 is an expression of the complement of the portion of frames that get transmitted successfully given that there is a transmission, derived from the following relations:

$$P_{otherSum} = P_{otherTx} + P_{otherCx},$$

$$P_{otherTx} = P_{otherSum}(1 - p_o), P_{otherCx} = P_{otherSum}p_o.$$  

Therefore, $\frac{P_{otherCx}}{P_{otherTx}} = \frac{p_o}{1 - p_o}$, from which 3.13 follows.

To illustrate the derivation of $p_o$, we will give a small example. Assume a WLAN with two nodes A and B. A and B have both managed to transmit 9 frames successfully. In their tenth attempt, node A’s transmission collided with node B’s transmission. From the node’s point of view, the probability of collision is 10%. However, from the system’s point of view, the wireless medium has witnessed 18 successful transmissions, and only one collision. Therefore, the probability of collision of nodes from an outsider’s point of view is 5.26%. For simplicity’s sake, we neglect the portion of collisions that are caused by 3 or more same-time transmissions because of their very low probability of occurrence. This example illustrates the derivation of $p_o$ which equals

$$p_o = \frac{p}{2(1 - p) + p}$$

$$= \frac{p}{2 - p}$$  \hspace{1cm} (3.14)
Now, the main quantities of the service time \(1/\mu\) can be derived directly:

\[
\text{node} \times = \text{Cycle} \times T_c 
\]

(3.15)

\[
\text{other} \times = (P_{\text{other}} - J^o - P_i) T_c
\]

(3.16)

\[
\text{Backoff} = \bar{W} + \text{Cycle} \times \bar{W}_p
\]

(3.17)

\[
\text{node} T x = (1 - p^m) T_s
\]

(3.18)

\[
\text{other} T x = (P_{\text{other}} T_s - P_s) T_s
\]

(3.19)

A detailed example of \(\mu\) with numerical results will be given in Section 3.7.2.

### 3.4.2 Service Time Distribution (Tagged Node Queuing Model)

We model the service time probability distribution function (PDF), \(b(t)\), as a shifted negative exponential distribution (n.e.d.), where the amount of shift is equal to the average frame length \(T_s\), as follows:

\[
b(t) = \mu_p \cdot e^{-\mu_p (t - T_s)} \cdot u(t - T_s)
\]

(3.20)
Chapter 3. An Enhanced Finite-Load Model for IEEE 802.11 Wireless LANs

Figure 3.4: Service Time Distribution Approximation for Typical $p$

The shift happens because any frame can not be discarded from the buffer (whether it be the result of a successful transmission or the exceeding of the maximum retransmissions threshold) unless it undergoes at least one transmission. Therefore, the service time of any frame can not be less than $T_s$, which yields a non-Markovian service time distribution. The function $u(t - T_s)$ is the Heaviside (or step) function shifted by $T_s$, and $\mu_p$ is the constant of the n.e.d. before shifting, expressed as a function of $T_s$ and the average service time, $\mu$, as follows,

$$\mu_p = \left( \frac{1}{\mu} - T_s \right)^{-1} \quad (3.21)$$

This distribution captures the nature of the true service time PDF computed as the cascaded convolution of the PDFs of the cumulative backoff-counter distribution, other nodes' transmission time distribution and the node's transmission time distribution. The resulting PDF will typically have the shape of a shifted negative exponential distribution. To validate this observation, Figures 3.4 and 3.5 show plots of the true service time PDF, obtained by
Figure 3.5: Service Time Distribution Approximation for Large $p$

Few papers have tried to derive the true service time distribution, and none, as yet, had it derived completely. In general, the concept is simple: the service time distribution could be derived by convolving the distributions of the five components of the service time. But the final expression is too complex to be derived, or to be manipulated in a mathematical derivation (exponentially increasing time to solve system in MATLAB). And even if the final expression is derived in the s-domain or the z-domain, it is impossibly complex to convert back to the time-domain. It was for this reason that we preferred to model the service time with a shifted exponential distribution. One key observation is that the variance of the actual service time is larger than that of the simplified distribution we proposed, although the mean is the same. To compensate for the error introduced, we apply a variance of $V = 2V\mu$. 
Applying the Laplace Transform to $b(t)$, we obtain

$$B(s) = \int_0^\infty e^{-st}b(t)dt = \int_0^\infty e^{-st}\mu_p e^{-\mu_p(t-T_s)}u(t-T_s)dt = \frac{\mu_p e^{-sT_s}}{\mu_p + s}$$  \hspace{1cm} (3.22)

From which we can derive the PGF of the new service time PDF, as follows

$$\phi(z) = B[\lambda(1-z)] = \frac{\mu_p e^{-\lambda(1-z)T_s}}{\lambda(1-z) + \mu_p}$$  \hspace{1cm} (3.23)

Let $Q(z)$ be the PGF of the steady state probability, $\pi_k$, the latter indicating that there are $k$ frames in the node queue. Considering that we are using Poisson arrivals, we have

$$Q(z) = \frac{(1-\rho)(1-z)\phi(z)}{\phi(z) - z} = \frac{(1-\frac{\lambda}{\mu})(1-z)e^{\lambda T_s(1-z)}}{e^{\lambda T_s(1-z)} - z\left[1 + \frac{\lambda}{\mu_p}(1-z)\right]}$$  \hspace{1cm} (3.24)

Now taking moments of $Q(z)$ will give us the steady-state probabilities (the queue length distribution), as follows:

$$\pi_k = \frac{(Q(z))^k}{k!}\bigg|_{z=0}$$  \hspace{1cm} (3.25)

Another quantity of interest is the average queue size $N_q$,

$$N_q = \rho + \frac{\rho^2 + 2A^2V_\mu}{2(1-\rho)}$$  \hspace{1cm} (3.26)

The quantity $V_\mu$ is the variance of the service time distribution, which is defined as
Chapter 3. An Enhanced Finite-Load Model for IEEE 802.11 Wireless LANs

follows

\[
V_\mu = \int_{-\infty}^{\infty} b(t)(t - \frac{1}{\mu})^2 dt = \int_{T_s}^{\infty} \mu_p e^{-\mu_p(t-T_s)}(t - \frac{1}{\mu})^2 dt
\]

\[
= T_s^2 + \frac{2T_s}{\mu_p} + \frac{2}{\mu^2_p} - \frac{2T_s}{\mu} - \frac{2}{\mu \mu_p} + \frac{1}{\mu^2} \tag{3.27}
\]

3.4.3 Throughput and Delay

The throughput of the network, also called utilization, is defined as the number of data bytes that are successfully transmitted per second. Throughput is an important criterion in some situations; it allows us to measure the network responsiveness to the load it is subject to. In the case of networks with non-saturated finite-load networks, calculating throughput is trivial since it is essentially equal to the input rate.

On the other hand, delay computation is more consequential, since delay tends to increase dramatically as the network moves towards saturation. In queuing theory, delay is defined as the waiting time \( T_q \) of a queued frame from the instant it enters the queue until the start of service. Similar to average queue size \( N_q \), the average waiting time \( T_q \) is one of the Pollaczek-Khintchine formulae, expressed as follows

\[
T_q = \frac{1}{\lambda} \left[ \rho^2 + 2\lambda^2 V_\mu \right] \tag{3.28}
\]

In our wireless networks, delay is defined as the waiting time \( W_q \) of a frame from the instant that it enters the queue until the start of being successfully transmitted. It follows
then that the waiting time of a frame $W_q$ is equal to $T_q$ plus the service time $1/\mu$, minus the time it takes to successfully transmit a frame, $T_s$. Therefore the average waiting time, or delay, has the following expression

$$W_q = T_q + \frac{1}{\mu} - T_s$$  \hspace{1cm} (3.29)

The next and important step of our analysis is to derive formulas for the important probabilities, $P_{idle}$ and $P_{busy}$. For that, we revert to the whole-network queuing model.

### 3.4.4 Bursty Source

In the case where we need to simulate a Bursty Source in each node, we can simplify the analysis and still get good approximations by using an M/G/1 queuing model with batch arrivals. The batches inter-arrival time is distributed exponentially with a rate of $\lambda_b$, and each batch has an arbitrary number of frames, with an average of $B$ frames/batch and a variance of $\sigma^2_b$. The EFL model can be used as it is, and all of the above analysis is still valid with the M/G/1 queue with batch arrivals, except that now $\lambda$ is equal to $B\lambda_b$. Also, the expressions of the waiting time, and of the average queue size change. The following is the waiting time expression:

$$T_{q}^{\text{Batch}} = \frac{\lambda_b B^2}{2\mu^2 (1 - \rho)} \left( 1 + \frac{B \sigma^2 + \sigma^2_b/\mu^2}{B^2/\mu^2} \right) + \frac{(B - 1)}{2\mu} + \frac{\sigma^2/\mu}{2B}$$ \hspace{1cm} (3.30)

The average queue size can be derived by using Little’s Law $N_{q}^{\text{Batch}} = \lambda T_{q}^{\text{Batch}}$. 

3.5 Whole Network Queuing Model

In the Whole Network Queuing model, the shared wireless medium is considered as the server. The service time is modeled as a shifted exponential distribution.

3.5.1 Estimation of the Network Service Time and Utilization Rate

The average service time of this model \(1/\mu_{\text{net}}\) has three components: the average time of all backoff processes for all stations (nodes), the average frame length \(T_s\) expressed in time slots, and the delay overhead introduced by collisions. The first of these three components is the average time it takes for the first backoff-counter to fire from the group of stations present in the network. The expression for this can be derived from probability theory and combinatorics, as in [3], as follows:

\[
\begin{align*}
P_j &= P\left[ j \text{ slots} = \min_{i \in \mathcal{N}} (\text{backoff\_counter}(i)) \right] \\
&= \sum_{k=1}^{N} \binom{N}{k} \left( \frac{1}{2W} \right)^k \left( \frac{2W - j}{2W} \right)^{N-k} 
\end{align*}
\]  

(3.31)

where \(N\) is the number of stations in the network, and \(j\) is the minimum number of time slots elapsed since the medium became idle until the first backoff-counter (among \(N\) counters) fires. Let \(T_{\text{net}}^N\) be the expected value of the random variable \(P_j^N\), i.e.

\[
T_{\text{net}}^N = E\left[P_j^N\right] = \sum_{i=1}^{2W} iP_i^N
\]  

(3.32)
Chapter 3. An Enhanced Finite-Load Model for IEEE 802.11 Wireless LANs

Then, the expression for the average network service time is:

\[
\frac{1}{\mu_{net}} = T_{net}^N + T_s + p(T_{net}^N + T_c)
\]

and following a derivation process similar to that of the tagged node model, we have

\[
\mu_{net} p = \frac{1}{\mu_{net} - T_s}
\]

The probability distribution function of the network service time is:

\[
p_{net}(t) = \mu_{net} e^{-\mu_{net}(t-T_s)} u(t-T_s)
\]

This leads to an expression similar to that of the tagged node model but with different parameters:

\[
Q_{net}(z) = \frac{(1 - \frac{\lambda N}{\mu_{net}})(1-z)e^{\lambda N T_s(z-1)}}{e^{\lambda N T_s(z-1)} - z \left[ 1 + \frac{\lambda N}{\mu_{net}} (1-z) \right]}
\]

Taking the moments of \( Q_{net}(z) \), we get the steady-state probabilities of the network \( \pi_{net_k} \), indicating that there are \( k \) frames in the network.

\[
\pi_{net_k} = \frac{(Q_{net}(z))^{k}}{k!}
\]

Note that in the network model, \( k \) frames can be queued all in one node or in up to \( k \) différent nodes, and \( k \) can be 0 indicating an empty network. This network model is
particularly useful for estimating network wide performance such as maximum throughput, and the impact of the combined load from all stations results on throughput, average delay and collision probability.

3.5.2 Estimation of $P_{idle}$ and $P_{busy}$

In this section, we complete our analytical model by deriving formulas for $P_{idle}$ and $P_{busy}$. Recall that these two important probabilities appear in the derivation of several key parameters such as the collision probability $p$ and average backoff window $\bar{W}$.

Let us define a service cycle as the time period that begins with the start of the wireless medium turning idle and ending with a successful transmission. The average service cycle time is equal to $1/\mu_{net}$, and it includes the time for resuming backoff counters for an average number of collisions. The number of idle slots is then the weighted sum of the probability of $i$ active stations at a given slot multiplied by the expected value of the first backoff counter to fire a count-down given $i$ active nodes. We can approximate this by having the weighted sum of the $\pi_{net_k}$ s multiplied by the expected value of the first backoff counter to fire ($T_{net_k}^k$). This is not absolutely accurate since $\pi_{net_k}$ s do not represent the probability of $i$ active stations, but it represents the steady state probabilities of the network queue. The network queue could have two or more frames belonging to the same node. But still this approximation yields an accurate enough value (with an average error value of 1%). So the equation for the average number of idle slots $I_s$ is:

$$I_s = \sum_{k=1}^{\infty} \left[ \pi_{net_k} \cdot \sum_{i=1}^{2^k} i \cdot p_i \right] = \sum_{k=1}^{\infty} \pi_{net_k} T_{net_k}^k$$  (3.39)
Then with this value, we derive the ratio of idle slots in each cycle:

$$R_s = \frac{I_s}{I_s + T_s + pT_c}$$  \hspace{1cm} (3.40)

and we have,

$$P_{idle} = 1 - \rho_{net} + \rho_{net} \cdot R_s = 1 - \rho_{net}(1 - R_s)$$  \hspace{1cm} (3.41)

$$P_{busy} = 1 - P_{idle}$$  \hspace{1cm} (3.42)

### 3.6 Multi-Load and Multi-Rate Wireless LANs

#### 3.6.1 Multi-Load Wireless LANs

In this section, we will describe how to alter the model in order to accommodate a WLAN with stations having different loads. Nodes having different loads will lead us to the conclusion that each node $j$ in the WLAN has its own set of variables. Therefore, the index $j$ will indicate that this variable is specific to node $j$. We will derive again the equations that are subject to change, and unless stated otherwise, the rest of the equations remain unchanged except for the addition of the index $j$.

