Performance evaluation and improvement of IEEE 802.11 MAC protocol in wireless ad-hoc networks

Li, Zhifei

2005

Li, Z. (2005). Performance evaluation and improvement of IEEE 802.11 MAC protocol in wireless ad-hoc networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/2376>

<https://doi.org/10.32657/10356/2376>

Nanyang Technological University

Downloaded on 26 Apr 2023 18:28:24 SGT

Performance Evaluation and Improvement of IEEE 802.11 MAC Protocol in Wireless Ad-hoc Networks

Li Zhifei

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement of the degree of
Doctor of Philosophy

2005

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Prof. Anil Kumar Gupta, for his professional guidance, invaluable assistance, constant encouragement and patience throughout my graduate study. I am grateful to Dr. Sukumar Nandi for his insightful discussions, comments and advice on various aspects of this work. I would also like to thank Prof. Amitabha Das who provided guidance in the later part of my research-work. It is also my great pleasure to thank and acknowledge the graduate students in the Center for Multimedia and Network Technology (CEMNET), Xie Jiawei, Zhou Zhi, Wang Huiqing, Ye Minghua, and Xiao Shu for their encouragements and camaraderie.

I am eternally indebted to my wife, He Zhuangxiang, and my other family members for their love, support and advice, which the words cannot describe.

Contents

Acknowledgments	i
Contents	ii
Abstract	vii
List of Figures	ix
List of Tables	xii
List of Abbreviations	xiv
Publications from this Research-work	xvi
1 Introduction	1
1.1 Wireless Ad-hoc Networks	1
1.2 Medium Access Control	2
1.2.1 Physical Layer Assumptions	3
1.2.2 Classifications of Network Topology	5
1.2.3 MAC Performance Metrics	7
1.2.4 Classifications of MAC Protocols	8
1.3 Basic Techniques in IEEE 802.11	9
1.4 Research Objectives and Contributions	13
1.4.1 Research Objectives	13
1.4.2 Main Contributions	13
1.5 Organization of the Thesis	15
2 Analytical Model for Single-hop Scenario	16
2.1 Introduction	16
2.2 Analytical Model	18
2.2.1 Modelling Binary Exponential Back-off	18
2.2.2 Throughput Analysis	19
2.2.3 Delay Analysis	22
2.2.4 Fairness Analysis	31
2.3 Performance Evaluation of IEEE 802.11	33

2.3.1	Throughput Results	34
2.3.2	Packet Drop Probability Results	36
2.3.3	Delay Results	36
2.3.4	Fairness Results	37
2.3.5	Summary of the Performance Evaluation	38
2.4	Conclusions	43
3	Analytical Model for Hidden-terminal Scenario	44
3.1	Introduction	44
3.2	Short-Term Unfairness in IEEE 802.11	45
3.2.1	Condition for Successful RTS/CTS Handshake	46
3.2.2	Explanation for the Short-term Unfairness	47
3.3	Analytical Modelling	48
3.3.1	Markov Chain Model	48
3.3.2	Basic Analysis of the Model	51
3.3.3	Derivation of the Metrics	53
3.4	Transition Probability Computation	55
3.4.1	Transition Probabilities out of a Col State	55
3.4.2	Transition Probabilities out of a T state	57
3.5	State Holding Time Computation	58
3.6	Numerical Results	60
3.6.1	Transition Probabilities and State Holding Time	61
3.6.2	Comparison between Analytical and Simulation Results	63
3.7	Conclusions	66
4	FMAC/CSR: Fair MAC with Cooperation between Sender and Receiver	67
4.1	Introduction	67
4.2	Unfairness in IEEE 802.11	70
4.2.1	Unfairness Due to Concealed Information Problem	70
4.2.2	Long-term Unfairness Due to Imprecise Collision Detection Mechanism	74
4.2.3	Short-term Unfairness due to Deficiencies of BEB	75
4.2.4	Remarks on the MAC Fairness	76
4.2.5	Impact of Unfairness	77
4.3	General Framework to Achieve MAC Fairness	78
4.3.1	Fairness Model	78
4.3.2	Compensation Model	79

4.4	Estimation Algorithm	80
4.4.1	Estimation of the Number of Active Flows (n)	80
4.4.2	Estimation of the Actual Share (w_x)	82
4.4.3	Detection of Unfairness	83
4.5	FMAC/CSR: Fair Medium Access Control using Cooperation between Sender and Receiver	84
4.5.1	Sender-based Differentiated Access	85
4.5.2	Motivation to use Receiver Feedback	86
4.5.3	Receiver Feedback: Implicit Restrictive-notification	88
4.5.4	Receiver Feedback: Explicit Aggressive-notification	89
4.6	Simulation Results	90
4.6.1	Hidden-terminal Topologies	91
4.6.2	Asymmetrical Information Topologies	93
4.6.3	Scenario-1 of Figure 4.2	94
4.6.4	Scenario-7 of Figure 4.5	96
4.6.5	High-contention Topology (Figure 4.6)	96
4.6.6	A Complex Topology	97
4.6.7	Verification of Estimation Algorithm	99
4.7	Discussions and Related Work	99
4.7.1	Further Comments on FMAC/CSR	99
4.7.2	Related Work	102
4.8	Conclusions	104
5	Enhanced Carrier Sensing (ECS)	105
5.1	Introduction	105
5.2	Imprecise-EIFS Problem in IEEE 802.11	107
5.2.1	Large-EIFS Problem	107
5.2.2	Small-EIFS Problem	109
5.2.3	General Scenario involving Imprecise-EIFS Problem	111
5.3	Enhanced Carrier Sensing (ECS)	111
5.3.1	Distinguishing among SR Frames	113
5.3.2	EIFS Values in ECS	116
5.4	Simulation Results	117
5.4.1	Performance for Scenarios with Two-flows	118
5.4.2	Performance for Complex Scenarios	124
5.5	Conclusions	127

6	Full Auto Rate (FAR)	128
6.1	Introduction	128
6.2	Motivation	130
6.2.1	Transmission Rate of the Control Frames	130
6.2.2	Maximum Throughput of IEEE 802.11	130
6.3	Full Auto Rate (FAR) Algorithm	132
6.3.1	VCS Issues in Multi-rate IEEE 802.11	132
6.3.2	Modified Virtual Carrier Sensing (MVCS)	134
6.3.3	Rate Adaptation of Control Frames	136
6.3.4	Full Auto Rate (FAR))	137
6.4	Analytical Modelling	138
6.5	Simulation Results	140
6.5.1	Single Flow within Single-hop	141
6.5.2	Multiple Flows within Single-hop	142
6.5.3	Multiple Flows within Multi-hop	143
6.5.4	Lessons Learnt from the Simulation	144
6.6	Conclusions	145
7	Conclusions and Future Work	146
7.1	Main Research Contributions	146
7.2	Future Work	149
7.2.1	Analytical Models for Multi-hop Networks	149
7.2.2	Fairness Scheduling in Multi-hop Networks	149
7.2.3	Integration of FMAC/CSR, ECS, and FAR	150
7.2.4	Cross Layer Optimization between MAC Layer and Routing Protocols	150
7.2.5	MAC Protocols in Sensor Networks	150
	References	152
A	Adaptive Transmission Control (ATC)	157
A.1	Introduction	157
A.2	Adaptive Transmission Control (ATC)	158
A.2.1	Sender Side Mechanism: CW Tuning	158
A.2.2	Sender Side Mechanism: Early Reset	159
A.2.3	Receiver Side Mechanism: Restrictive Notification	159
A.3	Simulation Results	160

A.3.1	Hidden-terminal Topology (Figure 4.4)	161
A.3.2	Scenario-1 of Figure 4.2	162
A.3.3	High-contention Topology (Figure 4.6)	162
A.3.4	Asymmetrical Information Topology (Figure 4.3)	163
A.4	Conclusions	164
B	Average Throughput Computation in Scenario-10	165
C	MAC Performance and its Interaction with Routing Mechanism in Scenario-11	168
C.1	Factors Affecting MAC Performance	168
C.1.1	Capture Effect	169
C.1.2	Hidden Terminal Problem	169
C.1.3	Imprecise-EIFS Problem	169
C.2	MAC Performance under Static Routing	170
C.2.1	Throughput Comparison	170
C.2.2	Fairness Comparison	171
C.3	Interaction between MAC Performance and Routing Mechanisms	173
C.3.1	Throughput Comparison	173
C.3.2	Fairness Comparison	176
C.4	Conclusions	177

Abstract

In recent years, wireless ad-hoc networks have attracted considerable research interest as they are easy to deploy and maintain. The unique characteristics of such networks make the design of an efficient Medium Access Control (MAC) protocol very challenging. As IEEE 802.11 is the de facto standard for wireless LANs, it is also popularly used in the wireless ad-hoc networks. However, as IEEE 802.11 is not designed for wireless ad-hoc networks, it suffers from many problems (e.g., unfairness). In this thesis, our main objective is to evaluate and improve the performance of IEEE 802.11 MAC protocol in wireless ad-hoc networks.

In the performance evaluation part, we first present an analytical model for the single-hop IEEE 802.11 networks. It is very different from the models reported in the literature, which have focused either on throughput or on delay. Our model is unique as it obtains the throughput, delay, and fairness all together. Using the proposed model, we carry out an extensive performance evaluation of IEEE 802.11 by varying the various system parameters. From the results, we arrive at several important conclusions that are contrary to those appeared in the ones published in the literature because these models are imprecise. However, the above model as well as most of the published models is not applicable in the hidden-terminal scenario. In light of this, we then propose an embedded-Markov chain model, which is able to precisely explain and predict the short-term unfairness presented in the hidden-terminal scenario.

To address the unfairness problem that has been analytically revealed in the performance evaluation part, we present two novel fairness schemes: Fair MAC by Cooperation between Sender and Receiver (FMAC/CSR), and Adaptive Transmission Control (ATC). The basic idea of the above schemes is that, based on the information at the senders as well as at the receivers, the flows detect the occurrence of unfairness in a dynamic and distributed manner, and then they contend for the medium with dynamic priorities so as to compensate for their past usages of the shared medium. FMAC/CSR and ATC differ from each other in the methods of compensation, and in the role that the receiver plays. Extensive simulation results show that the proposed schemes substantially improve the MAC fairness without unduly degrading the throughput.

We also focus on improving the performance of IEEE 802.11 by adopting more practical assumptions. The first practical assumption we adopt is that the sensing range should be greater than the transmission range. In this case, we find that the carrier sensing mechanism adopted in IEEE 802.11 leads to substantial unfairness and throughput degradation. We propose an Enhanced Carrier Sensing (ECS) mechanism, which greatly improves the performance. Based on ECS, we also study the interaction between the MAC and routing protocols. Another practical assumption we consider is that the MAC protocol should be able to dynamically adjust the transmission rate of the frames according to the channel conditions. We observe that all the published rate-adaptation schemes adapt the transmission rate for the Data frames only while the control frames are always transmitted at a very low rate, leading to considerable overhead. To reduce this overhead, we propose a Full Auto-Rate (FAR) scheme, which is able to adapt the transmission rate not only for the Data frames but also for the control frames.

Our work clearly demonstrates that IEEE 802.11 can be used as the MAC protocol for wireless ad-hoc networks. Our work also shows that the performance of IEEE 802.11 in wireless ad-hoc networks can be substantially improved if the proposed modifications are adopted.

List of Figures

1.1	Hidden-terminal Scenario	6
1.2	Multi-hop Scenario with Spatial-reuse	7
1.3	Virtual Carrier Sensing in IEEE 802.11	11
2.1	Illustration of Delay for Successful Packet	24
2.2	Illustration of Delay for Dropped Packet	27
2.3	Throughput under Variable CW_{min}	39
2.4	Throughput under Variable CW_{max}	39
2.5	Throughput under Variable Retry Limit m	39
2.6	Ratio P_{col}/P_{succ} under Variable CW_{min}	39
2.7	Drop Probability under Variable CW_{min}	39
2.8	Drop Probability under Variable CW_{max}	39
2.9	Drop Probability under Variable Retry Limit m	40
2.10	Average Delay D_{succ} under Variable CW_{min}	40
2.11	Average Delay D_{drop} under Variable CW_{min}	40
2.12	Average Delay D_{notify} under Variable CW_{min}	40
2.13	Average Delay $D_{intersucc}$ under Variable CW_{min}	40
2.14	Average Delay $D_{infinite}$ under Variable CW_{min}	40
2.15	Standard Deviation of D_{succ} under Variable CW_{min}	41
2.16	Standard Deviation of D_{drop} under Variable CW_{min}	41
2.17	Standard Deviation of D_{notify} under Variable CW_{min}	41
2.18	Average Delay D_{succ} under Variable CW_{max}	41
2.19	Average Delay D_{succ} under Variable Retry Limit m	41
2.20	Fairness under Variable CW_{min}	41
2.21	Fairness under Variable CW_{max}	42
2.22	Fairness under Variable Retry Limit m	42
3.1	Hidden-terminal Scenario	46
3.2	State Transitions from a Col State	50

3.3	State Transitions from a T State	50
3.4	State Transition Diagram for $m = 3$	51
3.5	Computation of $Pr(T_x Col, CW_x, CW_y)$	56
3.6	Transition Probabilities out of a Col State	56
3.7	Counting down of the Back-off Counter at the Waiting Node	57
3.8	Transition Probability out of a T State	59
3.9	State Holding Time	61
4.1	Relationship between Node S_A and Flow B	71
4.2	Category 1: Scenario-1	72
4.3	Category 2: Asymmetrical Information Scenarios	73
4.4	Category 3: Hidden-terminal Scenarios	73
4.5	Scenario-7: Imprecise Collision Detection Mechanism	75
4.6	Scenario-8: High Contention (all nodes hear each other)	75
4.7	Flow Contention Graph for Scenarios with two Contending Flows	76
4.8	Scenario where Senders Overhear RTS/Data	80
4.9	Overall Process Followed at the Senders	86
4.10	Scenario Illustrating the Aggressive-notification	90
4.11	Fairness Index in Hidden-terminal Topologies	92
4.12	Fairness Index in Asymmetrical Information Topologies	94
4.13	Fairness Index under Topology of Figure 4.2	95
4.14	Fairness Index under Scenario-7 (Figure 4.5)	96
4.15	Fairness Index in High-Contention Topology	97
4.16	Scenario-9: Complex Topology	98
4.17	Fairness Index in Complex Topology (Figure 4.16)	99
4.18	Comparison between Actual and Estimation of n	100
5.1	3-nodes with Two Single-hop Flows (Scenario-10)	109
5.2	Throughput under Large-EIFS Problem	109
5.3	Time Diagram Showing Frame Exchange Sequence	109
5.4	4-nodes with Two Single-hop Flows (Scenario-11)	110
5.5	Time Diagram when FES(A, B) is in Progress while D defers	110
5.6	Areas of Influence Due to FES(A, B)	112
5.7	Problems Arising in Different Areas while FES(A, B) is in Progress	112
5.8	PLCP Frame Format	114
5.9	Different EIFS Values in ECS	117

5.10	4-nodes with Two Single-hop Flows (Scenario-12)	119
5.11	Fairness Index Comparison for Scenario-12	120
5.12	4-nodes with Two Single-hop Flows	120
5.13	4-nodes with Two Single-hop Flows (Scenario-13)	121
5.14	Topology (SR=TR) Identical to Scenario-13 (ECS)	121
5.15	4-nodes with Two Single-hop Flows (Scenario-14)	122
5.16	Topology (SR=TR) Identical to Scenario-14 (ECS)	122
5.17	4-nodes with Two Single-hop Flows (Scenario-15)	123
5.18	Topology (SR=TR) Identical to Scenario-15 (ECS)	123
5.19	Chain Topology: 10 Nodes with 9 Flows	125
5.20	Double-ring Topology: 16 Nodes with 8 Flows	126
6.1	Scenario Showing the Range of Interference	135
6.2	Modified Virtual Carrier Sensing (MVCS)	135
6.3	Maximum Throughput in IEEE 802.11b	140
6.4	Throughput Ratio between RBAR with MVCS and RBAR	140
6.5	Throughput Ratio between FAR and RBAR with MVCS	140
6.6	Throughput Ratio between FAR and RBAR	140
6.7	Throughput under Single-flow within Single-hop	141
6.8	Throughput Ratio under Adaptive Rate Algorithms	142
6.9	Throughput under Multiple-flows within Single-hop	143
6.10	Throughput under Multiple-flows within Multi-hop	144
A.1	Fairness Index in Hidden-terminal Scenario	161
A.2	Fairness Index in Topology of Figure 4.2	162
A.3	Fairness Index in High-Contention Topology	163
B.1	Two-states Markov Chain for Scenario-10	165
C.1	4-nodes with Two Single-hop Flows	168
C.2	Fairness Comparison under Static Routing	172
C.3	Medium State Determined by Node D when FES(A, B) is in Progress	175
C.4	Fairness Comparison under S4	176

List of Tables

2.1	IEEE 802.11 Parameters used in the Calculation	34
2.2	Interdependence between Performance and Parameters	43
3.1	Re-designating the State Variables	51
3.2	Transition Probabilities out of a Col State	62
3.3	Transition Probabilities out of a T State	62
3.4	num(j,i) when the state j is a Col state	63
3.5	State Probabilities Comparison	64
3.6	Expected Holding Time Comparison	64
3.7	Expected First Passage Time Comparison	65
3.8	Comparison of the Metrics	66
4.1	Summary of Three Categories of FMAC/CSR	91
4.2	Throughput under Hidden-terminal Topologies	92
4.3	Throughput under Asymmetrical Information Topologies	94
4.4	Throughput under Topology of Figure 4.2	95
4.5	Throughput under Scenario-7 (Figure 4.5)	96
4.6	Throughput in High-contention Topology	97
4.7	Throughput in Complex Topology (Figure 4.16)	98
4.8	Aggregate Throughput in Complex Topology	98
5.1	Throughput Comparison for Scenario-10	118
5.2	Throughput Comparison for Scenario-11	119
5.3	Throughput Comparison for Scenario-12	119
5.4	Throughput Comparison for Scenario-13	121
5.5	Throughput Comparison for Scenario-14	122
5.6	Throughput Comparison for Scenario-15	124
5.7	Throughput Comparison for the Chain Topology	125
5.8	Performance Measures Comparison (Chain Topology)	125

5.9	Throughput Comparison for the Double-ring Topology	126
5.10	Performance Measures Comparison (Double-ring Topology)	126
6.1	Main Operation in Full Auto Rate	138
A.1	Comparison between FMAC/CSR and ATC	160
A.2	Three Categories of ATC	161
A.3	Throughput under Hidden-terminal Topology	162
A.4	Throughput under the Topology of Figure 4.2	162
A.5	Throughput in High-contention Topology	163
A.6	Throughput in Asymmetrical Information Topology	164
C.1	Four Different Scenarios	170
C.2	Average Throughputs under Static Routing	170
C.3	Throughputs under Different Routing Protocols	174

List of Abbreviations

ACK:	Acknowledgement
ATC:	Adaptive Transmission Control
BEB:	Binary Exponential Back-off
BER:	Bit Error Rate
CA:	Collision Avoidance
CBR:	Constant Bit Rate
CCA:	Clear Channel Assessment
CD:	Collision Detection
CDMA:	Code Division Multiple Access
COV:	Coefficient Of Variation
CR:	Contention Resolution
CS:	Carrier Sensing
CSMA:	Carrier Sense Multiple Access
CSMA/CA:	Carrier Sense Multiple Access with Collision Avoidance
CTS:	Clear To Send
CW:	Contention Window
DCF:	Distributed Coordination Function
DIFS:	DCF Inter-Frame Space
DSSS:	Direct Sequence Spread Spectrum
ECS:	Enhanced Carrier Sensing
EIFS:	Extended Inter-Frame Space
FAR:	Full Auto Rate
FCS:	Frame Check Sequence
FES:	Frame Exchange Sequence
FHSS:	Frequency Hopping Spread Spectrum
FMAC/CSR:	Fair MAC by Cooperation between Sender and Receiver
LAN:	Local Area Network

MAC: Medium Access Control
MANET: Mobile Ad-hoc Network
NAV: Network Allocation Vector
PCF: Point Coordination Function
PDU: Protocol Data Unit
PER: Packet Error Rate
PHY: PHYsical layer
PIFS: PCF Inter-Frame Space
PLCP: Physical Layer Convergence Procedure
QoS: Quality of Service
RF: Radio Frequency
RTS: Request To Send
SIFS: Short Inter-Frame Space
SNR: Signal to Noise Ratio
SR: Sensing Range
TR: Transmission Range
VCS: Virtual Carrier Sensing
WLAN: Wireless Local Area Network

Publications from this Research-work

Papers accepted/published in journals:

- Z. F. Li, S. Nandi, A. K. Gupta, “Modeling the Short-term Unfairness of IEEE 802.11 in Presence of Hidden Terminals”, accepted to appear in Elsevier Performance Evaluation.
- Z. F. Li, S. Nandi, A. K. Gupta, “Improving MAC Performance in Wireless Ad Hoc Networks using Enhanced Carrier Sensing (ECS)”, accepted to appear in Elsevier Computer Communications.
- Z. F. Li, A. Das, A. K. Gupta, S. Nandi, “A Full Auto Rate (FAR) MAC Protocol for Wireless Ad Hoc Networks”, accepted to appear in IEE Proceedings - Communications.
- Z. F. Li, S. Nandi, A. K. Gupta, “Improving fairness in IEEE 802.11 using enhanced carrier sensing”, IEE Proceedings - Communications – October 2004 – Volume 151, Issue 5, p. 467-472.

Papers published in conferences:

- Z. F. Li, S. Nandi, A. K. Gupta, “Achieving MAC Fairness in Multi-hop Wireless Ad hoc Networks using Adaptive Transmission Control”, The Ninth IEEE Symposium on Computers and Communications (ISCC 2004), Alexandria, EGYPT, June 28 - July 1, 2004
- Z. F. Li, S. Nandi, A. K. Gupta, “Modeling the Short-term Unfairness of IEEE 802.1 in Presence of Hidden Terminals”, The Third IFIP Networking Conference (Networking 2004), Athens, Greece, May 9- May 14, 2004.
- Z. F. Li, S. Nandi, A. K. Gupta, “Improving MAC Performance in Wireless Ad Hoc Networks using Enhanced Carrier Sensing (ECS)”, The Third IFIP Networking Conference (Networking 2004), Athens, Greece, May 9- May 14, 2004.

- Z. F. Li, S. Nandi, A. K. Gupta, “Study of IEEE 802.11 Fairness and its Interaction with Routing Mechanism”, The Fifth IFIP International Conference on Mobile and Wireless Communications Networks (MWCN 2003), Singapore, May 2003.

Papers under review

- Z. F. Li, A. K. Gupta, S. Nandi, “FMAC/CSR: a Fair MAC Protocol for Wireless Ad hoc Networks”.
- Z. F. Li, A. Das, A. K. Gupta, S. Nandi, “Performance Analysis of IEEE 802.11 DCF: Throughput, Delay, and Fairness”.

Chapter 1

Introduction

Recently, wireless networks are becoming more and more popular. Among them, wireless ad-hoc networks have attracted considerable research interest as they are easy to deploy and maintain. In this thesis, we mainly focus on the Medium Access Control (MAC) layer in wireless ad-hoc networks. In this chapter, we first present the background knowledge of the MAC protocols in general, followed by specific details of IEEE 802.11 MAC protocol. Then we state the research objectives, and highlight our main contributions. The organization of the thesis is also presented at the end of this chapter.

1.1 Wireless Ad-hoc Networks

The concept of wireless ad-hoc network originates from the packet radio networks [33, 31] developed during 1970s. Currently, the well-known examples of wireless ad-hoc networks are the mobile ad-hoc networks (MANETs) [28] and the wireless sensor networks [2]. Here, we give a brief introduction about the unique characteristics about the wireless ad-hoc networks.

In wireless ad-hoc networks, all the nodes communicate with each other through the wireless medium. Though the wireless medium has its advantages, it also brings some new challenges. The most important one is the high bit error rate. This is mainly due to the multi-path fading and Doppler frequency shift [51]. Specifically, the signal received by a node is a superposition of time-shifted and attenuated versions of the transmitted signal. As a result, the received signal power varies as a function of time. When the received signal strength drops below a certain threshold, the node is said to be in fade and thus error occurs.

In a wireless ad-hoc network, there is no infrastructure or centralized nodes. Therefore, all the nodes should take the same responsibilities and the algorithms designed for such a network

must operate in a distributed manner. The ad-hoc nature brings about many advantages, e.g., easy to deploy, robustness, self-organization, and self-healing. This makes it particularly attractive in many scenarios such as conferences, meetings, lectures, search and rescue operations, disaster recovery, and battlefields. However, the ad-hoc nature also brings many challenges. For example, since all the nodes are operated in a distributed manner, it is very difficult to get a global knowledge about the network and thus optimization algorithms that require a global knowledge may not be feasible in a wireless ad-hoc network.

In contrast to the last-hop wireless network (e.g., wireless LAN or GSM network), which uses wireless medium only in the last hop, all the links in a wireless ad-hoc network use the wireless medium. Due to the limited transmission range, a packet normally has to traverse multiple hops from the source to the destination. Therefore, wireless ad-hoc networks are normally multi-hop networks. The multi-hop property also brings about many challenges. For example, in the last-hop wireless network, no routing protocol is necessary in the wireless domain. In contrast, in a multi-hop ad-hoc network, routing protocols are indispensable to forward the packets from source to destination.

Another distinguished characteristic of wireless ad-hoc networks is mobility. Mobility results in the well-known Doppler frequency shift [51] and thus makes the propagation environment more complex. Moreover, due to mobility, the network topology may change frequently and unpredictably, making it difficult to find a route from the source to the destination. Mobility also makes the Quality of Service (QoS) support more challenging since the resources available to a node are time varying.

In summary, the unique characteristics (i.e., wireless medium, ad-hoc, multi-hop, and mobility) of wireless ad-hoc networks bring about many advantages as well as challenges, and thus this kind of networks has attracted substantial research interest during the past years.

1.2 Medium Access Control

In wireless networks, a MAC protocol moderates access to the shared medium by defining rules that allow the nodes to communicate with each other in an orderly and efficient way [12].

1.2.1 Physical Layer Assumptions

The design of a MAC protocol and its performance greatly depend on the assumptions made for the physical layer. The following assumptions are of great importance:

Single-channel: Due to the constraints of antennas and their size, most of the MAC protocols in wireless ad-hoc networks assume that only one channel is available at a node, though some research works (e.g., [30]) do explore the possibility of multiple channels. The single-channel here implies that a node can transmit (or receive) not more than one frame at a time. Throughout this thesis, the single-channel assumption will be adopted.

Half-duplex: In a single-channel wireless system, it is very difficult for a node to receive data when it is transmitting. This can be explained as follows [12]. When a node is transmitting, the strength of the received signal from its own transmission will be very high in comparison to that from any other station due to the fading along the distance, and therefore, the signal from other stations cannot be decoded clearly. This property also implies that the collision detection (CD) used in the wire-line Ethernet is not possible in wireless networks. Throughout this thesis, the half-duplex assumption is adopted.

Capture: As mentioned, in a single-channel system, it is not possible for a receiver to receive more than one frame simultaneously when there are multiple frames around it. However, in this case, it may be possible for a receiver to correctly receive *one* of the multiple frames around it, which is known as *capture*. Several capture models (e.g., [50]) have been developed to discuss the conditions under which a frame can be correctly received in the presence of other interfering frames. We adopt the most intuitive one, that is, a frame may be captured if both of the following conditions are satisfied: (i) The Signal to Noise Ratio (SNR) of the captured frame is greater than a threshold. (ii) The frame satisfying the first condition arrives at the receiving node earlier than any other frames.

To be able to capture a frame under strong interference from other frames, a node needs to be equipped with a more sensitive physical layer, and therefore certain nodes may not have the capture ability. In such situations, a collision occurs whenever there are multiple frames around a receiving node and all the frames will be destroyed. In fact, a majority of MAC protocols (e.g., [37], [6], and [20]) are designed under the assumption that a node does not have

the capture ability. In this thesis, unless explicitly mentioned, we assume that a node does not have the capture ability.

Sensing Range and Transmission Range: It is well known that the transmission power of the wireless signal attenuates over the distance traversed by the signal. Based on the attenuation, two ranges are defined: the *transmission range* (TR) and the *sensing range* (SR). Correspondingly, we call a frame detected by a node within TR as the *TR frame*, and a frame detected by a node out of TR but within SR as the *SR frame*. A TR frame can be received correctly if there is no transmission error. In contrast, an SR frame cannot be received correctly though it can be detected by the carrier sensing.

As we will show in this thesis, the ratio between SR and TR, i.e., SR/TR , plays an important role in determining the performance of MAC layer. Intuitively, SR/TR should be at least equal to one. In the literature, for simplicity, a majority of the MAC protocols (e.g., [37], [6], and [20]) are designed based on the assumption that SR/TR is equal to one. However, in practice, the SR/TR ratio normally should be greater than one [59]. In this thesis, we consider both cases. However, unless explicitly mentioned, we assume that the SR is equal to TR.

Symmetrical Channel and Identical Power: For simplicity, most works assume that the transmission power between two communicating nodes is identical. Also, they assume that the channel between the communicating pair is symmetrical, i.e., the channel quality does not depend on the direction. These assumptions imply that if node I can hear node J clearly, then node J can also hear node I clearly. This is different from the assumption made in some power control algorithms such as [32], in which the transmission power is adaptively tuned based on the channel conditions between the communication nodes. Throughout this thesis, the assumptions of symmetrical channel and identical power are adopted.

Omni-directional Antenna: In order to reduce the interference and thus improve the capacity, some research works (e.g., [14]) adopt directional antenna into ad-hoc networks. However, this brings about many new challenges and makes most of the MAC protocols difficult to use because they are designed for the omni-directional antennas. Throughout this thesis, the omni-directional antenna assumption is adopted.

Modulation Rate: Recently, some researchers have proposed adaptive modulation algorithms (e.g., [23]), which can dynamically change the modulation scheme (and thus the modulation rate) during the communication process according to different SNR conditions of the medium. However, this may increase the complexity of the physical layer. Therefore, most works assume that the modulation rate remains fixed during the whole communication process. In this thesis, we consider both cases. However, unless explicitly mentioned, we assume that the modulation rate is fixed.

1.2.2 Classifications of Network Topology

In addition to the physical layer assumptions, the performance of MAC protocol also greatly depends on the topology of the network. From the viewpoint of MAC layer, we can classify the network topology into three categories: single-hop networks, multi-hop networks without spatial-reuse, and multi-hop networks with spatial-reuse. Note that we distinguish between multi-hop and single-hop from the viewpoint of the *topology* rather than the *traffic*, since for the MAC layer all the traffic are *single-hop* though at the upper layers some of the traffic may be multi-hop.

Single-hop Networks: In a single-hop network, all the nodes in the network are within the range of each other. Therefore, all the nodes will share the same medium, and obviously spatial-reuse is impossible. Moreover, any activity on the shared medium will be detected by all the nodes. Due to this information synchronization, the MAC design as well as the performance evaluation for this kind of network is simpler compared to, for example, the hidden-terminal scenario discussed below.

Multi-hop Networks without Spatial-reuse: In this kind of networks, some of the nodes in the network cannot hear from each other, implying that it is a multi-hop network. However, all the nodes in the network share the common medium, and thus only two nodes can communicate with each other at any time. In other words, spatial-reuse is not possible in such a case.

As an example, the scenario shown in Figure 1.1 belongs to such a network. In the figure, there are three nodes, A, B and C. Nodes A and C cannot hear each other, and thus if A wants to communicate with C, the packet must be forwarded by node B, implying that it is a

multi-hop network. However, at any point of time, only two nodes can communicate with each other (i.e., A and B, or B and C), and thus spatial-reuse is impossible.

The above scenario is also known as the hidden-terminal scenario. Consider that there are two single-hop flows: from A to B, and from C to B. Since nodes A and C cannot hear each other, they may simultaneously try to communicate with the common node, i.e., node B, resulting in a collision. In such a situation, nodes A and C are referred as the hidden terminals of each other. The reason, that nodes A and C cannot hear from each other, may be that A and C are out of the range of each other. It may also be that there are some physical obstacles such as a wall between A and C.

In contrast to the single-hop scenario, in a multi-hop scenario, when a node is transmitting, the other nodes (which share the common medium with the transmitting node) may not know this transmission. Therefore, the information of the medium state obtained at a node may not be precise. Clearly, the MAC design as well as the performance evaluation becomes more complex than that for single-hop networks.

Figure 1.1: Hidden-terminal Scenario

Multi-hop Networks with Spatial-reuse: As stated above, in a multi-hop network, some of the nodes cannot hear from each other. However, different from the multi-hop networks without spatial-reuse, there may be more than one pair of nodes that can communicate at the same time, and therefore, spatial-reuse is possible.

For example, in the scenario of Figure 1.2, there are three single-hop flows: F_A , F_B , and F_C . Flows F_A and F_C are in the same contention range and thus share a common medium, while flows F_B and F_C share another common medium¹. Clearly, when flow F_A is transmitting, flow F_B can also transmit, showing the spatial-reuse in this scenario. However, when flow F_C is transmitting, both flows F_A and F_B cannot transmit since flow F_C is contending in the two mediums.

The protocol design in a multi-hop scenario with spatial-reuse becomes even more complex compared to the one without spatial-reuse. However, since the role of a MAC protocol is to

¹Note that the single-channel assumption is still applicable.

Figure 1.2: Multi-hop Scenario with Spatial-reuse

distribute a *common* medium among the contenders, it should deal with the access to the shared medium only, without being concerned with the spatial-reuse of the channel. There should be a subnetwork layer that is laid between the MAC layer and the network layer to deal with the issue of the spatial-reuse of the common channel. Since our main focus is on the MAC protocol, we will mainly consider the scenarios without spatial-reuse (i.e., the first two kinds of networks).

1.2.3 MAC Performance Metrics

Given the wide range of protocols that have been proposed, it is necessary to understand the metrics used to compare the MAC protocols. We present several widely used metrics here [12].

Capacity Utilization (or Throughput): Since the wireless bandwidth is very limited, it is important to maximize the utilization of the capacity. The ideal way to maximize the utilization is to adopt a scheduler, which can coordinate the transmissions at all the nodes in a global and optimal manner. However, in a wireless ad-hoc network, it is difficult to implement such a scheduler due to its inherently distributed nature. Therefore, sometimes the medium is idle and sometimes the medium experience collisions, both of which result in capacity wastage.

Fairness: A MAC protocol is fair if it does not exhibit preference to any single node when multiple nodes are trying to access the medium. However, since all the nodes in ad-hoc networks operate in a distributed manner without having precise information of the competing nodes, the MAC protocols suffer from unfairness.

Based on the *length* of the time over which we observe the system, the fairness can be defined on a short-term basis and a long-term basis. In the literature, it is not clear where the dividing line is between the short-term and long-term. In our work, we interpret the short-term as the time interval during which each node in the network can transmit three or four packets. The

short-term fairness implies long-term fairness, but not the vice versa [40]. In particular, under certain scenarios, though the bandwidth allocation is fair in a long-term (e.g., equal average throughput or delay), it is very unfair if we view the system from a short-term viewpoint. While the impact of long-term unfairness is obvious, the short-term fairness is very important for the adaptive traffic (e.g., TCP kind of traffic) and for the delay- or jitter-sensitive traffic [40].

QoS Support: With the convergence of voice, video, and data networks, it is necessary for the MAC protocols to support multimedia traffic. As a result, MAC protocols should be able to support different QoS classes for different kinds of traffic. Among the QoS parameters, packet delay is of great importance. The definition on fairness should be modified when traffic of multiple priorities is to be handled. Specifically, when multimedia traffic is supported, fairness is defined as being able to distribute bandwidth in proportion to the priorities. Clearly, fairness forms the basis for QoS support.

Other Metrics: Since most wireless devices have limited battery power, it is extremely important for MAC protocols to provide power saving features. Therefore, power consumption is an important performance metric. Another metric is the protocol stability [38], which is required when there are large variations in the amount of traffic offered by the higher layers to the MAC layer.

The maximization of the above performance metrics can also be viewed as the design objectives that a MAC protocol aims to achieve. However, the maximization of one metric may be conflicting with that of another. For example, maximizing throughput and achieving fairness are normally conflicting objectives in wireless ad-hoc networks [42]. Therefore, every MAC protocol tries to achieve a tradeoff among the above metrics. In our research work, we mainly consider three metrics: throughput, fairness, and QoS support (more specifically, packet delay).

1.2.4 Classifications of MAC Protocols

Generally, MAC protocols can be classified into two categories: fixed access and random access [12]. TDMA [16], FDMA [4] and CDMA [39] are the typical examples of the first class while Aloha [1] and CSMA family [38] belong to the second class. While the performance of fixed access is predictable, it is not suitable for burst traffic due to its low efficiency. Moreover, the fixed access

schemes need precise synchronization in the network. In the ad-hoc networks, since nodes inject traffic onto the medium in an ad-hoc manner and achieving global synchronization is very difficult, the random access method is preferred.

From another viewpoint, MAC protocols can be classified into the centralized MAC and the distributed MAC. Normally, fixed access protocols operate in a centralized manner while random access protocols operate in a distributed manner. In the ad-hoc networks, since no centralized nodes (such as base stations) are available, distributed MAC protocols are more promising. Therefore, in this thesis, we mainly focus on the *distributed random* MAC protocols. In particular, we focus on MAC protocols belonging to the CSMA family. Many protocols, such as MACA [37], MACAW [6], FAMA [20], and IEEE 802.11 [24], belong to this family. Among them, since IEEE 802.11 is the de-facto industry standard for Wireless LANs, most of the research-work on wireless ad-hoc networks adopt IEEE 802.11 as the MAC protocol even though it is not designed for such kind of networks. Our research-work is mainly based on IEEE 802.11.

1.3 Basic Techniques in IEEE 802.11

IEEE 802.11 [24] defines two MAC protocols, i.e., Point Coordination Function (PCF) and Distributed Coordination Function (DCF). However, only DCF is used in wireless ad-hoc networks since PCF requires base stations. DCF is a CSMA/CA based protocol, which normally have several components: Carrier Sensing (CS), Collision Avoidance (CA) and Contention Resolution (CR).

Collision Avoidance (CA): As collision detection (CD) is not possible in wireless networks, most of the MAC protocols designed for wireless ad-hoc networks adopt some kind of handshaking to decrease the chances of collision. The IEEE 802.11 DCF defines two kinds of handshaking, namely, the two-way and the four-way handshake. In the two-way handshake, the sender first transmits a Data frame to the receiver, which then responds with an ACK frame if it receives the Data correctly. It is evident that the two-way handshake suffers from the hidden-terminal problem (Figure 1.1). On the other hand, in the four-way handshake, before the transmission of a Data frame, the sender first sends out a Request To Send (RTS) frame. In response to this request, the receiver sends back a Clear To Send (CTS) frame if the medium

is determined to be idle. Then, the sender sends out the Data frame and the receiver responds with an ACK frame if it receives the Data frame without error. For the convenience, we call the exchange of Data/ACK or RTS/CTS/Data/ACK frames as a *frame exchange sequence* (FES). FES(A, B) represents a FES between nodes A and B, initiated by node A, implying that node A sends one *packet* successfully to node B. For simplicity, we call node A as the *transmitting node* and B as the *receiving node*, while all the other nodes that overhear any one of the frames are called the *waiting nodes*. Note that the word ‘packet’ implies the protocol data unit (PDU) of a higher layer whereas ‘frame’ is the MAC layer PDU.

In the four-way handshake, once the RTS/CTS has been completed successfully, the hidden-terminal problem does not arise any more. For example, in Figure 1.1, once node B sends back a CTS to node A, node C also hears this CTS and defers its transmission, avoiding the collision of the Data frame. The four-way handshake solves the hidden-terminal problem largely by introducing the RTS/CTS handshake before the actual Data frame is transmitted. However, it cannot eliminate the problem completely as the transmission of the RTS frames from the hidden terminals may collide, and thus the RTS/CTS handshake cannot always be successful. Due to the multi-hop nature of wireless ad-hoc networks, the hidden-terminal problem arises very commonly, and therefore the two-way handshake is not likely to be used. Thus, in our work, we mainly focus on the four-way handshake.

Carrier Sensing (CS): In an IEEE 802.11-based wireless ad-hoc networks, since all the nodes share a common channel, only one flow among the contending flows can transmit during a certain duration (note that two flows can transmit simultaneously if they are not contending). Therefore, whenever a frame exchange sequence between two nodes is in progress, it is extremely important to guarantee that all the other nodes in the interference range should defer their own transmission. This is achieved through the physical carrier sensing and virtual carrier sensing (VCS) mechanisms in IEEE 802.11. The physical carrier-sense mechanism is implemented at the physical layer. On the contrary, the VCS is implemented at the MAC layer. When any one of the CS mechanisms indicates a busy medium, the medium shall be considered busy; otherwise, it shall be considered idle.

For the operation of VCS, every frame should contain a *duration* field, which stores the time duration that the medium is to be reserved for the on-going sequence. In IEEE 802.11,

the duration value carried in a given frame should be large enough to allow the transmission of *all* the remaining frames in the sequence. When an overhearing node overhears a frame, it updates its variable called Network Allocation Vector (NAV) with the duration value carried in the overheard frame if the following conditions are satisfied: (i) the frame is received without error; (ii) the frame is not addressed to the node itself; and (iii) the duration value carried in the frame is greater than the current NAV value. The third condition is required since a node may overhear many frames corresponding to different exchange sequences. A node with a NAV greater than zero will defer its transmission as if the medium were physically busy, explaining why the mechanism is called the *virtual* carrier sensing. Figure 1.3 shows an example where the source node *Src* sends a packet to the destination node *Dst*. In the figure, node A is within the transmission range of node *Src* but out of the range of node *Dst*. On the other hand, node B is within the range of *Dst* but out of the range of *Src*. Also, the duration values carried in the RTS, CTS, Data, and ACK frame are represented by D_{RTS} , D_{CTS} , D_{Data} , and D_{ACK} , respectively. Note that the D_{ACK} is zero as the ACK is the last frame in the sequence.

Figure 1.3: Virtual Carrier Sensing in IEEE 802.11

Contention Resolution (CR): The CR is used as a rate control algorithm by a node to adjust its traffic rate by observing whether its transmission is successful or not. IEEE 802.11 adopts the well-known Binary Exponential Back-off (BEB) algorithm as its CR mechanism, which is described as follows. Whenever an attempt to transmit a packet fails, a retransmission is scheduled, unless a retry limit (say m) is reached. Every station maintains a Contention Window (CW) and a back-off counter. Before every transmission (including a retransmission), the station first backs off by a random time according to Equation (1.1). In the case that the back-off counter already contains a non-zero value, it is unnecessary to generate a new random back-off counter.

$$\text{Backoff Time} = \text{Random}() \times \sigma \quad (1.1)$$

where σ represents the slot time and is specified by the physical layer, and the *random* value is uniformly distributed over the range $[0, CW]$. For the *first* transmission attempt of a packet, the CW is set to CW_{min} . Whenever a retransmission is initiated, the CW is increased. The CW reaches to its *maximum* value after m' retries. After that, the CW remains at the maximum value until the retry limit (i.e., m) is reached. Normally, m is greater than m' . Once the retry limit is reached, the packet is dropped, and the CW is reset to CW_{min} . The CW is also reset to CW_{min} whenever a transmission attempt is successful. For convenience, we call a retransmission attempt as a *stage*. If we denote the *stage number* by index i , which is in the range $[0, m]$, the CW can be expressed as:

$$CW_i = \begin{cases} (CW_{min} + 1) \times 2^i - 1 & 0 \leq i < m' \\ (CW_{min} + 1) \times 2^{m'} - 1 & m' \leq i \leq m \end{cases} \quad (1.2)$$

When a station (say H) is transmitting a packet, the other stations *freeze* their back-off counter. After station H completes the transmission of the packet and thus the medium becomes idle, all the contending stations first defer for a DCF Inter-Frame Space (DIFS) period. Then, station H generates a new random back-off counter and backs off before it initiates another transmission. On the contrary, the other stations simply resume to count down from their *frozen* back-off counters. Contrary to a successful transmission, when a collision occurs, all the colliding stations will generate a new random counter.

Inter-frame Spaces: In order to provide differentiated access of the medium for different frames, IEEE 802.11 defines four Inter-frame Spaces (IFS). Among them, Short IFS (SIFS) is used before the transmission of CTS, Data, or ACK frame while the DCF IFS (DIFS) is applied before RTS's transmission or before back-off. Whenever an erroneous frame is received, the Extend IFS (EIFS) is used. The PCF IFS (PIFS) is not of our concern since we do not consider PCF.

1.4 Research Objectives and Contributions

1.4.1 Research Objectives

As mentioned, due to the unique characteristics of the wireless ad-hoc networks, the design and performance evaluation of the MAC protocol is a very challenging task. On the other hand, since IEEE 802.11 is the de-facto standard for the wireless LAN, it is an appropriate candidate for the wireless ad-hoc networks. In this thesis, our main objective is to evaluate and improve the performance of IEEE 802.11 MAC protocol in wireless ad-hoc networks.

In the performance evaluation part, we use both analytical and simulation techniques. The performance metrics include throughput, fairness, and QoS support (more specifically, packet delay). While fairness forms the basis for the QoS support, the improvement in the fairness is normally at the cost of throughput efficiency. Therefore, it is difficult for a MAC protocol to achieve the optimal throughput, fairness, and QoS support all together. Therefore, in the performance improvement part, we aim to design distributed MAC protocols, which can achieve fairness without unduly degrading the capacity utilization, or vice versa. As mentioned, the design and performance of a MAC protocol greatly depends on the assumptions made regarding the physical layer. We first develop MAC protocols under the simplified assumptions (i.e., $SR/TR = 1$, no capture capability, and fixed modulation rate), and then aim to improve the performance when more practical assumptions (i.e., $SR/TR > 1$, capture capability, and adaptive modulation rate) are adopted.

1.4.2 Main Contributions

Our main contributions in this thesis are as follows:

Analytical model for the single-hop scenario: In this part, we provide a simple analytical model to evaluate the throughput, delay, and fairness of IEEE 802.11 in the single-hop ad-hoc networks. In contrast to the published models, which have focused either on throughput or on delay, our model is unique as it obtains the above three metrics all together. Moreover, while most of the published models adopt the simplified infinite retries assumption, our model has considered the fact that a packet may be dropped after a given number of retries, and thus our model predicts the performance more precisely. Using the proposed model, we carry

out an extensive and insightful performance evaluation of IEEE 802.11 under different system parameters.

Analytical model for the hidden-terminal scenario: In the hidden-terminal scenario (Figure 1.1), IEEE 802.11 is fair on a long-term basis. However, it is very unfair on the short-term basis. In this part, we develop an embedded-Markov model to explain the short-term unfairness due to the hidden-terminal problem. The analytical model is validated through detailed simulation results. The results show that our model is able to precisely predict the short-term unfairness in the hidden-terminal scenario. Our model is unique as there is no published analytical model to explain the short-term unfairness of IEEE 802.11 in the hidden-terminal scenario.

Fairness Schemes (FMAC/CSR and ATC): In this part, we first systematically identify the reasons of unfairness in IEEE 802.11. We also propose a general framework to achieve fairness in IEEE 802.11. Following the framework, we propose two novel fairness schemes: Fair MAC by Cooperation between Sender and Receiver (FMAC/CSR), and Adaptive Transmission Control (ATC). While most of the published schemes are sender-based, meaning that a sender (of a flow) contends for the shared medium based on its own understanding of the contention on the medium, our proposed schemes utilize the information available at the sender as well as at the receiver. Extensive simulation results show that the proposed schemes substantially improve the MAC fairness without unduly degrading the throughput.

Enhanced Carrier Sensing (ECS): In the current IEEE 802.11 standards, whenever a node detects an erroneous frame on the medium, it defers the transmission by a fixed duration (i.e., EIFS). We show that this duration is sometimes smaller and sometimes larger than the desired period by which the transmission should be deferred, and it is the main cause of the unfairness and throughput degradation. We refer to this as *imprecise-EIFS problem*. In order to combat this problem, we propose an enhanced carrier sensing (ECS) mechanism, which intelligently distinguishes among the erroneous frames based on their lengths and defers the transmission accordingly. Simulation results show that the ECS improves the throughput and fairness substantially.

Based on the ECS, we also study the interaction between the MAC and routing mechanisms. From this study, we obtain several important observations.

Full Auto Rate (FAR): The multi-rate capability provided by the various physical layers in IEEE 802.11 requires to adapt the MAC layer transmission rate according to the channel conditions. Recently, several rate-adaptation algorithms have been proposed. However, all the schemes consider the rate adaptation for the Data frames only, while assuming that the control frames are always transmitted at a low basic rate, leading to considerable overhead. To reduce the overhead, we propose a new rate-adaptation algorithm, called Full Auto Rate (FAR), which is able to adapt the transmission rate of all the frames (both control and Data frames). Moreover, in order to cope with the issues involved in the virtual carrier sensing (VCS), we have proposed a modified virtual carrier sensing (MVCS) mechanism. Both analytical and simulation results show that the FAR greatly improves the performance of IEEE 802.11.

1.5 Organization of the Thesis

The remainder of this thesis is organized as follows. In Chapter 2, we present the analytical model for the single-hop ad-hoc network. In Chapter 3, we present the embedded-Markov model for the hidden-terminal scenario. In Chapter 4, FMAC/CSR is presented. ECS and FAR are presented in chapters 5 and 6, respectively. The thesis is concluded in Chapter 7. The work, which can be carried out to extend the work reported in this thesis, is also discussed in this chapter.

In Appendix A, we present the ATC fairness scheme. The probability calculation involved in the ECS is presented in Appendix B. Finally, the Appendix C presents the interaction between the MAC and routing mechanisms.

Chapter 2

Analytical Model for Single-hop Scenario

In this chapter, in the context of single-hop ad-hoc networks, we provide a simple analytical model to evaluate the throughput, delay, and fairness of IEEE 802.11.

2.1 Introduction

In the literature, many analytical models have focused on the derivation of the throughput (or capacity) of IEEE 802.11. For example, the authors in [10] derive the theoretical throughput of IEEE 802.11 by relating IEEE 802.11 to a p-persistent CSMA protocol. The authors in [53] derive the throughput of IEEE 802.11 by using an average analysis technique. However, the above models have oversimplified the calculation of the transmission probability τ (defined later), which is determined by the binary exponential back-off (BEB) algorithm and greatly affects the value of the throughput. In contrast, the author in [7] derives a more precise value of τ by modelling the stochastic process representing the back-off counter as a discrete-time Markov chain. However, it did not consider the frame retry limit used in IEEE 802.11. In light of this, the authors in [58] have extended the model by considering the retry limit.

While most of the published models focus on the throughput of IEEE 802.11, two recent works [11, 54] concentrate on the delay analysis. However, neither of them have considered the retry limit, resulting in a large error in the delay computation as shown in this chapter.

While the above models have focused on either throughput or delay analysis, to the best of our knowledge, there is no analytical model to quantify the unfairness in IEEE 802.11, though many works (e.g., [42, 43]) have focused on achieving fairness in IEEE 802.11. In this

chapter, we introduce an analytical model, which is able to calculate the throughput, delay, and fairness all together. Our model assumes that a station transmits in a randomly chosen slot time with a probability τ , and that the packet being transmitted experiences a collision with a probability p . In fact, τ and p characterize the main features of the binary exponential back-off (BEB) algorithm used in DCF, and they can be obtained using the published models (e.g., [7]). With the values of τ and p , we carry out the analysis for throughput, delay, and fairness. In the throughput analysis, we derive a closed-form solution for the saturation throughput. In addition, we also give a general analysis about the sensitivity of the throughput with respect to the system parameters. Such an analysis is not available in any of the published works. As for the delay analysis, while the published models [11, 54] assume infinite retrials for a given packet, our model considers the fact that a packet handed to the MAC layer may be transmitted successfully or be dropped after a finite number of retrials. In light of this, we derive five kinds of delay and the relationship among these delays. We also derive the error introduced by the infinite retrials assumption. Since in the delay analysis we get a probability mass function of the delay, we are able to carry out a fairness analysis. We show that the coefficient of variation (COV) of the delay is a good metric to measure fairness while the standard deviation is not. We also relate the COV with the well-known Jain's fairness index [29].

Using the proposed model, we carry out an extensive performance evaluation of IEEE 802.11 by varying the system parameters such as the minimum and maximum contention window, and the retry limit. Our results show that the throughput, delay, and fairness are quite sensitive to the system parameters being chosen, demonstrating the need of dynamic tuning of the system parameters in IEEE 802.11. This is contrary to the conclusion made in [7]. Moreover, for the optimum performance of different metrics, the values for the system parameters are inconsistent, implying that a tradeoff among the metrics must be made in the dynamic tuning. Finally, our results also show that the published models have largely overestimated the throughput or the packet delay, and thus lead to misleading conclusions.

The remainder of this chapter is organized as follows. The analytical model is presented in Section 2.2. Based on the model, a thorough performance evaluation is carried out in Section 2.3. The chapter is concluded in Section 2.4.

2.2 Analytical Model

For simplicity of the analysis, several popular assumptions have been made: (1) there are a finite number of stations (say n) in the network, and all the stations can hear each other (i.e., single-hop network); (2) stations are always backlogged, meaning that a station always has packets waiting to be transmitted; (3) ideal channel conditions (i.e., no transmission error or capture); (4) RTS/CTS handshake is used; and (5) the data packets have a fixed length.

2.2.1 Modelling Binary Exponential Back-off

In [7], by modelling the stochastic process representing the back-off counter at a given station as a discrete-time Markov chain, the values of two parameters that characterize the binary exponential back-off (BEB) algorithm are obtained. The first parameter is the *transmission probability*, τ , with which a station transmits in a randomly chosen slot time. The second parameter is the *conditional collision probability*, p , representing the probability of a collision experienced by a packet given that it is transmitted on the channel.

In the calculation of τ , the author in [7] did not consider the frame retry limit used in IEEE 802.11. In light of this, the authors in [58] have extended the model by considering the retry limit. According to [58], τ is expressed as follows,

$$\tau = \begin{cases} \frac{2(1-2p)(1-p^{m+1})}{W(1-(2p)^{m+1})(1-p)+(1-2p)(1-p^{m+1})} & m \leq m' \\ \frac{2(1-2p)(1-p^{m+1})}{W(1-(2p)^{m'+1})(1-p)+(1-2p)(1-p^{m+1})+W2^{m'}p^{m'+1}(1-2p)(1-p^{m-m'})} & m \geq m' \end{cases} \quad (2.1)$$

where $W = CW_{min} + 1$. As mentioned in Chapter 1, m is the retry limit¹ while m' is the stage number that the CW reaches its maximum value. Normally, $m \geq m'$.

Moreover, the relationship between τ and p is as follows [7],

$$p = 1 - (1 - \tau)^{n-1}. \quad (2.2)$$

Equations (2.1) and (2.2) represent a nonlinear system with two unknowns τ and p , which can be solved using numerical techniques. With these two parameters, many performance

¹In fact, there are two kinds of retry limit in IEEE 80211, i.e., long retry limit and short retry limit. Here, we refer to the long retry limit.

metrics can be obtained. In the following subsections, we discuss how the throughput, delay, and fairness can be obtained. Note that the discussion below can also be made based on any other model (e.g., [10]) as long as τ and p can be obtained from the model.

2.2.2 Throughput Analysis

Now, we calculate the normalized system throughput S , defined as the fraction of time that the channel is used to successfully transmit payload bits. The description in this subsection mainly follows from [7], except that we provide clearer insights in the model.

To compute S , we need to analyze what happens on the shared medium in a randomly chosen slot time. The slot may correspond to an idle slot, a collision, or a successful transmission. Let P_{idle} , P_{col} , and P_{succ} be the probabilities that a randomly chosen slot corresponds to an idle slot, a collision, or a successful transmission, respectively.

It is easy to see that P_{idle} is equal to the probability that *none* of the n stations transmits on the shared medium in a randomly chosen slot time. Therefore,

$$P_{idle} = (1 - \tau)^n. \quad (2.3)$$

Clearly, if there is a successful transmission in a randomly chosen slot time, it means that *exactly one* station transmits on the shared medium in the slot. Therefore, P_{succ} can be expressed as,

$$P_{succ} = n\tau(1 - \tau)^{n-1}. \quad (2.4)$$

Obviously, whenever more than one stations transmit on the shared medium in a randomly chosen slot time, a collision occurs. Therefore, P_{col} is,

$$P_{col} = 1 - (1 - \tau)^n - n\tau(1 - \tau)^{n-1}. \quad (2.5)$$

Now, let σ , T_{col} , and T_{succ} be the duration of the slot corresponding to an idle slot, a collision, or a successful transmission, respectively². Therefore, the average duration (represented by T_{avg}) that a randomly chosen slot lasts is,

²Note that the idle slot corresponds to the slot defined in IEEE 802.11 standards and its value is fixed, while the duration for the other two kinds of slots is variable.

$$T_{avg} = P_{idle}\sigma + P_{succ}T_{succ} + P_{col}T_{col}. \quad (2.6)$$

Now, the throughput S can be calculated as,

$$\begin{aligned} S &= \frac{E[\text{payload information transmitted in a slot time}]}{E[\text{length of a slot time}]} \\ &= \frac{P_{succ} \times E[P]}{T_{avg}} = \frac{P_{succ} \times E[P]}{P_{idle}\sigma + P_{succ}T_{succ} + P_{col}T_{col}} \end{aligned} \quad (2.7)$$

where $E[P]$ is the average payload size (in terms of time units, e.g., μs), and thus $P_{succ} \times E[P]$ is the average amount of payload information successfully transmitted in a randomly chosen slot time. Note that $E[P]$ is part of T_{succ} as the latter one includes the overhead such as RTS, CTS and ACK.

By dividing the numerator and denominator of Equation (2.7) by P_{succ} , the throughput can be expressed as follows,

$$S = \frac{E[P]}{T_{succ} + \frac{P_{col}}{P_{succ}}T_{col} + \frac{P_{idle}}{P_{succ}}\sigma}. \quad (2.8)$$

The above formula is quite intuitive. The denominator represents the average amount of time spent in order to observe a successful packet transmission on the channel. Specifically, we can view the packet transmissions on the channel as the *bernoulli trials* with a success probability P_{succ} . Therefore, the number of slots (including the slot for the successful transmission) required to observe a successful transmission is given by the geometric distribution. Clearly, the average number of slots required is simply $1/P_{succ}$. Out of these slots, the average numbers of idle and collision slots are P_{idle}/P_{succ} and P_{col}/P_{succ} , respectively. In addition, there is one slot for the successful transmission. Moreover, as mentioned before, the duration of the idle, collision, and successful transmission slots are σ , T_{col} , and T_{succ} , respectively. This explains Equation (2.8).

The throughput S can also be expressed as follows,

$$S = \frac{E[P]/T_{succ}}{1 + \frac{P_{col}}{P_{succ}} \times \frac{T_{col}}{T_{succ}} + \frac{P_{idle}}{P_{succ}} \times \frac{\sigma}{T_{succ}}}. \quad (2.9)$$

The sensitivity of the throughput to the system parameters (e.g., CW_{min} , CW_{max} , and m) can be easily analyzed from the above formula. Clearly, $E[P]$, T_{succ} , T_{col} , and σ are independent to the system parameters. On the other hand, as known from equations (2.3), (2.4) and (2.5), the values of P_{col} , P_{succ} , and P_{idle} depend on τ and p , which themselves depend on the system parameters (see equations (2.1) and (2.2)). Therefore, we only need to discuss the sensitivity of the throughput with respect to P_{col} , P_{succ} , and P_{idle} . From Equation (2.9), we know that the throughput S is sensitive to the value of P_{col}/P_{succ} only when the constant T_{col}/T_{succ} is large (compared to the first term of the denominator, i.e., unity). Similarly, S is sensitive to P_{idle}/P_{succ} only when the constant σ/T_{succ} is large. The sensitivity analysis gives an idea whether dynamic tuning of the system parameters is necessary or not. For example, if S is insensitive to the system parameters, it may not be necessary to do any dynamic tuning of the parameters.

As mentioned in [7], the above analysis applies to both the two-way and four-way handshakes. To specifically compute the throughput for a given handshake, we only need to specify the corresponding values of T_{col} and T_{succ} . Note that the idle slot time σ is specific to the physical layer, e.g., $20 \mu s$ for Direct Sequence Spread Spectrum (DSSS) [24].

The specific values of T_{col} and T_{succ} for the four-way handshake are as follows,

$$\begin{cases} T_{succ} &= RTS + CTS + Data + ACK + 3SIFS + DIFS \\ T_{col} &= RTS + DIFS + SIFS + CTS \end{cases} \quad (2.10)$$

It may be noticed that the above expression is different from that in [7]. First, the propagation delay δ is no longer included in the above formula since the Short Inter-Frame Space (SIFS) has already contained δ according to the standards [24]. Another difference is that in the expression of T_{col} , we have added an additional term, $SIFS + CTS$. As stated in [7], T_{col} is the period of time during which the channel is sensed busy by the *non-colliding* stations. After a collision, the *colliding*³ stations wait for a time equal to $CTSTimeout$ (or $ACKTimeout$ in the two-way handshake case). On the other hand, for the non-colliding stations, since they cannot interpret the contents of the colliding frames, they defer by an Extended Inter-Frame

³For example, when there is a collision between nodes A and B, and node C hears the collision, then node C is here called a non-colliding node while A and B are the colliding nodes.

Space (EIFS) value. Moreover, the EIFS value is equal to $DIFS + SIFS + TxTime(ACK)$, which is large enough for the complete transmission of the CTS or ACK frame (note that CTS has the same length as ACK). Therefore, if $CTSTimeout$ or $ACKTimeout$ is set to the time needed for the transmission of the CTS/ACK frame (which is true in the popularly adopted NS-2 simulator [15]), then, T_{col} for all the stations, whether involved in the collision or not, will be the same. Therefore, the $SIFS + CTS$ term should be included in T_{col} . Clearly, the above modifications are important to obtain a more precise value of the throughput. The model in [7] overestimates the throughput as it uses a smaller T_{col} .

2.2.3 Delay Analysis

In IEEE 802.11, there is no queue at the MAC layer itself, and thus IEEE 802.11 standards have not specified any queuing mechanisms. However, normally there should be a queue at the top of the MAC layer. Therefore, in general, the delay that a packet experiences should include two parts, i.e. the delay experienced in the queue and the delay experienced at the MAC layer. We only focus on the MAC layer delay. Since there are multiple stations contending for the shared medium and IEEE 802.11 DCF is a random access protocol, the MAC layer delay is a random value, which requires a detailed analysis. Five types of delay at the MAC layer are relevant to this analysis, which we will discuss in a little while.

As we know, in IEEE 802.11, if a packet is handed to the MAC layer from the upper layer, the packet may be transmitted successfully in one or several trials, or the packet will be dropped if the retry limit is reached. In both cases, the MAC layer will notify the upper layer about the final status of the packet (i.e., transmitted successfully or dropped). Accordingly, we can define the following four kinds of delays.

Delay D_{succ} : if a packet is successfully transmitted, what is the delay *from the instant* that the packet is handed to the MAC layer *to the instant* that the MAC layer successfully transmits the packet (and thus the upper layer gets a notification accordingly)?

Delay D_{drop} : if a packet is dropped, what is the delay *from the instant* that the packet is handed to the MAC layer *to the instant* that the MAC layer drops the packet after reaching the retry limit (and thus the upper layer gets a notification accordingly)?

Delay D_{notify} : combining the above two cases, what is the delay *from the instant* that the packet is handed to the MAC layer *to the instant* that the upper layer gets a notification from the MAC layer about the final status of the packet (i.e., transmitted successfully or dropped)?

Delay $D_{intersucc}$: from the viewpoint of an upper layer at a given station, what is the delay between two successful packet transmissions?

Note that in the definition of $D_{intersucc}$, the successful transmissions must belong to the given station only. Also, as shown later, $D_{intersucc}$ is directly related to the throughput that a single station gets, from which we can obtain the total system throughput S that has been derived in the previous subsection.

At last, to show the error introduced by the infinite retries assumption [11, 54], we are also interested in the following delay:

Delay $D_{infinite}$: if we assume that a packet is always retransmitted until it is transmitted successfully (i.e., infinite retries), what is the delay *from the instant* that the packet is handed to the MAC layer *to the instant* that the MAC layer successfully transmits the packet (and thus the upper layer gets a notification accordingly)?