In the case of different loads, the probability of no transmission $P[NT]$ becomes different for each node, therefore:

$$P[NT]_j = 1 - \rho_j \tau_j P^j_{\text{backoff}}$$  \hspace{1cm} (3.43)
and

\[ p_j = 1 - \prod_{i=1, i \neq j}^{N} P[NT]_j \]  

(3.44)

It follows then that the equations of \( \tau_j \), \( I^j_w \), \( \overline{W}_p^j \), \( \overline{W}_j \), \( P^j_{\text{backoff}} \), etc..., use the variables \( p_j \), \( \rho_j \), etc ..., that are specific to node \( j \).

For the average service time estimation, several equations have to be changed. The number of successful transmissions and collisions from other stations \( P_{\text{otherTx}} \) and \( P_{\text{otherCx}} \) become

\[ P_{\text{otherTx}}^j = \sum_{i=1, i \neq j}^{N} \rho_j \]  

(3.45)

and

\[ P_{\text{otherCx}}^j = \frac{P_o^j}{1 - P_o} \sum_{i=1, i \neq j}^{N} \rho_j \]  

(3.46)

The probability of collisions arising from other stations \( p_o \) becomes

\[ p_o^j = \frac{\sum_{i=1, i \neq j}^{N} p_j}{2(N-1) - \sum_{i=1, i \neq j}^{N} p_j} \]  

(3.47)

These changes will enable us to derive the \( \mu_j \) and \( \rho_j \) for each node. On the other hand, it was worth noting that having a different \( \rho_j \) for each node means that we applied the Node Queuing Model \( N \) times for each node in our calculations, while still applying the Wireless Medium Queueing Model only once for the whole wireless network. Therefore,
The same can be done for the average queue length.

The last change to be made is to replace the $W$ in Equation 3.31 to get $P^N_j$ with the average of all average backoff counters $W_{Av}$. The equation in question becomes

$$P^N_j = P \left[ j \text{ slots} = \min_{1 < i < k} (\text{backoff.counter}(i)) \right]$$

$$= \sum_{k=1}^{N} \binom{N}{k} \left( \frac{1}{2W_{Av}} \right)^k \left( \frac{2W_{Av} - j}{2W_{Av}} \right)^{N-k}$$

with

$$W_{Av} = \frac{\sum_{i=1}^{N} W_j}{N}$$

3.6.2 Multi-Rate Wireless LANs

It is interesting to note that wireless LANs with nodes having different physical rates (i.e. different bandwidth) have basically the same behavior as single-rate WLANs. The backoff process in the MAC is independent from the PHY rate, and therefore wireless nodes are synchronized to slot boundaries by pre-defined parameters relating to the underlying technology in operation. The only quantity that change in multi-rate WLANs is the success duration $T_s$, which becomes specific to each node ($T^j_i$). Therefore, a logic similar
to the one in Section 3.6.1 can be used to derive the now different $\mu_j$ and $\rho_j$ quantities.

### 3.7 Simulation versus Numerical Results

In this section, we compare numerical results obtained from the analytical model with results obtained from a simulator. We have implemented a detailed simulator based on the IEEE 802.11 standard and implementation notes. For further validation, we have compared our results with commercial simulators such as OPNET to ensure accuracy. Furthermore, we have verified that the simulator operating in saturation matches Bianchi's $p_{sat}$ and other quantities to a high level of accuracy (confidence interval above 95%) [9].

In our constructed system, the probability of collision $p$ parameter is the most sensitive to the choice of random seeds in the initialization process. The main reason is that $p$ is derived from $P[NT]$ raised to the power $N - 1$ (see equation 3.1) where $N$ is the number of stations in the network. Therefore, small differences between the simulated $P[NT]$ and the analytical $P[NT]$, amplify the corresponding difference between the simulated $p$ and analytical $p$. In real systems, the probability $p$ is typically in the range $[0.01, 0.2]$.

Except for the collision probability $p$, we found all other quantities to be stable, and give very accurate results when compared to simulation. Most of the calculated results and simulations overlap with exact values for large spans of $\lambda$, representing the input rate. For large values of $\lambda$, the network utilization $\rho$ approaches 1 and the laws of queuing theory do not hold accurately. Only in this high utilization range do we observe the largest divergence between our analytical model and simulation results.
3.7.1 Numerical Example of the Service Time

The following example shows how good each component of $\mu$ is represented in our model. The example shown in Table 3.1 if for a network with 20 stations or nodes, where $\text{CWMIN}$ is equal to 32, and the load per-station is equal to 22 frames/sec. In combination with these values, setting $\rho$ to equal 0.2 causes the maximum margin of error between simulation and our model, especially in the collision probability $p$. In Table 3.1, the EFL (Enhanced Finite Load ) model refers to the model developed in this chapter, which is compared to the model given in [14]. All the quantities in the table are given in terms of time-slot units. See Table 3.2 for details.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Simulation</th>
<th>EFL Model</th>
<th>Tickoo &amp; Sikdar [14]</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodeCx</td>
<td>4.232877</td>
<td>4.1656</td>
<td>9.8937</td>
</tr>
<tr>
<td>otherCx</td>
<td>8.668607</td>
<td>6.8908</td>
<td>87.7486</td>
</tr>
<tr>
<td>backoff</td>
<td>21.841324</td>
<td>19.9435</td>
<td>25.9287</td>
</tr>
<tr>
<td>nodeTx</td>
<td>94.978311</td>
<td>94.9540</td>
<td>95</td>
</tr>
<tr>
<td>otherTx</td>
<td>326.374658</td>
<td>340.2596</td>
<td>842.5684</td>
</tr>
<tr>
<td>Service Time ($1/\mu$)</td>
<td>453.704449</td>
<td>449.8883</td>
<td>1061.1</td>
</tr>
<tr>
<td>Collision probability $p$</td>
<td>0.1480</td>
<td>0.1483</td>
<td>0.2919</td>
</tr>
<tr>
<td>Utilization $\rho$</td>
<td>0.199770</td>
<td>0.2051</td>
<td>0.4669</td>
</tr>
</tbody>
</table>

Table 3.1: Numerical Example of the components of the Service Time

3.7.2 Results

In this section, we present our simulation and model results. The WLAN is assumed to operate in DCF with no “hidden station” problems. Whether the WLAN operates in ad-hoc or infrastructure mode does not matter in our scenarios because we assume all the stations are in each other’s range. Figures 3.6 to 3.8 show plots of numerical results from our EFL
analytical model compared to simulations, to results from the analytical model in [14], and to results from a de-coupled Tagged Node Queuing Model (i.e. Node Queuing Model only, without the help of a Wireless Medium Queueing Model and its feedback $P_{bus}$). All plots and simulations used the same parameter set given in Table 3.2. Figure 3.6 plots the utilization $\rho$ versus input load. Figure 3.7 plots the collision probability $p$ versus input load, with the saturation collision probability derived from [9] as reference. We can see that as $\rho \to 1$, our collision probability converges to Bianchi’s saturation collision probability. Figure 3.8 plots the average service time $1/\mu$ versus input load. Figure 3.9 plots the average waiting time, or delay, versus input load. All plots cover three WLAN sizes: 4 stations, 12 stations, and 20 stations. All nodes have the same input rate/load $\lambda$, and the data frame payload size is fixed at 2304 bytes, whihc is the MTU for a WLAN. The parameters $T_c$ and $T_s$ have been calculated as the approximate sum of RTS, CTS, frame size, physical and MAC headers, ACK, and DIFS. For simplicity, we have quantized the frame lengths to align with slot boundaries. The error due to this quantization is negligible especially considering that frame length is much larger than the slot size.

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>11 Mbps</th>
<th>Slot Time</th>
<th>20 microseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CW_{min}$</td>
<td>32</td>
<td>$T_c$</td>
<td>24 slots</td>
</tr>
<tr>
<td>$CW_{max}$</td>
<td>1024</td>
<td>$T_s$</td>
<td>95 slots</td>
</tr>
<tr>
<td>Retransmissions</td>
<td>4</td>
<td>2304 bytes</td>
<td>MAC Header</td>
</tr>
<tr>
<td>PHY Header</td>
<td>192 bits</td>
<td>RTS Size</td>
<td>112 bits</td>
</tr>
<tr>
<td>RTS Size</td>
<td>160 bits</td>
<td>MAC Header</td>
<td>272 bits</td>
</tr>
<tr>
<td>ACK Size</td>
<td>112 bits</td>
<td>DIFS Time</td>
<td>50 microseconds</td>
</tr>
</tbody>
</table>

Table 3.2: Simulation and Mathematical Models Parameters
Chapter 3. An Enhanced Finite-Load Model for IEEE 802.11 Wireless LANs

The figures show clearly that our EFL model is very accurate especially when the number of stations in the network is less than 20, which is typically the case for real WLANs. The accuracy degrades slightly as the number of nodes increases, but the model is still highly precise, especially for the key quantities: $\rho$ (utilization), and $1/\mu$ (service time).

In Figures 3.6 to 3.9, we compare our results against the model proposed by Tickoo and Sikdar [14]. Their analysis, although considers the node queuing model only, actually produces results with the same trend as our results, albeit with less accuracy. In Figure 3.6, for example, our model and the one in [14] follow the same trend as the load is increased. However, the model in [14] overestimates the utilization by 20% for 4 stations (especially at higher loads, e.g. 90 frames/sec), and by more than 50% for 20 stations when the load is above 20 frames/sec. In all cases, our model matches the simulation results to within a maximum of 5%. Similar observations can be deduced from graphs 3.7, 3.8, and 3.9. In the Figure 3.10, we have plotted the waiting time $T_q$ for a WLAN with batch arrivals for 4 stations.

Now, we compare the EFL model results with simulations results for a Multi-Load WLAN in which the network has 4 stations (nodes), each having a different load. In this setup, the load of node A varies while the load of the other nodes remains constant. Specifically, the load on node B, C, and D is 70, 50 and 120 frames/sec, respectively. The rest of the parameters are the same as in Table 3.2. In Figure 3.11, we plot the utilization $\rho_j$, the probability of collision $p_j$, and the average service time $1/\mu_j$ for all 4 stations as the load on node A increases. Figure 3.12 shows the average waiting-time for each of the nodes.
It is interesting to note that the utilization rate of the wireless medium was plotted in the first plot of Figure 3.11 to support our claim that a node can not saturate before the WLAN saturates. This is harder to show when simulating a homogeneous-load WLAN.

3.8 Conclusion

In this chapter, we have developed a novel finite-load queuing model for more accurate analysis of 802.11 WLANs. When compared to other models in the literature, our model provides substantially more accurate results that closely match those obtained from highly detailed WLAN simulators. The accuracy of our model is attributed mainly to the use of a coupled queuing model which proved to be very useful for studying the effect of varying parameters on the micro-scale (station level) and on the macro-scale (network level). We have derived closed form equations for the $Q(z)$ functions representing the steady-state probabilities. Our model takes for input the load, the number of stations, the contention window parameters, the bandwidth, and the frame size, and generates all the average and steady state probabilities. The proposed model can be easily extended to get the throughput and utilization of the network. The model is very flexible and allows also for easily changing the input traffic model from Poisson to other useful distributions by modifying the expression for $Q(z)$, and applying a G/G/1 queuing model.
Figure 3.6: Utilization vs. load for WLAN with 4, 12, and 20 stations
Figure 3.7: Collision probability, $p$ vs. load for WLAN with 4, 12, and 20 stations
Figure 3.8: Service time vs. load for WLAN with 4, 12, and 20 stations
Figure 3.9: Waiting time $T_Q$ vs. load for WLAN with 4, 12, and 20 stations

Figure 3.10: Waiting time $T_Q$ vs. load for WLAN with 4 stations with Batch Arrivals
Figure 3.11: Key Quantities vs. load for a Multi-Load WLAN with 4 stations
Figure 3.12: Waiting time $T_Q$ vs. load for a Multi-Load WLAN with 4 stations
Chapter 4

A Finite-Load Model for QoS-Enabled IEEE 802.11e WLANs

4.1 Introduction

The rapid development of switching architectures, new smart antennas, and robust security, has lead to the wide proliferation of wireless local area networks (WLANs) into the enterprise and consumer markets. Currently, WLANs have become more than just a niche solution of extending the enterprise or institution network to unwired areas. For businesses, WLANs offer true economic advantage by helping enterprises maintain a flexible reusable local wireless networks that can improve employee mobility.

Some of the rising application drivers for WLANs are voice/video over WLAN and high bandwidth TCP connections. As it became commonly evident, joining data and multimedia traffic in one system has forced us to think about classes of service. The 802.11e
committee has been formed to address this very issue. However, before main stream IT
and VoIP applications migrate to the QoS-enabled wireless LANs, good modeling tools
should be available in order to study and enhance the performance of these multimedia
QoS-enabled wireless networks.

In this chapter, we extend the finite-load model of Chapter 3 to 802.11e WLANs and
substantiate the validity of the model by comparing its results to those obtained from a very
realistic WLAN simulator. Our approach is based on combining two queuing models. The
first model represents each station's queue in the network as an M/G/1 queue, while the
second models the shared wireless medium, but also as an M/G/1 queue (channel queue).
The channel queue has a total rate of \( N\lambda \), where \( N \) is the total number of queues and \( \lambda \) is
the incoming traffic rate per queue. In the second model, the network is considered to be
the queuing system, with the frames from all the queues as its input.