In the following subsections, we discuss how to compute the above delays. In the calculations, we assume that the packet handed to the MAC layer must experience the back-off process, though the back-off process is unnecessary if the medium is idle at the instant the packet is handed to the MAC layer. However, under the saturation condition, the probability of such an event should be very small and thus we can safely ignore it.

2.2.3.1 Derivation of D_{succ}

Clearly, D_{succ} depends upon the number of back-off stages that a packet experiences, as well as upon the back-off counter generated at each stage. Figure 2.1 gives an example of the back-off process experienced by a packet at a given station. In the figure, the shaded blocks represent the events in which the given node is involved. In the example, the packet experiences three stages (from stage 0 to stage 2) before it is transmitted successfully, and the back-off counter (represented by B_0 , B_1 , and B_2 , respectively) generated at the stages are 3, 4, and 2, respectively.

Figure 2.1: Illustration of Delay for Successful Packet

From the figure, it is easy to find that the back-off counter at the given station decrements by one after every slot time if the slot corresponds to: (1) an idle slot, (2) a collision involving *other* stations, or (3) a successful transmission involving *other* stations⁴. Since we know that the duration of the idle, collision, and successful transmission slots are σ , T_{col} , and T_{succ} , respectively, we can easily get the delay that is contributed by the idle slots, collisions involving other stations, and successful transmissions of other stations. For example, in the above example, during the back-off process for the given packet, the number of idle slots is 4, while the numbers of collisions and successful transmissions among other stations are 3 and 2, respectively. Therefore, the delay contributed by the back-off slots is $4\sigma + 3T_{col} + 2T_{succ}$. In addition to this, the delay D_{succ} should include the time spent on the collisions that the given station *itself* is involved, e.g., $2 \times T_{col}$ in the above example. Finally, the D_{succ} should also include the time needed for the successful transmission of the given packet, i.e., T_{succ} . Therefore, the delay D_{succ} is $(4\sigma + 3T_{col} + 2T_{succ}) + 2T_{col} + T_{succ}$. In the following, we provide a formal method to calculate D_{succ} for a general case.

Let B_i be the random back-off counter generated at stage i , and let $B(j)$ be the *sum* of the back-off slots generated at stages from 0 to j . Therefore,

$$B(j) = \sum_{i=0}^j B_i. \quad (2.11)$$

We know that B_i at stage i follows the uniform distribution in the range $[0, CW_i]$, where CW_i is expressed by Equation (1.2). It is easy to get the probability mass function (pmf) of $B(j)$, which is simply a *convolution* of the pmfs of all B_i where $i \in [0, j]$. Clearly, $B(j)$ is in the range of $[0, B(j)_{max}]$, where $B(j)_{max} = \sum_{i=0}^j CW_i$.

⁴In fact, according to IEEE 802.11, corresponding to a collision or a successful transmission, the counters at the stations do not decrement. However, the imprecision caused by such an approximation is insignificant since T_{col} and T_{succ} are much greater than the idle slot time (i.e., σ).

Since the probability that a transmission experiences a collision is p (calculated in Section 2.2.1), the probability that a packet is successfully transmitted at stage j (starting from stage 0) is,

$$P_{succ}(j) = p^j \times (1 - p) \quad 0 \leq j \leq m. \quad (2.12)$$

Moreover, the probability that a packet is dropped because the retry limit (i.e., m) is reached is,

$$P_{drop} = 1 - \sum_{j=0}^m p^j \times (1 - p) = p^{m+1}. \quad (2.13)$$

Let $P_{succ}(j, b)$ be the probability that the packet is successfully transmitted at stage j ($0 \leq j \leq m$), and the sum of the back-off slots generated up to stage j is equal to b . Therefore, $P_{succ}(j, b)$ can be expressed as,

$$P_{succ}(j, b) = P_{succ}(j) \times Pr(B(j) = b) \quad 0 \leq j \leq m. \quad (2.14)$$

Similarly, let $P_{drop}(b)$ be the probability that a packet is dropped, and the sum of the back-off slots generated up to stage m is equal to b . Therefore, $P_{drop}(b)$ can be expressed as,

$$P_{drop}(b) = P_{drop} \times Pr(B(m) = b). \quad (2.15)$$

Since D_{succ} represents the delay of a packet that is successfully transmitted, we should only consider the packets that are successfully transmitted while excluding the packets that are dropped. Therefore, we define the following conditional probability, $P'_{succ}(j, b)$, which represents the probability that the back-off process for a packet ends at stage j and the sum of the back-off slots generated up to stage j is b , given that the packet is successfully transmitted. Therefore,

$$P'_{succ}(j, b) = \frac{P_{succ}(j, b)}{1 - P_{drop}}. \quad (2.16)$$

As a generalization of the example given in Figure 2.1, the corresponding delay, $D_{succ}(j, b)$, which represents the delay of a packet that is successfully transmitted at stage j and the sum of the back-off slots generated up to stage j is b , is as follows,

$$D_{succ}(j, b) = b \times T_{avg} + j \times T_{col} + T_{succ}. \quad (2.17)$$

Note that there is an approximation in the above equation, that is, we have simply used $b \times T_{avg}$ to represent the duration of the b number of back-off slots. In fact, if, among the b number of back-off slots, there are b_{col} number of slots corresponding to collisions, b_{succ} number of slots corresponding to successful transmissions, and b_{idle} number of idle slots, the delay contributed by the b number of back-off slots is $(b_{col}T_{col} + b_{succ}T_{succ} + b_{idle}\sigma)$. The random variables b_{col} , b_{succ} , b_{idle} , and b can be characterized by a multinomial probability distribution. However, it makes the analysis extremely complex. Moreover, the error introduced by the above approximation is very small if b is relatively large, which is normally true in IEEE 802.11. For example, when the minimum contention window is 32, the average value b should be 16 even if all the packets experience only one stage (i.e., stage 0). Therefore, the above approximation is reasonable and will be used throughout this chapter.

Equations (2.16) and (2.17) express the probability mass function of the delay D_{succ} in terms of the stage number (i.e., j) and the sum of the number of back-off slots (i.e., b). Therefore, the average of D_{succ} is,

$$\overline{D_{succ}} = \sum_{j=0}^m \sum_{b=0}^{B(j)_{max}} D_{succ}(j, b) \times P'_{succ}(j, b). \quad (2.18)$$

After a few simple steps, we get the following,

$$\overline{D_{succ}} = T_{succ} + \frac{1}{1 - P_{drop}} \sum_{j=0}^m [T_{avg} \times \overline{B(j)} + j \times T_{col}] P_{succ}(j) \quad (2.19)$$

where $\overline{B(j)}$ is the average of $B(j)$, and can be expressed as,

$$\overline{B(j)} = \frac{B(j)_{max}}{2} = \frac{\sum_{i=0}^j CW_i}{2}. \quad (2.20)$$

From Equation (2.19), we can easily calculate the value of $\overline{D_{succ}}$ under any given set of system parameters.

With the probability mass function of the delay expressed by equations (2.16) and (2.17), we can compute other important metrics, such as the standard deviation of the delay.

2.2.3.2 Derivation of D_{drop}

Similar to the derivation of D_{succ} , we present an example in Figure 2.2 for the case that a packet will be dropped. As shown in the figure, in contrast to the case of a successful transmission, if a packet is to be dropped, the back-off process experienced by the packet must reach the retry limit (i.e., stage m).

Figure 2.2: Illustration of Delay for Dropped Packet

To compute D_{drop} , which represents the delay of a packet that is dropped, we define the following conditional probability as done in the derivation of D_{succ} ,

$$P'_{drop}(b) = \frac{P_{drop}(b)}{P_{drop}} = Pr(B(m) = b) \quad (2.21)$$

where $P_{drop}(b)$, P_{drop} , and $B(m)$ have been defined in equations (2.15), (2.13), and (2.11), respectively.

The corresponding delay, using the same approximation as in equation (2.17), can be expressed as follows,

$$D_{drop}(b) = b \times T_{avg} + m \times T_{col} + T_{col}. \quad (2.22)$$

The above two equations express the probability mass function of the delay D_{drop} . The average of D_{drop} is,

$$\overline{D_{drop}} = \sum_{b=0}^{B(m)_{max}} D_{drop}(b) \times P'_{drop}(b). \quad (2.23)$$

After some simple steps, we get the following expression,

$$\overline{D_{drop}} = T_{col} \times (m + 1) + T_{avg} \times \overline{B(m)} \quad (2.24)$$

where $\overline{B(m)}$ is the average value of $B(m)$, and is equal to $B(m)_{max}/2$.

The variance of the delay can be obtained as follows (note that all B_i are independent to each other),

$$\begin{aligned} Var(D_{drop}) &= T_{avg}^2 \times Var(B(m)) = T_{avg}^2 \times Var\left(\sum_{i=0}^m B_i\right) \\ &= T_{avg}^2 \times \sum_{i=0}^m Var(B_i) \end{aligned} \quad (2.25)$$

where $Var(B_i) = (CW_i^2 + 2CW_i)/12$.

2.2.3.3 Derivation of D_{notify}

The probability mass function of the delay D_{notify} is expressed by equations (2.14), (2.17), (2.15) and (2.22). Note that we should not use the conditional probabilities expressed in equations (2.16) and (2.21) since D_{notify} represents the delay for a general packet (that is either dropped or successfully transmitted). The average of D_{notify} can be obtained as follows,

$$\begin{aligned} \overline{D_{notify}} &= \sum_{j=0}^m \sum_{b=0}^{B(j)_{max}} [D_{succ}(j, b) \times P_{succ}(j, b)] \\ &\quad + \sum_{b=0}^{B(m)_{max}} [D_{drop}(b) \times P_{drop}(b)]. \end{aligned} \quad (2.26)$$

After some simple steps, we get the following expression,

$$\overline{D_{notify}} = (1 - P_{drop})\overline{D_{succ}} + P_{drop}\overline{D_{drop}}. \quad (2.27)$$

The above result is quite intuitive. The first term expresses the fact that the packet is successfully transmitted with a probability $(1 - P_{drop})$ and the corresponding average delay is $\overline{D_{succ}}$, while the second term expresses the fact that the packet is dropped with a probability P_{drop} and the corresponding average delay is $\overline{D_{drop}}$. We can also compute the standard deviation of D_{notify} .

2.2.3.4 Derivation of $D_{intersucc}$

In contrast to D_{succ} , D_{drop} , and D_{notify} discussed above, we are not able to get the probability mass function of the delay $D_{intersucc}$. However, we can get the average value of $D_{intersucc}$.

Specifically, we can view the packet transmissions at a given station as a *bernoulli experiment* with the probability of failure P_{drop} , and the probability of success $(1 - P_{drop})$. Therefore, to observe a successful transmission at the given station, the number of trials required follows a geometric distribution. Clearly, the average number of trials required is simply $\frac{1}{1-P_{drop}}$. Among these trials, a fraction of P_{drop} trials fail and the packet is dropped, and each of the failing trial lasts $\overline{D_{drop}}$ duration. Obviously, among the trials the last trial must be successful, and it lasts $\overline{D_{succ}}$ duration. Therefore, the average delay $\overline{D_{intersucc}}$ can be obtained as,

$$\overline{D_{intersucc}} = \frac{P_{drop}}{1 - P_{drop}} \overline{D_{drop}} + \frac{1 - P_{drop}}{1 - P_{drop}} \overline{D_{succ}} = \frac{\overline{D_{notify}}}{1 - P_{drop}}. \quad (2.28)$$

The last expression is quite intuitive. To observe a successful transmission at a given station, on an average, the station gets $1/(1 - P_{drop})$ notifications and the average duration between two consecutive notifications is $\overline{D_{notify}}$.

Interestingly, using the above result, we can also obtain the throughput at any single station, which is simply equal to $E(P)/\overline{D_{intersucc}}$. Observing that every station gets the same throughput in a long term, if there are n number of stations in the network, then the total system throughput S can be obtained as follows,

$$S = \frac{E(P)}{\overline{D_{intersucc}}} \times n = \frac{E(P)}{\overline{D_{succ}} + \frac{P_{drop}}{1-P_{drop}} \overline{D_{drop}}} \times n. \quad (2.29)$$

The above equation relates the delay, throughput, and packet loss probability, and thus it may be of great importance in developing dynamic control algorithms that aim to achieve a tradeoff among the delay, throughput, and packet loss probability.

2.2.3.5 Derivation of $D_{infinite}$

Now, we derive the delay under the infinite retries assumption that is adopted in [11, 54]. Let $P_{infinite}(j, b)$ be the probability that a packet is successfully transmitted at stage j ($0 \leq j < \infty$), and the sum of the back-off slots generated up to stage j is equal to b . Therefore,

$$P_{infinite}(j, b) = P_{succ}(j) \times Pr(B(j) = b) \quad j \in [0, \infty) \quad (2.30)$$

where $P_{succ}(j)$ is defined in Equation (2.12) except that the range of j is changed to $[0, \infty)$, and $B(j)$ is defined in Equation (2.11). Clearly, the corresponding delay is as follows,

$$D_{infinite}(j, b) = b \times T_{avg} + j \times T_{col} + T_{succ} \quad j \in [0, \infty). \quad (2.31)$$

The above two formulas form the probability mass function of the delay $D_{infinite}$. Therefore, the average of $D_{infinite}$ is,

$$\begin{aligned} \overline{D_{infinite}} &= \sum_{j=0}^{\infty} \sum_{b=0}^{B(j)_{max}} [D_{infinite}(j, b) \times P_{infinite}(j, b)] \\ &= \sum_{j=0}^m \sum_{b=0}^{B(j)_{max}} [D_{infinite}(j, b) \times P_{infinite}(j, b)] \\ &\quad + \sum_{j=m+1}^{\infty} \sum_{b=0}^{B(j)_{max}} [D_{infinite}(j, b) \times P_{infinite}(j, b)]. \end{aligned} \quad (2.32)$$

After several simple steps, we can find that the first term is simply equal to $(1 - P_{drop})\overline{D_{succ}}$. This is very intuitive and can be explained as follows. Under the infinite retrials assumption, the probability that a packet is successfully transmitted at a stage later than stage m is equal to the probability that the packet is dropped if the retry limit is m . Therefore, the probability that a packet is transmitted successfully at one of the earliest m stages is $(1 - P_{drop})$, and the average delay of such a packet is $\overline{D_{succ}}$. This explains the first term. After some simple steps, the second term (represented by T_2) of Equation (2.32), can be expressed as follows,

$$T_2 = P_{drop} [T_{succ} + (m + \frac{1}{1-p})T_{col} + (\frac{B(m)_{max}}{2} + \frac{CW_{max}}{2} \frac{1}{1-p})T_{avg}]. \quad (2.33)$$

The above term is also quite intuitive. If a packet experiences more than m stages (with a probability P_{drop}), it will first experience m collisions. Then, the average number of additional collisions experienced is simply $1/(1-p)$ due to the *memoryless* property of the geometric distribution. The above arguments explain the term $(m + \frac{1}{1-p})T_{col}$ in Equation (2.33). Now, we explain the term $(\frac{B(m)_{max}}{2} + \frac{CW_{max}}{2} \frac{1}{1-p})T_{avg}$. Specifically, $B(m)_{max}/2$ represents the average number of back-off slots in the first m stages. Then, at every additional stage, on an average $CW_{max}/2$ slots are involved, and the average number of additional stages is simply $1/(1-p)$. In

the above discussion, we have assumed that once the CW reaches its maximum value (CW_{max}), it does not increase any more in the remaining stages. From equations (2.32) and (2.33), we obtain,

$$\overline{D_{infinite}} = (1 - P_{drop})\overline{D_{succ}} + T_2. \quad (2.34)$$

We can also express the average delay as follows,

$$\overline{D_{infinite}} = (1 - P_{drop})\overline{D_{succ}} + P_{drop}\overline{D_{drop}} + T_3 = \overline{D_{notify}} + T_3 \quad (2.35)$$

where

$$T_3 = P_{drop}[T_{succ} + \frac{p}{1-p}T_{col} + \frac{CW_{max}}{2} \frac{1}{1-p}T_{avg}]. \quad (2.36)$$

We notice that $\overline{D_{infinite}}$ is always larger than $\overline{D_{notify}}$ as well as $\overline{D_{succ}}$ (since $\overline{D_{notify}}$ is always larger than $\overline{D_{succ}}$). Therefore, in general $\overline{D_{infinite}}$ overestimates the delay. In fact, as shown later by the numerical results, the results of $\overline{D_{infinite}}$ is very misleading whenever p (or P_{drop}) is large.

From equations (2.30) and (2.31), we can also numerically obtain the standard deviation of $D_{infinite}$.

2.2.4 Fairness Analysis

IEEE 802.11 is fair in the *long-term* (e.g., equal average delay or throughput) in a *single-hop* network⁵. However, due to the randomness involved in the binary exponential back-off (BEB) algorithm, substantial *short-term* unfairness remains even in a single-hop scenario. In this subsection, we aim to measure the short-term unfairness in the single-hop scenario.

In general, unfairness can be measured in terms of the throughput or the packet delay. To measure the short-term unfairness through the packet delay, an ideal way is to first get the probability distribution of the delay experienced at all the stations, and then see how variable the delay is. Clearly, higher variability implies a more unfair system. However, to get such a

⁵Note that in a *multi-hop* scenario, IEEE 802.11 cannot maintain the long-term fairness among the contending stations as will be shown in Chapter 4.

distribution, we need to model the system state in a way that there is at least one separate parameter used to represent the state of each station, making the state space very large if the number of stations is large. On the other hand, if we assume that the probability distribution of the delay experienced at a given station is similar to the distribution of the delay experienced at all the stations, the problem becomes much easier as we only need to consider the states at a typical station. We adopt this approach as it is much simpler but still able to reveal the general trend of the unfairness, though it may not give a very precise measurement of the unfairness.

In the previous subsection we have already obtained the probability distribution of the delay (e.g., D_{succ}) experienced at a typical station. Therefore, we now only need to develop a metric to quantify the variability of the delay and thus reflect the unfairness. The standard deviation is a good candidate, however, it is not independent of scale (i.e., the unit of measurement) [29] as shown by the following example. Considering there are two different schemes to distribute a common resource to three users, the delays experienced by the users under the two schemes are as follows:

Scheme I = {1, 2, 3}

Scheme II = {2, 4, 6}

Clearly, the two schemes deliver the same fairness as far as the delay is concerned. However, the standard deviation of the delay under the two schemes are $\sqrt{2}$ and $2\sqrt{2}$, respectively. Therefore, if we use the standard deviation as a metric to reflect unfairness, we will arrive at a misleading conclusion: the second scheme is two times more unfair than the first one. It is clear that the standard deviation is not a good metric to reflect the fairness as it is not scale independent. Now, we look at the coefficient of variation (COV) of the delay, which can also reflect the variability of the delay. The COV is defined as,

$$COV = \frac{\text{Standard Deviation}}{\text{Average}}. \quad (2.37)$$

Clearly, the COV under the above two schemes are the same, i.e., $\frac{\sqrt{2}}{2}$, implying that they deliver the same fairness. In fact, as mentioned in [29], COV in many senses is a good metric to reflect fairness. In general, the larger the COV is, the broader the delay spreads, and thus the

more unfair the scheme is. Numerically, we can easily find the value of the COV of the delay from the results in the previous subsection.

Since the Jain's index [29] is a more popular metric for the unfairness, now we discuss how to relate the COV to the Jain's index. If there are n number of users sharing the common resource, and user i gets a proportion of x_i , the Jain's index is defined as follows,

$$\text{Jain's Index} = \frac{[\sum_i x_i]^2}{n \sum_i x_i^2}. \quad (2.38)$$

On the other hand, the square of the COV is as follows,

$$COV^2 = \frac{\text{Variance}}{\text{Average}^2} = \frac{\overline{x^2} - (\bar{x})^2}{(\bar{x})^2}. \quad (2.39)$$

Now, we can get the relationship between the COV and the Jain' index as follows,

$$\begin{aligned} \text{Jain's Index} &= \frac{[\sum_i x_i]^2}{n \sum_i x_i^2} = \frac{[\frac{1}{n} \sum_i x_i]^2}{\frac{1}{n} \sum_i x_i^2} = \frac{(\bar{x})^2}{\overline{x^2}} = \frac{(\bar{x})^2}{(\bar{x})^2 + \overline{x^2} - (\bar{x})^2} \\ &= \frac{1}{1 + \frac{\overline{x^2} - (\bar{x})^2}{(\bar{x})^2}} = \frac{1}{1 + COV^2}. \end{aligned} \quad (2.40)$$

Clearly, while the COV is boundless, the Jain's index is in the range of $[0, 1]$. However, note that the Jain's index obtained from the COV in our case is not able to tell exactly how unfair the system is since we obtain it through the delay distribution at a typical station rather than through a distribution of the delay experienced at all the contending stations.

2.3 Performance Evaluation of IEEE 802.11

In this section, we present the performance results for the four-way handshake in IEEE 802.11 DCF under the Direct Sequence Spread Spectrum (DSSS) physical layer [24]. Table 2.1 lists the values of the parameters used in the calculation. Rather than using the same transmission rate (i.e., 1 Mbps) for both the Data and control frames as in [7], to closely follow the standards, in our calculation the transmission rate for the Data frame is 2 Mbps while that for the control frames is 1 Mbps. Using the parameters listed in the above table, we can calculate $E(P)$, T_{succ} , and T_{col} , which are 4096 μs , 5440 μs , and 716 μs , respectively.

Table 2.1: IEEE 802.11 Parameters used in the Calculation

Payload of data packet	1024 bytes
Data	1024 bytes + MAC header + PHY header
RTS	20 bytes + PHY header
CTS	14 bytes + PHY header
ACK	14 bytes + PHY header
PHY Header	192 us
MAC Header	28 bytes
Basic rate	1 Mbps
Data rate	2 Mbps
Slot time	20 us
SIFS	10 us
DIFS	50 us
EIFS	364 us

As shown in the previous section, the performance depends on four parameters: n , CW_{min} , CW_{max} , and m . In all the following performance graphs, the horizontal axis represents the number of stations in the network (i.e., n). The vertical axis represents one of the four performance metrics (throughput, packet drop probability, delay, or fairness) while in a graph one of the three system parameters (i.e., CW_{min} , CW_{max} , or m) is varied and the remaining two parameters are fixed at their standard values. The standard values [24] are as follows: $CW_{min} = 31$, $CW_{max} = 1023$, and $m = 6$. The results are validated through simulation under NS-2. In particular, for the results presented in each graph, we have done the simulation for several selected set of parameters to check whether the general trends presented in the paper correctly reflect those obtained from the simulation.

2.3.1 Throughput Results

Figures 2.3-2.5 present the throughput results. From the figures, we can make three main observations: (i) Compared to the throughput presented in [7], the throughput obtained here is much smaller; (ii) In general, the larger the CW_{min} , CW_{max} , and m are, the larger the throughput is. On the other hand, when the network size (i.e., n) increases, the throughput normally decreases; (iii) The throughput is quite sensitive to the system parameters (i.e., CW_{min} , CW_{max} , and m) as well as to the network size, which is contrary to the observation made in [7].

There are two reasons behind the first observation. Firstly, since the model in [7] has not considered the retry limit (i.e., it assumes infinite retrials), it overestimates the throughput as

mentioned in [58]. In fact, this can also be easily verified from Figure 2.5, which shows that the throughput increases with the increase of the retry limit m . The second reason is as follows. As mentioned in the discussion for Equation (2.10), T_{col} in our calculation is larger than the one in [7]. On the other hand, T_{succ} is smaller than that in [7] since we use a higher transmission rate for the Data frame. As a result, in our calculation, the time spent on collisions increases while the time spent on the payload transmission decreases. The above arguments explain why the throughput obtained here is smaller than that in [7].

To explain the second observation (i.e., how the throughput changes with the system parameters), we first need to look at the values of T_{col}/T_{succ} and σ/T_{succ} as discussed in the context of Equation (2.9). In our calculation, σ/T_{succ} is very small (i.e., 20/5440) compared to the unity, and thus the throughput S is insensitive to the value of P_{idle}/P_{succ} . On the contrary, T_{col}/T_{succ} is relatively large (i.e., 716/5440), and thus S is sensitive to the value of P_{col}/P_{succ} . Therefore, to explain the second observation, we only need to look at the value of P_{col}/P_{succ} under different system parameters. Clearly, the larger P_{col}/P_{succ} is, the smaller the throughput is. To exemplify this, Figure 2.6 presents the value of P_{col}/P_{succ} when the CW_{min} is varied. We notice that the value increases as the CW_{min} decreases or as n increases, explaining why the throughput in Figure 2.3 decreases in these two cases.

Now, let us explain the third observation. As mentioned, T_{col}/T_{succ} is quite large in our case, and thus the throughput S is very sensitive to the system parameters. In contrast, the T_{col}/T_{succ} is very small (i.e., 8/191.36) in [7], and thus the throughput S presented there is insensitive to the system parameters.

Here, we would like to point out that in the emerging IEEE 802.11 standards (e.g., IEEE 802.11-b, -a and -g), the T_{col}/T_{succ} is becoming larger because the transmission rate of the Data frames (which determines T_{succ}) is increasing (e.g., can be as high as 54 Mbps) while the transmission rate of the control frames (which determines T_{col}) remains small. Therefore, we expect the throughput in the emerging standards to be more sensitive to the system parameters, and thus the dynamic tuning of the system parameters will be of great importance in these standards.

2.3.2 Packet Drop Probability Results

Figures 2.7-2.9 present the packet drop probability. From the figures, we find that, the drop probability increases when CW_{min} , CW_{max} , or m decrease. This is due to the fact that when CW_{min} , CW_{max} , or m decrease, the conditional collision probability p increases. We also notice that the drop probability is quite large (e.g., larger than 0.5) when the network size is large. This observation will be useful when we explain the large difference between the delay D_{notify} and $D_{infinite}$ later.

2.3.3 Delay Results

In figures 2.10-2.14, we first present the five kinds of delay by varying the CW_{min} . From the results, we find that: (i) In general, in the first three kinds of delay (i.e., D_{succ} , D_{drop} , and D_{notify}), the larger the CW_{min} is, the larger is the delay; (ii) in contrast, $D_{intersucc}$ and $D_{infinite}$ decrease when CW_{min} increases; (iii) D_{drop} is much higher than D_{succ} (note that the vertical axis have different scales), while D_{notify} stands in between the above two; (iv) $D_{infinite}$ is much larger than D_{succ} and D_{notify} , showing the error introduced by the assumption of the infinite retrials.

The first observation is quite intuitive, and can be explained by the fact that a larger CW_{min} should result in a larger average number of back-off slots. Also, the trend of $D_{intersucc}$ with respect to CW_{min} coincides with the trend of throughput with respect to CW_{min} . In fact, using Equation (2.29), we can obtain similar throughput results as those presented in Figure 2.3. The trend of $D_{infinite}$ with respect to CW_{min} is similar to that obtained in [11], and gives the same misleading conclusion that a larger CW_{min} results in a smaller delay. Now, we explain why the trend corresponding to $D_{infinite}$ is contrary to that of D_{notify} (as well as of D_{succ} and D_{drop}). As known from Equation (2.36), the difference between $D_{infinite}$ and D_{notify} becomes larger when the conditional collision probability p (or P_{drop}) becomes larger. Clearly, when CW_{min} is smaller, p is larger, resulting in a larger error in the calculation of $D_{infinite}$. On the contrary, the error is much smaller when CW_{min} is larger as p becomes smaller. Therefore, $D_{infinite}$ becomes larger under smaller CW_{min} . In conclusion, whenever the conditional collision probability (or the packet drop probability P_{drop}) is large, the assumption of infinite retrials will result in a large error in the calculation of the delay.

Figures 2.15-2.17 present the standard deviation of the first three kinds of delay (i.e., D_{succ} , D_{drop} , and D_{notify}) for different values of CW_{min} . In general, they follow the similar trend as the average delay. D_{notify} has a larger standard deviation than the other two since it considers the delay of all the packets (i.e., dropped as well as successfully transmitted), and since there is a large difference between the delay of the packets being dropped and those being transmitted successfully.

In the above, we have presented the delay results when CW_{min} is variable. Similarly, we have also obtained the results when CW_{max} or m varies. However, here we only present the average of D_{succ} in figures 2.18-2.19. In general, the larger the CW_{max} and m are, the larger is the average delay. Under variable CW_{max} or m , the average of other kinds of delay and their standard deviations, in general, have the trend similar to those under variable CW_{min} .

2.3.4 Fairness Results

Figures 2.20-2.22 present the Coefficient of Variation (COV) of D_{succ} for different values of CW_{min} , CW_{max} , and m , respectively. To recall, a larger COV implies a more unfair system. From the figures, we can clearly make two observations: (i) In general, the larger the CW_{min} is and the smaller the CW_{max} and m are, the more fair the system is; (ii) As the network size (i.e., n) increases, first the system becomes more unfair, and then after a certain value of n it becomes more fair.

We now explain the first observation. In IEEE 802.11, whenever a station transmits a packet successfully, it resets its contention window to CW_{min} . Clearly, if CW_{min} is small, the station that has just transmitted a packet successfully is more likely to get control of the medium again. On the other hand, if CW_{min} is large, it gives other stations more opportunity to transmit on the medium, explaining why the system becomes more fair when CW_{min} is larger. Now, we consider the fairness with respect to CW_{max} . Clearly, the contention window at a station will become CW_{max} only when the station has experienced collisions successively. Therefore, if CW_{max} is large, the colliding stations, though they have already suffered a lot, will be more unfairly treated, explaining why the system is more unfair when the CW_{max} is larger. As for the retry limit m , if it is small, a station that has experienced collision(s) will drop the packet early and will reset its CW to CW_{min} , and then will schedule the transmission of a new packet. On

the contrary, if m is large, the station may have to schedule the retransmission of a packet more number of times, which leads to a larger contention window, and in turn, short-term unfairness. Therefore, in this manner, the early dropping of a colliding packet contributes to the fairness, though it certainly results in more number of collisions as the stations become more aggressive. This explains why the system is more fair when m is small. From observation (i), we may reach an interesting conclusion regarding the back-off policy: to achieve better fairness, we should always use a medium-size CW without any retrial for the unsuccessful transmission.

For the second observation, we first explain the reason why the system becomes more unfair when we begin to increase n (e.g., from 2 to 17). This is due to the randomness involved in the binary exponential back-off (BEB) algorithm. In particular, some stations may experience collisions consecutively while other stations can transmit packets without experiencing any collision during this period. With the increase of n , collisions are more likely to happen, explaining the increasing amount of unfairness. In fact, we have also observed this phenomena in the simulation results presented in Chapter 4. However, when the collision probability increases beyond a certain value along with the increase of n , most of the stations will experience collisions before they can transmit any packet successfully, and therefore, it tends to become more fair for all the stations. In the extreme case, when the number of stations becomes infinite, the collision probability tends to unity, and no station can transmit any packet successfully. Therefore, every station has zero throughput (or infinite delay), implying absolute fairness among the contending stations!

2.3.5 Summary of the Performance Evaluation

In this subsection, we give a brief summary of the performance of IEEE 802.11 in single-hop networks. With respect to the throughput, we find that it is quite sensitive to the chosen system parameters. This would be particularly true in the emerging IEEE 802.11 standards. Therefore, dynamic tuning of the system parameters is of great interest to achieve optimal performance. We also find that the throughput calculated in the model of [7] is higher than that in our model, which is more precise. As for the delay, we show that the first four kinds of delay under the finite retrials condition have very distinct values, showing the importance of distinguishing among them. We also find that the $D_{infinite}$ (i.e., delay under the infinite retrials assumption) has a

Figure 2.3: Throughput under Variable CW_{min} Figure 2.4: Throughput under Variable CW_{max} Figure 2.5: Throughput under Variable Retry Limit m Figure 2.6: Ratio P_{col}/P_{succ} under Variable CW_{min} Figure 2.7: Drop Probability under Variable CW_{min} Figure 2.8: Drop Probability under Variable CW_{max}

Figure 2.9: Drop Probability under Variable Retry Limit m Figure 2.10: Average Delay D_{succ} under Variable CW_{min} Figure 2.11: Average Delay D_{drop} under Variable CW_{min} Figure 2.12: Average Delay D_{notify} under Variable CW_{min} Figure 2.13: Average Delay $D_{intersucc}$ under Variable CW_{min} Figure 2.14: Average Delay $D_{infinite}$ under Variable CW_{min}

Figure 2.15: Standard Deviation of D_{succ} under Variable CW_{min}

Figure 2.16: Standard Deviation of D_{drop} under Variable CW_{min}

Figure 2.17: Standard Deviation of D_{notify} under Variable CW_{min}

Figure 2.18: Average Delay D_{succ} under Variable CW_{max}

Figure 2.19: Average Delay D_{succ} under Variable Retry Limit m

Figure 2.20: Fairness under Variable CW_{min}

Figure 2.21: Fairness under Variable CW_{max}

Figure 2.22: Fairness under Variable Retry Limit m

large error, and thus leads to misleading observations. For the fairness performance, we get an interesting conclusion about the back-off policy: to achieve better fairness, a medium-sized contention window should be used without any retransmission of the unsuccessful transmissions.

Table 2.2 summarizes how the system parameters should be selected (in terms of “large” or “small”) to achieve the desired performance (i.e., higher throughput, smaller delay, and better fairness). By “smaller delay”, we mean a smaller value of the average as well as the standard deviation of D_{succ} , D_{drop} , and D_{notify} . As shown before, when D_{succ} , D_{drop} , and D_{notify} are smaller, $D_{intersucc}$ and $D_{infinite}$ are larger. Since $D_{intersucc}$ is directly reflected by the throughput and $D_{infinite}$ is misleading, we do not need to consider these delays when optimizing the delay performance. Also, since a given set of system parameters, which yields a higher throughput, results in a smaller value of the packet drop probability, we do not need to present the desirable system parameters for a smaller packet drop probability.

From the table, we find that a set of system parameters delivering a good performance for one metric, may deliver a bad performance in terms of another metric. For example, if we aim to achieve a higher throughput, the table shows that we should use a large value of CW_{max} and m . However, the table also tells us that we should use a small value of CW_{max} and m if we aim to achieve better fairness. In other words, a trade-off among the metrics must be made in the dynamic tuning of the system parameters.

Table 2.2: Interdependence between Performance and Parameters

Parameter \ Metric	Higher Throuhgput	Smaller Delay	Better Fairness
$CW_{\min}$	Large	Small	Large
$CW_{\max}$	Large	Small	Small
m	Large	Small	Small

2.4 Conclusions

In this chapter, we have proposed a simple analytical model to evaluate the throughput, delay, and fairness performance in single-hop IEEE 802.11 networks. Using the proposed model, we have carried out an extensive performance evaluation of IEEE 802.11 under three different system parameters (i.e., $CW_{\min}$, $CW_{\max}$, and retry limit m). Our results show that the throughput, delay, and fairness are quite sensitive to these system parameters, and thus dynamic tuning of the parameters is of great interest to achieve the optimal performance. Moreover, the optimal values of the parameters may not be consistent with each other for different performance metrics, implying that some tradeoff among the metrics must be made in the dynamic tuning. Finally, our results also show that the published models have largely overestimated the throughput as well as the packet delay.

Chapter 3

Analytical Model for Hidden-terminal Scenario

In the previous chapter, we have developed an analytical model for the single-hop ad-hoc networks. In this chapter, we develop an analytical model to evaluate the unfairness of IEEE 802.11 in the presence of hidden terminals.

3.1 Introduction

For simplicity, most of the analytical models (e.g., [7, 53, 10] evaluating the performance of IEEE 802.11 have assumed that there are no hidden terminals in the network. Therefore, these models apply only to single-hop ad-hoc networks. Moreover, these models focus on either throughput or packet delay, and they do not throw any light on fairness performance. While analyzing the throughput or delay, these models assume that all the nodes in the system have the same average behavior. Therefore, the models only need to analyze the performance at a typical or tagged node. Though the fairness can also be analyzed using the above method as we have done in the previous chapter, the analysis may not be very precise. To predict the unfairness *precisely*, the model must take into account the difference in the behavior of various nodes. In other words, the fairness captures the relative behavior, and can be predicted only by comparing the behavior at different nodes. Therefore, in the representation of the system state, we need to include one or more separate state variables for each of the nodes. Clearly, when the number of nodes increases, the state space becomes very large, making the model extremely complex.

Recently, the analytical model presented in [57] has focused on the throughput of a scenario with hidden terminals. However, similar to the models for single-hop networks, it first finds the throughput at a typical station, and then gets the total system throughput by multiplying it by the number of stations in the network. Therefore, the model cannot be used to predict the unfairness of IEEE 802.11 in the hidden terminal scenario.

In this chapter, we develop an analytical model to evaluate the short-term unfairness due to the hidden-terminal problem. We measure the short-term fairness by evaluating the following two metrics: (i) once a node gets control of the medium, what is the average number of packets this node can transmit *consecutively* without collision, (ii) once a node loses its control of the medium, what is the average time the node has to wait before it gets control to the medium again. The first metric reflects on how long a node can *capture* the medium, whereas the second one reflects on how long a node may be *starved*.

Using an embedded Markov chain model, the above two metrics are measured based on the concepts of “*expected state holding time*” and “*expected first passage time*”. Moreover, to predict the *short-term* behavior precisely, our model captures the protocol in a very detailed manner and uses separate parameters to represent the state of each station. The analytical model is validated by simulation results. The results show that our model is able to precisely predict the short-term unfairness of IEEE 802.11 in presence of hidden terminals.

The remainder of the chapter is organized as follows. In Section 3.2, through a simple simulation, we show that IEEE 802.11 exhibits considerable short-term unfairness in the presence of hidden-terminals. In Section 3.3, the Markov chain model is described. Sections 3.4 and 3.5 deal with the computation of the transition probabilities and the expected state holding time. Section 3.6 presents the analytical and simulation results, and the chapter is concluded in Section 3.7.

3.2 Short-Term Unfairness in IEEE 802.11

In order to show the short-term unfairness in the presence of hidden-terminal, we simulate the simple hidden-terminal scenario depicted in Figure 3.1.

The simulation environment is as follows. NS-2 with CMU wireless extensions [15] and static routing is used. For each single-hop flow, a Constant Bit Rate (CBR) traffic generates

Figure 3.1: Hidden-terminal Scenario

200 packets per second. Each packet is 1000-bytes long, resulting in a traffic source rate of 1.6 Mbps. The raw bandwidth is set with 2 Mbps, leading to a *maximum* throughput of about 1.4 Mbps due to the overhead of IEEE 802.11. The source rate is made greater than the medium capacity as unfairness issue occurs only when the system is overloaded. The sensing range is equal to the transmission range.

The simulation results show that each flow gets an *average* throughput of about 0.67 Mbps, which means that the two flows share the medium fairly on a *long-term* basis¹. However, if we compute the values of the two *short-term* metrics defined in Section 3.1, the protocol exhibits substantial unfairness. For metric (i), we find that, on an average, once a node gets control of the medium, it can transmit about 6.4 packets *consecutively* without collision. For metric (ii), once a node (say node C) loses control of the medium, it has to wait for the other node (i.e., A) to transmit about 27 packets before it gets control of the medium again. However, this does not mean that node A can transmit 27 packets *consecutively* without collision. We illustrate this using the following example. Consider that after node C *loses* control of the medium, node A gets control of the medium and it transmits one or more packets consecutively. Then, one or more collisions occur. After the collision(s), node A again gets control and transmits one or more packets consecutively. This kind of process may repeat several times before node C gets control of the medium, explaining why the values of the two metrics *differ* so much. In addition to the average values, we also observed the corresponding *maximum* values, which are equal to 35 and 160, respectively, for the two metrics. This shows how much short-term unfairness is ingrained in the current IEEE 802.11, which is unacceptable for jitter-sensitive traffic. In the next two subsections, we explain the short-term unfairness in an intuitive way.

3.2.1 Condition for Successful RTS/CTS Handshake

As pointed out in Chapter 1, the four-way handshake solves the hidden-terminal problem largely by introducing the RTS/CTS handshake before the actual Data frame is transmitted. However,

¹Note that the aggregate throughput is less than 1.4Mbps since a large number of collisions involved in this scenario.

it cannot eliminate the problem completely as the transmission of the RTS frames from the hidden terminals may collide, and thus the RTS/CTS handshake may not always be successful. We now derive the condition under which the RTS/CTS handshake can be successful when two hidden nodes (A and C) contend for the medium (Figure 3.1). The condition is as follows: after a collision or a successful FES, the difference between the back-off counters at the two nodes should be large enough for node B to send back a CTS to node A (C) before node C (A) starts sending its RTS. The minimum time difference required is equal to the transmission time of RTS plus a Short Inter-Frame Space (SIFS). This can be expressed as,

$$|Z| > Len = TxTime(RTS) + SIFS \quad (3.1)$$

where Z is the difference between the back-off counters of nodes A and C. Len is equal to about 19 slots when the slot time is $20 \mu s$ for DSSS [24]. It is easy to see that the condition is difficult to satisfy when the CWs at the contending nodes are small (e.g., 31). Note that in the above analysis, we have omitted the propagation delay, which is negligible. We also assume that the two nodes begin to back-off at the same time, i.e., after a collision, the time difference between the two nodes detecting the collision is neglected.

3.2.2 Explanation for the Short-term Unfairness

Now, let us explain how the hidden-terminal problem causes short-term unfairness. Consider the situation that the CWs at nodes A and C are very small (e.g., 31). As discussed above, under such situation the condition illustrated in Equation (3.1) is difficult to satisfy, i.e., the transmission of RTSs of nodes A and C may overlap partially, and as a result they collide. The collision may occur several times until the CWs are large enough to allow either node to get control of the medium. In particular, one of the two nodes (let us say, node A) may select a small back-off time from its CW, while the other node (i.e., C) selects a large value. The difference between these two values may be large enough to satisfy the condition illustrated in Equation (3.1), and thus the FES(A, B) can be successfully completed. Once the FES(A, B) is completed, node A resets its CW and backs-off before initiating another FES. However, the remaining back-off counter at node C may be large compared to the back-off counter at node A, which is drawn from the range $[0, CW_{min}]$. In that case, nodes A and B may exchange several

more FESs before node C's back-off counter reduces to a small value. Clearly, the freezing mechanism used in IEEE 802.11 causes short-term unfairness.

Whenever the back-off counter at node C reduces to a small value, node C contends for the medium. However, as the CW at node A is equal to CW_{min} , the contention is most likely to result in a collision. After the collision, node A doubles its CW from CW_{min} whereas node C doubles its CW from a larger value (at least 63). Therefore, the CW at node C is greater than that at A, and node A is more likely to get control of the medium *again*. This is obviously unfair for node C since A has already transmitted several packets while C is starved during this period. Moreover, this process (i.e., several packet transmissions by node A, followed by collisions, and then packet transmissions by node A again) may repeat several times, leading to starvation at node C for a long period (compared to the time needed for a FES).

However, several mechanisms incorporated in the IEEE 802.11 prevent node C from starving *completely*, such as: (i) after every FES, node A will back-off before initiating another FES, which gives node C a chance to contend for the medium with node A; (ii) the CW at node C will be reset to CW_{min} after the retry limit m is reached. Moreover, once node C gets control of the medium, it will capture the medium for a long time in a similar manner. As a result, the long-term fairness between these two flows is ensured.

Though the above analysis explains how the hidden-terminal problem causes short-term unfairness, we still need to evaluate the unfairness *quantitatively*, which is carried out in the following sections.

3.3 Analytical Modelling

In this section, we model the hidden-terminal scenario in Figure 3.1 using an *embedded* Markov chain. In the discussion, the reader must differentiate between the words *stage* and *state*, which are very often used.

3.3.1 Markov Chain Model

At any point of time, the *medium* is in one of the following five states: T_A , T_B , T_C , Col and $Idle$, where T_A means that node A is getting the control of the medium and transmitting its packet, i.e., FES(A, B) is in progress. Similarly, T_B and T_C correspond to nodes B and C,

respectively. However, in our considered scenario, T_B does not arise, as node B does not have any data packets to send. State *Col* means that there is a collision on the medium, while state *Idle* means that there is no transmissions or collisions over the medium. As our objective is to analyze the short-term fairness rather than the capacity utilization of the medium, we do not need to consider the *Idle* state. As a result, only three states of the medium are considered: T_A , T_C , and *Col*. When the medium is either in T_A or in T_C , we simply say, for the convenience of the presentation, that the medium is in a T state.

As mentioned before, to predict the unfairness *precisely*, we should model the system state in a way that there is at least one separate parameter used to represent the state of every contending station (i.e., nodes A and C in our case). Since the transition probabilities among the T_A , T_C , and *Col* states depend on the values of CWs at nodes A and C, which, in turn, are determined by the corresponding *stages*, we use the *stage* number at a station to represent the state of the station. Therefore, the *system state* can be modelled using three random variables: state of the *medium*, *stage* at node A, and *stage* at node C. Therefore, the system states are (T_A, k, l) , (T_C, k, l) , and (Col, k, l) , where k and l denote the *stages* at nodes A and C, respectively. Obviously, $k, l = 0, \dots, (m - 1)$. Note that we must use *stages* rather than the values of CW to represent the system state, because, as shown by Equation (1.2), the CW does not change once stage m' is reached. When the medium is in state T_A , it is easy to see that the stage at the *transmitting* node (i.e., node A) must be zero, that is, only $(T_A, 0, l)$ system states are possible. Similarly, when the medium is in state T_C , the possible system states are $(T_C, k, 0)$. Therefore, if the retry limit is m , there are m^2 number of *Col* states, and m number of states corresponding to *each* of T_A and T_C , and thus the number of all possible system states is:

$$N_{state} = m^2 + 2m. \quad (3.2)$$

From a *Col* state, whenever a *transition* occurs, the system can enter anyone of the three kinds of states as shown in Figure 3.2. If the *next* state is also a *Col* state, both of the stages, k and l , are incremented by one, except that the stage is reset to zero whenever the retry limit m is reached. On the other hand, if the next state is a T state, the stage at the *transmitting* node is reset while the other stage remains unchanged.

Figure 3.2: State Transitions from a *Col* State

In state T_A , whenever node A transmits another *packet*, it is natural to view this event as a *self-transition*. However, if we model the system in such a way, the transition probabilities depend upon the remaining back-off counter at the waiting node (i.e., node C), which in turn, depends upon how many times this time counter has been *frozen* and decremented, i.e., how many self-transitions have occurred in the T_A state. This requires *memorizing* the *history* to obtain the *current* transition probabilities, which violates the *memoryless* requirement of the Markov chain. Therefore, we do *not* treat this event as a state transition. Rather, whenever in state T_A , a state transition occurs *only* when the system enters a *Col* state or a T_C state. Therefore, whenever the system enters a T_A state, the time that the system will *remain* in that state depends on the number of packets that node A can transmit consecutively before that node C controls the medium or that a collision occurs. Similar explanation applies for the state T_C . The state transition at a T state is illustrated in Figure 3.3. It is easy to see that the chain obtained in such a way is an *embedded* Markov chain.

Figure 3.3: State Transitions from a *T* State

To illustrate the model more clearly, we present the complete state transition diagram in Figure 3.4, where we assume $m = 3$. The diagram is primarily based on the transitions diagrams shown in figures 3.2 and 3.3. We realize that the diagram is quite complex even in such a simple case, and it is difficult to draw it on paper when m is large since the number of states is $m^2 + 2m$.

Figure 3.4: State Transition Diagram for $m = 3$

3.3.2 Basic Analysis of the Model

For the convenience of presentation, rather than representing the system states using three variables as above, we assign a *single* variable to represent the states, by ordering them as indicated in Table 3.1.

Table 3.1: Re-designating the State Variables

States	Single Variable Range
$(T_A, 0, 0)$ to $(T_A, 0, m-1)$	$[0, m-1]$
$(T_C, 0, 0)$ to $(T_C, m-1, 0)$	$[m, 2m-1]$
$(Col, 0, 0)$ to $(Col, m-1, m-1)$	$[2m, m^2+2m-1]$

For the embedded Markov chain, we need to know the transition probability matrix, P . Once we get the matrix P , we can find the *steady state* probability vector, π , by solving the following equation:

$$\begin{cases} \pi = \pi \cdot P & \pi = [\pi_0, \pi_1, \pi_2, \dots], P = [p_{ij}] \\ \sum_i \pi_i = 1 & i \in [0, m^2 + 2m - 1] \end{cases}. \quad (3.3)$$

In a *pure discrete* Markov chain, if the *time intervals* between two *consecutive* transitions (including self-transition) are always *the same*, the steady state probability π_i reflects the proportion of the *time* that the system is in state i . However, in our model, since the interval

between a T state and its next state is a *random* variable, π_i can only tell us the probability that the system *enters* state i whenever a *transition* occurs [19]. To get the *time average* state probability of being in state i , we must first analyze the *holding time* of state i .

Let μ_{ji} denote the average time that the system will remain in state i once a *transition* from state j to i occurs. The μ_{ji} is given by:

$$\mu_{ji} = \begin{cases} 0 & p_{ji} = 0 \\ 1 & p_{ji} > 0, i \in [2m, m^2 + 2m - 1] \\ r \times \text{num}(j, i) & p_{ji} > 0, i \in [0, 2m - 1] \end{cases} \quad (3.4)$$

Whenever the transition probability p_{ji} is zero, the μ_{ji} must also be zero. The holding time in a *Col* state, since it does not depend upon the previous or the present state, is assigned a *unit* value as indicated in the second line of the above formula. For expressing the holding time of a T state, let us define $\text{num}(j, i)$, which denotes the average number of packets the transmitting node can transmit consecutively once the system reaches the T state (i.e., state i) from state j . Clearly, the holding time of a T state is proportional to $\text{num}(j, i)$. Moreover, we define the ‘*FES time*’ as the time needed for a FES to be completed, and ‘*Col time*’ as the time needed to detect the collision², while r is the *ratio* between the ‘*FES time*’ and the ‘*Col time*’. Therefore, whenever the system enters a T state, the average holding time can be represented by $r \times \text{num}(j, i)$, if the time required to detect a collision is unity. This explains the last line of Equation (3.4). From μ_{ji} , which corresponds to a *transition*, we can get μ_i , the expected holding time of state i , as follows:

$$\mu_i = \sum_j \frac{\pi_j \times p_{ji}}{\pi_i} \mu_{ji}. \quad (3.5)$$

It is easy to see that the μ_i corresponding to a *Col* state is always equal to *one*. Corresponding to a T state, we define $\text{num}(i)$ in Equation (3.6), which denotes the average number of packets the *transmitting* node can transmit consecutively once the system reaches state i .

$$\text{num}(i) = \sum_j \frac{\pi_j \times p_{ji}}{\pi_i} \text{num}(j, i) \quad i \in [0, 2m - 1]. \quad (3.6)$$

As a result, formula (3.5) can be replaced by:

²Note that ‘*FES time*’ and ‘*Col time*’ are equal to T_{succ} and T_{col} , respectively, which have been defined in Equation (2.10).

$$\mu_i = \begin{cases} num(i) \times r & i \in [0, 2m - 1] \\ 1 & i \in [2m, m^2 + 2m - 1] \end{cases} \quad (3.7)$$

Now, we can get ρ_i , which represents the *time average* state probability of state i , as follows [19]:

$$\rho_i = \frac{\pi_i \times \mu_i}{\sum_j \pi_j \times \mu_j} \quad i, j \in [0, m^2 + 2m - 1]. \quad (3.8)$$

3.3.3 Derivation of the Metrics

Based on the above analysis, we can obtain the two metrics defined in Section 3.1. To recall, the *first* metric is, once a node gets control of the medium, what is the average number of packets this node can transmit consecutively without any collision. In our Markov chain model, let us say for node A, it is *not* possible for the system to travel from one T_A state to another T_A state without visiting a T_C or Col state. Moreover, once the system enters a given T_A state (let us say, state i), the average number of units of ‘*FES time*’ for which the system *remains* in that T_A state, is simply equal to $num(i)$, which is defined by Equation (3.6). Therefore, the first metric can be obtained by taking average of $num(i)$ corresponding to all T_A states:

$$H_{metric-1} = \sum_{i=0}^{m-1} \frac{\pi_i}{\pi(A)} \times num(i) \quad \text{where } \pi(A) = \sum_{i=0}^{m-1} \pi_i. \quad (3.9)$$

Since the behavior is *identical* at nodes A and C, the $H_{metric-1}$ is also applicable to node C.

Now, we recall that the *second* metric is, once a node *loses* its control of the medium, what is the average time the node has to wait before it gets control to the medium again. Since there are only two nodes (i.e., A and C) contending for the medium (Figure 3.1), the metric, let us say for node A, can be replaced by: once the medium is controlled by node A, what is the average number of packets that A can transmit before the medium is controlled by node C. This simply implies that once the system enters a T_A state, what is the average number of units of ‘*FES time*’ that the system can remain in *any* T_A state (via visiting Col state in-between) before the system enters a T_C state. Note that the *second* metric allows a visit to a Col state in-between two T_A states, which is *not* permitted in the *first* metric.

Here we use the concept of “*expected first passage time*” [19], which means that if the system *starts* in a given T_A state (let us say, state i), what is the expected time after which the system will enter *any* T_C state for the *first* time. The expected first passage time, V_i , can be expressed as follows [19]:

$$V_i = \begin{cases} 0 & i \in [m, 2m-1] \\ R_i + \sum_j p'_{ij} V_j & i \in [0, m-1] \cup [2m, m^2 + 2m - 1] \end{cases} \quad (3.10)$$

where R_i is the *immediate reward* once the system enters state i . For a T_A state, R_i is equal to the corresponding $num(i)$. Since all T_C states are the *trapping* states, R_i is zero for these states [19]. We should also assign R_i with zero for all *Col* states, as our objective is to obtain the number of FESs, rather than the number of collisions. Therefore, R_i is,

$$R_i = \begin{cases} num(i) & i \in [0, m-1] \\ 0 & i \in [m, m^2 + 2m - 1] \end{cases} . \quad (3.11)$$

In Equation (3.10), p'_{ij} is a *modified* value of the transition probability p_{ij} . Since the T_C states are considered as *trapping* states, the transition probabilities out of a T_C state is set to zero, whereas the *self-transition* probability for each T_C state is set to one. Other transition probabilities remain unchanged. Therefore, the p'_{ij} :

$$p'_{ij} = \begin{cases} 0 & i \in [m, 2m-1], j \neq i \\ 1 & i \in [m, 2m-1], j = i \\ p_{ij} & i \in [0, m-1] \cup [2m, m^2 + 2m - 1] \end{cases} . \quad (3.12)$$

Obviously, V_i corresponding to the *trapping* states (i.e., all T_C states) should be zero. For all the other states, V_i is equal to the ‘immediate reward’ R_i plus the expected reward earned from whatever state is entered *next*. This explains the Equation (3.10).

Now, the second metric can be obtained by taking average over V_i values corresponding to all the T_A states:

$$H_{metric-2} = \sum_{i=0}^{m-1} \frac{\pi_i}{\pi(A)} \times V_i \quad \text{where } \pi(A) = \sum_{i=0}^{m-1} \pi_i. \quad (3.13)$$

The $H_{metric-2}$ computed for node A, is also applicable to node C.

To solve the above equations, we *only* need to know the transition probability matrix P and the $num(j, i)$, which are computed in sections 3.4 and 3.5, respectively.

3.4 Transition Probability Computation

The transition probabilities depend upon the value of the back-off counters at nodes A and C. In each of the subsections 3.4.1 and 3.4.2, we first compute the transition probabilities in general terms based on the possible ranges of the back-off counters, and then obtain the specific transition probabilities.

3.4.1 Transition Probabilities out of a Col State

Assuming two nodes are contending for the medium, after a collision, each node will generate a random number from its CW according to the *uniform* distribution. Let us denote the two random numbers by X and Y , and their corresponding CWs by CW_x and CW_y . Moreover, we denote T_x as a T state under which the node, whose contention window is CW_x , gets control of the medium, while T_y is a T state corresponding to the node with the CW_y . It is clear that the *difference* between X and Y determines what is the next state and what is the corresponding transition probability. As in Formula (3.1), we refer to this difference as Z , i.e., $Z = X - Y$. Since one of the CWs must be smaller than or equal to the other one, we simply assume $CW_x \leq CW_y$ and thus Z is in the range $[-CW_y, CW_x]$. As a result, all the possible transition probabilities from *Col* states can be expressed as follows in terms of Z :

$$\begin{aligned} Pr(T_x | (Col, CW_x, CW_y)) &= Pr(-CW_y \leq Z < -Len) \\ Pr(Col | (Col, CW_x, CW_y)) &= Pr(-Len \leq Z \leq Len) \\ Pr(T_y | (Col, CW_x, CW_y)) &= Pr(Len < Z \leq CW_x) \end{aligned} \quad (3.14)$$

In the above conditional probabilities, the *present* states (e.g., (Col, CW_x, CW_y)) have been described using three variables whereas the *next* states is simply described by T_x , T_y , or *Col* since the transition probabilities do not depends on the CWs in the next states. The distribution function of Z can be easily derived as follows:

$$Pr(Z = z) = \begin{cases} (CW_y + z + 1)P_{xy} & z \in [-CW_y, CW_x - CW_y] \\ (CW_x + 1)P_{xy} & z \in [CW_x - CW_y, 0] \\ (CW_x - z + 1)P_{xy} & z \in [0, CW_x] \end{cases} \quad (3.15)$$

where $P_{xy} = 1/((1 + CW_x)(1 + CW_y))$.

We can compute the transition probabilities using equations (3.14) and (3.15). It is easy to see that the transition probabilities depend upon the value of Len with respect to those of CW_x , CW_y , and $(CW_x - CW_y)$. We here show in Figure 3.5 how to obtain the $Pr(T_x | (Col, CW_x, CW_y))$.

Similarly, we obtained all the other probabilities under various conditions. However, we directly present the results in Figure 3.6 under the assumption: $0 < Len < CW_{min}$, which is true in the considered case where $Len = 19$ and $CW_{min} = 31$.

<p>When $Len \in [0, CW_y - CW_x)$</p> $\begin{aligned} \Pr(T_x Col, CW_x, CW_y) &= \sum_{z=-CW_x}^{-Len-1} P(z) \\ &= P_{\mathcal{P}} \left\{ \sum_{z=-CW_x}^{CW_x - CW_y - 1} (CW_y + z + 1) + \sum_{z=CW_x - CW_y}^{-Len-1} (CW_x + 1) \right\} \\ &= \frac{CW_y - Len - CW_x / 2}{CW_y + 1} \end{aligned}$
<p>When $Len \in [CW_y - CW_x, CW_y)$</p> $\begin{aligned} \Pr(T_x Col, CW_x, CW_y) &= \sum_{z=-CW_x}^{-Len-1} P(z) \\ &= P_{\mathcal{P}} \sum_{z=-CW_x}^{-Len-1} (CW_y + z + 1) = \frac{(CW_y - Len)(1 + CW_y - Len)}{2(1 + CW_x)(1 + CW_y)} \end{aligned}$
<p>When $Len \in [CW_y, \infty)$</p> $\Pr(T_x Col, CW_x, CW_y) = \sum_{z=-CW_x}^{-Len-1} P(z) = 0$

Figure 3.5: Computation of $Pr(T_x | Col, CW_x, CW_y)$

<p>If $CW_x = CW_y = cw$:</p> $\begin{aligned} \Pr(Col (Col, cw, cw)) &= \frac{1 + cw + 2cw \times Len + Len - Len^2}{(1 + cw)^2} \\ \Pr(T_y (Col, cw, cw)) &= \Pr(T_x (Col, cw, cw)) \\ &= \frac{(cw - len)(1 + cw - len)}{2(cw + 1)^2} \end{aligned}$
<p>If $CW_x < CW_y$:</p> $\begin{aligned} \Pr(T_x (Col, CW_x, CW_y)) &= \frac{CW_y - Len - CW_x / 2}{CW_y + 1} \\ \Pr(T_y (Col, CW_x, CW_y)) &= \frac{(CW_x - Len)(1 + CW_x - Len)}{2(1 + CW_x)(1 + CW_y)} \\ \Pr(Col (Col, CW_x, CW_y)) &= \frac{1 + (3 - Len) \times Len / 2 + CW_x(1 + 2Len)}{(1 + CW_y)(1 + CW_x)} \end{aligned}$

Figure 3.6: Transition Probabilities out of a *Col* State

Now, it is easy to compute the transition probabilities out of a given *Col* state. This is achieved by translating the stages at the two nodes to the corresponding CWs, and then, assign

the *smaller* CW to CW_x and the *greater* one to CW_y .

3.4.2 Transition Probabilities out of a T state

Once the system enters a T state, the *transmitting* node may transmit one or several packets before the *waiting* node's back-off counter reduces to zero. Figure 3.7 shows how the back-off counter at the waiting node decrements.

In the diagram, U denotes the back-off counter at the waiting node when the system enters a T state, and $num(\geq 1)$ denotes the number of packets transmitted by the transmitting node before the back-off counter at the *waiting* node reduces to zero. S_k represents the back-off counter at the transmitting node corresponding to the k^{th} consecutive FES. Now, let us consider how much the U will be reduced during *each* 'FES time'. Clearly, during each S_k , the waiting node counts down its back-off counter. From the time, the transmitting node initiates a RTS, to the time, the receiving node begins to send back a CTS, the waiting node also counts down its back-off counter. This period is equal to Len . As soon as the waiting node overhears the CTS, it *freezes* its back-off counter. Therefore, during each 'FES time', the waiting node reduces its back-off counter by $S_k + Len$. When the *last* FES (i.e., the num^{th} FES) is completed, the transmitting node generates a new random number (let us say, S), while the waiting node simply counts down its frozen back-off counter. Let W denote the remaining value of the frozen back-off counter, which can be expressed as:

$$W = U - \sum_{k=1}^{num} (S_k + Len). \quad (3.16)$$

To find the next state and the corresponding transition probability, we need to know the *ranges* and the *distributions* of S and W . It is easy to see that S is *uniformly* distributed over the range $[0, CW_{min}]$. However, it is not so trivial to get the distribution of W . It is intuitive that, when the CW at the waiting node (let us say, CW_u), from which U is drawn, is large, the num is also large. Though, to get the distribution of W , computationally it is possible to

Figure 3.7: Counting down of the Back-off Counter at the Waiting Node

consider all values of CW_u , for simplicity, we consider only two cases here: $CW_u = CW_{min}$ and $CW_u > CW_{min}$.

When $CW_u = CW_{min}$, the *num* is most of the time *unity*, therefore, Equation (3.16) can be simplified as:

$$W = U - (S_1 + Len). \quad (3.17)$$

We notice that the $(U - S_1)$ must be greater than Len , *otherwise* the system will *not* enter the T state. Moreover, since U is drawn from the range $[0, CW_{min}]$, the range of S_1 must be $[0, CW_{min} - Len]$. Therefore, from Equation (3.17), W must be in the range of $[0, CW_{min} - Len]$. Since both S_1 and U are *uniformly* distributed, the distribution of W can be derived as follows:

$$Pr(W = w) = \frac{2(1 + CW_{min} - Len - w)}{(1 + CW_{min} - Len)(2 + CW_{min} - Len)}. \quad (3.18)$$

When $CW_u > CW_{min}$, we notice that W must be smaller than $CW_{min} + Len$, *otherwise*, the transmitting node will be able to transmit one *more* packet. Since in this case, it is difficult to obtain the distribution of W , we approximate it with a uniform distribution in the range of $[0, CW_{min}]$.

Once we know the *ranges* and the *distributions* of the back-off counters (i.e., S and W), we can obtain the transition probabilities as done before. However, we here present the results directly in Figure 3.8.

Now, the transition probabilities can be found according to the following rule: if the *stage* at the *waiting* node is equal to zero, the probabilities corresponding to $CW_u = CW_{min}$ are used, otherwise, the other probabilities are used.