We will show that the combined model provides a powerful analytical tool for estimating
the performance of the individual queues and the entire WLAN (wireless channel). For
example, the model can be used to determine what traffic load levels from the queues
will cause the network to saturate. Also, when the queues generate different loads (i.e.
they have different \( \lambda \)'s), an individual queue can not saturate before the wireless network
saturates.

The reader can refer to Section 2.2 for an overview of the technology used in the 802.11e
MAC, and to Section 2.4 to know more about related work of 802.11e finite-load models.
Figure 4.1: Tagged Node Queuing Model

Figure 4.2: Network Queuing Model
4.2 System Model and Assumptions

We have based our finite-load analysis on an approach that requires the interaction between two distinct queuing models, one representing the *tagged node* (or user) view and the other representing the *whole network* (or WLAN medium) view. In the former, the node is a special wireless station which is (mathematically) isolated from the network by modeling other nodes’ interactions (such as collisions and frame transmissions) as a cumulative delay in the service time of the frame ready for transmission in the tagged node. The tagged node server essentially models the processing done at the IEEE 802.11 MAC and PHY layers (see Figure 4.1). By contrast, the whole network model has the shared wireless LAN medium as the server (Figure 4.2).

Our analysis leads to a set of non-linear equations that we solve to derive the steady-state probabilities that will be applied to M/G/1 queuing models for the tagged node and the whole network.

In congruence with similar analysis in the literature, we will make the following simplifying assumptions. We assume all stations, or nodes, are within each others reach and there are no hidden terminal problems. We also ignore the effects of bit-errors due to noise. Therefore, transmitted frames are lost only as a result of collisions caused by other simultaneous transmissions. This also implies that each node has an infinite buffer, which, with today’s buffer sizes, is almost realistic. We assume that the packets/frames arrive to the buffer according to a Poisson process with rate $\lambda$, and are queued in a FIFO manner. In addition, after transmission and successful reception, the frames are destroyed on
the receiver's side and do not enter any queue again. We also assume constant collision probability \( p \) for a given set of input parameters.

### 4.3 Basic Analysis

IEEE 802.11e standard specifies the use of four distinct queues, one for each access category. To simplify presentation and avoid unnecessary complicated formulas, we will base our analysis on the behavioral study of two queues, labeled A and B. Queue A will be assigned to the higher priority AC while queue B will be assigned to the lower priority AC. We will show in a later section how the model can be easily extended to an arbitrary number of priority queues.

Remark on notation: We will refer to priority queues with different expressions. In general, queue A, priority A queue, class A, and class A queue, all point to the same QoS queue with the highest possible priority.

In many cases the expressions for both AC queues are identical, in which case, we will show the derivations for the A queue only, and the expression for queue B will be stated with no further explanation.

### 4.3.1 Estimation of the Collision Probability

Let \( P[NT_A] \) be the probability that the A queue, in a finitely-loaded (or non-saturated) wireless network of \( N \) nodes \( (N = N_A + N_B) \), is not transmitting in a given idle slot. Then as was established in [14], the probability of no transmission is:
\[ P[NT_A] = P[NT_A|QE_A]P[QE_A] + P[NT_A|QNE_A]P[QNE_A] \]
\[ = 1.(1 - \rho_A) + \rho_A(1 - \tau_A \cdot P_{\text{backoff}}) \] (4.1)
\[ = 1 - \rho_A \cdot \tau_A \cdot P_{\text{backoff}} \]

where \( P[QE_A] \) denotes the probability that queue A is empty, \( P[QNE_A] \) is the probability that queue A is non-empty, and \( \rho_A = \lambda_A / \mu_A \) is the utilization parameter for AC_A (Access Category A). Clearly, \( P[QNE_A] = \rho_A \) and \( P[QE_A] = 1 - \rho_A \). Note that \( P[NT_A|QNE_A] \) is expressed as the complement of the product \( \tau_A \cdot P_{\text{backoff}} \), where \( \tau_A \) is the probability of frame transmission by queue A during a given time slot and \( P_{\text{backoff}} \) is the probability that a frame arrives from higher layers then waits through the backoff process in the station’s A queue. Therefore, \( \tau_A \cdot P_{\text{backoff}} \) represents the probability of transmission given that the frame goes through a backoff process. On the other hand, if an arriving frame finds the queue empty and the medium idle, then it gets transmitted directly without going through the backoff process. The same applies to priority B queue, in which the final expression for \( P[NT_B] \) is,
\[ P[NT_B] = 1 - \rho_B \cdot \tau_B \cdot P_{\text{backoff}} \] (4.2)

The backoff probability for AC_A, \( P_{\text{backoff}} \) can be computed from the probabilities for the busy and idle periods, using the approach given in Chapter 3. Let \( P_{\text{busy}} \) be the probability that the medium is busy in a given time slot and let \( P_{\text{idle}} \) be the complement of \( P_{\text{busy}} \). Then the portion of the transmitted frames that will go through a backoff process is given by:
\[ P_{\text{backoff}} = (P_{\text{busy}} + \rho A P_{\text{idle}}) \]  

(4.3)

and for AC\(_B\),

\[ P_{\text{backoff}} = (P_{\text{busy}} + \rho B P_{\text{idle}}) \]  

(4.4)

In other words, the probability \( P_{\text{backoff}} \) represents the portion of the frames that will go through a backoff period because either the medium was busy (with probability \( P_{\text{busy}} \)), or because the priority A queue had some frames buffered ahead of the incoming frames while the medium is idle (\( \rho A P_{\text{idle}} \)). The complement of \( P_{\text{backoff}} \) is the portion of the frames from queue A that will be transmitted directly without going through a backoff process, because the queue buffers are empty and the medium is idle.

### 4.3.2 Different AIFS Periods and Slot Occupancy Weighting

Robinson and Randhawa proposed a saturation model for the 802.11 wireless networks that accounts for the different AIFS periods in the backoff process of each priority class [3]. Lower priority queues have larger AIFS periods than higher priority queues. These differences among AIFS periods were modeled by separating the network idle time into distinct contention zones, and therefore distinct collision zones, one for each priority (or access category) level (equations 11-13, in [3]).

After the medium becomes idle, and as the time increases in slot number, then the network progresses from one contention zone to the other, as lower priority queues are now
able to backoff and transmit due to the end of their AIFS period. For example, in zone 1 only frames from the A queues can contend for channel. This will continue until the AIFS_B timer expires, and frames from the B queues join in the contention for the medium, thus moving the system into contention Zone 2. Every contention zone has its own characteristic transmission and collision probabilities.

To obtain the final average collision probability ($p_A$ or $p_B$), the collision probabilities of the contention zones are weighted by $b_i$, a discrete probability function, representing Slot Occupancy, and then summed. The slot occupancy p.d.f. $b_i$ is modeled as a Markov chain (see Figure 4.3), where each state represents a timeslot. The transition from state $i$ to state $i+1$ represents the probability of no transmission in the previous slot, and the transition from state $i$ to the initial state represents the probability of transmission in slot $i$.

Using this technique as part of our model, we summarize the probabilities already defined in [9] for the sake of completeness. Assuming that there exists an AIFS period difference between priority A and priority B queues, let $P_{Tr, Zone_i}$ be the probability of transmission from any queue in Zone $i$, then

![Figure 4.3: Slot Occupancy Markov Chain (taken from [3])](image-url)
\[ P_{T, Zone_1} = 1 - P[NT_A]^{N_A} \]  
(4.5)

and

\[ P_{T, Zone_2} = 1 - P[NT_A]^{N_A} \cdot P[NT_B]^{N_B} \]  
(4.6)

Let \( PC_{Zone_i} \) be the probability of collision for priority \( x \) queue, in contention Zone \( i \).

Then

\[ PC_{A, Zone_1} = 1 - P[NT_A]^{N_A - 1} \]  
(4.7)

and

\[ PC_{A, Zone_2} = 1 - P[NT_A]^{N_A - 1} \cdot P[NT_B]^{N_B} \]  
(4.8)

Also, for priority B,

\[ PC_{B, Zone_2} = 1 - P[NT_A]^{N_A} \cdot P[NT_B]^{N_B - 1} \]  
(4.9)

The reason why there is two collision probabilities for priority A queues, while only one for priority B queues, is that contending A queues span two zones, Zone 1 and Zone 2. Priority B queues contribute to the contention only in Zone 2 after the expiry of their AIFS period. The solution to the Slot Occupancy Markov Chain, after defining the needed probabilities is
\[ b_0 = \frac{1}{\min(CW_{\text{maxA}}, CW_{\text{maxB}}) \left( 1 + \sum_{i=1}^{n} \prod_{j=1}^{i} (1 - P_T, Zone_j) \right)} \] (4.10)

and

\[ b_1 = (1 - P_T, Zone_{i-1}) \cdot b_{i-1} \] (4.11)

Therefore the final average collision probabilities \( p_A \) and \( p_B \) are then expressed as

\[ p_A = \sum_{i=1}^{\min(CW_{\text{maxA}}, CW_{\text{maxB}})} P_c A Zone_i \cdot b_i \] (4.12)

and

\[ p_B = P_c B Zone_2 \] (4.13)

### 4.4 Tagged Node Queuing Model

As explained earlier, our finite load analysis is based on the interaction of two queuing models; one based on a user view (node model) and the other on the network/channel view (network model). In this section we derive important expressions for the node model. The network model is analyzed in the next section. Because the service time distribution for this model is not Markovian, we use an M/G/1 queuing model.

This section’s model considers the node’s queue as the server. All other messages, transmissions, and interference from other nodes’ queues or from co-located queues sharing the wireless medium will be modeled as delays in the service time.
4.4.1 Service Time Estimation (Node Queuing Model)

In this section, we will derive a compound equation for the expected service-time \((1/\mu_x)\) of queue \(x\) for the Tagged Node Queue server, using the basic queuing relation \(\rho_x = \lambda_x/\mu_x\).

Essentially, the server represents the processing of the MAC and physical layers of an 802.11e station, and the server queue contains frames received from its higher layers. The service time is defined by the frame length distribution, the backoff process, the number of stations (transmissions and collisions of other nodes), and the collision probability \(p_x\).

We start by defining the transmission cycle as the time it takes a queue to transmit a frame, whether this attempt results in a collision or in a successful attempt. The total time it takes a queue to transmit a frame successfully may consist of several cycles each ending in a collision except for the last one. The expected number of cycles a node takes to transmit a frame successfully is

\[
(1 - p_x) \left(1 + 2p_x + 3p_x^2 + \ldots\right)^m = 1 + p_x + p_x^2 + \ldots + p_x^{m-1} = \sum_{k=0}^{m-1} p_x^k,
\]

with \(m\) being the maximum number of retransmissions. Therefore, the expected number of transmission cycles that end in collisions is

\[
CycleC_x = \sum_{k=1}^{m-1} p_x^k = \sum_{k=1}^{m-1} p_x^k,
\]

(4.14)

In each transmission cycle, the queue counts down backoff slots, but may be interrupted by either collisions or successful transmissions from other queues, and finally ends the cycle either with a collision or a successful transmission of its own.
Chapter 4. A Finite-Load Model for QoS-Enabled IEEE 802.11e WLANs

The service time can be expressed in terms of its component delays as follows:

\[ \frac{1}{\mu_x} = \text{queueC}_x + \text{otherC}_x + \text{backoff}_x + \text{queueT}_x + \text{otherT}_x \]  

(4.15)

Thus, the service time is comprised of the accumulated backoff through the expected number of cycles (say C cycles), the accumulated successful transmissions from other queues (\(\text{otherT}_x\)) in C cycles, the accumulated collisions (\(\text{otherC}_x\)) from other queues in C cycles, the accumulated collisions made by the tagged node queue (\(\text{queueC}_x\)) throughout C cycles, and the successful transmission (\(\text{queueT}_x\)) or discarding of the frame by the tagged node queue when the maximum number of retransmissions is exceeded.

Now, let the probability \(P_{eb}^A\) denote the portion of frames that arrive to an empty queue A when the medium is busy. These frames will experience only a portion of a transmission from other queues, an average period of \(T_s/2\) in case of a successful transmission, and \(T_c/2\) in case of a collision. Here, \(T_s\) is the time it takes to complete a successful transmission, including the PHY headers overhead, RTS, CTS, and ACK frames, and finally the frame payload. \(T_c\) is the time it takes to detect a collision, including PHY headers, RTS frame, and then some deference period. Then,

\[ P_{eb}^A = (1 - \rho_A)P_{\text{busy}}\left[\rho_A + (1 - \rho_A)\frac{N_A - 1}{N_A}\right] \]

(4.16)

also for class B,

\[ P_{eb}^B = (1 - \rho_B)P_{\text{busy}}\left[\rho_B + (1 - \rho_B)\frac{N_B - 1}{N_B}\right] \]

(4.17)
Chapter 4. A Finite-Load Model for QoS-Enabled IEEE 802.11e WLANs

The factor $\rho_A + (1 - \rho_A)^{N_A - 1}_{N_A}$ is included to adjust the impact of $P_{busy}$ from the point of view of the tagged node queue. When the queue utilization $\rho_A$ is small and the Tagged Queue is empty, then the real contributors to $P_{busy}$ are the other $N_A - 1$ queue. But as $\rho_A$ gets larger, this equation becomes non-linear because of the inter-dependence between $\rho_A$ and $P_{busy}$.

Given the fairness of the backoff process over a long time period, each queue will have a chance to transmit given that its buffers are not empty. In this case, the number of successful transmissions ($P^A_{otherTx}$) and collisions ($P^A_{otherCx}$) arising from other queues during the service time of a single frame are given by the following probabilities,

$$P^A_{otherTx} = \rho_A(N_A - 1) + Ratio_{AB}\rho_BN_B$$

(4.18)

and for class B,

$$P^B_{otherTx} = \rho_B(N_B - 1) + Ratio_{BA}\rho_AN_A$$

(4.19)

and

$$P^A_{otherCx} = [\rho_A(N_A - 1) + Ratio_{AB}\rho_BN_B]\frac{P^A_0}{1 - P^A_0}$$

(4.20)

also for class B,

$$P^B_{otherCx} = [\rho_B(N_B - 1) + Ratio_{BA}\rho_AN_A]\frac{P^B_0}{1 - P^B_0}$$

(4.21)

The ratios $Ratio_{AB}$ and $Ratio_{BA}$ are quantities that indicate how each class "views" the other class in terms of transmissions. Since class A has higher priority over B, the class
B will see much more transmissions from A queues than from other B queues. Therefore, class B views class A as having more queues, and the \( \text{Ratio}_{BA} \) captures this effect. In contrast, class A views class B as having fewer transmissions which will consider class B to have less queues. Hence the inclusion of \( \text{Ratio}_{AB} \) in the above formula. The next section will explain how these ratios are derived.

The probabilities \( p^A_o \) and \( p^B_o \) represent the portions of transmissions from other nodes that result in collisions. But because of the existence of different zones, these probabilities have to go through the same process as \( p_A \) and \( p_B \). However, a simple and logical approximation with an acceptable margin of error, is to get the average probability of collision \( p = (p_A + p_B)/2 \) and proceed as if the probability of collisions from other stations is the same for all classes. The expressions for \( p^A_o \) and \( p^B_o \) are

\[
p^A_o = p^B_o = \frac{p}{2 - p}
\]

(4.22)

\( P^A_{otherTx} \) is an expression of the complement of the portion of frames that get transmitted successfully given that there is a transmission, derived from the following relations.

\[
P^A_{otherSum} = P^A_{otherTx} + P^A_{otherCx}
\]

(4.23)

\[
P^A_{otherTx} = P^A_{otherSum} \cdot (1 - p^A_o)
\]

(4.24)

and
\[ P^A_{\text{otherC}_x} = P^A_{\text{other Sum}}P^A_o \]  
(4.25)

Therefore,
\[ \frac{P^A_{\text{otherC}_x}}{P^A_{\text{otherTx}}} = \frac{P^A_o}{1 - P^A_o} \]  
(4.26)

from which the equations of \( P^A_o \) and \( P^B_o \) follow.

Now, the main quantities of the service time \( 1/\mu_A \) for class A are derived directly:

\[ \text{queueC}_x = \text{CycleC}_x + T_c \]  
(4.27)

\[ \text{otherC}_x = (P^A_{\text{otherC}_x} - \frac{P^A_o P^A_{eb}}{2(1 - P^A_o)})T_c \]  
(4.28)

\[ \text{backoff}_A = \overline{W}^A + \text{CycleC}_x + \overline{W}^A_p \]  
(4.29)

\[ \text{queueT}_x = (1 - P^A_o)T_s \]  
(4.30)

\[ \text{otherT}_x = (P^A_{\text{otherT}_x} - \frac{P^A_{eb}}{2})T_s \]  
(4.31)

\[ \frac{1}{\mu_A} = \text{queueC}_x + \text{otherC}_x + \text{backoff}_A + \text{queueT}_x + \text{otherT}_x \]  
(4.32)

and for class B,
\[
\frac{1}{\mu_B} = queueC_{xB} + otherC_{xB} + backoff_{fB} + queueT_{xB} + otherT_{xB} \quad (4.33)
\]

### 4.4.2 Ratio Selection

Broadly speaking, \( Ratio_{AB} \) and \( Ratio_{BA} \) are a measure of the presence of one priority class relative to the other. Each ratio is the proportion of the number of successful transmissions from one class divided by the proportion of the number of successful transmissions from the other class while traversing all possible combinations of the backoff counters.

Each ratio is a direct product of several parameters: the contention window size of each class, the AIFS period specific to each class, the number of retransmissions, the utilization \( \rho \) of each queue, and the number of total queues contending for the medium.

We present a small example to clarify the above concept. Let there be 1 station in the network with 2 queues, the first for priority A, and the second for priority B. Both queues have \( CW_{\text{MIN}} \) equal to 3 slots, with no retransmissions, and an AIFS difference of 1 slot. Because there are no retransmissions, doubling the Contention Window size is not possible even if collision occurs. We also assume that the queue is continuously backlogged, i.e. always has some packets to send, and that the backoff counters are decremented at the start of each idle time slot. We define a round of the backoff counters when both counters have just been re-generated (i.e. re-initialized with a random start value).

As illustrated in Figure 4.4, every time the medium gets idle, queue A can start decreasing its backoff counter by one, while queue B finishes its AIFS period, marked by an X in
Round 1: If, initially, queue A chooses a value of 2 for its backoff counter, and queue B chooses a value of 1, then collision will follow. This combination of backoff counters for the A and B queues will be denoted (2, 1). In the first timeslot after the medium gets idle, queue A will decrement its counter from 2 to 1, while queue B is still waiting for the end of its AIFS. At the beginning of the second timeslot, both queues A and B will have their backoff counters equal to 1, i.e. (1, 1) counter combination, and will be both decremented to 0 within the same timeslot, resulting in a definite collision between their transmissions. This marks the end of this round, and both counters will be initialized again in the following timeslot. Therefore, the combination (2, 1) of the backoff counter will end up in a collision.

Round 2: Assuming that in the next round, queues A and B choose the combination (3, 1). Then in the first timeslot, the counter combination changes to (2, 1), then to (1, 0) in the second timeslot. Thus, in the second timeslot, queue B will have a successful transmission, while queue A counter gets decremented to 1. Since only queue B counter has been regenerated after the successful transmission, then we are still in the same round, and queue A counter has a residual value of 1, which we call it residual count (1, 0). Whatever the
value that queue B chooses afterwards, queue A will transmit successfully before the end of the AIFS period of queue B, thus concluding that the combination (3, 1) will lead to 1 success for A and 1 success for B.

**Round 3:** For the next round, assume we start with the combination (1, 1), then following the same logic as before, queue A will have a successful transmission, resulting in a residual count (0, 1). Afterwards, queue A will choose a new value, which will lead us to three possibilities: a) either queue A initializes to 1, leading to a (1, 1) combination which takes us to where we started in the 3rd round (recurrence); b) A initializes to 2 which will lead to collision and the round ends; c) A initializes to 3, in which case queue B will have a successful transmission with a residual of (1, 0), as in round 2.

The possibility of having the same counter combination recurring happening many times in one round, such as the possibility of having the same residual (0, 1) occurring many times in the same round, has led us to define a new quantity $R$, short for “recurrent” as follows

$$R = \sum_{i=1}^{\infty} \frac{i}{\text{Total}_c}$$

(4.34)

where $\text{Total}_c$ is the total number of combinations which is in this case equal to 3, meaning the number of members in the set $(x,y)$: $1 \leq x \leq 3, 1 \leq y \leq 3$, where $y$ is fixed and only $x$ is being regenerated.

In the case where recurrence is possible, such as in round 3 where residual (0, 1) can happen again as a success for the higher priority, then all recurrences are counted as $R$
which translates into having the total successes of queue A in the combination (1, 1) equal to $1 + R$. In this example, $R$ is equal to 0.75. Table 4.1 shows all the combinations of the above example.

After calculating the total successes of both queues for all combinations, we define

$$\text{Ratio}_{AB} = \frac{\text{Successes}_B}{\text{Successes}_A} = \frac{4}{19.75} = 0.202$$

and

$$\text{Ratio}_{BA} = \frac{\text{Successes}_A}{\text{Successes}_B} = \frac{1}{\text{Ratio}_{AB}} = 4.937$$

The ratios calculated above match approximately the ratio values measured in simulations, which are $\text{Ratio}_{A_{BSim}} = 0.2152$ and $\text{Ratio}_{B_{ASim}} = 4.462$.

<table>
<thead>
<tr>
<th>Combinations</th>
<th>Queue A Successes</th>
<th>Collisions</th>
<th>Queue B Successes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>1.75</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(1,2)</td>
<td>1.75</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(1,3)</td>
<td>1.75</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(2,1)</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(2,2)</td>
<td>3.75</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(2,3)</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(3,1)</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(3,2)</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(3,3)</td>
<td>3.75</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>19.75</strong></td>
<td><strong>5</strong></td>
<td><strong>4</strong></td>
</tr>
</tbody>
</table>

Table 4.1: Successes and Collisions of the (3,3) pair combinations

On the other hand, when the contention window $\text{CWMIN}$ is greater than 3, then more complications arise. Assume the same example as above, but with $\text{CWMIN}$ equal to 4
instead of 3 for both priorities, and let’s study the combination (4, 4), also called counter pair. Then in the first timeslot, the counters become 3 for class A, and class B counter remains at 4. After 3 timeslots, class A queue successfully transmits the frame, which leaves a residual of (0,1). Since we have 4 possibilities, let’s examine the case where priority A queue chooses 4. The counter pair becomes (4, 1), which leads to a priority B successful transmission in the slot after the next, leaving a residual of (2, 0). Assuming that priority B backoff procedure chooses a value of 2 for its counter, the counter pair becomes (2, 2). This pair leads again to a residual of (0, 1), returning us to our start point. This re-entrancy complicates matters substantially, making the combinational calculations of the correct and exact ratios almost impossible.

In both the above mentioned examples, we assumed that the queues are backlogged. But as a finite-load model, this is not always the case. $\text{Ratio}_{AB}$ and $\text{Ratio}_{BA}$ have a strong relationship with the utilization rate $\rho$ of their respective queues. When $\rho_A$ and $\rho_B$ are small enough (i.e. light load supplied to the WLAN), then $\text{Ratio}_{AB}$ and $\text{Ratio}_{BA}$ get to be equal approximately to 1. It is when $\rho_A$ and $\rho_B$ increase, that both ratios diverge, to finally have one equal the inverse of the other, in the manner explained above. We have extracted an empirical relationship between the utilization rates and the ratios as follows, if $\rho_B < 0.98$:

$$\text{Ratio}_{AB} = 1 - (1 - \text{Ratio}_{ABSat})\rho_A$$  \hspace{1cm} (4.35)

$$\text{Ratio}_{BA} = 1 + (1 - \text{Ratio}_{ABSat})\rho_A$$  \hspace{1cm} (4.36)
on the other hand, if $\rho_B > 0.98$, meaning that class B queues are saturated, then

$$\text{Ratio}_{AB} = 1 + \text{Ratio}_{ABS_{at}} - \rho_A \quad (4.37)$$

$$\text{Ratio}_{BA} = \frac{1}{1 + \text{Ratio}_{ABS_{at}} - \rho_A} \quad (4.38)$$

$\text{Ratio}_{ABS_{at}}$ is the ratio that exists when the queues are backlogged, i.e. saturated. Retransmissions and the number of queues are also factors of influence on the two ratio quantities. All in all, too many complex parameters affect the value of the ratios, and therefore a mathematical solution for this problem is still under study, and is out of the scope of this thesis.

### 4.4.3 Service Time Distribution

We model the service time probability distribution function (PDF), $b_x(t)$, of any queue, as a shifted negative exponential distribution (n.e.d.), where the amount of shift is equal to the average frame length $T_s$, as follows:

$$b_A(t) = \mu_p^A e^{-\mu_p^A (t-T_s)} u(t-T_s) \quad (4.39)$$

and for class B,

$$b_B(t) = \mu_p^B e^{-\mu_p^B (t-T_s)} u(t-T_s) \quad (4.40)$$
For conciseness's sake, and because the analysis continues just as in Chapter 3, we'll only mention the final equations, which apply for both ACs. We will also drop the indexes.

The n.e.d. constant $\mu_p$ is:

$$\mu_p = \left(\frac{1}{\mu} - T_s\right)^{-1} \quad (4.41)$$

The PGF of the queue length distribution is:

$$Q(z) = \frac{(1 - \rho)(1 - z)\phi(z)}{\phi(z) - z} = \frac{(1 - \frac{\lambda}{\mu})(1 - z)e^{\lambda T_s(z-1)}}{e^{\lambda T_s(z-1)} - z\left[1 + \frac{\lambda}{\mu} (1 - z)\right]} \quad (4.42)$$

The average queue size $N_q$ is:

$$N_q = \rho + \frac{\rho^2 + \lambda^2 V_\mu}{2(1 - \rho)} \quad (4.43)$$

The variance $V_\mu$ is:

$$V_\mu = \frac{1}{\mu_p^2} \quad (4.44)$$

The average system time $T_q$ is:

$$T_q = \frac{1}{\lambda} \left[\frac{\rho^2 + \lambda^2 V_\mu}{2(1 - \rho)}\right] \quad (4.45)$$

And finally, the average network delay $V_\mu$ is:
\[ W_q = T_q + \frac{1}{\mu} - T_s \]  

(4.46)

### 4.5 Whole Network Queuing Model

We adopt a similar methodology for the derivation of the expressions for \( P_{\text{idle}} \) and \( P_{\text{busy}} \) as in Chapter 3, given that the highest priority class is equal in presence with the lower priorities at light loads and is the most present at high loads (the "Ratio" component), it was sufficient to use only class A average backoff counter value \( \overline{W}_A \) in the expression of \( P_j^N \). The error introduced in this simplification is very small, especially that \( P_{\text{idle}} \) and \( P_{\text{busy}} \), which are the target probabilities of this section, are stable quantities and do not change with small variations of \( \overline{W}_A \).