3.5 State Holding Time Computation

Now, we show how to compute $num(j, i)$, the average number of packets the transmitting node can transmit consecutively once the system reaches a T state (i.e., state i) from state j . If the *previous* state (i.e., state j) is also a T state, both the *stage number* in the present state must

<p>If $CW_u = CW_{\min}$:</p> $\Pr(T CW_u = CW_{\min}) = \frac{2(CW_{\min} - Len)}{3(1 + CW_{\min})}$ $\Pr(Col CW_u = CW_{\min}) = \frac{3 + CW_{\min} + 2Len}{3(1 + CW_{\min})}$ <hr style="border-top: 1px dotted black;"/> <p>If $CW_u > CW_{\min}$:</p> $\Pr(T CW_u > CW_{\min}) = \frac{(CW_{\min} - Len)(1 + CW_{\min} - Len)}{2 + (3 + CW_{\min})CW_{\min} + Len + 2Len \times CW_{\min} - Len^2}$ $\Pr(Col CW_u > CW_{\min}) = \frac{2(1 + CW_{\min} + Len + 2Len \times CW_{\min} - Len^2)}{2 + (3 + CW_{\min})CW_{\min} + Len + 2Len \times CW_{\min} - Len^2}$
--

Figure 3.8: Transition Probability out of a T State

be zero (see Figure 3.3), and therefore the $num(j, i)$ is mostly unity. However, if the previous state is a Col state, computation of $num(j, i)$ is not so trivial.

In Section 3.4.2, when the system enters a T state, the back-off counter at the waiting node is represented by U while the back-off counter at the transmitting node is represented by S_1 (see Figure 3.7). The corresponding CWs are CW_{s_1} and CW_u , respectively. It is easy to see that, when the system is in the state j (i.e., previous state), the CWs are also CW_{s_1} and CW_u . Moreover, we have concluded that, during each ‘FES time’, the waiting node reduces its back-off counter by $S_k + Len$. Since S_k (when $k > 1$) is uniformly distributed over the range $[0, CW_{\min}]$, the average of S_k (when $k > 1$) is $(CW_{\min} + 1)/2$. However, as the CW_{s_1} may not be the same as $CW_{\min}$, we need to compute the average value of S_1 *separately*. When the *last* FES (i.e., the num^{th} FES) is completed, the remaining back-off counter at the waiting node is W , which is a very small value as seen from Section 3.4.2, and so its average value can be neglected. Therefore, the $num(j, i)$ can be expressed as:

$$\begin{aligned}
 num(j, i) &= 1 + \frac{Avg(U) - (Avg(S_1) + Len) - Avg(W)}{Avg(S_k; k > 1) + Len} \\
 &\approx 1 + \frac{Avg(U) - Avg(S_1) - Len}{(CW_{\min} + 1)/2 + Len}.
 \end{aligned} \tag{3.19}$$

In order to solve Equation (3.19), we *only* need to know the average values of U and S_1 . Since we are considering the average values of U and S_1 under the condition that the system

enters a T state (i.e., $U - S_1 > Len$), we must first derive the *conditional* distribution of U and S_1 , from which we can obtain their average values. In the following we illustrate how to compute the average value of S_1 . The *conditional* distribution of S_1 is:

$$\begin{aligned} Pr(S_1 = s | (U - S_1 > Len)) &= \frac{Pr(S_1 = s, U - s > Len)}{Pr(U - S_1 > Len)} \\ &= \frac{Pr(S_1 = s) \times Pr(U > s + Len)}{Pr(U - S_1 > Len)} \\ &= \frac{P_{s_1} \times P_u \times (CW_u - Len - s)}{Pr(U - S_1 > Len)} \end{aligned} \quad (3.20)$$

where $P_{s_1} = 1/(CW_{s_1} + 1)$ and $P_u = 1/(CW_u + 1)$.

In the above formula, $Pr(U - S_1 > Len)$ is equal to $(CW_u - Len - CW_{s_1}/2)/(CW_u + 1)$ if $CW_u > CW_{s_1}$, which can be easily computed from the formula (3.15). Therefore,

$$Pr(S_1 = s | (U - S_1 > Len)) = \frac{2(Len + s - CW_u)}{(1 + CW_{s_1})(2Len + CW_{s_1} - 2CW_u)}. \quad (3.21)$$

The average of S_1 under the condition $U - S_1 > Len$, which is represented by $E(S_1 | \dots)$, is:

$$\begin{aligned} E(S_1 | (U - S_1 > Len)) &= \sum_{s=0}^{CW_{s_1}} Pr(S_1 = s | (U - S_1 > Len)) \\ &= \frac{CW_{s_1}(3CW_u - 3Len - 2CW_{s_1} - 1)}{6(CW_u - Len - CW_{s_1}/2)}. \end{aligned} \quad (3.22)$$

Similarly, we can get the average value of U , and then using Equation (3.19) we can evaluate $num(j, i)$. The results are directly presented in Figure 3.9.

Now, it is easy to compute a specific $num(j, i)$ when state j is a *Col* state. To achieve this, the stages in state j are first translated to the corresponding CW, and then the CW belonging to the *present transmitting* node is assigned to CW_{s_1} and the other one to CW_u .

3.6 Numerical Results

In Section 3.6.1, values of the transition probabilities and $num(j, i)$ from the Markov chain are presented. Based on these results, in Section 3.6.2, we compute the metrics and compare them to those obtained from the simulation.

<p>When $CW_u > CW_{s_1}$:</p> $E(S_1 (U - S_1 > Len)) = \frac{CW_{s_1}(3CW_u - 3Len - 2CW_{s_1} - 1)}{6(CW_u - Len - CW_{s_1}/2)}$ $E(U (U - S_1 > Len)) = \frac{1}{12}(6(1 + CW_u + Len) + CW_{s_1}(3 + \frac{2 + CW_{s_1}}{CW_{s_1} - 2CW_u + 2Len}))$ $num(j,i) = \frac{-2CW_{s_1}^2 - 6(CW_u - Len)(2 + CW_{min} + CW_u + Len) + CW_{s_1}(5 + 3CW_{min} + 6CW_u)}{3(1 + CW_{min} + 2Len)(CW_{s_1} - 2CW_u + 2Len)}$ <hr style="border-top: 1px dashed black;"/> <p>When $CW_u \leq CW_{s_1}$:</p> $E(S_1 (U - S_1 > Len)) = \frac{1}{3}(-1 + CW_u - Len)$ $E(U (U - S_1 > Len)) = \frac{1}{3}(1 + 2CW_u + Len)$ $num(j,i) = \frac{7 + 2CW_u + 3CW_{min} + 4Len}{3(1 + CW_{min} + 2Len)}$
--

Figure 3.9: State Holding Time

3.6.1 Transition Probabilities and State Holding Time

Transition Probabilities out of *Col* States: The transition probabilities from the *Col* states are presented in Table 3.2, which are computed from the formulas derived in Section 3.4.1. From the results, we can see that, when the CWs are small (e.g., 31), the probability of the system transiting to another *Col* state is very large (e.g., 0.848). To resolve collisions, we must reduce this probability, so that the system is more likely to enter a *T* state. From the table we see that the transition probability from a *Col* state to another *Col* state is small in the following two situations: (i) when both the CWs are large, (ii) when the difference between the two CWs is large. However, in such situations, if the system *does* enter a *T* state, the system will remain in that *T* state for a long time (which is clearly brought out by the later results), which results in short-term unfairness. This shows the *conflict* between resolving collisions and achieving short-term fairness of IEEE 802.11 in the hidden-terminal scenario.

Transition Probabilities out of *T* States: Similarly, we compute the transition probabilities out of a *T* state (see Table 3.3) using the results of Section 3.4.2. We see that the transition probability from a *T* state to another *T* state is small in comparison to the transition probability from a *T* state to a *Col* state. This shows, after a node loses control of the medium,

Table 3.2: Transition Probabilities out of a Col State

CW_x	CW_y	$\Pr(T_x Col)$	$\Pr(T_y Col)$	$\Pr(Col Col)$
31	31	0.076	0.076	0.848
	63	0.445	0.038	0.517
	127	0.723	0.019	0.258
	255	0.861	0.010	0.129
	511	0.931	0.005	0.064
	1023	0.965	0.002	0.033
63	63	0.241	0.241	0.518
	127	0.598	0.121	0.281
	255	0.799	0.060	0.141
	511	0.900	0.030	0.070
	1023	0.950	0.015	0.035
127	127	0.359	0.359	0.282
	255	0.674	0.180	0.146
	511	0.837	0.090	0.073
	1023	0.918	0.045	0.037
255	255	0.426	0.426	0.148
	511	0.712	0.213	0.075
	1023	0.856	0.107	0.037
511	511	0.462	0.462	0.076
	1023	0.731	0.231	0.038
1023	1023	0.481	0.481	0.038

Table 3.3: Transition Probabilities out of a T State

CW_u	$\Pr(Col T)$	$\Pr(T T)$
31	0.75	0.25
>31	0.918	0.082

the system is more likely to experience collisions before either node gets control of the medium.

State Holding Time $num(j, i)$: As discussed in Section 3.5, the $num(j, i)$ is mostly equal to unity when the previous state (i.e., state j) is also a T state. When the previous state is a Col state, the values of $num(j, i)$ are presented in Table 3.4. From the table, we can see that when the two CWs are very large (e.g., both are 1023), or the difference between the two CWs is very large (e.g., one is 31 and the other one is 1023), the $num(j, i)$ is very large (e.g., 10.581 and 15.137, respectively). This *strengthens* our earlier conclusion that resolving collisions and achieving short-term fairness are contradictory objectives when using IEEE 802.11 in a hidden terminal scenario.

Table 3.4: $\text{num}(j,i)$ when the state j is a Col state

$CW_{s,i}$	CW_u	Avg(S_i)	Avg(U)	$\text{num}(j,i)$
≥ 31	31	3.667	27.333	1.133
31	63	12.509	47.754	1.464
	127	14.578	80.790	2.349
	255	15.113	145.057	4.170
	511	15.321	273.161	7.824
	1023	15.414	529.207	15.137
≥ 63	63	14.333	48.667	1.438
63	127	27.039	87.020	2.171
	255	29.831	152.416	3.960
	511	30.759	280.879	7.603
	1023	31.149	537.075	14.912
≥ 127	127	35.667	91.333	2.048
127	255	55.586	165.293	3.592
	511	60.314	295.657	7.181
	1023	62.048	552.524	14.471
≥ 255	255	78.333	176.667	3.267
255	511	112.517	321.759	6.435
	1023	121.269	582.135	13.625
≥ 511	511	163.667	347.333	5.705
511	1023	226.315	634.657	12.124
1023	1023	334.333	688.667	10.581

3.6.2 Comparison between Analytical and Simulation Results

In this section, we evaluate the equations derived in Section 3.3, and the analytical results are compared with those obtained from the simulation. The simulation environment is the same as described in Section 3.2. The results correspond to the case when $CW_{min}=31$, $Len=19$, $m' = 5$, and $m = 6$, which are typically used in IEEE 802.11. The results presented here are calculated through several scripts which analyze the trace file generated by the NS2 simulator. Since our main objective is to analyze the behavior at the T states, and the behavior at nodes A and C are *identical*, we only present the results for the T_A states.

State Probabilities: Table 3.5 presents the values of π_i and ρ_i . To recall, π_i represents the proportion of *transitions* entering state i , while ρ_i reflects the proportion of *time* spent in state i . We see that the analytical results are quite close to those of simulation. We also notice that, though the sum of π_i is quite small (about 0.24), the sum of ρ_i is quite large (0.496). Since the T_C states also have the *same* values, the total fraction amount of *time* spent in T states is about 0.992, which implies that only a very small amount of time is spent in the large number of *Col* states. The reason is that the ratio (i.e., r) between the ‘*FES time*’ and the ‘*Col time*’

is very large. From here, we can conclude that, though the hidden-terminal problem leads to considerable collisions, it does not result in substantial capacity wastage if the *short* RTS/CTS frames are used before the *long* Data frame is transmitted.

Table 3.5: State Probabilities Comparison

From		$(T_A, 0, 0)$	$(T_A, 0, 1)$	$(T_A, 0, 2)$	$(T_A, 0, 3)$	$(T_A, 0, 4)$	$(T_A, 0, 5)$	$(T_A, 0, 6)$	Total
π_i	Model	0.025	0.031	0.037	0.038	0.037	0.035	0.032	0.236
	Simulation	0.025	0.038	0.039	0.039	0.037	0.034	0.031	0.243
ρ_i	Model	0.008	0.014	0.025	0.046	0.088	0.165	0.150	0.496
	Simulation	0.008	0.015	0.023	0.045	0.088	0.167	0.150	0.496

Expected State Holding Time: Table 3.6 presents the results of $num(i)$, which denotes the average number of packets node A can transmit consecutively once the system enters a given T_A state (i.e., state i). We see that the results match very closely. As the stage at the waiting node (i.e., node C) increases, the $num(i)$ also increases, which indicates that it becomes more unfair for node C.

Table 3.6: Expected Holding Time Comparison

$num(i)$	$(T_A, 0, 0)$	$(T_A, 0, 1)$	$(T_A, 0, 2)$	$(T_A, 0, 3)$	$(T_A, 0, 4)$	$(T_A, 0, 5)$	$(T_A, 0, 6)$
Model	1.006	1.450	2.143	3.830	7.496	14.858	14.893
Simulation	1.002	1.206	1.829	3.616	7.461	15.336	15.348

Expected First Passage Time: Table 3.7 presents the results of V_i , which represents, if the system starts in a given T_A state (i.e., state i), what is the expected number of FESs after which the system will enter any T_C state for the first time. Again, the results obtained from the model are quite close to those from simulation. When the stage at the waiting node (i.e., node C) increases (from 0 to 4), the V_i also increases, which implies that it becomes more unfair for node C. However, when the stage at the node C further increases (i.e., from 4 to 5, and then 6), the V_i decreases. First, let us explain why the V_i corresponding to $(T_A, 0, 6)$ state is small. We recall that V_i is equal to the immediate reward (i.e., $num(i)$) plus the expected reward earned from whatever state is entered next. When the system departs from the $(T_A, 0, 6)$ state, the system enters the state $(Col, 1, 0)$ where the stage at node C has been reset. From this Col state, the system is more likely to enter a T_C state, in comparison to, from other Col states. For instance, when the system transits to a Col state from the $(T_A, 0, 5)$ state, the stage at

node C will not be reset, and the probability of transiting to a T_C state is small. This implies that after leaving $(T_A, 0, 6)$ state, the expected reward earned from the future states is smaller in comparison to that after leaving the state $(T_A, 0, 5)$. Therefore, the V_i corresponding to $(T_A, 0, 6)$ is small compared to the $(T_A, 0, 5)$ state. Now, we explain why the V_i corresponding to the $(T_A, 0, 5)$ state is smaller than that for the $(T_A, 0, 4)$ state. Since the state $(T_A, 0, 6)$ is a *future* state of $(T_A, 0, 5)$, a small V_i for $(T_A, 0, 6)$ will also affect the V_i for the $(T_A, 0, 5)$ state. However, the effect of the reset behavior decreases rapidly as the stage at the waiting node becomes smaller than 4. From the above discussion, it is clear that *resetting CW mechanism* adopted in the IEEE 802.11 improves the short-term fairness. This is also consistent with the statement made in Chapter 2 that a smaller retry-limit leads to better fairness.

To improve the short-term fairness, in addition to the standard resetting mechanism, we suggest that the CW should also be reset whenever the short-term unfairness occurs. In fact, as we show in the Adaptive Transmission Control (ATC) fairness scheme presented in Appendix A, such a mechanism (called early-reset) does improve the short-term fairness performance substantially.

Table 3.7: Expected First Passage Time Comparison

V_i	$(T_A, 0, 0)$	$(T_A, 0, 1)$	$(T_A, 0, 2)$	$(T_A, 0, 3)$	$(T_A, 0, 4)$	$(T_A, 0, 5)$	$(T_A, 0, 6)$
Model	11.796	25.416	31.081	34.059	34.404	30.273	17.661
Simulation	10.413	23.584	29.998	34.031	34.896	31.430	18.555

Metrics: Table 3.8 presents the values of the two metrics defined earlier. We also present the corresponding maximum values. Again, the analytical results match the simulation results. Note that the values of the two metrics *differ* largely. The reason is that whenever the system departs from a T_A state, the probability that the system enters a Col state is very large (see Table 3.3). After the collision(s), the system is more likely to enter another T_A state (rather than a T_C state) as the CW at node A is smaller than that at node C. This may be repeated several times, resulting in such a large difference.

General Applications of the Model: The above results correspond to the case when $CW_{min}=31$, $Len=19$, $m' = 5$, and $m = 6$. By varying parameters m' , m , CW_{min} , and Len , the *short-term* behavior for other scenarios can be obtained. For example, by making

Table 3.8: Comparison of the Metrics

From	metric-1	Max metric-1	metric-2	Max metric-2
Model	6.683	NA	27.379	NA
Simulation	6.413	35	27.090	160

$Len = TxTime(Data) + SIFS$, we can model the *two-way* handshake in the presence of hidden terminals. On the other hand, by making $Len=0$, we can model *anyone* of the two handshakes *without* hidden terminals. By varying m' , m , and CW_{min} , we can model different physical layers, such as FSSS, DSSS and IR [24]. Moreover, since the model can predict the *short-term* behavior *precisely*, it would also predict the *long-term* behavior accurately.

3.7 Conclusions

In this chapter, we have presented an analytical model that is *extremely accurate* in predicting the short-term unfairness of IEEE 802.11 in the hidden-terminal scenario. To the best of our knowledge, no such analytical model is available in the literature. Our model is mainly based on the evaluation of the *first passage time* and *expected state holding time*. From our results, we arrive at the following several conclusions: (1) IEEE 802.11 exhibits substantial short-term unfairness in presence of hidden terminals; (2) resolving collisions and achieving short-term fairness are contradictory objectives when using IEEE 802.11 in a hidden terminal scenario; (3) though the hidden-terminal problem leads to considerable collisions, it does not result in substantial capacity wastage if the four-way handshake is adopted; and (4) the resetting of the contention window improves the short-term fairness.

Chapter 4

FMAC/CSR: Fair MAC with Cooperation between Sender and Receiver

In the previous two chapters, we have analytically shown the unfairness in the single-hop and hidden-terminal scenarios under IEEE 802.11. In this chapter, we systematically address the unfairness issue in the networks where spatial-reuse is not possible, and propose a MAC scheme, FMAC/CSR, to achieve fairness in IEEE 802.11. We also introduce another MAC scheme, Adaptive Transmission Control (ATC), which requires less modifications to IEEE 802.11 than FMAC/CSR does. However, the fairness under ATC is not as good as that under FMAC/CSR. The details of ATC are presented in Appendix A.

4.1 Introduction

In the literature, the fairness issue in wireless ad-hoc networks has been addressed in two different manners. One is to design a scheduler (e.g., [42]), which is overlaid on the top of the MAC layer, to achieve a global fairness in a *multi-hop* wireless ad-hoc network with spatial reuse. Obviously, this approach requires some global information of the network (e.g., topology). The other manner is to address the fairness issue at the MAC layer itself (e.g., [43]), which operates in a distributed manner but may only achieve a local fairness in a neighborhood where spatial-reuse is not possible. In fact, the fair scheduling in a multi-hop network with spatial-reuse and the fair medium access in a neighborhood without spatial-reuse are quite different issues and may need to be addressed separately.

In this thesis, we address the issue of fairness at the MAC layer for the network without spatial-reuse. Most of the published schemes in this category are sender-based, meaning that a sender of a flow¹ contends for the shared medium based on its own understanding of the contention on the medium. However, as clearly shown in this chapter, the senders of the flows that contend for the common medium may not be in the hearing range of each other, and thus the information obtained at the senders may not reflect the medium contention precisely. We call it *concealed information problem*, since the information needed for the fair contention is *concealed* from the senders. Our results show that the concealed information problem results in substantial unfairness in IEEE 802.11 as it is a sender-centric protocol. Moreover, due to this problem, any enhanced fair scheme (e.g., [18]) that is purely based on the information at the sender may not achieve fairness or sometimes it may even degrade the performance due to the wrong decisions made after the incorrect understanding of the contention. In addition to the concealed information problem, we have identified several other causes of unfairness in IEEE 802.11: imprecise collision detection, and freezing mechanism and randomness in the binary exponential back-off (BEB).

To achieve fairness, we propose a general MAC fairness framework, which includes three components: a fairness model to define fair shares, a compensation model to compensate for the over-use and under-use with respect to the fair shares, and a distributed algorithm to realize the above two models. Clearly, the last component is the key to achieve fairness due to the distributed nature of the ad-hoc networks. The distributed algorithm proposed in this chapter is called FMAC/CSR, which achieves fairness using the *Cooperation* between the *Sender* and *Receiver*. We also introduce here another distributed algorithm, called Adaptive Transmission Control (ATC). In FMAC/CSR and ATC, the sender as well as the receiver of a flow (say, x), in a *distributed* manner, estimate the number of *active* flows (say n) within the contention range, and the actual bandwidth share (say w_x) received by the flow. The estimate of n is a good indication of the contention degree, and therefore, it can be used by the flow to dynamically determine its fair share (say ϕ_x). Due to the unfairness in IEEE 802.11, the actual share (i.e., w_x) may deviate from the fair share (i.e., ϕ_x). Based on the deviation, a flow detects

¹Note that all the flows seen from the MAC layer are *single-hop* as mentioned in Chapter 1.

the unfairness, and thus determines which mode it should enter into out of the three modes: *aggressive*, *restrictive*, and *normal*, indicating how the sender and receiver of the flow should behave when contending for the medium. FMAC/CSR and ATC differ from each other in the sense that the behavior of the sender as well as the receiver under the two schemes is different in a given contention mode.

In this chapter, we discuss how the sender and receiver under FMAC/CSR behave in different contention modes, while the behavior under ATC will be discussed in Appendix A. Specifically, in FMAC/CSR, according to the mode a flow enters into, the sender contends for the medium with different priorities. On the other hand, the receiver affects the sender's behavior in the contention through either implicit or explicit feedback to the sender. Specifically, an implicit feedback, called *restrictive-notification*, is used by the receiver to slow down the sender whenever it finds (based on its own understanding) that the flow should enter the restrictive mode. On the other hand, whenever the receiver finds that the flow should enter the aggressive mode and now it is the right time for the sender to contend for the medium, an explicit feedback, called *aggressive-notification*, may be transmitted so as to expedite the sender's contention for the medium. However, when the receiver thinks that the flow has got a fair share of the medium (and thus the flow should enter the normal mode), no special action is taken by the receiver. Clearly, in FMAC/CSR, the sender of a flow even now plays a more active role than the receiver does.

In the design of FMAC/CSR, two issues need very careful consideration. (i) Due to the concealed information problem, the decision (e.g., the mode that a flow should enter) made by the sender may be different from that made by the receiver since they make the decision based on their own understanding. Therefore, rules must be defined that the sender and receiver should follow whenever such a discrepancy occurs. (ii) Benefits of cooperation among flows may be undermined due to the incorrect information at the nodes, and therefore, the events such as medium being idle, collisions, unnecessary explicit feedback, and deadlock that result in throughput degradation need much careful consideration.

Extensive simulation results show that the FMAC/CSR achieves fairness in all the scenarios where spatial-reuse is not possible. Meanwhile, it does not unduly degrade the throughput, and

it even improves the throughput in certain scenarios. These properties make our FMAC/CSR unique compared to the large number of published papers on MAC fairness.

The remainder of the chapter is organized as follows. The causes of unfairness are systematically identified in Section 4.2. In Section 4.3, we present a general MAC fairness framework. A distributed estimation algorithm and the FMAC/CSR are presented in sections 4.4 and 4.5, respectively. The simulation results are presented in Section 4.6. Comments on FMAC/CSR and a thorough literature review on fairness are given in Section 4.7. The chapter is concluded in Section 4.8.

4.2 Unfairness in IEEE 802.11

Though there has been a substantial research-work on MAC fairness, the causes of unfairness have not been identified very clearly. In this section, we systematically identify the causes of unfairness in IEEE 802.11. To emphasize, as our focus is to study the fairness issue at the *MAC layer*, we only consider the scenarios in which spatial reuse is not possible, i.e., all the flows in the network are contending with each other. Moreover, all the flows are single-hop. For each flow, a Constant Bit Rate (CBR) traffic is adopted and the traffic rate is large enough for a single flow to occupy the entire channel capacity as unfairness occurs only when the system is over-loaded.

4.2.1 Unfairness Due to Concealed Information Problem

Since the CSMA/CA-based MAC protocols (e.g., IEEE 802.11) normally adopt handshakes, when a flow contends for the medium, the contention is at both the sender and the receiver. Therefore, two flows are contending with each other if either the sender or the receiver of one flow is within the transmission range² of the sender or the receiver of the other flow. In this subsection, we aim to identify all the possible scenarios, which have two contending flows and exhibit short-term or long-term unfairness. The discussion also applies when more than two flows are involved.

For convenience, we define $F(X, Y)$ as a single-hop flow between nodes X and Y , where node X is the sender and node Y is the receiver. Let us consider the situation that two flows, say $F(S_A, R_A)$ and $F(S_B, R_B)$, contend for a common medium. We first assume that S_A, R_A ,

²It should be the sensing range if the sensing range is greater than the transmission range.

S_B , R_B are different nodes. Based on whether S_A is in the range of S_B and R_B , we present the relationship between node S_A and *flow* B in Figure 4.1. In the figure, a dotted line connecting two nodes indicates that these two nodes are in the range of each other. It is easy to see that there are four possibilities as far as the relationship between node S_A and flow B is concerned. This is also true for the relationship between node R_A and *flow* B. As a result, the number of possible scenarios having different relationships between *flows* A and B is sixteen (4×4). However, in the scenario that both S_A and R_A are not in the range of any side of flow B, flows A and B do not contend with each other any more. Therefore, the number of possible scenarios reduces to fifteen. Moreover, as known from our simulation (though not presented in the thesis), in the scenarios that the *senders* are in the range of each other (the cases labelled as (2) and (4) in Figure 4.1), the flows can contend very fairly (except the two scenarios discussed in sections 4.2.2 and 4.2.3). There are eight (2×4) scenarios where the two *senders* are within range of each other. As a result, seven ($15 - 8$) scenarios remains to be studied for unfairness.

Based on the relationship between the senders and the receivers belonging to different flows, the seven scenarios can be further classified into three categories. In the *first* category, neither of the senders is within the range of the receiver of the other flow. The scenario shown in Figure 4.2 belongs to this category. The *second* category is that only one of the senders is within the range of the receiver of the other flow. The two scenarios shown in Figure 4.3 where S_B is within the range of the R_A belong to such a category. Note that two other scenarios, where S_A is within the range of the R_B , also belong this category. However, we do not show them in the figure as they have the same problem as those in Figure 4.3. In the *third* category, both the senders are within the range of the receiver of the other flow. The scenario-4 and -5 in Figure 4.4 belong to this category.

Figure 4.1: Relationship between Node S_A and Flow B

In the above, we have assumed that S_A , R_A , S_B , R_B are different nodes. However, if the two flows just involve *three* different nodes, the only scenario where the senders are not within

range of each other is scenario-6 shown in Figure 4.4, which is also known as the hidden-terminal scenario. Obviously, if the two flows just involve *two* nodes, the two senders are always within the range of each other. We do not consider these scenarios since the fairness can simply be achieved by using fair queuing algorithms developed for the wire-line networks.

Now, we will show how the unfairness takes place in the above scenarios (figures 4.2, 4.3, and 4.4).

Short-term Unfairness in Scenario-1: In Figure 4.2, the distance between the senders of the two flows is *three* hops, which is in fact the *maximum* distance between two *senders* if they are contending for the same medium. Now, we explain the reasons of the unfairness in this scenario (the simulation results are presented in Section 4.6.3). Consider the situation that flow A is in progress and thus node R_B is not able to respond to any request from node S_B . However, since node S_B does not know about the ongoing transmission of flow A, S_B may *futilely* retry, resulting in a large CW at the node. As for the node S_A , after it transmits the packet, it will reset its CW and contend for the medium again. Since the CW at node S_B becomes very large, node S_A may transmit several packets *consecutively* before node S_B gets control of the medium, leading to short-term unfairness. However, several mechanisms incorporated in IEEE 802.11 prevent flow B from starving *completely*, such as: (i) after every packet transmission, node S_A will back-off before initiating another transmission, which gives node S_B a chance to contend for the medium; (ii) the CW at node S_B will be reset to CW_{min} after the retry limit is reached. Moreover, once node S_B gets control of the medium, it will capture the medium for a long time in a similar manner. As a result, the long-term fairness between these two flows are ensured in IEEE 802.11³.

Figure 4.2: Category 1: Scenario-1

Long-term Unfairness in Asymmetrical Information Scenarios (Figure 4.3): In this category, the sender, who is in the range of the receiver of the other flow, can get more information of the contention than the other sender does. This has been referred as the asymmetrical

³According to the results presented in [6], in the MACAW protocol, flow B will be completely starved in such a scenario.

information problem [36], which results in substantial long-term unfairness as explained below. Whenever S_A has completed transmission of a packet, both S_A and S_B begin to contend for the medium. In such a situation, the only condition that S_A can win is that R_A responds with a CTS before S_B begins to send out a RTS. Otherwise, S_B will definitely win the contention even it begins to send out the RTS later than S_A does. This is obviously unfair for flow A. Now, let us consider the situation that flow B is transmitting a packet. Node S_A cannot even identify when the transmission of the packet will be end. Therefore, during the period that S_B is transmitting, S_A will contend *futilely*, resulting in a large CW and thus leading to unfairness for S_A . In summary, after every transmission (transmitted either by flow A or B), S_A will always be treated unfairly, resulting in long-term unfairness. Note that though the receivers in the scenario-2 and -3 have different information about the contention, the two scenarios will show the same performance since the receiver in IEEE 802.11 does not play an active role in the contention.

Figure 4.3: Category 2: Asymmetrical Information Scenarios

Short-term Unfairness in Hidden-terminal Scenarios: In Chapter 3, we have already shown that IEEE 802.11 exhibits substantial short-term unfairness in the hidden-terminal scenario while the long-term fairness is ensured. Again, the three scenarios of Figure 4.4 show the same performance though the receivers have different degrees of information of the contention on the medium.

Figure 4.4: Category 3: Hidden-terminal Scenarios

Concealed Information Problem: From all the above scenarios, we notice that the main problem leading to unfairness in the CSMA/CA protocols is that the senders cannot obtain

precise information about the contention on the medium. In other words, the state information required for fair contention is *concealed* from the sender, and therefore we call it the *concealed information problem*. Here, the state information a sender needs is: when exactly the *contention period* occurs on the shared medium. A contention period *starts* after the transmission of an ACK frame on the medium, and *ends* when the transmission of a RTS frame begins on the medium.

In scenario-1 (Figure 4.2), after one flow finishes the transmission of a packet (i.e., the receiver of the flow sends an ACK frame), the sender of the other flow can identify *neither* the beginning *nor* the ending of the contention period. In the hidden-terminal topologies (Figure 4.4), though the senders can identify the beginning of the contention period (i.e., whenever an ACK frame is transmitted on the medium, both the senders know this event), they cannot identify the ending of the contention period (i.e., a sender does not know whether the other sender has already begun to transmit a RTS). In the asymmetrical information scenarios (Figure 4.3), whenever flow B transmits a packet on the medium, S_A can identify *neither* the beginning *nor* the ending of the contention period. Conversely, whenever flow A transmits a packet, both S_A and S_B cannot identify the ending of the contention period since they are hidden from each other.

4.2.2 Long-term Unfairness Due to Imprecise Collision Detection Mechanism

In this subsection, we will show that the collision detection mechanism used in IEEE 802.11 results in long-term unfairness even when the senders are in the range of each other. Figure 4.5 presents such a scenario. Our simulation results (presented in Section 4.6.4) show that flow B gets more bandwidth share than flow A. Let us consider the situation that nodes S_A and S_B choose the same back-off value before contending. Then, each of them initiates a RTS at the same time, resulting in a collision. Since node R_A is in the range of node S_B , R_A detects this collision and thus it will not send a CTS to S_A . On the contrary, since node R_B is out of the range of S_A , R_B will *not* detect the collision and therefore it will respond with a CTS to S_B . In summary, whenever there is a collision between the two RTSs, S_B *always* wins the contention, explaining the unfairness for flow A.

Figure 4.5: Scenario-7: Imprecise Collision Detection Mechanism

4.2.3 Short-term Unfairness due to Deficiencies of BEB

Due to the deficiencies of the BEB, the short-term unfairness may occur even when all the senders are within the range of each other and all the nodes have the same understanding when a collision occurs.

As analytically shown in Chapter 2, the randomness involved in the BEB is one of such deficiencies causing unfairness when the network size is relatively large. In fact, we have also observed this from the simulation. For example, in the scenario shown in Figure 4.6, where there are six nodes and five flows⁴, we found that the system is able to deliver long-term fairness among the flows, but substantial short-term unfairness exists (the simulation results presented in Section 4.6.5).

In addition to the randomness of BEB, the freezing mechanism used in IEEE 802.11 is also a major reason of short-term unfairness. For example, if a node has generated a very large back-off counter, due to the freezing mechanism, another node may transmit many packets consecutively before this node's back-off counter is reduced to zero, leading to short-term unfairness. This have been clearly shown in the hidden-terminal scenario discussed in Chapter 3.

Figure 4.6: Scenario-8: High Contention (all nodes hear each other)

⁴Of course, one can choose a *higher* number of flows to show the short-term unfairness in a high contention scenario.

4.2.4 Remarks on the MAC Fairness

So far, we have systematically identified the possible scenarios that show unfairness due to concealed information problem (figures 4.2, 4.3, and 4.4), imprecise collision detection mechanism (Figure 4.5), or deficiency of BEB (Figure 4.6). Now, we would make four important remarks on the fairness issue.

Remark 1: A *flow-contention graph* technique that has been popularly used in the literature is not an appropriate means for the study of MAC fairness. It is easy to see that all the scenarios with two contending flows discussed above will have the same flow-contention graph (Figure 4.7), but totally different performance results are obtained for these scenarios. Therefore, any fairness scheme (e.g., [17, 43]) based on such a graph may not work very well.

Figure 4.7: Flow Contention Graph for Scenarios with two Contending Flows

Remark 2: In contrast to a purely sender-based protocol (e.g., IEEE 802.11), if we were to design a *purely receiver-based* algorithm, it would suffer from the similar problems because the two *receivers* may not be within the range of each other when two flows are contending with each other. Therefore, to achieve fairness, the protocol should employ some sort of cooperation between sender and receiver.

Remark 3: In a given scenario, whether short-term or long-term unfairness will occur largely depends upon the topology. When the topology involves asymmetry, long-term unfairness occurs. Conversely, in a symmetrical topology, the short-term unfairness occurs while the long-term fairness is ensured.

Remark 4: In the above, using specific scenarios, we have identified several problems that cause unfairness in IEEE 802.11. However, we should note that these problems are quite general and ingrained in a large number of scenarios. Also, though in each scenario we have explained only one main problem that causes unfairness, it is common that a single scenario involves more than one problem. For example, in the concealed information scenarios, the randomness and the freezing mechanism also contribute to the unfairness.

4.2.5 Impact of Unfairness

While the impact of long-term unfairness is obvious, as shown in [40, 60], the short-term unfairness at MAC layer may lead to a large delay jitter, and thus it substantially degrades the performance of the delay- or jitter-sensitive traffic (such as video and audio). The short-term unfairness also has a great impact on the adaptive traffic (e.g., TCP kind of traffic). Let us consider an example that there is a TCP flow from node S_A to S_B in the hidden-terminal scenario (see scenario-6 in Figure 4.4). From the simulation trace, we found that the TCP Data packets (from S_A to S_B) and the TCP Ack packets (from S_B to S_A) will always be transmitted in *bursts*. That is, in the steady-state, node S_A sends as many TCP Data packets as permitted by the congestion window. Then, node S_B transmits all the TCP Ack packets in a burst back to S_A . Through detailed analysis of the trace records, we found that this is mainly due to the short-term MAC unfairness between nodes S_A and S_B , i.e., once a node (say S_A) gets control of the medium, it may capture the medium for a long time before the other node (i.e., node S_B) transmits any packet. According to [40], the above phenomenon has also been observed in the wireless LAN (e.g., WaveLAN), and has been referred as *ACK compression*. The ACK compression results in packet loss at the congested nodes. Also, the link may be idle for some duration between the bursts of TCP Data and ACK packets.

In addition to the negative effects on jitter and TCP performance, the short-term unfairness may also greatly affect the behavior of on-demand routing protocols. For example, if a node cannot transmit a packet after several retries due to the short-term unfairness, the MAC layer discards the packet and conveys this event to the routing protocol. As the routing protocol (e.g., AODV) interprets the discarding of a packet as an indication of link-breakage, the node discards all the packets in the queue and initiates a new route discovery process, though the link is still available. Even worse, in such a situation, the new route discovery process is difficult to be successful due to the unreliable nature of broadcast packets in IEEE 802.11 and due to the exponential back-off in the route discovery process itself. Therefore, a flow may be starved for a considerable time. In fact, the above phenomenon has been observed very frequently in our simulation when AODV is used. It is clear that it is extremely important to achieve both short-term and long-term fairness at the MAC layer.

4.3 General Framework to Achieve MAC Fairness

In order to achieve MAC fairness in general, we propose a framework, which includes the following three components. *Firstly*, we should define a fairness model, which determines how much share a user should get to maintain fairness, and how long the duration be over which the fairness is measured. *Secondly*, since the actual share may deviate from its fair share, a compensation model is needed to compensate the share of the user based on its past usage of the medium. *Thirdly*, we should have a distributed algorithm in place, which tries to realize the above two models, and thus achieves fairness.

4.3.1 Fairness Model

The objective of a fairness algorithm is to ensure that a *user* gets a fair share (say, ϕ_x for user x) of the bandwidth over every certain duration (say T_{cyc}). Based on whether the *flows* or *nodes* are defined as *users*, *flow-level* or *node-level* fairness is desired. Since a node may have multiple flows associated with it, the flow-level fairness has finer granularity than the node-level fairness, and therefore we consider flow-level fairness here. The value of T_{cyc} determines the duration over which the fairness is desired. For example, if T_{cyc} is large, the algorithm is aiming to achieve the long-term fairness, which however does not necessarily imply short-term fairness. On the other hand, if T_{cyc} is small, the objective is to achieve the short-term fairness, which automatically gives rise to the long-term fairness. We aim to provide short-term fairness (implying long-term fairness). If the fairness is defined as the equal share of the bandwidth among n active⁵ flows (obviously n varies with time), it is reasonable to assign T_{cyc} with the time required to transmit n packets. Moreover, if all the packets are assumed to have the same length, the T_{cyc} can simply be replaced by a *transmission window* (say W_{cyc}), which is equal to n . Obviously, T_{cyc} or W_{cyc} is more complex to define if the fairness model needs to provide differentiated service among the active flows. Moreover, how to decide the fair share (ϕ_x for flow x) is in itself an issue. At a *higher* layer, the value of ϕ_x should be determined based on the application requirement. However, as our focus is to achieve fairness in the contention for the shared medium, we simply assume that all the applications have the same requirements.

⁵A flow is referred to be *active* whenever it has at least one packet waiting to send.

Therefore, at the MAC layer, it is reasonable to assign ϕ_x with $1/n$ when n flows are active, implying that every *active* flow should transmit *exactly* one packet whenever n packets are transmitted over the medium. In summary, this can be formalized as,

$$\begin{cases} W_{cyc} = n \\ \phi_x = 1/n \end{cases} \quad (4.1)$$

W_{cyc} and ϕ_x , defined as above, change with the contention degree (i.e., n), rather than being pre-defined as in most of the literature (e.g., [18]), and thus are very *adaptive* to the *dynamic* network conditions.

4.3.2 Compensation Model

If the actual share (say, w_x for flow x) is always equal to the fair share ϕ_x during every window W_{cyc} , a system following the above fairness model behaves like a *dynamic* TDMA protocol, which has desirable fairness. By *dynamic*, we mean that the number of flows sharing the common medium is dynamic and that the transmission order among the flows from one cycle to another is also dynamic. However, in IEEE 802.11, w_x may deviate from ϕ_x , especially in the wireless ad-hoc networks due to the unfairness discussed in Section 4.2. In particular, during a given window of length W_{cyc} , a flow may transmit more than one packet. We refer to this as the *over-use* of the medium by the flow. On the other hand, if a flow does not transmit any packet during the window, we call it *under-use*. Naturally, *normal-use* refers to the case when a flow transmits *exactly* one packet in the window. To compensate for the over-use and under-use in the previous window, and thus make w_x as close to ϕ_x as possible, an active flow should adjust its rate as early as possible. Specifically, from the contention viewpoint, an active flow should be in one of the three modes: *aggressive*, *restrictive*, and *normal* at any given time. If a flow has *under-used* during the previous window, it should give itself more opportunity in contending for the medium, and thus should enter the aggressive mode. On the other hand, if a flow has *over-used* the medium, it should enter the restrictive mode. However, if a flow gets its *fair* share, it should operate in the normal mode. Note that in the above compensation model we have not specified how much a flow should be compensated in the restrictive and aggressive modes, since that depends on the MAC protocol used as well as on the implementation.

Clearly, the above fairness and compensation models are based on the knowledge of n (the number of active flows) and w_x (the actual share of flow x). However, in ad-hoc networks, the sender as well as receiver of a flow has to *estimate* these two values in a *distributed* manner. Moreover, due to the distributed nature of ad-hoc networks, how to design a medium contention algorithm that can achieve fairness by utilizing the estimated information is a demanding challenge. The two issues are discussed in sections 4.4 and 4.5, respectively.

4.4 Estimation Algorithm

We first present a simple observation in IEEE 802.11. Whenever a flow transmits a *packet* over the medium, the *other* nodes within the contention range can generally overhear the RTS/Data, the CTS/ACK, or all the four *frames*⁶. For example, in the scenario of Figure 4.8, whenever flow A transmits a packet, node S_B can overhear the RTS/Data frames. In the hidden-terminal scenario (Figure 4.4), node S_B can overhear the CTS/ACK frames transmitted by flow A. In the scenario of Figure 4.6, where all the nodes are within one-hop, every node can overhear all the four frames. However, the above observation about overhearing is not always true. For example, in scenario-1 (Figure 4.2), whenever flow A transmits a packet, node S_B *cannot* overhear any of the four frames. This is also true for node S_A in the asymmetrical information scenarios (Figure 4.3) when flow B transmits a packet. For simplicity, we ignore these violation cases until to Section 4.5.2. Also, we assume that the sensing range is equal to the transmission range, and we will come back to this point in Section 4.7.1.

Figure 4.8: Scenario where Senders Overhear RTS/Data

4.4.1 Estimation of the Number of Active Flows (n)

In order to estimate the number of active flows within the contention range, every node maintains an ID list of the active flows. Whenever a node hears/overhears a frame (RTS/CTS/Data/ACK)⁷,

⁶The word ‘*packet*’ implies the protocol data unit (PDU) of a higher layer whereas ‘*frame*’ is the MAC layer PDU.

⁷Note that the correctness of the estimation does not rely on the overhearing of the Data/ACK frames.

it inserts the corresponding *flow ID*⁸ into the list if the ID does not exist in the list. In the case that the ID exists, the node simply refreshes the time of the entry containing this ID. With the above list, every node can easily get the number of active flows by counting the number of *unique* IDs in its list. Clearly, in such an algorithm, it is crucial to prevent stale entries, and therefore an entry is deleted after a timeout interval that is long enough to transmit a certain number (say W_e) of packets. Now, we discuss how to *adaptively* determine the value of W_e . It is intuitive that, to get the current estimate (let us say n_e) of n , the W_e must be at least equal to the previous estimate (say n'_e) of n , therefore,

$$\begin{cases} W_e \geq n'_e \\ n_e = \text{the number of unique IDs} \end{cases} \quad (4.2)$$

Clearly, the value of W_e affects the precision of n_e . Sometimes, n_e is greater than n , and we call this as an *over-estimation*. In contrast, the *under-estimation* refers to the case, where n_e is smaller than n . Both over-estimation and under-estimation may lead to bandwidth wastage. Specifically, when over-estimation occurs, the flows are restrictive and thus more slots are *idle*. On the other hand, when under-estimation occurs, the flows are aggressive and thus increasing the likelihood of *collision*. Now, we consider three situations involving the imprecise estimation. (i) When a flow becomes active, the other nodes are *not* aware of this, until the flow transmits its *first* frame on the medium. This results in the under-estimation; (ii) After an active flow transmits its *last* packet in the queue and thus becomes inactive, its ID will *not* be deleted by the other nodes until a timeout interval long enough to transmit W_e number of packets. This results in the over-estimation; (iii) Due to short-term unfairness, an active flow may not be able to initiate any transmission in a given window W_e . As a result, though the flow is active, other nodes delete the corresponding ID, leading to under-estimation.

In our algorithm, whenever a flow becomes active, the flow enters the aggressive mode and thus transmits its first RTS immediately. Therefore, the under-estimation of the first case, if occurring, will last for a very short duration. In order to cope with the over-estimation problem in the second case, we should use a value of W_e as *small* as possible. On the other hand, a large value of W_e should be used to cope with the under-estimation problem in the third case.

⁸If we aim to achieve the node-level fairness for the senders as in IEEE 802.11 or if we assume that a node can act as the sender at the most for one flow, we can simply use the sender ID as the flow ID.

To cope with this dilemma, we propose *inactive-notification mechanism*. Specifically, whenever a flow sends out its *last* packet in its queue (i.e., switching from active to inactive), the sender of the flow uses a bit in the RTS/Data to tell other nodes that the flow is becoming inactive. Moreover, the receiver also piggybacks this notification in the *responding* CTS/ACK frames, as some nodes may not overhear the RTS/Data. All the nodes that hear the notification, rather than inserting or refreshing the flow ID, should *delete* this flow ID from the list. Clearly, the inactive-notification mechanism greatly solves the over-estimation problem. Therefore, we can use a large W_e value to avoid the under-estimation problem. In the simulations, we found that the following values of W_e are better suited:

$$W_e = \begin{cases} 6 \times n'_e & \text{when } n'_e \leq 10 \\ 4 \times n'_e & \text{when } n'_e > 10 \end{cases} . \quad (4.3)$$

The above carefully designed timeout mechanism greatly prevents the imprecise estimation due to the following reasons: delayed transmission of the first frame of an active flow, stale entries, and short-term unfairness. However, it cannot solve the imprecise estimation due to the *absence* of overheard information in the violation cases (e.g., scenarios in figures 4.2 and 4.3). We will propose two novel algorithms in Section 4.5 to cope with this problem.

4.4.2 Estimation of the Actual Share (w_x)

We now discuss how to estimate w_x , the actual proportion shared by flow x during the previous window W_{cyc} . This can be done by maintaining a transmission history at *each* node. Note that this transmission history is totally different from the ID list maintained for the estimation of n . We have intentionally designed the estimation algorithm in this manner so that it also applies when the sensing range is greater than the transmission range as discussed later in Section 4.7.1. Now, we discuss how to update the transmission history. Whenever a node hears/overhears a Data or ACK frame transmitted over the medium, the node will *append* the flow ID into its history. If a node overhears both the Data and ACK frames belonging to the same *handshaking* for a single packet, it should add the ID only *once*. By maintaining such a history, a sender or a receiver can easily know its flow's actual share w_x during the latest window by simply checking how many times (say l) its *own* ID appears in the window W_{cyc} . Therefore,

$$Estimation(w_x) = l/n. \quad (4.4)$$

As an example, let us consider the history (beginning with the most recent entry), which is as follows,

$$\{A, B, A, C, B, A, D, E, C, \dots\}$$

where A, B, C, D , and E are the flow IDs. Assuming that the n (estimated from a separate ID list) is equal to 5, then the estimated actual shares w_x for flows A, B, C, D , and E are $2/5, 2/5, 1/5, 0$, and 0 , respectively, since we need only look at the most recent transmission window (with *five* entries), i.e., $\{A, B, A, C, B\}$. In fact, in estimating w_x at the sender and receiver of flow x , the two nodes do not need to know the *exact* flow IDs of the packets transmitted by *other* flows. For example, when estimating w_A at S_A and R_A , the history $\{A, B, A, C, B\}$, if replaced by $\{A, -1, A, -1, -1\}$, will still work. Obviously, a node always knows the flow ID of a packet if the node is the sender or receiver of the flow. The above properties are very important when the Data/ACK is not interpretable to the overhearing nodes due to collisions or a large sensing range as discussed in Section 4.7.1.

4.4.3 Detection of Unfairness

We now discuss how to detect the unfairness based on the deviation of a flow's actual share from its fair share. Let us again consider the example above (with transmission history $\{A, B, A, C, B, A, D, E, C, \dots\}$). Since the fair share (ϕ_x) for each flow should be $1/5$, flows A and B have over-used the medium during the latest window (with five entries), while flow C has had the normal-use of the medium. Flows D and E , however, have under-used the medium.

Based on the history, a sender or receiver of a flow can also measure *how much* the flow has over-used or under-used the medium recently. With this measurement, a flow decides the *degree* by which it should be aggressive or restrictive, which is represented by N_a and N_r , respectively, and can be derived as follows. First, the flow (through sender or receiver) checks its actual share in the latest transmission window (with n entries). If it has had normal-use of the medium in the window, N_a and N_r are set equal to zero (e.g., at flow C in the above example). On the other hand, if the flow has under-used the medium in the window, N_a is initialized to *one*.

In this case, the flow will then check the next transmission window by sliding by one entry in the history (i.e., the window of n entries after skipping the most recent entry). If it still has under-used the medium during this window, N_a is incremented by one. This process continues until the flow finds its share becoming normal in a transmission window. In the example given above, N_a at flows D and E are equal to 2 and 3, respectively. Similarly, we can get the value of N_r if a flow has over-used the medium during the latest transmission window. For instance, N_r at flows A and B are equal to 3 and 2, respectively.

4.5 FMAC/CSR: Fair Medium Access Control using Cooperation between Sender and Receiver

Using the estimation algorithm proposed in the previous section, the sender as well as the receiver of a flow obtain the following information: the number of active flows n , the actual share (w_x) of the flow, the contention mode that the flow should enter, and the corresponding degree of aggressiveness/restrictiveness. Clearly, the estimated values at the sender and the receiver of a flow may be different due to imprecise estimation. In this section, we will introduce a novel medium contention algorithm, called FMAC/CSR, which exploits the information available at the sender and receiver to achieve fairness. In FMAC/CSR, a node will act according to whether it is the sender or the receiver of the flow. Specifically, a sender contends for the medium with different priorities according to the mode and the corresponding degree that the flow should use. On the other hand, the receiver affects the sender's behavior in the contention through either an implicit or an explicit feedback.

In the design of FMAC/CSR, two issues need very careful consideration. (i) As mentioned, the decision (e.g., the mode that a flow should enter) made by the sender may be different from that made by the receiver. Therefore, rules must be defined that the sender and receiver should follow whenever such a discrepancy occurs. (ii) Since cooperation (between the sender and receiver of a flow, as well as among different flows) based on imprecise information is greatly involved in FMAC/CSR, the events such as medium being unduly idle, collisions, unnecessary explicit feedback, and deadlocks that degrade the throughput should be carefully avoided. This is the key to achieve fairness without unduly degrading throughput.

4.5.1 Sender-based Differentiated Access

To compensate for the under or over usage of a flow, a sender should contend for the medium with different priorities. In general, this can be realized using two different approaches. The first approach is that the sender will not transmit until the flow has the highest priority to access the medium among the contending flows. This is a *deterministic* approach, which achieves the desirable fairness, but potentially leads to substantial throughput degrade [36]. The second approach is to assign different contention windows to the senders. This is a *probabilistic* approach, which achieves a worse fairness but has a better throughput performance [18]. In order to achieve fairness without unduly degrading the throughput, we use a hybrid approach of the above two as will be clear below.

Now, we define six rules that a sender should follow while contending for the medium whenever the medium becomes idle⁹. (1) If the flow should be in the aggressive mode, the sender will back-off by generating a random value from $[0, X]$ where $X = \max(n, 2n - N_a)$. (2) If a flow should be in the normal mode, the random back-off value will be generated from $[2n, CW]$, where CW is the contention window. (3) If the flow should be in the restrictive mode, the sender will first defer by a time equal to $(N_r + 1) \times T_{succ}$ where T_{succ} is the time needed to successfully transmit a *packet* as defined in Equation (2.10). Then, the sender will back-off by generating a random value from $[2n, CW \times N_r]$. (4) Irrespective of the mode that the flow is in, whenever the back-off timer expires and the medium is idle, the sender will transmit. (5) Whenever the medium becomes idle again after a busy state, the mode of the flow and the corresponding degree of aggressiveness/restrictiveness will be recomputed. Also, the back-off value will be regenerated. (6) The Contention Window (CW) is manipulated as in the Binary Exponential Back-off (BEB) of IEEE 802.11. The overall process followed at a sender is summarized in Figure 4.9.

Now, we make three comments on the above rules. (i) Among the flows that are in the *same* mode but have different degrees of aggressiveness/restrictiveness, a *probabilistic* approach is adopted. For example, as known from the rule 1, a flow having a larger aggressive degree N_a will have a larger probability to transmit. Conversely, as known from the rule 3, a flow having a

⁹In fact, the contention begins only if the medium has been idle for a duration equal to DIFS or EIFS [24].

larger restrictive degree N_r will have a smaller probability in accessing the medium. (ii) Among the flows that are in the *different* modes, a *deterministic* approach has been used. For example, as seen from rules 1 and 2, the flows that are in the aggressive mode will get higher priority than those in the normal mode since X must be smaller than $2n$. However, our scheme is not *fully* deterministic since a flow, irrespective of the mode it is in, may transmit whenever the back-off timer expires as indicated by rule 4. This is crucial to prevent the throughput degradation when cooperation between flows is involved. Consider that there are two flows, and due to imprecise estimation, both of them operate in the restrictive mode. If a fully deterministic method is used, both the senders will defer since they are gracefully waiting for the other sender to transmit. Therefore, the medium will be unduly idle or even worse, a deadlock will occur, resulting in substantial throughput degradation. (iii) The rule 5 is completely different from the freezing mechanism used in IEEE 802.11, which is one of the main reasons of unfairness as discussed in Section 4.2. On the other hand, as known from rule 6, the increasing and decreasing policy of the CW in the BEB algorithm is kept for its efficiency in resolving collisions.

Figure 4.9: Overall Process Followed at the Senders

4.5.2 Motivation to use Receiver Feedback

The above sender-based mechanism can achieve fairness if the estimated information at a sender is always correct. However, as a sender of a flow may not overhear any frames transmitted by

some other *contending* flows, two problems arise. The *first* one is that since the sender may get an incorrect estimation of the contention, it may not be aware of the situation whenever the flow has over-used/under-used the medium, and thus the compensation will not be performed. The *second* problem is that whenever one of the other contending flows transmits a packet over the medium, the given sender cannot identify the contention period, and thus it cannot contend efficiently. Due to the above two problems, a purely sender-based approach cannot always achieve fairness and sometimes it may even degrade the performance, as clearly shown by the following two examples.

In scenario-1 (Figure 4.2), since node S_A cannot overhear any of the frames transmitted by flow B, the n estimated at node S_A is always equal to one. Therefore, the S_A determines that the fair share for flow A should be one. Moreover, since the transmission history at node S_A will *never* contain flow B's ID, the actual share of flow A estimated at node S_A is also always equal to one. As a result, S_A will behave as if flow A were always in the *normal* mode. The above discussion also applies to S_B , and thus no compensation will be done. As a result, the sender-based algorithm in this topology will deliver the similar performance (e.g., substantial short-term unfairness) as that under IEEE 802.11. On the other hand, in the asymmetrical information topologies (see Figure 4.3), suppose that flow B has transmitted two or more packets consecutively, flow B will enter the restrictive mode and thus S_B will defer its transmission. However, since S_A cannot identify the contention period, it cannot contend efficiently. Specifically, when flow B is transmitting, S_A contends futilely, resulting in a large CW. However, after flow B finishes its transmission (i.e., a new contention period begins) and enters the restrictive mode, S_A will not take this opportunity to transmit since it is still deferring. Therefore, the medium will be *idle* for a *long* time and then flow B may get control of the medium *again*. Therefore, in this topology, the sender-based algorithm cannot achieve fairness and it even reduces the throughput of IEEE 802.11. All the above observations are confirmed by the simulation results in Section 4.6.

To solve the above problems, we recall that two flows are contending with each other only if either the sender or the receiver of one flow is within the range of the sender or the receiver of the other flow. Therefore, *in the case that the sender of a flow cannot overhear any frames*

transmitted by some other contending flows, the receiver must be able to overhear some frames transmitted by those contending flows. In light of this, we propose two general *receiver feedback* mechanisms. In particular, whenever a receiver, based on its own understanding, notices that the flow has *over-used* the medium in the latest transmission window (and thus the flow should enter the restrictive mode), the receiver will try to slow down the sender by sending out a restrictive-notification to the sender. On the other hand, when a receiver finds that an *active*¹⁰ flow has *under-used* the medium and now it is the right time (i.e., the beginning of a new contention period) for the sender to contend for the medium, the receiver may send out an aggressive-notification to expedite the sender's contention. However, when the receiver thinks that the flow has got a fair share, no special action is taken by the receiver. Clearly, in our FMAC/CSR, the sender still plays a more active role than the receiver does.

4.5.3 Receiver Feedback: Implicit Restrictive-notification

If a flow has over-used the medium, it must have just transmitted a packet. Therefore, the restrictive-notification can always be piggybacked in an ACK frame in an *implicit* manner. The remaining issues for this mechanism include: what information should be feedbacked, and what decision the sender should make after receiving the notification.

Regarding the issue of what the feedback information should be, there are several options: the number of active flows n , the actual share (w_x) of the flow, and the degree of restrictiveness N_r of the flow. Let us consider that the receiver feedbacks its estimated n to the sender. Therefore, a sender will have two estimates of n and it should choose the *maximum* one, since both the sender and the receiver may under-estimate n in different scenarios. Using this method, in the scenario-1 (Figure 4.2), n used at node S_A is equal to two, and S_A will determine that the fair share for flow A should be $1/2$, which reflects the real state of the contention. However, since the transmission history at S_A will *never* contain flow B's ID, S_A operates as if the flow was *always* in the *restrictive* mode. The same is true for S_B , leading to drastic throughput degrade. Therefore, the feedback of n may not be a good choice. We propose that the receiver should feedback to the sender with the degree of restrictiveness N_r , since N_r reflects both n and w_x . Of course, one can explore some other possibilities.

¹⁰Note that a receiver can know whether the flow is active with the help of the inactive-notification mechanism introduced in Section 4.4.1.

Whenever the sender receives a restrictive-notification, if it is already in the restrictive mode with a degree equal or greater than the one contained in the notification, it will simply ignore this notification¹¹. Otherwise, the sender will put itself in the restrictive mode with the degree contained in the notification.

4.5.4 Receiver Feedback: Explicit Aggressive-notification

If a flow has under-used the medium in the latest window, it must have *not* transmitted any packet recently, and there is no way to *piggyback* the aggressive-notification. Therefore, the aggressive-notification must be sent to the sender in an *explicit* manner. We can define a frame similar to RTS to carry the notification. Upon receiving the notification, if the sender is in the restrictive mode, it ignores the notification. Otherwise, it contends for the medium as per the aggressive mode with a degree of aggressiveness N_a that is contained in the notification.

Two issues need further investigation. *First*, we should try to avoid the collision of the frames carrying aggressive-notification as there may be more than one receiver intending to send out a notification at the same time. *Second*, since the *explicit* transmission of an aggressive-notification is a wastage of bandwidth, we should not use the aggressive-notification whenever it is unnecessary. In particular, if the sender is already aware that the flow should be aggressive and the sender can identify the contention period, the receiver should not send out a notification. Let us consider the scenario presented in Figure 4.10. The nodes R_A , R_B , S_C , S_D , R_C , and R_D are within the range of each other. In such a scenario, the sender S_C have the same understanding of the contention as the receiver R_C does, and thus R_C should not send out a notification when flow C has under-used the medium. This is also true for R_D . In contrast, if flow A or B has under-used the medium, the notification should always be sent by R_A and R_B . Moreover, if there is another contending flow that is in normal/restrictive mode, the notification should be sent out *before* the sender of that flow transmits its RTS frame. In summary, for the *second* issue, we need to investigate some mechanisms, which can intelligently decide whether or not an aggressive-notification should be sent, and if it is to be sent, *when* to send it.

To cope with the first issue discussed above, before a receiver sends out an aggressive-notification, it will first back off by a random value. To cope with the second issue, the priorities

¹¹This is to prevent the unnecessary use of the receiver information in the scenario where the sender has complete information about the contending flows, e.g., in the high-contention scenario of Figure 4.6.

Figure 4.10: Scenario Illustrating the Aggressive-notification

for the medium access should be in the following order (from highest to lowest): senders of the flows in the aggressive mode, receivers of the flows in the aggressive mode, senders of the flows in the normal mode, and then, senders of the flows in the restrictive mode. Since the senders of the flows that are in aggressive mode will back-off with a maximum value equal to $2n$ as discussed in Section 4.5.1, the random back-off value for the aggressive-notification can be within the range $[2n, Y]$ where $Y = \max(3n, 4n - N_a)$. Clearly, the maximum value a receiver will back-off before sending out the aggressive-notification is $4n$. Therefore, all the senders of the flows that are in normal/restrictive mode should generate a random back-off value which is at least $4n$ rather than $2n$ as used in Section 4.5.1. In this way, we avoid the unnecessary transmission of aggressive-notification but do not compromise on the effectiveness of this mechanism. For example, in the scenario of Figure 4.10, the receiver R_C will not send out an aggressive-notification whenever flow C should be in the aggressive mode since the sender S_C will send out a RTS before the receiver R_C 's back-off timer expires. On the other hand, the receiver R_A will always send out a notification to its sender S_A whenever the flow A should be aggressive.

Summary of FMAC/CSR: In Table 4.1, we summarize the FMAC/CSR. FMAC/CSR-1 refers to the FMAC/CSR that only adopts the differentiated access at the senders, while FMAC/CSR-2 refers to FMAC/CSR-1 plus the restrictive-notification mechanism. Obviously, FMAC/CSR-3 refers to the FMAC/CSR-2 plus the aggressive-notification mechanism.

4.6 Simulation Results

In this section, we present the simulation results to compare our FMAC/CSR algorithm with IEEE 802.11. The simulations parameters used is the same as in Chapter 3. The well-known Jain's index [29] is used as the main measure, which is defined as follows:

Table 4.1: Summary of Three Categories of FMAC/CSR

Mode	Side	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Aggressive	Sender	<i>Back off:</i> [0, max(n, 2n-N _a)]	<i>Back off:</i> [0, max(n, 2n-N _a)]	<i>Back off:</i> [0, max(n, 2n-N _a)]
	Receiver	NA	NA	<i>First Back off:</i> [2n, max(3n, 4n-N _a)] <i>Then send Aggressive-notification</i>
Normal	Sender	<i>Back off:</i> [2n, CW]	<i>Back off:</i> [2n, CW]	<i>Back off:</i> [4n, CW]
	Receiver	NA	NA	NA
Restrictive	Sender	<i>First Defer:</i> TxTime(pkt) x (N _r + 1) <i>Then Back off:</i> [2n, N _r x CW]	<i>First Defer:</i> TxTime(pkt) x (N _r + 1) <i>Then Back off:</i> [2n, N _r x CW]	<i>First Defer:</i> TxTime(pkt) x (N _r + 1) <i>Then Back off:</i> [4n, N _r x CW]
	Receiver	NA	<i>Piggyback:</i> Restrictive-notification	<i>Piggyback:</i> Restrictive-notification

$$F_J = (\sum_{i=1}^N \gamma_i)^2 / (N \sum_{i=1}^N \gamma_i^2) \quad (4.5)$$

where N is the total number of flows that share the wireless medium, and γ_i is the fraction of the bandwidth utilized by flow i over a certain number of packets transmitted, say w , called *fairness measurement window*. As the computation of γ_i depends on w , the value of Jain's index also depends on w , though w does not appear in the formula directly. Generally, F_J value increases with w . *Absolute fairness* is achieved when $F_J = 1$ while the *absolute unfairness* is achieved when $F_J = 1/N$. As in [40], the index has been *averaged* over all *sliding* windows of w packets, which occur in the simulation run.

4.6.1 Hidden-terminal Topologies

The Jain's index for the hidden-terminal topologies (Figure 4.4) is presented in Figure 4.11. For IEEE 802.11, when w is small (e.g., 2), the index is very small (about 0.52) compared to the absolute fairness (i.e., unity), implying substantial short-term unfairness. On the other hand, when w is very large, the index is close to unity (though not shown in the figure due to the constraint of the figure size), implying the long-term fairness. Compared to IEEE 802.11, FMAC/CSR-1 greatly improves the fairness. However, FMAC/CSR-2 and FMAC/CSR-3, which incorporates the receiver-feedback mechanism(s), do not show any advantage in comparison to FMAC/CSR-1. The reason is that the senders in this topology can get a precise estimation, and thus the receiver feedback mechanisms do not play any roles.