The expression for \( P_j^N \) becomes

\[
P_j^N = P \left[ j \text{ slots} = \min_{i \in iN} (\text{backoff\_counter}(i)) \right] \\
= \sum_{k=1}^{N} \binom{N}{k} \left( \frac{1}{2 \overline{W}_A} \right)^k \left( \frac{2 \overline{W}_A - j}{2 \overline{W}_A} \right)^{N-k} 
\]  

(4.47)

Where \( N \) is the total number of queues \((N_A + N_B)\) in the network, and \( j \) is the minimum number of time slots elapsed since the medium became idle until the first backoff-counter (among \( N \) counters) fires.
4.6 Extending the Model to Multiple Priorities

The model introduced in this chapter was derived assuming two priority classes. We now show it can be easily extendible to multiple priorities by only modifying key quantities. The IEEE 802.11e standard defines 4 priorities (or Access Categories). As to the collision probability $p_x$, the same steps should be followed as before, by dividing the timeslots into more contention zones depending on the AIFS periods of the multiple priorities. For each contention zone, if applicable for priorities A, B, C, and D in a specific contention zone, the following quantities become

\[ P_{c, Zone_i} = 1 - P[NT_A]^{N_i-1}.P[NT_B]^{N_i}.P[NT_C]^{N_i}.P[NT_D]^{N_i} \]  \hspace{1cm} (4.48)

\[ P_{Tr, Zone_i} = 1 - P[NT_A]^{N_i}.P[NT_B]^{N_i}.P[NT_C]^{N_i}.P[NT_D]^{N_i} \]  \hspace{1cm} (4.49)

Then the $P_{Tr, Zone_i}$ probabilities are used to calculate the Slot Occupancy probabilities $b_i$, which in turn are used to weight the $P_{c, Zone_i}$ probabilities to get the final average collision probabilities $p_x$.

Additionally, the number of successful transmissions from other queues $P_{otherTx}^x$ should be adjusted to take into consideration the presence of more priorities. The expression becomes

\[ P_{otherTx}^A = \rho_A(N_A - 1) + Ratio_{AB}\rho_BN_B + Ratio_{AC}\rho_CN_C + Ratio_{AD}\rho_DN_D \]  \hspace{1cm} (4.50)
The same should be done to $P^A_{otherCx}$ as follows

$$P^A_{otherCx} = [\rho_A(N_A - 1) + \text{Ratio}_{AB}\rho_BN_B + \text{Ratio}_{AC}\rho_CN_C + \text{Ratio}_{AD}\rho_DN_D] \frac{p^A_0}{1 - p^A_0} \quad (4.51)$$

### 4.7 Numerical Results

In this section, we compare numerical results obtained from the analytical model with results obtained from our simulator. We have implemented a detailed simulator of the IEEE 802.11e standard and its associated devices. We have validated our simulator by comparing our results with commercial simulators such as OPNET to ensure accuracy. We have also verified that the simulator operating in saturation matches Robinson and Rhandawa’s collision probabilities in saturation $\overline{p_A}$ and $\overline{p_B}$ for each queue and other quantities to a high level of accuracy (confidence interval above 95%)[3].

Except for the collision probabilities $\rho_x$, we found all other quantities to be stable, and give very accurate results when compared with simulations. Most often, the calculated results and simulations overlap with exact values for large spans of $\lambda$s. For large values of $\lambda$, the network utilization $\rho_x$ approaches 1 and the laws of queuing theory do not hold accurately. In this high utilization range, we observe the largest divergence between our analytical model and simulation results.
4.7.1 Numerical Example for Service Time

The following example shows how well each component of $\mu$ is represented in our model. Although there are some minimal differences, we note that simulation results vary with the initial random seed, and that some of our proposed equations are good estimations rather than real probabilities. The example is given with 12 nodes in the network, $C\text{Wmin}_A$ is equal to 8, $C\text{Wmin}_B$ is equal to 32, $C\text{Wmax}_A$ is equal to 128, $C\text{Wmax}_B$ is equal to 1024, AI$FSA$ is equal to 0 and AI$FSB$ is equal to 1, and the load is equal to 50 frames/sec. All quantities units are represented in time slots. See Table 3 for details.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Simulation of A</th>
<th>EFLM of A</th>
<th>Simulation of B</th>
<th>EFLM of B</th>
</tr>
</thead>
<tbody>
<tr>
<td>node$C_x$</td>
<td>15.03</td>
<td>14.78</td>
<td>19.05</td>
<td>20.06</td>
</tr>
<tr>
<td>other$C_x$</td>
<td>26.28</td>
<td>31.05</td>
<td>230.42</td>
<td>238.94</td>
</tr>
<tr>
<td>backoff</td>
<td>19.64</td>
<td>12.96</td>
<td>108.8</td>
<td>137.4</td>
</tr>
<tr>
<td>node$T_x$</td>
<td>94.16</td>
<td>94.18</td>
<td>93.28</td>
<td>92.87</td>
</tr>
<tr>
<td>other$T_x$</td>
<td>367.79</td>
<td>376.24</td>
<td>3025.18</td>
<td>3035.05</td>
</tr>
<tr>
<td>$1/\mu$</td>
<td>522.93</td>
<td>529.21</td>
<td>3476.75</td>
<td>3524.3</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.3873</td>
<td>0.3866</td>
<td>0.4472</td>
<td>0.4676</td>
</tr>
<tr>
<td>Utilization $\rho$</td>
<td>0.2646</td>
<td>0.2589</td>
<td>0.999</td>
<td>0.999</td>
</tr>
</tbody>
</table>

Table 4.2: Numerical Example of the components of the Service Time

4.7.2 Results

Figures 4.5 to 4.8 show plots of numerical results from our extended finite-load analytical model for QoS-Enabled WLANs compared to simulations. All plots and simulations used the same parameter set given in Table 4.3. Figure 4.5 plots the utilization rate $\rho$ versus input load for queues A and B. Figure 4.6 plots the collision probability $p$ versus input load for both queues. Figure 4.7 plots the average service time $1/\mu$ versus input load for both...
Chapter 4. A Finite-Load Model for QoS-Enabled IEEE 802.11e WLANs

queues. Figure 4.7 plots the average waiting time $W_q$ versus input load for both queues respectively.

All plots cover three WLAN sizes: 4 stations, 12 stations, and 20 stations, each station hosting one class A queue, and one class B queue. All queues have the same input rate/load $\lambda$, which is the same for both priorities, and the data frame payload size is fixed at 2304 bytes, which is the MTU for WLAN. The parameters $T_c$ and $T_s$ have been calculated as the approximate sum of RTS, CTS, frame size, physical and MAC headers, ACK, and DIFS. For simplicity, we have quantized the frame lengths to align with slot boundaries. The error due to this quantization is negligible especially considering that frame length is much larger than the slot size.

The figures show clearly that our EFL model is very accurate especially when the number of stations in the network is less than 20, which is typically the case for real WLANs. The accuracy degrades slightly as the number of nodes, and therefore queues, increases, but the model is still highly precise, especially for the key quantities: $\rho$ (utilization), and $1/\mu$ (service time).

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>11 Mbps</th>
<th>Slot Time</th>
<th>20 microseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWminA</td>
<td>8</td>
<td>Frame Size</td>
<td>2304 bytes</td>
</tr>
<tr>
<td>CWmaxA</td>
<td>128</td>
<td>Tc</td>
<td>24 slots</td>
</tr>
<tr>
<td>CWminB</td>
<td>32</td>
<td>Ts</td>
<td>95 slots</td>
</tr>
<tr>
<td>CWmaxB</td>
<td>1024</td>
<td>MAC Header</td>
<td>272 bits</td>
</tr>
<tr>
<td>Retransmissions</td>
<td>4</td>
<td>CTS Size</td>
<td>112 bits</td>
</tr>
<tr>
<td>PHY Header</td>
<td>192 bits</td>
<td>DIFS Time</td>
<td>50 microseconds</td>
</tr>
<tr>
<td>RTS Size</td>
<td>160 bits</td>
<td>AIFS A Period</td>
<td>0 slot + DIFS</td>
</tr>
<tr>
<td>ACK Size</td>
<td>112 bits</td>
<td>AIFS B Period</td>
<td>1 slot + DIFS</td>
</tr>
</tbody>
</table>

Table 4.3: Simulation and Mathematical model Parameters
4.8 Conclusion

In this chapter, we have extended the EFL model in the previous chapter to QoS-Enabled WLANs. The model has more accuracy compared to the other models in the literature when matching with results obtained by a detailed WLAN simulator. Our model is also the first to explain in details the constituting components of the service time in a probabilistic multi-priority system. The accuracy of our model is attributed mainly to the use of a coupled queuing model which proved to be very useful for studying the effect of varying parameters on the micro-scale (queue level) and on the macro-scale (network level). The proposed model gives numerical results for the average Waiting Time (Delay) and the average Queue Length (Buffer Size) which are crucial information for enforcing QoS guarantees. The model is very flexible and allows also for easily changing the input traffic model from Poisson to other useful distributions by modifying the expression for $Q(z)$, and applying a $G/G/1$ queuing model.
Figure 4.5: Utilization vs. load for WLAN with 4, 12, and 20 stations
Chapter 4. A Finite-Load Model for QoS-Enabled IEEE 802.11e WLANs

Figure 4.6: Collision probability, $p$ vs. load for WLAN with 4, 12, and 20 stations
Figure 4.7: Service time vs. load for WLAN with 4, 12, and 20 stations
Figure 4.8: Waiting time, or Delay, $W_q$ for Priority A vs. load for WLAN with 4, 12, and 20 stations
Chapter 5

An Improved Saturation Model for

IEEE 802.11e WLANs

5.1 Introduction

As mentioned before, IEEE 802.11e standard introduces several enhancements to the MAC layer of the original 802.11 standard. The EDCA MAC function is built on top of the distributed coordination function (DCF) of the original standard. The EDCA introduces class specific contention window sizes and initial deferment time (Inter Frame Space, IFS) [36].

There have been several efforts to analyze the behaviour of the EDCA mechanism. These efforts mostly consider the saturation conditions in which all stations and queues are always backlogged. In particular we found models developed by Robinson and Randhawa [3] and Xiao [25] noteworthy. However, the model developed by [25] only considers the
contention window size for prioritizing frames, ignoring the effect of different IFS times for different traffic classes (using Arbitration IFS: AIFS). The other important model, developed by Robinson and Randhawa, considered both parameters and provides a solution for combining the effects of both parameters. However, this model assumes a post collision behaviour that does not reflect the reality of 802.11 or 802.11e networks. For further material on related work, the reader can refer to Section 2.5.

In 802.11 when a collision happens stations who are involved in collision will wait for an Ack Timeout after the end of collision before they can realize that a collision happened, they will then resume normal contention; the other stations only wait until the end of the collided transmission and resume operation normally. This is different from assumptions in [3] that presume non-colliding stations will wait for an EIFS before resuming normal contention. Apart from this assumption, the study published in [3] presents a novel way of modeling the behaviour of contending stations with different priorities. However, [3] only presents a case with two priorities and a generalized model is not offered.

The ideas presented in [3] are the basis for our model. We correct the assumptions made in [3] and generalize the model for an arbitrary number of priorities instead of only two. Our model provides a complete solution for modeling 802.11e EDCA under saturation mode [36].

The reader can refer to Section 2.2 for information about the underlying technology of 802.11e networks.
5.2 Model Development

The EDCA function is a contention-based method in which stations desiring to access the medium have to wait for a random duration of time before they are allowed to start a transmission on the channel. This random waiting time is different for each priority and consists of two components: first the IFS which is a predetermined value for each priority level, and the second is the contention window size specified by priority-specific parameters $CWMIN$ and $CWMAX$, defined as the minimum and maximum sizes of the contention window, which define the backoff procedure. Each queue that wants to access the medium has to sense it idle for AIFS plus a randomly chosen backoff duration. The backoff duration is chosen randomly between 1 and $CW$ (Contention Window). $CW$ is initially set to $CWMIN$ but after each collision it is doubled until it reaches $CWMAX$. A successful transmission will reset $CW$ to its minimum.

Given the above description of the EDCA function, we introduce our model by mathematically defining the backoff process behaviour. We assume saturation situation in which all queues are always backlogged. As a result, all transmissions consist of two stages: waiting for the AIFS expiry, and waiting for the backoff counter to reach zero. To model the backoff process we first model the process for one queue with given EDCA parameters. This could be done following the same method described in earlier works by Bianchi [9] and Robinson [3]. As described in [9] and [3], the backoff value can be modeled as a two dimensional Markov Process.
For such a model to be correct there are two fundamental assumptions. One is that in each idle slot, each queue may attempt to transmit with an independent and constant priority. The other assumption is that at each transmission attempt, regardless of the number of collisions suffered, a packet may collide with an independent and constant probability $p$. However, since in EDCA the slots after a busy period are not all available to all priorities due to the AIFS differentiation mechanism, the number of contending stations may be different in each slot. Therefore to overcome this complexity we use an average conditional collision probability $p$ (as is used in [3]). The model that yields from such assumptions is a bi-dimensional Markov chain (refer to [9] for a detailed examination of this model).

From this model we derive the following equation that describes $\tau$ as a function of $p$ for each priority level (therefore we dropped the priority subscript):

$$b_{0,k} = \frac{CW_{min} + 1 - k}{CW_{min}} b_{0,1}$$  \hspace{1cm} (5.1)

$$b_{1,k} = b_{0,1} p^i$$  \hspace{1cm} (5.2)

$$\sum_{i=0}^{m} \sum_{k=1}^{W_i} b_{i,k} = 1$$  \hspace{1cm} (5.3)

$$\tau = \sum_{i=0}^{m} b_{i,1}$$  \hspace{1cm} (5.4)

The probabilities $b_{i,j}$ are the steady-state probabilities of the two-dimensional Markov Chain that models the backoff process [9][3]. Obviously, the $b_{i,j}$ probabilities are strongly
dependent on \( p \). To solve for \( p \) and we need to find \( p \) in each slot after a busy period and using the second set of equations we can derive \( p \) and consequently the throughput.