Figure 4.11: Fairness Index in Hidden-terminal Topologies

Since *maximizing capacity utilization* and *achieving fairness* are generally two *conflicting* objectives in wireless ad-hoc networks [42], when evaluating a fairness algorithm, it is also necessary to look at the throughput results to ensure that the algorithm does not unduly degrade the throughput. Table 4.2 presents the throughputs under different schemes. While FMAC/CSR greatly improves the fairness, it also improves the aggregate throughput. This shows the advantage of our FMAC/CSR. Note that the aggregate throughput under FMAC/CSR-3 degrades insignificantly compared to those of FMAC/CSR-1 and FMAC/CSR-2 due to the reason that under normal and restrictive modes in FMAC/CSR-3, the minimum back-off time at the sender is $4n$ rather than $2n$ (see Table 4.1). Now, we explain why the throughput under FMAC/CSR is greater than that under IEEE 802.11. As discussed before, under IEEE 802.11, whenever the two senders contend for the medium, their RTSs are very likely to collide, leading to wastage of bandwidth. On the contrary, under FMAC/CSR, since the flows can cooperate in a distributed manner (e.g., when one flow is in the restrictive mode, then the other one is in the aggressive mode), the chances of collisions and the medium being idle are greatly reduced, explaining the throughput improvements.

Table 4.2: Throughput under Hidden-terminal Topologies

Throughput (Mbps)		IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Average	S_A to R_A	0.678	0.720	0.720	0.717
	S_B to R_B	0.676	0.720	0.720	0.717
Aggregate		1.354	1.440	1.440	1.434

4.6.2 Asymmetrical Information Topologies

As long-term unfairness is known to occur in the asymmetrical information topologies (Figure 4.3), we first present the throughput results in Table 4.3. Clearly, under IEEE 802.11, flow B gets almost the entire bandwidth while flow A is almost starved. The reason of long-term unfairness has been explained in Section 4.2.1. Under FMAC/CSR-1, the fairness substantially improves, however, it does not achieve long-term fairness completely. Moreover, the aggregate throughput degrades. The reason for these two observations has been explained in Section 4.5.2.

The fairness as well as the aggregate throughput substantially improves under FMAC/CSR-2, which incorporates the restrictive-notification mechanism. However, the performance does not improve any further under FMAC/CSR-3, which includes the aggressive-notification mechanism. To explain why FMAC/CSR-2 improves the performance compared to FMAC/CSR-1, consider that, after node S_B transmits two packets consecutively, flow B enters the restrictive mode and thus S_B defers its transmission by a time needed for transmitting two packets. Therefore, once node S_A gets control of the medium, it can also transmit two packets consecutively after which node S_B will begin to contend for the medium. If the restrictive-notification mechanism is used (as in FMAC/CSR-2), node S_A will receive a notification and thus defer. On the contrary, without the restrictive-notification (as in FMAC/CSR-1), S_A will futilely retry. In other words, the restrictive-notification mechanism helps node S_A to identify the contention period, explaining the performance improvements in FMAC/CSR-2. In this manner, the restrictive-notification mechanism has accomplished the task (i.e., helping S_A to identify the contention period) that the aggressive-notification mechanism is supposed to do, and therefore the aggressive-notification mechanism introduced in FMAC/CSR-3 does not show any additional advantage.

The Jain's index is presented in Figure 4.12. FMAC/CSR-2 achieves short-term fairness, and thus automatically ensures long-term fairness. Since FMAC/CSR-2 and FMAC/CSR-3 show similar fairness, we do not present the index under FMAC/CSR-3 to avoid the cluttering of the graph.

Table 4.3: Throughput under Asymmetrical Information Topologies

Throughput (Mbps)		IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Average	S_A to R_A	0.073	0.538	0.718	0.716
	S_B to R_B	1.345	0.628	0.718	0.716
Aggregate		1.418	1.166	1.436	1.432

Figure 4.12: Fairness Index in Asymmetrical Information Topologies

4.6.3 Scenario-1 of Figure 4.2

In this scenario, under IEEE 802.11, each of the two flows gets only about 0.29 Mbps, which is much less than the expected 0.7 Mbps. The reason is as follows. Consider the situation that node S_A sends out a RTS to node R_A , and then node R_A sends back the CTS. As node S_B is unaware of this CTS, it may send out a RTS, leading to a collision at node R_B . A node (e.g., node R_B) that detects a collision will defer for the Extended Inter-Frame Space (EIFS) duration, which is much smaller than the time needed for the complete transmission of a Data frame. Since the RTS/CTS handshaking is successful for node S_A , it will transmit the Data frame. During the transmission, S_B may initiate its RTS again. If the RTS arrives at node R_B after the EIFS deferment is over, R_B will respond with CTS, which will collide with the Data frame from node S_A to R_A , and thus node R_A will *discard* the Data frame and defer by an EIFS. Now, for node S_B , since the RTS/CTS is successful, it transmits the Data frame. The Data frame is likely to be destroyed by node R_A 's CTS in a similar manner, resulting in substantial bandwidth wastage. The dual busy tone multiple access (DBTMA) [22] can solve this problem, but two additional busy-tone channels are required. For simplicity, here

we make a slight modification in IEEE 802.11, that is, whenever a node detects a collision, rather than deferring by an standard EIFS duration, it will defer for a large enough duration enabling transmission of a Data frame. We call this as the *Large-Col-EIFS* mechanism. In this subsection, we present the results assuming that this mechanism is included. Though the pros and cons of the *Large-Col-EIFS* need further investigation, it is also used in the scenario of Figure 4.16 as the scenario also contains the case that two senders are three hops away. The aggregate throughput is presented in Table 4.4 and the Jain's index is presented in Figure 4.13.

Table 4.4: Throughput under Topology of Figure 4.2

Throughput (Mbps)		IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Average	S _A to R _A	0.703	0.701	0.532	0.582
	S _B to R _B	0.707	0.700	0.530	0.582
Aggregate		1.410	1.401	1.062	1.164

Figure 4.13: Fairness Index under Topology of Figure 4.2

We notice that the aggregate throughput in IEEE 802.11 with *Large-Col-EIFS* mechanism greatly improves. As for the fairness, IEEE 802.11 exhibits substantial short-term unfairness. Moreover, the FMAC/CSR-1 does not improve the fairness or throughput. On the contrary, the FMAC/CSR-2, which incorporates the restrictive-notification mechanism, greatly improves the fairness. However, under FMAC/CSR-2, the short-term unfairness remains and the aggregate throughput degrades substantially. The reason is again due to the fact that, when one flow enters the restrictive mode, the sender of the other flow cannot identify the contention period

and thus cannot recognize the opportunity to transmit. The FMAC/CSR-3, which includes the aggressive-notification, improves the short-term fairness as well as the aggregate throughput compared to FMAC/CSR-2.

4.6.4 Scenario-7 of Figure 4.5

The throughput results are presented in Table 4.5 and the Jain's index is presented in Figure 4.14. It is easy to see that, under IEEE 802.11, flow B gets more bandwidth than flow A, due to the imprecise collision detection as explained in Section 4.2.2. Under FMAC/CSR-1, the fairness improves. Again, the FMAC/CSR-2 and FMAC/CSR-3, which incorporate the receiver-feedback mechanisms, do not show advantage in comparison to the FMAC/CSR-1 as the senders can get the complete information about the contending flows.

Table 4.5: Throughput under Scenario-7 (Figure 4.5)

Throughput (M bps)		IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Average	S _A to R _A	0.672	0.720	0.720	0.717
	S _B to R _B	0.766	0.720	0.720	0.717
Aggregate		1.438	1.440	1.440	1.434

Figure 4.14: Fairness Index under Scenario-7 (Figure 4.5)

4.6.5 High-contention Topology (Figure 4.6)

For the high-contention scenario, the Jain's index is displayed in Figure 4.15 and the throughput results are presented in Table 4.6. The IEEE 802.11 exhibits substantial short-term unfairness.

In contrast, FMAC/CSR-1 greatly improves the short-term fairness compared to IEEE 802.11, and again the receiver feedback mechanisms do not play any role here.

Similar fairness benefits were observed when the number of contending nodes is *larger* than five but the nodes were within the one hop distance.

Figure 4.15: Fairness Index in High-Contention Topology

Table 4.6: Throughput in High-contention Topology

Throughput (M bps)	IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Aggregate	1.410	1.405	1.405	1.370

4.6.6 A Complex Topology

In this subsection, we consider a complex topology shown in Figure 4.16, which combines the main scenarios discussed so far. For convenience of explaining the relative positions of the nodes, we have drawn two concentric circles (dotted-line) in the figure. The distance between the sender and receiver of each flow is 200 meters, except for the flow from node 16 to 17, where the distance is set in a way such that the two nodes are in the ranges of all the nodes on the inner circle but out of the ranges of all the nodes on the outer circle. The diameter of the inner circle is 200 meters. Therefore, the diameter of the outer circle is 600 meters. The angle between any two neighboring flows is 45 degrees. The *Large-Col-EIFS* mechanism discussed before is adopted in this topology.

The throughput results are presented in tables 4.7 and 4.8, while the Jain's index is displayed in Figure 4.17. In Table 4.7, the flow ID is the same as the node ID of the flow's sender. Under

IEEE 802.11, four flows (i.e., flows from node 0 to 1, node 2 to 3, node 4 to 5, and node 6 to 7) starve, while the flow from node 16 to 17 gets much higher throughput than the remaining flows. On the other hand, the bandwidth under FMAC/CSR-3 is distributed quite evenly among the flows. From Table 4.8, we notice that the aggregate throughputs under all three FMAC/CSR schemes improve compared to that under IEEE 802.11, showing the merits of our FMAC/CSR in complex scenarios. The FMAC/CSR schemes also achieve short-term fairness as shown by the Jain's Index in Figure 4.17.

Since the above topology involves all the causes of unfairness discussed in Section 4.2, the FMAC/CSR should be able to deliver similar performance (i.e., achieving fairness without unduly degrading throughput) in a randomly generated topology as long as all the single-hop flows are contending with each other (i.e., without spatial-reuse).

Figure 4.16: Scenario-9: Complex Topology

Table 4.7: Throughput in Complex Topology (Figure 4.16)

Flow ID	0	2	4	6	8	10	12	14	16
IEEE 802.11	0.000	0.000	0.000	0.000	0.156	0.193	0.195	0.134	0.509
FMAC/CSR-3	0.149	0.146	0.149	0.143	0.128	0.128	0.128	0.121	0.153

Table 4.8: Aggregate Throughput in Complex Topology

Throughput (M bps)	IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Aggregate	1.187	1.258	1.210	1.244

Figure 4.17: Fairness Index in Complex Topology (Figure 4.16)

4.6.7 Verification of Estimation Algorithm

So far, using Constant Bit Rate (CBR) traffic, we have seen that our FMAC/CSR substantially improve the fairness. From the simulation traces, we also found that the estimation of n is quite precise except for the imprecise-estimation due to the absence of overheard information (e.g., in scenarios of figures 4.2 and 4.3). Here, we evaluate the precision of the estimate of n when the traffic is dynamically changing (i.e., exponential On/Off traffic) to show that the timeout mechanism designed in Section 4.4.1 can work well under the dynamic conditions. The topology of Figure 4.6 is used for study. The average “on” time is 0.3 s, whereas the average “off” time is 0.7 s. In the “on” state, CBR traffic is generated as explained earlier. We dynamically record n_e and n throughout the simulation time. The interval between two consecutive samples is 0.005 s, which is very close to the time needed for the transmission of a packet. Figure 4.18 presents the results during a typical duration of 2 seconds. We notice that n_e is equal to n most of the time.

4.7 Discussions and Related Work

4.7.1 Further Comments on FMAC/CSR

FMAC/CSR when Sensing Range > Transmission Range: In FMAC/CSR, a sender or receiver of a flow needs to know the number of active flows (i.e., n) within the contention range and the flow’s actual share (i.e., w_x), which can only be estimated in a distributed manner. In Section 4.4, we have proposed an estimation algorithm by assuming that the transmission

Figure 4.18: Comparison between Actual and Estimation of n

range (TR) is equal to the sensing range (SR). In practice, the SR may be greater than the TR . Therefore, our estimation algorithm, as well as other algorithms ([18, 36, 35], and even the *virtual carrier sensing* in IEEE 802.11) based on overhearing may not work in such a scenario. Note that the unfairness problem (in Section 4.2), the fairness framework (in Section 4.3), and the FMAC/CSR itself (in Section 4.5) are general and applicable even when $SR > TR$.

Now, we discuss how to modify our estimation algorithm when $SR > TR$. First of all, we should note that the transmission range of RTS/CTS frames is *much greater* than that of Data/ACK frames since the length of the RTS/CTS is very short compared to that of Data frame, and since the RTS/CTS are always transmitted at the *lowest* physical rate. This has also been pointed out in [3]. On the contrary, the sensing range for all the frames (RTS/CTS/Data/ACK) should be the same since the range only depends on the carrier sensing energy threshold. Therefore, if the threshold is set to an energy level such that the RTS/CTS are always interpretable, then the transmission range of the RTS/CTS frames is equal to the common sensing range. In fact, as clearly shown in [49], this is true for the emerging standards IEEE 802.11a/h. In such a situation, whenever a *packet* is transmitted over the medium, a node within the sensing range will always overhear the RTS/CTS frames *clearly* (but not true for Data/ACK frames). Therefore, if an algorithm assumes $SR = TR$ but only relies on the overhearing of RTS/CTS frames, it can work even when the common SR is greater than the TR for the Data/ACK frames. This is true in our estimation algorithm for the number of active flows n as it only relies on the overhearing of RTS/CTS. Therefore, when $SR > TR$, we only need to modify the estimation algorithm for the actual share w_x . We should note that a sender

or receiver of a flow under our FMAC/CSR needs only to know the actual share of the flow itself. Moreover, the estimation of w_x at the sender or the receiver of flow x does not need to know the exact flow ID of the packets transmitted by other flows. Obviously, a node will always know the flow ID of a packet if the node is the sender or the receiver of the flow. Therefore, we can modify the estimation algorithm for the actual share w_x as follows. Whenever a node detects a Data/ACK frame but it cannot interpret the contents of the frame¹², it should add an invalid ID (e.g., -1) into its transmission history. To cope with the situation that a node detects the Data and ACK frames both of which are not interpretable and belong to a same handshake, the node can start a timer on detecting a Data frame, and will not add '-1' ID into its history *another* time if an ACK frame is detected before the timer expires. We can further prevent the wrong inserting of '-1' ID by maintaining timers when RTS/CTS is clearly overheard. With the above modification, our estimation algorithm can work well when the SR is greater than the TR of Data/ACK frames. The above discussion also applies to the situation that a collision or wireless error is detected since in these cases also the frame contents are not interpretable. Clearly, the key reason that our estimation algorithm can be easily modified to work when $SR > TR$ is that we have intentionally estimated n and w_x separately by maintaining two different lists, as mentioned before.

In the case that the sensing range is even greater than the transmission range of RTS/CTS (e.g., under IEEE 802.11b), the above modified estimation algorithm for the actual share still works, though the estimation of n may not always be precise. However, with our receiver-feedback mechanisms, any ill effects will occur only when both the sender and receiver have wrong estimation of n . Alternatively, we can get an estimate of n using some heuristics methods as in [8, 10].

FMAC/CSR under Variable Packet Length and Different Flow Priorities: As done in many research-work, we have assumed that the packets have the same length and the applications have the same priority, which are practical assumptions in wireless networks [41]. However, our FMAC/CSR can be easily extended for the scenarios with variable packet length

¹²The type (Data or ACK) of a frame can be identified based on the duration for which the medium is busy due to the transmission of the frame as we will show in the ECS in Chapter 5.

and different flow priorities. Again, the main modification needed is to refine the fairness model to reflect such a case.

Flow-level Fairness in FMAC/CSR: In the description of FMAC/CSR in Section 4.5, we have implicitly assumed that a node can act as the sender of at the most for one flow. Therefore, the flow-level fairness is identical to the *node-level* fairness for the *senders*, which is the fairness model followed in IEEE 802.11. To achieve flow-level fairness, IEEE 802.11 needs several modifications. Firstly, a flow ID should be added into the frames, since a sender or receiver or both no longer uniquely represent a flow. Secondly, a separate contention window and back-off timer should be used for each flow. With these, FMAC/CSR can achieve the flow-level fairness. Note that if an algorithm is able to achieve flow-level fairness, it can always achieve node-level fairness.

4.7.2 Related Work

The fairness problem in the random wireless MAC protocols was first highlighted in [6]. Recently, the fairness issue has been extensively addressed. Specifically, several works (e.g., [44] and the references therein) have addressed the unfairness problem due to the *location-dependent transmission error* in wireless LANs. More recently, assuming that there is no wireless error, a lot of works has addressed the fairness issue in wireless ad-hoc networks. Based on the information needed and the fairness objective (global or local), these works can be categorized into two classes. The first category (e.g., [42, 41]) aims to design an ideal centralized scheduling algorithm, which is overlaid on the top of the MAC layer, to address the unfairness problem due to the *location-dependent contention* in *multi-hop* wireless ad-hoc networks with spatial-reuse. Though these scheduling algorithms may achieve a network-wide fairness, they normally need global information (e.g., flow-contention graph or network topology). Moreover, whenever the network status changes (e.g., the topology changes due to mobility, or a flow enters or leaves the network), this information should be broadcasted in the network and a new scheduling decision has to be made. On the contrary, some other works (e.g., [18, 36, 43, 56]) aim to achieve fairness at the MAC layer itself. Our work belongs to this category. Though an algorithm following this approach may only achieve local fairness, it is desirable that it can operate in a distributed manner. Moreover, to *practically* implement an ideal centralized scheduling algorithm, a distributed

MAC layer fair algorithm is required to approximate the ideal algorithm. Therefore, whether the objective is to achieve local fairness or global fairness, MAC layer fairness is essential.

Now, we review the works that followed the second approach. In [43], the authors have developed a fully distributed MAC protocol, called Proportionally Fair Contention Resolution, which dynamically adjusts the probability (based on its observation of the medium states, e.g., collision, idle, or busy) with which a sender accesses the medium. Based on the information in the overheard frames (rather than only the medium states), the works in [18, 56] try to make the medium access more adaptive. Specifically, the work in [56] aims to emulate the Self-Clocked Fair Queueing [21] in wireless ad-hoc networks by adaptively choosing a back-off value proportional to the finish tag of a packet to be transmitted. The work in [18] dynamically tunes the contention window of a node based on its estimation of the sharing of the medium. More recently, another work [17] that aims to achieve MAC fairness by generalizing the approach in [43] using a game theory framework. However, it derives the fairness algorithm based on a flow-contention graph, which is inappropriate as remarked in Section 4.2.4. All the above schemes are sender-based, which cannot *always* achieve fairness since the information at the sender is not always precise. On the other hand, the authors in [36] have exploited the receiver information to implement a FIFO queue among the contending nodes. However, they have considered the asymmetrical information topology only, and how the sender and receiver should cooperate is largely unaddressed. Moreover, their scheme results in substantial throughput degrade as a fully deterministic method is adopted. Even worse, as shown in [55], due to the deterministic nature, deadlocks are likely to occur in a complex scenario. While all the above works (except [43, 17]) rely on the correctness of the estimation based on the overheard frames, but none of them describes how to estimate correctly when the sensing range is greater than the transmission range in which case the overheard frames may not always be interpretable and thus the estimation is not precise.

Compared to all the above works on MAC fairness, our FMAC/CSR is quite unique due to the following reasons: (1) it exploits the information available both at the sender and the receiver, (2) it achieves fairness without degrading the throughput in all the scenarios where spatial reuse is not possible, and (3) it is applicable even when the sensing range is greater than the transmission range.

4.8 Conclusions

In this chapter, we have proposed a novel medium access control protocol, called FMAC/CSR, which exploits information available at the senders and receivers to achieves MAC fairness. The simulation results show that the FMAC/CSR greatly improves the fairness without unduly degrading the throughput.

The main contributions include: *(i)* identification of general reasons that lead to MAC unfairness, e.g., concealed information problem, imprecise collision detection mechanism, and deficiency of BEB; *(ii)* definition of a general MAC fairness framework; *(iii)* proposal of a simple but efficient algorithm, which can dynamically estimate the number of *active* flows as well as the actual share of a flow; *(iv)* proposal of the FMAC/CSR, which incorporates a suite of mechanisms: differentiated access at the senders, restrictive-notification, and aggressive-notification.

Chapter 5

Enhanced Carrier Sensing (ECS)

In the previous chapters, for simplicity, we have assumed that the transmission range is equal to the sensing range. In this chapter, under the assumption that the sensing range is greater than the transmission range, we propose an enhanced carrier sensing (ECS) mechanism to improve the performance (i.e., throughput and fairness) of IEEE 802.11.

5.1 Introduction

As mentioned before, IEEE 802.11 is a CSMA/CA based MAC protocol, which normally includes several components: carrier sensing (CS), collision avoidance (CA), and contention resolution (CR). In the literature, there has been a lot of research-work on improving fairness or throughput by modifying the CA or CR mechanism. For example, [5] mainly focuses on the CA part to improve the fairness, while [18, 43, 56] aim to achieve fairness through modifying the CR algorithm of IEEE 802.11. In contrast, [6, 20, 37] aim to enhance the CA handshake to improve the throughput by preventing collisions as much as possible, while [10] aims to make the CR algorithm more adaptive and thus improve the capacity utilization. It is necessary to point out that each of the above work either improves the fairness at the cost of the throughput, or vice versa. Moreover, to the best of our knowledge, there is no research-work focusing on the carrier sensing (CS) part to improve the performance of the MAC protocol in ad-hoc networks. In this chapter, we aim to improve the performance (both fairness and throughput) by enhancing the CS mechanism in IEEE 802.11.

As mentioned in Chapter 1, in an IEEE 802.11-based wireless ad-hoc networks, since all the nodes share a common medium, only one flow among the contending flows can transmit during

a certain duration. Therefore, whenever a frame exchange sequence between two nodes is in progress, it is extremely important to guarantee that all the other nodes in the interference range should defer their own transmission. In IEEE 802.11, whenever a node detects a physical carrier on the medium, it does not transmit. This is known as the *physical carrier sensing*. Moreover, even after the medium physically becomes idle, the node may need to defer further to allow the transmission of the *remaining* frames in the on-going sequence. This is known as the *virtual carrier sensing* (VCS). Under VCS, every frame will carry a duration value indicating the time by which the overhearing nodes should defer. Clearly, whenever a node overhears a frame, the node needs to interpret the contents of the frame to get the exact length of the duration by which it should defer. However, this cannot be achieved in the following two cases: (i) If a frame experiences a transmission error, a node that overhears this frame cannot interpret the contents of the frame; (ii) Since the carrier sensing range is normally greater than the transmission range [59], whenever a frame is transmitted over the medium, a node out of the transmission range but within the sensing range of this transmission will detect this sensing range (SR) frame but it cannot interpret the contents. The above two cases occur very frequently in wireless networks and have a great impact on the performance of IEEE 802.11. Therefore, an important issue is that *whenever a node detects an erroneous frame, how long should the node defer its transmission?* To the best of our knowledge, this issue has not been addressed in the literature.

In the current IEEE 802.11 standards, whenever a node detects an erroneous frame on the medium, it defers the transmission by a *fixed* duration indicated by the Extended Inter-Frame Space (EIFS) constant. In this chapter, we show that in some situations, the EIFS value is too large compared to the desired value, while in some other situations the EIFS value is too small. Respectively, we refer to these two cases as large-EIFS and small-EIFS problems. The two problems, together, are referred as the imprecise-EIFS problem. The imprecise-EIFS problem leads to immense unfairness and throughput degradation. Specifically, when the small-EIFS problem occurs, a node may begin its transmission even though some other nodes in the interference range are still transmitting, resulting in collisions (and thus throughput degradation). On the other hand, when the large-EIFS problem occurs, the medium may be

unnecessarily idle and the node may experience unfairness as it defer a longer duration than other nodes. In order to solve the imprecise-EIFS problem, we propose an enhanced carrier sensing (ECS) mechanism in which the EIFS value is made variable in an adaptive manner. The simulation results show that our ECS greatly improves the fairness as well as the capacity utilization.

The rest of the chapter is organized as follows. In Section 5.2, the small- and large-EIFS problems are illustrated with the help of simple examples. The enhanced carrier sensing (ECS) mechanism is proposed in Section 5.3. The performance of ECS is studied in Section 5.4, and the chapter is concluded in Section 5.5.

5.2 Imprecise-EIFS Problem in IEEE 802.11

In this section, we demonstrate the imprecise-EIFS problem through the simulation in two simple scenarios. Though in the introduction section of this chapter, we have mentioned two types of *erroneous* frame (i.e., a TR frame experiencing a transmission error, or an SR frame), we do not explicitly consider the case of a TR frame with a transmission error since in this case similar problems are resulted as in the case of an SR frame.

The simulation environment is the same as that used in the previous chapters except that the transmission range and the sensing range are 250 and 550 meters, respectively.

5.2.1 Large-EIFS Problem

To explain the large-EIFS problem, scenario shown in Figure 5.1 is used, where the distance between two neighboring nodes is 200 meters. Therefore, nodes A and C are out of TR but within the SR of each other. There are two single-hop flows, and we expect them to share the bandwidth equally, i.e., each flow should have a throughput of about 0.7 Mbps. However, from the results in Figure 5.2, we find that the flow from B to C gets about 1.15 Mbps whereas the flow from A to B gets only about 0.25 Mbps, which shows how much unfair the IEEE 802.11 is in this simple scenario.

Now, we explain the reason of the above unfairness. IEEE 802.11 defines how a node should defer its transmission while a FFS between two other nodes is in progress. Figure 5.3 (a) demonstrates the process how node C defers its transmission while FFS(A, B) is in progress.

First, node A sends a RTS to B. Since node C is out of the TR but within the SR of node A, it gets an SR frame. After the completion of this frame, node C defers its transmission with EIFS. Then, node B sends a CTS to node A. Since node C is within the TR of node B, it gets a TR frame and updates the Network Allocation Vector (NAV) with the duration contained in the CTS frame whose value is equal to $SIFS + TxTime(Data) + SIFS + TxTime(ACK)$. After this, node A sends the Data frame to node B. After the completion of this Data frame, node C defers its transmission with EIFS. At last, node B sends an ACK to node A. Since the duration field in the ACK frame is zero, the NAV of C will be updated to zero. At this point, all the three nodes will defer their transmission by DIFS, and then begin to contend for the medium using the random back-off. In the above scenario, all the nodes get up-to-date information of the medium, avoiding collision and guaranteeing fair access.

Now, let us consider another scenario shown in Figure 5.3 (b), in which node A is deferring while FES(B, C) is going on. Since node A receives the TR frames corresponding to RTS and Data frames transmitted by node B, node A gets the precise state information of the medium up to the point when node C sends an ACK to node B¹. Corresponding to this ACK, since node A is in the sensing range of node C, node A gets an SR frame and defers its transmission with EIFS. We can see that, after the completion of the ACK, before node A backs-off by a random period, it has to defer by EIFS duration rather than by DIFS as done by the other nodes (e.g., nodes B and C). Since EIFS is equal to $SIFS + TxTime(ACK) + DIFS$ [24], the deferment at node A is certainly of much longer duration than that at the other nodes. We call this large-EIFS problem because the EIFS value is larger than it should be to reflect the state of the medium. In fact, the EIFS value should be equal to DIFS in this case.

Summarizing the results of the above two cases, we see that whenever a FES(A, B) is successfully completed, nodes A and B start contending for the medium at about the same time. However, whenever a FES(B, C) is completed, node A contends for the medium ($EIFS - DIFS$) duration *later* than node B does, which is clearly very unfair for A. This explains the unfairness between the two single-hop flows. However, the above arguments do not explain why the throughput between the two flows differs so much. The average throughputs can be obtained

¹Note that in Figure 5.3 (b) we do not indicate the EIFS deferment at node A when it detects the SR frame corresponding to node C's CTS because the remaining NAV value is greater than the EIFS.

by computing the relative frequencies of FES(A, B) and FES(B, C). In Appendix B, we present a Markov-chain model to obtain the average throughputs of the two flows, validating the results shown in Figure 5.2.

Figure 5.1: 3-nodes with Two Single-hop Flows (Scenario-10)

Figure 5.2: Throughput under Large-EIFS Problem

Figure 5.3: Time Diagram Showing Frame Exchange Sequence

5.2.2 Small-EIFS Problem

While the large-EIFS problem causes substantial unfairness as shown in the previous subsection, here we will show that the small-EIFS problem results in substantial throughput degrade. The

scenario of Figure 5.4 is used, where the distance between two neighboring nodes is again 200 meters. The simulation results show that the two flows share the bandwidth equally (in a long term). However, each of them only gets about 0.31 Mbps, resulting in an aggregate throughput about 0.62 Mbps, which is much smaller than the capacity (i.e., about 1.4 Mbps).

To explain why the throughput degrades so much, let us consider the situation where nodes C and D are deferring their transmission, while FES(A, B) is in progress (see Figure 5.5). After node B sends a CTS to A, node D gets an SR frame and defers its transmission with EIFS. Then, node A begins to send its Data frame. However, since node D is out of the SR of A, node D cannot hear this Data frame and thus it determines that the medium be idle though in fact it is busy. Moreover, since the EIFS value is much smaller than the transmission time of the Data frame, after deferring for the EIFS duration, node D backs-off and then sends out a frame even though node A is still transmitting the Data frame. The two transmissions result in a collision at node B and thus node B discards the frames. Likewise, due to symmetry in the topology, node C is also likely to drop the Data frame of a FES(D, C), explaining why the throughputs degrade so much. We call this the small-EIFS problem because the EIFS value is smaller than it should be to reflect the state of the medium. In fact, in this situation, the EIFS value should be large enough to allow for the complete transmission of the Data frame. Note that in this scenario, as indicated in Figure 5.5, the large-EIFS problem also occurs at node D after node B sends the ACK to node A.

Figure 5.4: 4-nodes with Two Single-hop Flows (Scenario-11)

Figure 5.5: Time Diagram when FES(A, B) is in Progress while D defers

In the above discussion, we assumed that the nodes do not have the capture ability and the static routing is used. In Appendix C, by varying the factors including the capture ability,

with and without imprecise-EIFS problem, and the routing protocols, we have conducted an extensive simulation study for this topology. Our study shows that all the three factors greatly affect the performance.

5.2.3 General Scenario involving Imprecise-EIFS Problem

So far using specific examples, we have demonstrated how the large-EIFS and small-EIFS problems result in substantial unfairness and throughput degrade. In fact, which specific type of problem that a node will suffer from depends upon what kind of frame it detects. What kind of frame a node detects during an ongoing FES, in turn, depends upon the location of the node with respect to the location of the two nodes between whom the FES is in progress. For example, if a node gets an SR frame corresponding to a CTS, it may suffer from the small-EIFS problem. If a node gets an SR frame corresponding to an ACK, it may suffer from the large-EIFS problem. However, if a node gets an SR frame corresponding to a Data frame, the large- or small-EIFS problem does not arise because the EIFS value caters for the time needed for the next frame (i.e., ACK) to pass through. This is also true when an SR frame corresponding to a RTS is detected, since the next frame after a RTS is a CTS frame, and the length of the CTS is equal to an ACK. Figure 5.6 shows a generalized scenario with imprecise-EIFS problem where FES(A, B) is in progress. We classify the entire region into seven areas. Nodes within different areas get different type of frames and thus may suffer from different type of problems. The results are summarized in Figure 5.7, where $T(\text{frame})$ denotes a TR frame and $S(\text{frame})$ denotes an SR frame. Note that in the area 6, a node will not suffer from the small-EIFS problem after detecting an SR frame of CTS, since the node can also detect the SR frame corresponding to the Data.