5.3 Slot Occupancy

We assume that each priority level may have its own AIFS and Contention Window. To examine the contention behaviour after busy periods, we assume that there are \( K \) possible priority levels with each slot containing \( n_j \{ j : (1, K) \} \) contending stations (or queues). Each priority level has an AIFS value equal to \( j \) slots, meaning that priority level \( j \) will start contention in the \( j \)'th idle slot. For example if there are only three priorities 1, 4, and 7; slots 1-3 will only have \( n_1 \) stations contending, while slots 4-6 will have \( n_1 + n_4 \) stations, slot 7 and after will have all \( n_1 + n_4 + n_7 \) stations contending. Note that we set \( n_j \) for other priorities to zero.

We model the contention slots after a busy period in a separate Markov Chain in order to derive the slot occupancy and probabilities of collision in each slot. Our model is a generalization of the model presented in [3]. The following Figure 5.1 describes our model. \( W_{min} \) is the minimum of all maximum contention window sizes (\( CW_{max} \)), and \( M \) is the lowest priority.

If \( n_i \) stations of priority \( i \) exist, then the equations that govern this model are as follows. For the first zone, where only the highest priority is active, the expression of the probability of transmission is:
Chapter 5. An Improved Saturation Model for IEEE 802.11e WLANs

Figure 5.1: Modeling Slot Occupancy for different contention zones

\[ P_{z=1}^{lr} = 1 - (1 - \tau_1)^{n_1} \]  
\[ (5.5) \]

For the second zone,

\[ P_{z=2}^{lr} = 1 - (1 - \tau_1)^{n_1}(1 - \tau_2)^{n_2} \]  
\[ (5.6) \]

It follows that, for an arbitrary zone, the general expression is

\[ P_{z=M}^{lr} = \begin{cases} 
1 - \prod_{j=1}^{z} (1 - \tau_j)^{n_j} & \text{if } 1 \leq z \leq M \\
1 - \prod_{j=1}^{M} (1 - \tau_j)^{n_j} & \text{if } z > M
\end{cases} \]  
\[ (5.7) \]

Knowing \( P_{z=M}^{lr} \) and assuming that the Markov chain is in equilibrium, the relationship between the slot occupancy steady-state probabilities is

\[ b_i = (1 - P_{z=i-1}^{lr}) \cdot b_{i-1} \]  
\[ (5.8) \]
And with the condition that

$$\sum_{i=0}^{W_{\text{min}}} b_i = 1 \quad (5.9)$$

It follows that

$$b_0 \cdot (1 - P_{z=1}^{\text{cr}}) = \sum_{i=1}^{M} b_i \cdot P_{z=1}^{\text{cr}} + \sum_{i=M+1}^{W_{\text{min}}} b_i \cdot P_{z=M}^{\text{cr}} \quad (5.10)$$

Therefore leading to

$$b_o = \frac{1}{1 + \sum_{i=1}^{W_{\text{min}}} \prod_{j=1}^{i} (1 - P_{z=j}^{\text{cr}})} \quad (5.11)$$

Next, we derive the average conditional collision probability $\bar{P}_j$ for each priority $j$ (probability that a transmitted packet of class $j$ encounters collision). As before, we give the example of two priorities, then the general expression for the probability. We know the probability of a collision happening amongst class $j$ stations in zone $z$, where $j = 1$ and $z = 1$:

$$P_{c_{z=1}}^{j=1} = 1 - (1 - \tau_1)^{n_1-1} \quad (5.12)$$

The second priority is not active in this region, so it follows that

$$P_{c_{z=1}}^{j=2} = 0 \quad (5.13)$$

In the second zone,
Chapter 5. An Improved Saturation Model for IEEE 802.11e WLANs

\[ P_{c_{z=2}}^{j=1} = 1 - (1 - \tau_1)^{n_1-1} \cdot (1 - \tau_2)^{n_2} \]  \hspace{1cm} (5.14)

and,

\[ P_{c_{z=2}}^{j=2} = 1 - (1 - \tau_1)^{n_1} \cdot (1 - \tau_2)^{n_2-1} \]  \hspace{1cm} (5.15)

The general expression for the conditional collision probability is

\[ P_{c_z}^j = 1 - (1 - \tau_j)^{n_j-1} \cdot \prod_{k=1,k\neq j}^Z (1 - \tau_k)^{n_k} \]  \hspace{1cm} (5.16)

Knowing the probability of collision within stations for a specific zone, we can find the average conditional probability of collision for class \( j \) stations (probability that in any zone, a collision occurs by class \( j \) stations' transmission):

\[ \bar{P}_j = \sum_{i=1}^{W_{\min}} b_i \cdot P_{c_{z=1}}^j \]  \hspace{1cm} (5.17)

Using the above equations we can derive the average collision and transmission probabilities for each priority class \( (\bar{P}_j, \tau_j) \). Using these probabilities we can derive the achievable throughput for each class as well as the total throughput of the system.

5.4 Throughput Analysis

To find the total throughput we need to find out the average probability of successful transmissions as well as the average probability of collisions in each slot \( i \) \( (P_{z=1}^{S}, P_{z=1}^{C}) \). We
can then use these probabilities and find the expected length of the transmission duration (including the time spent in probable collisions). For this purpose we find the average of the transmission periods initiated in each slot weighted according to the slot occupancy probabilities found in the previous section. The expected length of the transmission duration is expressed as follows:

\[
E_D = \sum_{i=1}^{W_{\text{min}}} b_i \cdot (\delta \cdot (1 - P_{zwi}^S - P_{zwi}^{Col}) + P_{zwi}^S \cdot T^i + P_{zwi}^{Col} \cdot T^{col})
\]  \hspace{1cm} (5.18)

To find the average probabilities of successful transmissions or collisions in each zone \(i\), we can use the probability of transmission in each slot, \(\tau_j\) for class \(j\), found in the previous section.

The probability of a given class \(j\) transmission in zone \(i\) is successful is expressed as follows:

\[
P_{j,zi}^S = \begin{cases} 
\tau_j \cdot \frac{1 - p_{\text{err}}}{1 - \tau_j} & \text{if } j \leq i \\
0 & \text{otherwise} 
\end{cases}
\]  \hspace{1cm} (5.19)

The probability of any given transmission in zone \(i\) is successful is written as follows:

\[
P_{zwi}^S = \sum_{j=1}^{i} n_j \cdot P_{j,zi}^S
\]  \hspace{1cm} (5.20)

The probability of any given transmission in zone \(i\) encounters a collision is expressed as follows:

\[
P_{zwi}^{Col} = P_{zwi}^{Tr} - P_{zwi}^S
\]  \hspace{1cm} (5.21)
Chapter 5. An Improved Saturation Model for IEEE 802.11e WLANs

The expected payload size indicates how much useful data is transmitted during the expected length of the transmission duration. The quantity $P_L$ refers to the average payload size in a frame. The expected payload size is:

$$E_p^j = n_j \cdot P_L \sum_{i=1}^{W_{\text{min}}} b_i \cdot P_s^i$$  \hspace{1cm} (5.22)

Using the expected length of transmission duration and the expected payload length we can derive the throughput for class $j$:

$$T_j = \frac{E_p^j}{E_D}$$  \hspace{1cm} (5.23)

5.5 Simulation Results

In this section, we present the simulation results, which are generated using a very accurate discrete-event simulator, using the parameters in Table 5.1 and Table 5.2. The reader can also refer to [36]. As the figures show, the mathematical model results and the simulation results are almost identical. In Figures 5.2 and 5.4, we plotted the collision probability $p$ versus the number of stations for four priorities (A being the higher priority). In Figures 5.3 and 5.5, we plotted the Saturation Throughput versus the number of stations with a different set of input parameters. Figures 5.2 and 5.3, labeled Batch 1, are plotted with the parameters in Table 5.1, and Figures 5.4 and 5.5, labeled Batch 2, are plotted with the parameters in Table 5.2.
### Table 5.1: Simulation Parameters for Batch 1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Payload</td>
<td>12000 bits</td>
<td>Bit Rate</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>MAC header</td>
<td>224 bits</td>
<td>Slot Time</td>
<td>20 µs</td>
</tr>
<tr>
<td>PHY header</td>
<td>192 bits</td>
<td>SIFS</td>
<td>10 µs</td>
</tr>
<tr>
<td>ACK</td>
<td>112 bits + PHY</td>
<td>Retry limit</td>
<td>4</td>
</tr>
<tr>
<td>RTS</td>
<td>160 bits + PHY</td>
<td>CTS</td>
<td>112 bits + PHY</td>
</tr>
<tr>
<td>AIFS1</td>
<td>SIFS + Slot Time</td>
<td>AIFS2</td>
<td>SIFS + Slot Time</td>
</tr>
<tr>
<td>AIFS3</td>
<td>SIFS + Slot Time</td>
<td>AIFS4</td>
<td>SIFS + Slot Time</td>
</tr>
<tr>
<td>CW_MIN1</td>
<td>8</td>
<td>CW_MIN2</td>
<td>16</td>
</tr>
<tr>
<td>CW_MIN3</td>
<td>24</td>
<td>CW_MIN3</td>
<td>32</td>
</tr>
<tr>
<td>CW_MAX1</td>
<td>16</td>
<td>CW_MAX2</td>
<td>32</td>
</tr>
<tr>
<td>CW_MAX3</td>
<td>48</td>
<td>CW_MAX4</td>
<td>64</td>
</tr>
</tbody>
</table>

### Table 5.2: Simulation Parameters for Batch 2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Payload</td>
<td>12000 bits</td>
<td>Bit Rate</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>MAC header</td>
<td>224 bits</td>
<td>Slot Time</td>
<td>20 µs</td>
</tr>
<tr>
<td>PHY header</td>
<td>192 bits</td>
<td>SIFS</td>
<td>10 µs</td>
</tr>
<tr>
<td>ACK</td>
<td>112 bits + PHY</td>
<td>Retry limit</td>
<td>4</td>
</tr>
<tr>
<td>RTS</td>
<td>160 bits + PHY</td>
<td>CTS</td>
<td>112 bits + PHY</td>
</tr>
<tr>
<td>AIFS1</td>
<td>SIFS + Slot Time</td>
<td>AIFS2</td>
<td>SIFS + 2*Slot Time</td>
</tr>
<tr>
<td>AIFS3</td>
<td>SIFS + 3*Slot Time</td>
<td>AIFS4</td>
<td>SIFS + 4*Slot Time</td>
</tr>
<tr>
<td>CW_MIN1</td>
<td>32</td>
<td>CW_MIN2</td>
<td>32</td>
</tr>
<tr>
<td>CW_MIN3</td>
<td>32</td>
<td>CW_MIN3</td>
<td>32</td>
</tr>
<tr>
<td>CW_MAX1</td>
<td>64</td>
<td>CW_MAX2</td>
<td>64</td>
</tr>
<tr>
<td>CW_MAX3</td>
<td>64</td>
<td>CW_MAX4</td>
<td>64</td>
</tr>
</tbody>
</table>
Chapter 5. An Improved Saturation Model for IEEE 802.11e WLANs

Figure 5.2: Collision Probability vs. Number of Stations for Batch 1

Figure 5.3: Saturation Throughput vs. Number of Stations for Batch 1
Chapter 5. An Improved Saturation Model for IEEE 802.11e WLANs

Collision Probability $p$ vs. Number of Stations

![Collision Probability vs. Number of Stations for Batch 2](image)

Figure 5.4: Collision Probability vs. Number of Stations for Batch 2

Throughput in Kbits vs. Number of Stations

![Saturation Throughput vs. Number of Stations for Batch 2](image)

Figure 5.5: Saturation Throughput vs. Number of Stations for Batch 2
Chapter 6

Adaptive Contention-Window MAC Algorithms for IEEE 802.11e Wireless LANs

6.1 Introduction

The rapid development of switching architectures, new smart antennas, and robust security, has lead to the wide proliferation of wireless local area networks (WLANs) into the enterprise and consumer markets. Currently, WLANs have become more than just a niche solution of extending the enterprise or institution network to unwired areas. For businesses, WLANs offer true economic advantage by helping enterprises maintain a flexible reusable local wireless networks that can be moved with the enterprise, or to improve employee mobility.
Some of the major problems that have been challenging WLANs, and wireless networks in general, are the imperfections and the time-varying characteristics of wireless channels. The high bit error rate, fading effect, and multipath propagation of the same signal, all contribute to the complexity of handling wireless channels. This motivates the development of new techniques to improve the performance. These new techniques rely on adapting the MAC to the present conditions of the wireless network.

In this chapter, we study the effect of changing the Contention Window Parameters (CW) on the Saturation Throughput (ST) of the wireless network. We use a modified version of the mathematical model developed in [3], which we published in [36], in one of our algorithms to optimize the performance.

The reader can refer to Section 2.2 for information about the underlying technology of 802.11e WLANs. Also, to know more about the related work, the reader can refer to Section 2.5.

### 6.2 Effect of the CW Parameters on the Saturation Throughput

The 802.11 MAC overhead includes access deference, transmission of RTS, CTS, and ACK packets which are specified by timing parameters that ensure correct network operation. When added to collisions, these parameters constitute a significant bandwidth and delay overheads, especially when the number of contending stations increases or demand more bandwidth. In this research we are concerned with the problem of the low Saturation
Throughput phenomenon, both for access point and wireless (client) station. In the follow­
ing discussion, the term “network” will be used to refer to area covered by a single Access 
Point and its associated stations.

The Saturation Throughput gets divided among the stations that are active in the net­
work. In [9][12], Bianchi showed that beyond a certain traffic load threshold, the through­
put of an 802.11 WLAN remains constant for a constant number of stations. The through­
put value at this point is called saturation throughput. The value of the Threshold depends 
on the network dynamics, and can vary considerably from one network to the other.

In the following we investigate the impact of the CW period on the performance of the 
saturation throughput. When executing the backoff procedure, a station chooses a random 
value with a uniform distribution over the interval \([1, \text{CWMIN} + 1]\) for the first transmission 
attempt. After each collision, the contention window is doubled, which means that the 
random value is chosen from the interval \([1, 2^i(\text{CWMIN}+1)]\), after the \(i^{th}\) collision. But 
the maximum limit of the contention window is not allowed to grow indefinitely, as the 
minimum of the two quantities \(2^i(\text{CWMIN}+1)\) and \(\text{CWMAX}\) is chosen. Therefore the two 
values \(\text{CWMIN}\) and \(\text{CWMAX}\) define completely the backoff procedure.

In Figures 6.1 and 6.2, the network is constructed of stations that have backlogged 
queues i.e. the buffers of each station never get empty. The parameters of the network 
are those shown in Table 6.1 unless otherwise specified. The Contention Window pa-
rameters, \(\text{CWMIN}\) and \(\text{CWMAX}\) are 7 and 15 respectively. As shown in Figure 6.1(a), the 
Saturation Throughput goes practically to zero as the number of stations increases in the
network. This is due to the Probability of Collision which approaches 1 as the number of stations in the network increases as shown in Figure 6.2(a).

To illustrate the effect of the Contention Window on the Saturation Throughput, we change $C_{\text{Wmin}}$ and $C_{\text{Wmax}}$ values to 15 and 31 respectively. The result is shown in Figures 6.1(b) and 6.2(b). This seemingly minor change in the Contention Window parameters greatly enhances the Saturation Throughput, increasing it by almost ten folds when the number of stations in the network reaches 50.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Payload</td>
<td>8000 bits</td>
<td>Bit Rate</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>MAC header</td>
<td>224 bits</td>
<td>Slot Time</td>
<td>20 $\mu$s</td>
</tr>
<tr>
<td>PHY header</td>
<td>192 bits</td>
<td>SIFS</td>
<td>10 $\mu$s</td>
</tr>
<tr>
<td>ACK</td>
<td>112 bits + PHY</td>
<td>Retry limit</td>
<td>5</td>
</tr>
<tr>
<td>RTS</td>
<td>160 bits + PHY</td>
<td>AIFS1</td>
<td>SIFS + Slot Time</td>
</tr>
<tr>
<td>CTS</td>
<td>112 bits + PHY</td>
<td>AIFS2</td>
<td>SIFS + 2*Slot Time</td>
</tr>
</tbody>
</table>

Table 6.1: Simulation parameters

Finding an optimal values of $C_{\text{Wmin}}$ and $C_{\text{Wmax}}$ for a single AC may not be hard, however with multiple ACs the problem becomes harder to solve, because we have to find a set of optimal $C_{\text{Wmin}}$ and $C_{\text{Wmax}}$ pairs and not just a single pair. To understand this, suppose there are $n$ discrete possible values of $C_{\text{Wmin}}$ and $k$ possible priority levels, then number of possible permutations for $k$ $C_{\text{Wmin}}$ values, one for each priority levels are $n!/(n-k)!$. This means that for 2 priority levels and 32 discrete possible values of each $C_{\text{Wmin}}$ number of possible solutions are $32!/(32-2)! = 992$. However, when we change the number of priority levels to 8 then the solution space drastically increased to $32!/(32-8)! =$
Chapter 6. Adaptive Contention-Window MAC Algorithms for IEEE 802.11e Wireless LANs

Figure 6.1: Effect of $C_{\text{MIN}}$ and $C_{\text{MAX}}$ on the Saturation Throughput

(a) Saturation Throughput (Mbps) with $C_{\text{MIN}}, C_{\text{MAX}} = (7, 15)$

(b) Saturation Throughput (Mbps) with $C_{\text{MIN}}, C_{\text{MAX}} = (15, 31)$
Figure 6.2: Effect of \( \text{CW}_{\text{MIN}} \) and \( \text{CW}_{\text{MAX}} \) on the Probability of Collision
4.2410e+011. In this study, we have developed algorithms to find the near optimal values of $C\text{WMIN}$ and $C\text{WMAX}$ for each priority level.

6.3 MAC Adaptation Algorithms

So far, three different algorithms have been developed for handling the optimal Contention Window parameters problem. In all three algorithms, the Access Point takes the responsibility of collecting information, computing throughputs, calculating Contention Window parameters, and then broadcasting the results back to the stations in the network, using Beacon frames. In this section, we will describe the developed algorithms. The three AP algorithms are capable of overcoming the “Saturation Throughput” problem. All algorithms are based on the same fundamental concept of using adaptive CW size control for maximizing the throughput. The three algorithms differ mainly in the adaptation mechanism.

6.3.1 The Station-Count Algorithm (SCA)

The Station-Count Algorithm establishes a linear equation between the CW parameters and the number of stations in the network. It takes the number of stations having the highest priority and multiplies by a constant to get the new value of $C\text{WMIN}[\text{highest}]$, then for the subsequent priority, it adds the number of stations of this priority multiplied by a constant to the $C\text{WMIN}$ of the highest priority, and so on. Algorithm 1 illustrates the SCA. The Station-Count Algorithm is a very simple algorithm. It is a deterministic algorithm, and doesn’t have the kind of uncertainty that the Derivative Algorithm has. The SCA works
as follows. For the highest priority AC, SCA multiplies the number of stations using this
AC by a constant to get the value of \( C_{WMIN} \). The \( C_{WMIN} \) for other ACs are calculated
recursively as follows
\[
C_{WMIN}[i - 1] = C_{WMIN}[i] + C2
\]
where \( C2 \) is a user defined parameter and AC \( i-1 \) has lower priority than AC \( i \). This
recursive calculation assures that higher priority ACs always have lower value of CW
as compared to low priority ACs. SCA also keeps the ratio \( C_{WMAX}[AC]/C_{WMIN}[AC] \)
constant. It performs better than the static case of the 802.11e (i.e. same network having
same parameters except for the CW parameters)

**Algorithm 1 - The Station Count Algorithm**

\[
\begin{align*}
\text{CWmin[highest AC]} &= C1 \times \text{Stations[highest AC]} \\
\text{FOR } i &= \text{highest AC} - 1; i > 0; i-- \\
& \quad \text{CWmin}[i] = \text{CWmin}[i+1] + C2 \times \text{Stations}[i] \\
& \text{END FOR} \\
\text{FOR } i &= \text{highest AC}; i > 0; i-- \\
& \quad \text{CWmax}[i] = \text{CWmin}[i] \times \text{CWratio}[i] \\
& \text{END FOR} \\
& \text{UPDATE data structures in the beacon frame} \\
& \text{SEND beacon frame}
\end{align*}
\]
6.3.2 The Throughput Derivative Algorithm (TDA)

After each CW change, the Throughput Derivative Algorithm employs measurements of the throughput taken in the Access Point (AP). It then tentatively changes the CWMIN of the system, and observes the derivative of the throughput. CWMAX is also changed while keeping the same ratio (new CWMAX/ new CWMIN), as the original (CWMAX / CWMIN) ratio.

The throughput derivative is taken over the present and a few past measurements of the throughput. We used the last 80 measurements in our simulations, with the frequency of 1 beacon/second. Algorithm 2 presents the Throughput Derivative Algorithm. Note that the AP, by controlling the CW parameters is essentially trying to track the optimum CW size for maximizing the WLAN throughput. In reality, the TDA only calculates CWMIN[AC] for each access category and calculate CWMAX[AC] by keeping the ratio CWMAX[AC]/CWMIN[AC] constant. The AP updates the values of CW parameters in every beacon frame.

The operation of the TDA is best described with Figure 6.3, where the throughput derivative is plotted versus time. We have divided the plot into four quarters for illustrative purposes:

1. The first quarter Q1 shows the derivative increasing, therefore the difference between the current derivative measurement and the past derivative measurement is positive, and the value of the current derivative is also positive, representing a throughput improvement. In such a case, the TDA keeps adding the predefined increment mul-
tiplied by the sign of each AC (\textit{diff\_sign}), to the CW parameter values in subsequent beacon frames.

2. In the second quarter Q2, although the difference between the current derivative measurement and the past derivative measurement is negative, but the value of the current derivative is positive, therefore it means the throughput is still increasing. The sign of \textit{diff\_sign} is not reversed.

3. In the third quarter Q3, the difference between the current derivative measurement (P2) and the past derivative measurement (P1) is negative and the value of the current derivative (P2) is negative, indicating a throughput decrease. This is the only case where the sign of \textit{diff\_sign} is reversed.

4. In the fourth quarter Q4, the value of the current derivative is negative, but the difference between the current derivative measurement and the past derivative measurement is positive. While the current derivative measurement is negative indicating a decrease in the throughput, it also indicates that the throughput is decreasing less sharply than the past measurement since the difference of the current and past derivative measurements is positive. Therefore, it means that the addition of the increment multiplied by the current sign of the \textit{diff\_sign} variable is working towards increasing the throughput. This is why the sign of \textit{diff\_sign} is not reversed in the fourth quarter.

The TDA has a real potential for being a smart solution for the ever-changing condition of the wireless medium, as well as the wireless network.
**Algorithm 2 - The Throughput Derivative Algorithm**

% mean is the average value of throughput measurements
% for last few observations

\[
\text{derivative}[AC] = \frac{\text{mean}[AC] - \text{old}_\text{mean}[AC]}{\text{beacon}\_\text{int}}
\]

IF derivative[AC] < old_derivative[AC] & derivative[AC] < 0 THEN

\[
\text{diff}_\text{sign}[AC] *= -1
\]

% Update every AC with the following equations. CWratio[AC]
% is a constant value

\[
\text{CWmin}[AC] = \text{CWmin}[AC] + \text{diff}_\text{sign}[AC] \times \text{step}
\]
\[
\text{CWmax}[AC] = \text{CWmin}[AC] \times \text{CWratio}[AC]
\]

UPDATE data structures in the beacon frame
SEND beacon frame

---

Figure 6.3: The TDA Operation Zones
6.3.3 The Analytical Algorithm

The Analytical Algorithm uses mathematical calculations to estimate the optimum CW parameters. It is based on a modified version of Robinson's model [3], which we published in [36], and is a typical optimization problem. Deriving the equations from the analytical model, we came out with a system of non-linear equations, with as many unknowns, and is presented in Algorithm 3.

The Analytical model is very accurate and provides the optimum value of the CW parameters right away, without further searching. It may even give some surprising result, such as interchanging of the CW parameters values between the highest and lowest priority, which will still lead to the highest possible throughput of the system.

Solving this system of non-linear equations may take long time. For example, it takes up to 2 seconds on a 2 GHz PC, which is not practical for a wireless Access Point, unless accelerated by hardware, or unless performing the calculations offline.

The algorithm changes with predetermined increments the CW parameters of all priorities, evaluates the throughput with these values and then records it in memory. Then a simple search of memory for the largest throughput yields the optimum we're searching for. In this system, only two priorities exist, priority A and priority B. The minPriorityA and maxPriorityA values are the range of priority A that we want the algorithm to search for the optimum in. The same applies to the minPriorityB and maxPriorityB values. The "incr" constant is the increment we want to add to the CW parameters on every evaluation.
Algorithm 3 - The Analytical Algorithm for access categories A and B

\[
\begin{align*}
\text{CWmin}[A] &= \text{minPriorityA}; \\
\text{CWmin}[B] &= \text{minPriorityB}; \\
\text{Maxthrup} &= \text{Sat_thrup}(A, B, CWmin[A], CWmin[B]); \\
\end{align*}
\]

FOR \( i = \text{minPriorityA}; i < \text{maxPriorityA}; i = i + \text{incr} \)
  
  FOR \( j = \text{minPriorityA}; j < \text{maxPriorityA}; j = j + \text{incr} \)
    
    \[
    \text{Thrup} = \text{Sat_thrup}(A, B, i, j);
    \]
    
    IF \( \text{Maxthrup} < \text{Thrup} \) THEN
      
      \[
      \begin{align*}
      \text{Maxthrup} &= \text{Thrup}; \\
      \text{CWmin}[A] &= i; \\
      \text{CWmin}[B] &= j; \\
      \end{align*}
      \]
    END IF
  END FOR
END FOR

UPDATE data structures in the beacon frame
SEND beacon frame

It is to be noted here that full enumeration of all possible solutions results in a large computation time. However, this can be reduced by properly adjusting the range \([\text{minPriorityA}, \text{maxPriorityA}]\) and the value of constant “\text{incr}” in the analytical algorithm. Currently we propose to use Analytical Algorithm for at most two priority levels (ACs), however, we are working on enhancing the algorithm to accommodate all possible priority levels.

### 6.4 Simulation Results

All the simulations and the analysis were done with the same parameters as in Table 6.1, except for priority 2 (or B) which the \text{CWmin} is equal to 32, and the \text{CWmax} is equal to 1024. It’s worth noting that the CW Parameters in Table 6.1 are the initial values. These will be changed later by the adaptive algorithm. In the Adaptive EDCA simulations, the Analytical Algorithm decides which CW parameters are the most suitable for the network. All the queues of all the stations are backlogged. Also unless it is mentioned in the figure,
all simulation scenarios use 50 user stations and a single AP. We have compared our adaptive algorithms with the standard IEEE 802.11e EDCA static parameter scheme; we call it static in our simulation results.

Figure 6.4 shows the ST of the Static EDCA and of the Adaptive EDCA using the Station-Count Algorithm plotted versus time. It was also simulated for 50 stations and shows approximately 7 times improvement of the ST. Figure 6.5 shows the change in the Contention Window Parameters, specifically in \( C_{\text{MIN}} \), when the SCA is activated at \( t = 1000 \) sec. In this scenario, we applied the TD algorithm to the CW of the higher priority, and fixed the CW of the lower priority with respect to the higher one. The TD algorithm can be applied separately to the two priorities, but it seems that only the CW of the higher priority will significantly change the ST.

Figure 6.6 shows the results of the throughput using the Throughput Derivative Algorithm: the Saturation Throughput (ST) is plotted versus time for both Static EDCA and Adaptive EDCA using the TD Algorithm. This figure has been plotted for 50 stations, and shows approximately 7 times improvement of the ST. The TDA algorithm has been enabled to work when time reaches 1000 seconds, and the difference is obvious before and after this point. However, since the TDA algorithm has the same improvement as the SCA in this case, it is worth noting that the TDA performs generally much better than the SCA, because of its adaptability and flexibility.

Figure 6.7 shows the same throughput but after being smoothed by a low-pass filter to prepare it for the use of the TDA. Figure 6.8 shows the derivative values derived from the
throughput that the TDA uses to calculate the changes in the CW parameters. Figure 6.9 shows the change in the Contention Window Parameters, specifically in $\text{CWmin}$, when the TDA is activated at $t = 1000$ sec.

Figure 6.10 shows the results of the simulation of the Analytical Algorithm along with the analysis and simulation results for the Static EDCA. This figure shows approximately 7 times improvement over the Static EDCA for 50 stations for the highest priority. Figure 6.11 shows the simulation results for priority B.

Figure 6.12 shows the proof of optimization for Priority A. In this figure, we plotted all the analysis results for the different CW parameters that the Analytical Algorithm has chosen, and compared them with the Adaptive simulation results. It shows that for different number of stations, the Analytical Algorithm chooses the optimal throughput value for Priority A and tracks the optimum of all the curves. We have chosen to optimize for Priority A instead of the Overall Throughput because of the focus of our group on QoS and Multimedia. Optimizing for the Overall Throughput is just as easy, and required to set a flag in both the analysis and the simulations.

### 6.5 Conclusion

In this chapter, we proposed centralized MAC adaptation algorithms for 802.11e wireless networks. The algorithms change the Contention Window Parameters following the present condition of the network. This study shows clearly the advantage of the adaptive schemes over the fixed-parameter 802.11e wireless networks.
The analytical algorithm calculates the maximum theoretical throughput that the network can have under optimal conditions. It could be used as a benchmark for other adaptive schemes.
Chapter 6. Adaptive Contention-Window MAC Algorithms for IEEE 802.11e Wireless LANs

Figure 6.4: Static vs. Station-Count Algorithm - Throughput

Figure 6.5: Static vs. Station-Count Algorithm - CWmin Change
Chapter 6. Adaptive Contention-Window MAC Algorithms for IEEE 802.11e Wireless LANs

Figure 6.6: Static vs. Throughput Derivative Algorithm - Throughput

Figure 6.7: Static vs. Throughput Derivative Algorithm - Low Pass Filtered Throughput
Figure 6.8: Static vs. Throughput Derivative Algorithm - Derivative

Figure 6.9: Static vs. Throughput Derivative Algorithm - CW\textsubscript{MIN} Change
Chapter 6. Adaptive Contention-Window MAC Algorithms for IEEE 802.11e
Wireless LANs

Static vs. Adaptive EDCA for Priority A Throughput

![Graph showing Static vs. Adaptive EDCA for Priority A Throughput]

Figure 6.10: Static vs. Analytical Algorithm for Priority A

Static vs. Adaptive EDCA for Priority B Throughput

![Graph showing Static vs. Adaptive EDCA for Priority B Throughput]

Figure 6.11: Static vs. Analytical Algorithm for Priority B
Chapter 6. Adaptive Contention-Window MAC Algorithms for IEEE 802.11e Wireless LANs

Figure 6.12: Proof of Optimization for the Analytical Algorithm
Chapter 7

Conclusions and Future Research

In this thesis, we proposed a new analytical model for finite-load wireless LANs, based on IEEE 802.11 and 802.11e MACs. An elaborate and detailed framework has given the model a high degree of accuracy which was observed when matching the results with a highly detailed WLAN simulator. Because of the ability to compute results for any amount (and type) of input traffic, our Enhanced Finite-Load (EFL) model approximates WLANs behaviour more realistically than Saturated models which assume backlogged queues. The highly-accurate Saturation model that we propose can help study congested WLANs, where multiple users are attempting to access the wireless network with high volumes of traffic.

The models are very accurate from the MAC layer point of view. However, the models currently do not include all details of PHY layer effects. Key aspects of PHY effects such as multi-rate operation and bit errors can be easily incorporated in our model.
The models and the proposed algorithms can be used in a variety of applications, ranging from wireless multimedia to network design. The models can help researchers understand better the complex behavior of probabilistic system such as WLANs. The models can be easily integrated as algorithmic procedures in an Access Point to help the scheduler take its decisions, and enforce Quality-of-Service guarantees or Access Control policies.

For example, the Enhanced Finite-Load models are especially efficient if used in deciding Access Control. As the shift from finite-load to saturation happens quite rapidly and in an exponential way, the Access Point can accept a new traffic flow or reject it if it causes a degradation in network performance. The AP can use the EFL to take such decisions and many other applications are also possible.

The main objectives of this work were to develop an accurate modeling platform for 802.11 and 802.11e WLANs under finite-load and saturation conditions, and the enhancement of the Saturation Throughput in 802.11e WLANs.

In Chapter 3, we presented an accurate and fairly complicated mathematical model for 802.11 WLANs called the “Enhanced Finite-Load Model (EFLM) for IEEE 802.11 Wireless LANs”. EFLM uses a dual queuing model where a node is represented by one queuing model, and the wireless medium by the other. The model introduces many probabilities and quantities that were not used in previously published finite-load models, such as the probability of direct transmissions $P_{\text{backoff}}$, and the probability of witnessing incomplete transmissions from other stations $P_x$. The model yields a set of non-linear equations, which can be solved with the algorithm listed in Appendix A.
Chapter 7. Conclusions and Future Research

Chapter 4 presented a mathematical finite-load model for 802.11e WLANs, and introduced new quantities to account for multiple priorities, such as the ratio of the presence of the traffic from one priority compared to another, and contention zones that depend on the AIFS periods of the priorities. The model was constructed with the same concepts as EFLM in Chapter 3, but the extension of the EFLM model to 802.11e WLANs is more complicated because of the prioritization mechanisms that were introduced in [7] such as the AIFS, and the different CW parameters for each AC.

In Chapter 5, we presented a mathematical model for saturated 802.11e WLANs, that corrects the model proposed by Robinson and Randhawa in [3]. Due to a misinterpretation of the 802.11e draft, the model had a flaw. We removed the probabilities of post-collision that Robinson and Randhawa had included, and re-built the model based on the new assumptions.

Chapter 6 presented three algorithms, that were devised to enhance the Saturation Throughput in saturated 802.11e WLANs. The proposed algorithms are based on adaptively changing the Contention Window (CW) parameters, trying to find the optimum CW parameter values. The Station-Count Algorithm (SCA) is a very simple algorithm and one of the earliest of our work. The Analytical Algorithm uses the mathematical model of Chapter 5 to find directly the optimum values. The Throughput Derivative Algorithm (TDA) is the algorithm that has the most potential, due mainly to its extreme adaptability and flexibility.

We will also list the possible research problems that could constitute a continuation of the work in this thesis.
Chapter 7. Conclusions and Future Research

1. In Chapters 3, 4 and 5, the mathematical models of 802.11 and 802.11e WLANs assume ideal wireless channel conditions. In real networks, different stations can send with different bit rates. To address the issue of differing bit rates, the stations in the models can be extended to have different success and collision times, $T_s$ and $T_c$. But when two stations with two different collision times collide, what is the resulting collision time? Surely, the longer one, but what are the probabilities of such collisions happening, and how could the average collision time be calculated from the stations' collision times? Some of these questions have been partially answered in [9] and [12]. But much more work is still needed.

2. The same can be noted for bit errors. In real networks, the radio links that are established between pairs of stations will certainly undergo bit errors. Compared to differing bit rates, it seems easier to solve the issue of bit errors, where stations respond to bit errors the same way they respond to collisions. However, unlike the probability of collision which depends on the number of stations in the network, the bit error probability is usually assumed constant for a given radio link, and can differ from one link to another.

3. Chapter 4 has introduced the concept of the ratio of the presence of one priority compared to another, called $Ratio_{XY}$. This ratio quantifies the presence of priority X compared to the presence of priority Y, which can be explained as the ratio of the number of transmissions from all queues of priority X over the number of transmissions from all queues of priority Y in a unit time. This ratio consists of several components that affect its value, and their inter-dependence is still a mystery. It
would be interesting to research the correct expression of this quantity.

4. In Chapter 6, the algorithms also use the ideal-condition assumptions. More robust and adaptive algorithms can be devised to handle realistic channel conditions, such as bit errors, and different bit rates.
Bibliography


[26] T.-C. Tsai, M.-J. Wu, “An Analytical Model for IEEE 802.11e EDCA”, in *IEEE International Conference on Communications*, vol. 5, pp. 3474 - 3478, May 2005


IEEE International Conference on Local Computer Networks, pp. 682-690, October 2003


[38] E. Winands, T. Denteneer, J. Resing, R. Rietman, “A Finite-Source Feedback Queueing Network as a Model for the IEEE 802.11 Distributed Coordination Function”, in The Fifth European Wireless Conference, February 2004


Appendix A

MATLAB Algorithm for Solving

Finite-Load Models - The Non-Linear

Set of Equations

All these equations form a non-linear set. This set could be solved with MATLAB. Since this set is not the usual one of \( k \) equations with \( k \) unknowns, a different approach should be taken. We used iterations to solve this set, and organized the equations into a large function. This function takes as argument a vector of values, and returns the same vector with the new updated values. Say the function's name is FiniteLoad, then:

\[
[p, P_{idle}, \rho, \overline{W}, \overline{W}_p, \pi_{net_k}] = \text{FiniteLoad}(p, P_{idle}, \rho, \overline{W}, \overline{W}_p, \pi_{net_k})
\]

We laid the equations in FiniteLoad in the following order (Equation numbers are different for 802.11e WLANs):

1. \( I_w \) (Eq. 3.5)
2. $\bar{W}_p$ (Eq. 3.6)

3. $\bar{W}$ (Eq. 3.7)

4. $I_s$ (Eq. 3.39)

5. $\pi_{net_k}$ (Eq. 3.38)

6. $P_{idle}$ (Eq. 3.41)

7. $P_{busy}$ (Eq. 3.42)

8. $P[NT]$ (Eq. 3.2)

9. $\rho$ (Eq. 3.1)

10. $1/\mu$ (Eq. 3.10)

11. $\rho = \lambda/\mu$

Note that $\rho$ was placed the last equation in FiniteLoad. The return vector returns the new values computed with the old ones. Proceeding with solving this set, we devised a smart algorithm that takes advantage of the structure of FiniteLoad. The algorithm goes as follows:

We should note that if the arrival rate is larger than the service rate, the system won’t converge. This is why we placed the SaturationCondition If statement in the algorithm. The algorithm assigns in each of the SolveLoops, and in the FinalLoop, $\rho$ to a fixed value (the third member of vector X which is $\rho$ is assigned to a fixed value in each iteration). The other arguments of X are passed to FiniteLoad and then saved into X again from the previous iteration.
Algorithm 4 - The Non-Linear Set of Equation Solving Algorithm

\[ Xi = [p_{init}, P_{idleInit}, ro_{init}, Wp_{init}, W_{init}, Pi_{knetInit}] \]
\[ d = 0.5 \]
\[ Oldro = 0.5 \]
\[ Minim = 1 \]
\[ Bestro = 0 \]

IterationsLoop: For \( j = 1 \) to \( 9 \)
\[ d = d / 2 \]
\[ X = Xi \]

SolveLoop1: For \( i = 1 \) to \( 5 \)
\[ X(3) = Oldro + d \]
\[ X = \text{FiniteLoad}(X) \]
End SolveLoop1

\[ \text{Diff1} = \text{AbsoluteValue}(\text{Oldro} + d - X(3)) \]
\[ X = Xi \]

SolveLoop2: For \( i = 1 \) to \( 5 \)
\[ X(3) = Oldro - d \]
\[ X = \text{FiniteLoad}(X) \]
End SolveLoop2

\[ \text{Diff2} = \text{AbsoluteValue}(\text{Oldro} - d - X(3)) \]

If \( \text{Diff1} > \text{Diff2} \)
\[ Oldro = Oldro - d \]
Else
\[ Oldro = Oldro + d \]
End If

End IterationsLoop

SaturationCondition: If \( \text{Minim} > 0.01 \)
\[ \text{Bestro} = 0.999 \]
End If SaturationCondition

\[ X = Xi \]

FinalLoop: For \( i = 1 \) to \( 6 \)
\[ X(3) = \text{Bestro} \]
\[ X = \text{FiniteLoad}(X) \]
End FinalLoop
The algorithm works as follows: it searches for the value of \( \rho \) in the range \([0,1]\) that causes the least error compared with \(X(3)\). First, the algorithm evaluates two values of \( \rho \), 0.75 and 0.25. Then it computes the differences of the results with 0.75 and 0.25 respectively, and stores the differences in Diff1 and Diff2. Say for example Diff2 is smaller. The algorithm takes 0.25 then as the center, and the next two values of \( \rho \) that are evaluated are 0.125 and 0.375. As the algorithm goes through its iterations, it closes on the value of \( \rho \) that causes the least error from the constant value that is fed to it.