5.3 Enhanced Carrier Sensing (ECS)

The basic idea behind EIFS deferment is as follows. When a node detects an SR frame, it assumes that this frame should be one of the frames belonging to an ongoing frame exchange sequence between two *other* nodes, and therefore, there may be another frame transmission in the near future between the same pair of nodes. In order to avoid interfering with the ongoing exchange, the node that detects the frame should defer its transmission by a certain duration.

Figure 5.6: Areas of Influence Due to FES(A, B)

Area 1: T(RTS) ->T(CTS) ->T(Data) ->T(ACK) no large- or small-EIFS problem
Area 2: T(RTS) ->S(CTS) ->T(Data) ->S(ACK) large-EIFS problem after S(ACK)
Area 3: S(RTS) ->T(CTS) ->S(Data) ->T(ACK) no large- or small-EIFS problem
Area 4: S(RTS) ->S(Data) no large- or small-EIFS problem
Area 5: S(CTS) ->S(ACK) small-EIFS problem after S(CTS) and large-EIFS problem after S(ACK)
Area 6: S(RTS) ->S(CTS) ->S(Data) ->S(ACK) large-EIFS problem after S(ACK)
Area 7: Cannot detect any TR or SR frame no large- or small-EIFS problem

Figure 5.7: Problems Arising in Different Areas while FES(A, B) is in Progress

Since the *next* frame in the ongoing frame exchange sequence can be of any type (i.e., CTS, Data, or ACK), the transmission time required by this next frame may differ substantially. However, the IEEE 802.11 does not distinguish among different SR frames and uses the same constant EIFS value in all the cases, which results in large- and small-EIFS problems. Therefore, if somehow a node can distinguish among different type of SR frames and adopts different value of EIFS accordingly, these two problems can be greatly reduced and thus the performance (i.e., fairness and throughput) can be improved. In the light of this discussion, we make two proposals as follows:

1. Whenever a node detects an SR frame, it should try to identify the type of that frame.
2. EIFS value should be directly linked to the type of the SR frame detected.

Since our CS mechanism tries to identify the type of an SR frame rather than just sensing its presence, we call it Enhanced Carrier Sensing (ECS). In Section 5.3.1 we discuss a scheme, which can be used to distinguish among the frames. Once the type of the frame is detected, how to choose the corresponding EIFS value is discussed in Section 5.3.2.

5.3.1 Distinguishing among SR Frames

Basically, there are two methods in which we can distinguish among different type of SR frames. The first method is to use the recent history of frames observed on the medium to decide the type of the current SR frame as the frames are transmitted in a specific order. However, this method is very complex and the recent history may not provide very precise information as there may be multiple FESs in progress at any given time. The second method is to get the information from the frame itself. However, it is not trivial as the node cannot interpret the contents of an SR frame. We propose that the various type of frames should have different lengths (in terms of bytes), and based on the length of an SR frame observed on the medium, the type of the frame is identified.

We first need to differentiate the lengths of different type of frames. In IEEE 802.11 [24], the lengths corresponding to the control frames are as follows, RTS: 20 bytes, CTS: 14 bytes, ACK: 14 bytes. On the other hand, the header of a Data frame is 28 bytes, implying that the length of a Data frame must be greater than 28 bytes. In order to distinguish between CTS and ACK, the size of CTS should be increased by a few bytes. The reason why we increase the

size of CTS rather than ACK is that the extra bytes in the CTS may be used by the receiver to add some receiver status information, which may be useful to the sender before it sends out the Data frame. What is an appropriate number of bytes that should be added to the CTS depends upon the trade-off between the sensitivity of the physical layer and the additional overhead introduced due to these bytes. In our implementation, we change the length of the CTS to 17 bytes.

When a node gets an SR frame, to identify the type of the frame, the node only needs to detect the length (in terms of bytes) of that frame. Before discussing how to detect the length, we have to introduce the format of the physical layer frame in the IEEE 802.11 [24]. As presented in Figure 5.8, a Physical Layer Convergence Protocol (PLCP) frame includes three parts: PLCP Preamble, PLCP Header, and MAC Protocol Data Unit (PDU). The PLCP preamble is used by the receiver to synchronize, while the PLCP header contains information to help the receiver to decide the end of the frame. The MAC PDU corresponds to the MAC layer frame, e.g., a RTS frame.

Figure 5.8: PLCP Frame Format

Now, we discuss how to detect the length (in terms of bytes at the MAC layer) of an SR frame by considering two possibilities. The *first* possibility is that the PLCP header part of the frame can be correctly received by the physical layer but the MAC PDU part cannot be interpreted by the MAC layer. This is very likely due to the following two reasons: (i) In the IEEE 802.11, the PLCP header is always transmitted at the lowest transmission rate (i.e., 1 Mbps) while the MAC PDU in the same PLCP frame may be transmitted at a higher rate (e.g., 2 Mbps or even higher). Generally, the Bit Error Rate (BER) under the case of lower transmission rate should be smaller if the Signal to Noise Ratio (SNR) is assumed to be the same, which should be true during the transmission of an entire PLCP frame. Therefore, the BER for the PLCP header may be much smaller than that for the MAC PDU; (ii) Moreover, the length of the PLCP header (i.e., 6 bytes) is much smaller than that of the MAC PDU (e.g., 20 bytes for an

RTS frame). Due to the above two reasons, it is very likely that the PLCP header is correctly received while the MAC PDU is not. In such a situation, the MAC frame length (and thus the type) can be easily identified. Specifically, when the physical layer detects such a frame, it calculates the length of the MAC PDU based on the following two fields contained in the PLCP header: LENGTH field and SIGNAL field. The LENGTH field indicates the transmission time (in terms of μs) of the MAC PDU while the SIGNAL field indicates the transmission rate (in terms of 100 kbit/s) for the MAC PDU. Therefore, the length (in terms of bytes) of the MAC PDU can be detected if the PLCP header is received correctly. It is necessary to point out that the above discussion also applies for a TR frame containing transmission errors since the PLCP header of a TR frame should always be interpretable.

The *second* possibility is that, when an SR frame is detected, even the PLCP header part in the PLCP frame cannot be correctly received by the physical layer. In such a situation, to identify the type of the MAC frame, we can make use of the Clear Channel Assessment (CCA) mechanism [24]. Specifically, based on the length of the *time duration* that the CCA mechanism indicates a busy medium, if the transmission rate is fixed for all the frames, the MAC layer can calculate the length (in terms of bytes) and thus identify the type. However, when the transmission rate of the MAC PDUs varies, it may not work since two frames of different lengths (in terms of bytes), may have the same transmission time. Therefore, we should make the transmission time needed by various type of frames at different rates unique. It is easy to verify that the control frames among themselves satisfy this requirement if the length of the CTS frame has been changed as discussed above. However, a Data frame of a small length may be confused with the control frames. Therefore, whenever a data frame may be confused with a control frame, we should append enough dummy bytes in the Data frame, so that the transmission time needed by the Data frame is different from that of any control frame at any standard rate. Since there are only a few transmission rates (e.g., four rates in IEEE 802.11b) standardized, the method described above should be easily implemented. Therefore, even in the situation that the PLCP header is not received correctly and that the transmission rate is variable, the type of an SR frame can also be identified with the help of the CCA mechanism. However, for simplicity, in our simulation the transmission rate of the MAC PDU is fixed. Also,

the length of the Data frame is long enough (e.g, 1000 bytes) to avoid the confusion between a Data frame and a control frame.

Further comments: Whenever a node detects an SR frame, whether or not the PLCP header of that frame can be received correctly depends on how large the carrier sensing energy threshold [24] is. If the threshold is large enough, the PLCP header can always be interpreted whenever an SR frame is detected. What is the optimal value of the carrier sensing threshold is out of the scope of this thesis. However, according to the current standards [24], the case that the PLCP header is interpretable (i.e., the *first* possibility discussed before) is more likely to happen than the *second* possibility.

Also note that, while our ECS will get help from the physical layer to detect the type of a frame, it does not need any additional cooperation (compared to the current standards) between the MAC and physical layers, because the length of a frame (needed under the *first* possibility) and the CCA busy time (needed under the *second* possibility) are already conveyed to the MAC layer [24]. Therefore, the only change needed is to add some dummy bytes in the CTS frame, and sometimes in the Data frame if multi-rate is used under the *second* possibility as discussed before. Consequently, ECS can be easily incorporated in the IEEE 802.11.

5.3.2 EIFS Values in ECS

Based on the type of the frame observed on the medium, an EIFS value should be chosen accordingly. The basic rule for deciding the EIFS value is that it should be large enough to allow the complete transmission of the *next* frame in the sequence. Based on this rule, five different values of EIFS are defined in Figure 5.9. For a RTS type SR frame, the EIFS value is equal to SIFS + TxTime (CTS). When a node detects a CTS type SR frame, since the next frame in the sequence is a Data frame whose length may be variable and cannot be detected from the CTS type SR frame, we simply set the EIFS according to the maximum length allowed for the Data frames, i.e. Max-Data-Length. For a Data type SR frame, the EIFS value is set to SIFS + TxTime (ACK). Lastly, for an ACK type SR frame, since this is the last frame in the sequence, the EIFS value is set to DIFS.

In the discussion of the *second* possibility in Section 5.3.1, we have deliberately ignored two complex problems for convenience of description. The first problem occurs as follows. In the

wireless ad-hoc networks, due to the mobility and dynamic propagation characteristics (e.g., fading), when an SR frame is detected by a node, it may only be a part of a frame rather than being a complete frame. We call this partial-frame problem. In such a situation, we may not identify the type of the frame or misunderstand the type of the frame. Another type of problem may occur as follows. Because of the multi-hop scenario in the ad-hoc networks, spatial reuse of the bandwidth is very likely. Therefore, multiple frames may be there around a node at any given instant, resulting in a collision. As a result, the SR frame that a node detects may be an overlapped of several frames rather than a single frame. We call this overlapping-frames problem. In fact, it is very difficult for an MAC protocol to fully take care of these two problems. Therefore, whenever any of these two problems occur, we use the same value of EIFS as defined in the standard [24]. This is indicated by the last line in Figure 5.9. Note that the above two problems does not exist if the PLCP header of an SR frame can be interpreted, which is more likely as discussed before.

The ECS proposed above applies to both the four-way handshaking and the two-way handshaking defined in IEEE 802.11, though the four-way handshaking is more desirable for a multi-hop scenario.

$$\begin{aligned}
 \text{EIFS(RTS)} &= \text{SIFS} + \text{TxTime(CTS)}; \\
 \text{EIFS(CTS)} &= \text{SIFS} + \text{TxTime(Max-Data-Length)}; \\
 \text{EIFS(Data)} &= \text{SIFS} + \text{TxTime(ACK)}; \\
 \text{EIFS(ACK)} &= \text{DIFS}; \\
 \text{EIFS(collision or unknown type)} &= \text{Standard EIFS};
 \end{aligned}$$

Figure 5.9: Different EIFS Values in ECS

5.4 Simulation Results

In the performance evaluation of our Enhanced Carrier Sensing (ECS), the same simulation parameters are used as described in Section 5.2. In Section 5.4.1, the performance is studied for the simple scenarios having only two flows. Then, in Section 5.4.2, two complex scenarios (i.e., chain and double-ring topologies) are studied.

5.4.1 Performance for Scenarios with Two-flows

Scenario-10: This scenario (Figure 5.1) has been used to demonstrate the large-EIFS problem in Section 5.2. The average throughputs under the scenario are presented in Table 5.1. The IEEE 802.11 is very unfair due to the large-EIFS problem at node A. However, under our ECS, the two flows share the bandwidth equally, i.e., each obtains an average throughput of about 0.7 Mbps, and thus we conclude that the large-EIFS problem has been eliminated up to a great extent. Moreover, as shown in Table 5.1, the aggregate throughput under the ECS is slightly greater than that under IEEE 802.11. This can be explained as follows. Consider that after a successful transmission by node B, nodes A and B contend for the medium. Suppose that the node A generates a random back-off timer equal to 10 slots, while the node B generates a random back-off timer with 30 slots, the node A will get control of the medium. If the large-EIFS problem does not occur as in the case of ECS, the medium will be idle for only 10 slots and then node A begins to transmit. On the contrary, if the large-EIFS problem occurs as in the case of IEEE 802.11, the medium will be idle by an extra duration equal to $(EIFS - DIFS)$, i.e., 16 slots. In summary, since the ECS solves the large-EIFS problem, it reduces the idle time of the medium, explaining the throughput improvement in ECS. Though the absolute throughput improvement is not much (about 15 Kbps) in the given scenario, it should be much higher if a high-rate physical layer is used.

Table 5.1: Throughput Comparison for Scenario-10

Throughput (Mbps)	IEEE 802.11	ECS
A to B	0.254	0.705
B to C	1.154	0.718
Aggregate	1.408	1.423

Scenario-11: This scenario (Figure 5.4) has been used to demonstrate the small-EIFS problem in Section 5.2. The average throughputs are presented in Table 5.2. Under the IEEE 802.11, due to the small-EIFS problem, the aggregate throughput is very small (i.e., 0.621 Mbps). However, under our ECS, the aggregate throughput greatly improves (i.e., 1.334 Mbps) as the ECS solves the small-EIFS problem. Note that our ECS does not require capture ability at the physical layer. Also, note that the static routing is used in the simulation. In Appendix C,

we have done a detailed study for this topology where the capture is assumed and the dynamic routing is used. In fact, the performance of this scenario under ECS is identical to that of the hidden-terminal scenario under IEEE 802.11 assuming $SR = TR$.

Table 5.2: Throughput Comparison for Scenario-11

Throughput (Mbps)	IEEE 802.11	ECS
A to B	0.314	0.662
D to C	0.307	0.672
Aggregate	0.621	1.334

Scenario-12: Now, we discuss the results for the scenario shown in Figure 5.10. This scenario is similar to scenario-11 except that the direction of the flows is reversed. It is easy to see that after the node A sends back an ACK frame to node B, node C suffers from the large-EIFS problem if IEEE 802.11 is used. This is also true for node B after the node D sends back an ACK frame to node C. The average throughputs of the two flows are presented in Table 5.3. In contrast to scenario-10, the large-EIFS problem does not result in long-term unfairness in this scenario (i.e., otherwise the flows will have different average throughputs). However, it results in short-term unfairness as shown later.

Table 5.3 also shows that the aggregate throughput under the ECS improves marginally in comparison to the IEEE 802.11. This is because the ECS solves the large-EIFS problem and thus reduces the chance of the medium being idle, as explained in the discussion of scenario-10.

Figure 5.10: 4-nodes with Two Single-hop Flows (Scenario-12)

Table 5.3: Throughput Comparison for Scenario-12

Throughput (Mbps)	IEEE 802.11	ECS
B to A	0.708	0.719
C to D	0.702	0.710
Aggregate	1.410	1.429

In order to study fairness on a short-term basis, Figure 5.11 presents the Jain's index. It is easy to see that the ECS greatly improves the fairness compared to IEEE 802.11. The reason

that the large-EIFS problem leads to short-term unfairness is as follows. Consider that node B successfully transmits an upper-layer packet, and the chance that the node B gets control of the medium *again* is larger than that of node C, as node C suffers from the large-EIFS problem. On the other hand, once node C controls the medium, node C will also have high chance to transmit *consecutively*. Therefore, the large-EIFS problem results in short-term unfairness. However, since the large-EIFS problem occurs at nodes B and C with the same possibility, the long-term fairness between the two flows is ensured.

Figure 5.11: Fairness Index Comparison for Scenario-12

Similarly, the short-term unfairness will also occur in the scenario of Figure 5.12 where the flow between nodes A and B is reversed compared to the scenario in Figure 5.10. However, in Figure 5.12, it is the receivers (i.e., nodes B and D) that suffer from the large-EIFS problem, rather the senders (i.e., nodes B and C) as in Figure 5.10.

Figure 5.12: 4-nodes with Two Single-hop Flows

Scenario-13: Now, we discuss the results for the scenario shown in Figure 5.13, which is similar to scenario-11 except that the distance between nodes B and C is increased to 400 meters. Therefore, nodes B and C are out of the TR but within the SR of each other, and they suffer from the small-EIFS problem as indicated by area 5 in the generalized scenario (Figure 5.6). Table 5.4 presents the throughput results. The aggregate throughput under the IEEE 802.11 is 0.155 Mbps, which is much smaller than that under the ECS (i.e., 0.578 Mbps). However, even in under ECS, the aggregate throughput is still low compared to the maximum

throughput (i.e., about 1.4 Mbps). The low throughput is not due to any deficiency in the ECS. To show this, we need to consider the scenario presented in Figure 5.14 where the SR is assumed to be equal to TR (i.e., 250 m) and the distance between nodes B and C is changed to 200 meters. Note that the scenario of Figure 5.14 is exactly the same as that of Figure 4.2 presented in Chapter 4. Obviously, there is no imprecise-EIFS problem in the scenario of Figure 5.14. Moreover, if the ECS is able to solve the imprecise-EIFS problem in the scenario of Figure 5.13, the scenario should show the same performance as that of Figure 5.14, because when one flow is in progress, the other flow in both the scenarios will behave (e.g., defer) in the identical manner. This is verified by our simulation results.

The reason why the aggregate throughput is not close to 1.4 Mbps in the scenario of Figure 5.14 as well as in the scenario of Figure 5.13 (with ECS) has been clearly explained in Section 4.6.3 of Chapter 4.

Figure 5.13: 4-nodes with Two Single-hop Flows (Scenario-13)

Figure 5.14: Topology (SR=TR) Identical to Scenario-13 (ECS)

Table 5.4: Throughput Comparison for Scenario-13

Throughput (Mbps)	IEEE 802.11	ECS
A to B	0.079	0.290
D to C	0.076	0.288
Aggregate	0.155	0.578

Scenario-14: Now, we discuss the scenario shown in Figure 5.15, which is similar to scenario-13 except that the direction of the flow between nodes C and D is reversed. Contrary to the situation that both nodes B and C suffer from the small-EIFS problem in the scenario-13, only one node (i.e., node C) suffers from the small-EIFS problem in this scenario. The average throughputs are presented in Table 5.5. Under IEEE 802.11, the flow from A to B

is completely starved while the other flow gets the entire bandwidth. Note that in our simulation the static routing is being used, and therefore this starvation is not caused by the failure of the routing discovery process as will be discussed in Appendix C. The reason for the starvation is as follows. As indicated in the generalized scenario (Figure 5.6), whenever a FES(A, B) is in progress, the small-EIFS problem occurs at node C. In other words, whenever there is a Data transmission from A to B, it will always collide with node C's transmission, explaining why the flow from A to B is completely starved. Under the ECS, the flow from A to B gets about 0.075 Mbps, which is still much smaller than that of the other flow. However, again this is not due to the imprecise-EIFS problem, and therefore, the fairness cannot be ensured by the ECS. This can be validated by using an identical scenario (shown in Figure 5.16) where $SR = TR$, as done in the discussion of scenario-13. In fact, the scenario of Figure 5.16 is the same as the *asymmetrical information* scenario discussed before. The unfairness in such a scenario has been clearly explained in Chapter 4.

Figure 5.15: 4-nodes with Two Single-hop Flows (Scenario-14)

Figure 5.16: Topology (SR=TR) Identical to Scenario-14 (ECS)

Table 5.5: Throughput Comparison for Scenario-14

Throughput (Mbps)	IEEE 802.11	ECS
A to B	0.0	0.075
C to D	1.398	1.338
Aggregate	1.398	1.413

Table 5.5 also shows that the aggregate throughput under the ECS improves marginally in comparison to the IEEE 802.11. This can be explained as follows. Under the IEEE 802.11, whenever node B sends back the CTS frame to node A, node C will defer by the EIFS value. However, the Data frame transmitted by node A will be discarded by node B as node C will

send out a frame after it has deferred by the EIFS duration. Therefore, the bandwidth during the EIFS deferment at node C is wasted, explaining why the aggregate throughput under IEEE 802.11 is smaller than that in the ECS. Again, the improvement of ECS in the aggregate throughput should be much higher if a high-rate physical layer is used.

Scenario-15: Now, we discuss the scenario shown in Figure 5.17. In this scenario, node C is out of the TR but within the SR of nodes A and B, while node D is out of the SR of nodes A and B. As indicated by the generalized scenario (Figure 5.6), the node C is in the area 6 and it suffers from the large-EIFS problem after the node B sends back an ACK frame to node A. The average throughputs for this scenario are presented in Table 5.6. It is easy to see that the throughputs under the IEEE 802.11 are very similar to those in scenario-10, as they exhibit the same problem. However, under the ECS, though the fairness substantially improves, the throughputs of the two flows are still not very close to each other (i.e., one is 0.672 Mbps while the other is 0.766 Mbps). Again, this unfairness is not due to any deficiency of ECS, which can be verified by using an identical scenario of Figure 5.18 with $SR = TR$. As explained in Chapter 4, this unfairness is mainly due to the impreciseness of the collision detection mechanism used in IEEE 802.11.

Figure 5.17: 4-nodes with Two Single-hop Flows (Scenario-15)

Figure 5.18: Topology (SR=TR) Identical to Scenario-15 (ECS)

By referring to the generalized scenario discussed in Section 5.2, one can easily design some other scenarios (with two flows) to show the effects of the imprecise-EIFS problem and the advantage of our ECS. However, due to space limitation, we do not present any more such scenarios here.

Table 5.6: Throughput Comparison for Scenario-15

Throughput (Mbps)	IEEE 802.11	ECS
A to B	1.161	0.672
C to D	0.254	0.766
Aggregate	1.415	1.438

5.4.2 Performance for Complex Scenarios

Now, we study the scenarios that have more than two flows.

Scenario-16: Chain Topology

Figure 5.19 shows the chain topology with 10 nodes and 9 single-hop flows. The distance between any two neighboring nodes is 200 meters. The average throughputs are presented in Table 5.7. In the table, the flow ID is the same as the node ID of the flow's sender. We notice that the flow from node 5 to node 6 (i.e., flow with ID 5) is completely starved when IEEE 802.11 is used. However, under ECS, it gets about 0.008 Mbps, though it is still very small compared to the throughputs of other flows. To understand the performance more clearly, in Table 5.8, we present the aggregate throughput, the standard deviation among the average throughputs of the flows, and the Jain's fairness index. Note that the Jain's Index is computed using the average throughputs of the flows, which means that the fairness measurement window is equal to the total number of packets transmitted in the entire simulation time.

From Table 5.8, we find that, under ECS the standard deviation is smaller and the Jain's index is larger, showing that the ECS greatly improves the fairness compared to IEEE 802.11. However, we also notice that the improvement in fairness is at the cost of throughput, since the aggregate throughput under ECS is smaller than that under IEEE 802.11. First of all, the chain topology can be viewed as a combination of topologies in figures 5.1, 5.12, and 5.15. As shown before, the imprecise-EIFS problem in these topologies will lead to unfairness only, but not the throughput degradation. Therefore, solving the imprecise-EIFS problem in the chain topology does not mean the aggregate throughput will improve, though the fairness should definitely improve, which is true in our results.

In fact, in the chain topology (Figure 5.19), the aggregate throughput is directly related to the contention in the network. To maximize the throughput, one can just starve all the flows

except the flows 0, 4, and 8, which do not contend with each other, leading to an aggregate throughput of 4.2 Mbps (i.e., 1.4×3). It is easy to verify that, as long as the throughputs of the flows with IDs 1, 2, 3, 5, 6, and 7 are greater than zero, the aggregate throughput must be smaller than 4.2 Mbps, which is true under both IEEE 802.11 and ECS. Moreover, with the increase of the throughputs of these flows, the aggregate throughput will decrease. This explains why the aggregate throughput under ECS is smaller than that under the IEEE 802.11 as the flows 1, 2, 3, 5, 6, and 7 get more bandwidth under ECS. In fact, in multi-hop wireless ad-hoc networks where spatial-reuse is possible, maximizing throughput and achieving fairness are generally conflicting objectives as pointed out in [42].

We should also notice that, even the ECS does not distribute the bandwidth very fairly among the flows. Indeed, without deploying a scheduling algorithm (e.g., the one discussed in [42]) on the top of the MAC layer, it is very difficult for the MAC protocol itself to achieve the fairness among all the flows in a multi-hop scenario.

Figure 5.19: Chain Topology: 10 Nodes with 9 Flows

Table 5.7: Throughput Comparison for the Chain Topology

Flow ID	0	1	2	3	4	5	6	7	8
IEEE 802.11	0.517	0.054	0.157	0.131	0.55	0.0	0.242	0.202	0.967
ECS	0.334	0.178	0.322	0.187	0.175	0.008	0.329	0.445	0.638

Table 5.8: Performance Measures Comparison (Chain Topology)

Metrics	Aggregate	STD DEV	Jain's Index
IEEE 802.11	2.820	0.292	0.536
ECS	2.616	0.172	0.742

Scenario-17: Double-ring Topology

Figure 5.20 shows the double-ring topology with 16 nodes. The distance between the sender and the receiver of a flow is 200 meters. The diameter of the inner circle is also 200 meters, and thus the diameter of the outer circle is 600 meters. The angle between any two neighboring flows is 45 degrees.

The performance measures are presented in tables 5.9, and 5.10. Again, the ID of a flow is the same as the node ID of the flow's sender. In contrast to the chain topology, the ECS greatly improves the aggregate throughput in this topology. As for the fairness, both the standard deviation and Jain's index under ECS are slightly greater than those under IEEE 802.11, which may convey very conflicting conclusions. If the standard deviation alone is considered, it seems that the ECS is more unfair than IEEE 802.11. On the other hand, the Jain's index is indicating that the ECS is more fair. Because the throughputs under the two schemes differ by an order of magnitude, a larger standard deviation does not imply that ECS is more unfair. Therefore, standard deviation is not always a good measure of fairness as we have stated in Chapter 2.

We have also conducted the simulation for some more complex scenarios (e.g., grid topologies). It is found that the ECS always improves the performance of IEEE 802.11.

Figure 5.20: Double-ring Topology: 16 Nodes with 8 Flows

Table 5.9: Throughput Comparison for the Double-ring Topology

Flow ID	0	2	4	6	8	10	12	14
IEEE 802.11	0.0169	0.0214	0.0197	0.0195	0.0207	0.0206	0.0197	0.0209
ECS	0.1586	0.1587	0.1542	0.1517	0.1641	0.1627	0.1663	0.1483

Table 5.10: Performance Measures Comparison (Double-ring Topology)

Metrics	Aggregate	STD DEV	Jain's Index
IEEE 802.11	0.1594	0.0013	0.9957
ECS	1.2646	0.0059	0.9986

5.5 Conclusions

In this chapter, we have addressed the following issue: *whenever a node detects an erroneous frame (e.g., a sensing range frame or a transmission range frame with a wireless error), how long should the node defer its transmission to prevent the potential interference on the ongoing transmission over the shared medium?* This issue has not been addressed in the literature. In IEEE 802.11, whenever a node detects an erroneous frame, it always defers by a fixed duration (represented by EIFS). Due to this fixed EIFS value, we showed that two problems arise: small-EIFS problem and large-EIFS problem, which lead to considerable unfairness and throughput degradation. In order to solve the problems, we have proposed an enhanced carrier sensing (ECS) mechanism. In the ECS, the lengths of the frames are made different. Based on the length of a frame observed on the medium, the *type* of the frame can be detected, and the node defers the transmission for a duration accordingly, rather than by the fixed duration. In order to obtain the length of an erroneous frame, the ECS utilizes the information provided by the physical layer. However, the ECS does not need any enhancement at the physical layer or on the cooperation between the MAC and physical layers. Therefore, the ECS can be easily incorporated into IEEE 802.11. Our extensive simulation results have shown that the ECS improves the fairness as well as throughput drastically.

Compared to the large number of published works that focus on the collision avoidance (CA) or contention resolution (CR) algorithm to improve the fairness or throughput in the CSMA/CA-based MAC protocols (e.g., IEEE 802.11), our work is unique as we have focused on the carrier sensing (CS) mechanism to improve both the fairness and the throughput, which is not available in the literature.

Chapter 6

Full Auto Rate (FAR)

In the previous chapter, we have studied the performance under the assumption of $SR > TR$, which is true in practice in IEEE 802.11. In this chapter, we study the performance under the assumption that the transmission rate is adjusted adaptively, which is again practised in IEEE 802.11. We also propose a Full Auto Rate (FAR) protocol, which is able to adapt the transmission rate of all the frames (both control and Data frames).

6.1 Introduction

At the physical layer, the IEEE 802.11a [25], 802.11b [26] and 802.11g [27] standards provide the multi-rate capability. For example, data rates defined in IEEE 802.11b are 1, 2, 5.5, and 11 Mbps, while the 802.11a and 802.11g provide a data rate up to 54 Mbps. To fully make use of the multi-rate capability, the MAC protocol should choose a transmission rate in an adaptive manner, known as *link adaptation*, *rate adaptation*, or *auto rate*. The basic idea of rate adaptation is to dynamically estimate the channel conditions and then select a transmission rate that will give the optimum throughput under the given channel conditions. In IEEE 802.11 standards, the rate adaptation algorithm has been intentionally left open.

Recently, numerous algorithms [23, 52, 34, 45, 48, 13] have been proposed to support multi-rate operation in IEEE 802.11. These algorithms consider the rate adaptation of the Data frame only, while the control frames (RTS/CTS/ACK) are always transmitted at a very low basic rate (e.g., 1 or 2 Mbps in IEEE 802.11b). Based on whether the rate adaptation function for the Data frame is performed at the sender or the receiver side, these algorithms can be classified into two categories: sender-based and receiver-based. The algorithms in [34, 45, 48, 13] fall into the first

category while those in [23, 52] belong to the second category. The sender-based algorithms are simple and easy to incorporate into the standards. However, since it is the receiver (rather than the sender) that can perceive the channel quality, the receiver-based method adapts much better to the channel conditions. The receiver-based algorithms rely on the existence of the RTS/CTS handshaking. Specifically, based on the Signal to Noise Ratio (SNR) of the received RTS frame, the receiver estimates the channel conditions, and then in the CTS frame the receiver suggests a data rate that the sender should use for the transmission of the Data frame. In the ad-hoc networks, since the RTS/CTS handshaking is always used to cope with the common hidden-terminal problem, the receiver-based algorithm is more desirable. However, since the control frames (RTS/CTS) are always transmitted at a very low data rate, the bandwidth is greatly wasted by the control frames. Moreover, with the increase of the maximum data rate in the emerging standards (e.g., 54 Mbps in 802.11a and 802.11g), the proportion of the wastage increases. Therefore, it is of great importance to adapt the transmission rate of the control frames in ad-hoc networks.

In this chapter, we propose a new rate adaptation algorithm, called Full Auto Rate (FAR), which can adapt the transmission rate of both the Data and the control frames in the wireless ad-hoc networks. Our FAR combines the sender- and receiver-based methods proposed in the literature, that is, the rate adaptation of the RTS/CTS frames is done at the sending side of these frames while that for the Data/ACK frames is done at the receiving side of the frames. Moreover, in order to cope with the issues involved in the virtual carrier sensing (VCS) in multi-rate IEEE 802.11, we have proposed a modified virtual carrier sensing (MVCS) mechanism. Both analytical and simulation results show that the FAR greatly improves the performance of IEEE 802.11.

The remainder of the chapter is organized as follows. In Section 6.2, we present the motivation to adapt the transmission rate of the control frames. In Section 6.3, the full auto rate (FAR) protocol is presented. Sections 6.4 and 6.5 present the analytical and simulation results, respectively. The chapter is concluded in Section 6.6.

6.2 Motivation

6.2.1 Transmission Rate of the Control Frames

IEEE 802.11 specifies that all the control frames should be transmitted at one of the rates in the *basic rate set* so that all the potentially interfering nodes can decode the duration information carried in these frames [24]. Obviously, the basic rate set contains only one or several lowest transmission rates among all the supported rates. For example, in IEEE 802.11b that supports rates of 1, 2, 5.5 and 11 Mbps, the basic rate set normally contains only 1 and 2 Mbps¹. Another rule regarding the transmission rate of the control frames is as follows [24]: in order to allow the transmitting node to calculate the duration field value, the responding node should transmit its control response frame (either CTS or ACK) at the highest rate in the basic rate set, which is less than or equal to the rate of the received frame. Since there is no rule about how to adapt the transmission rate of the RTS frame, a conservative implementation will choose to transmit it at the lowest rate, i.e., 1 Mbps. As a result, the CTS will also be transmitted at 1 Mbps. On the other hand, according to the above rule, the ACK frame can be transmitted at 2 Mbps whenever the transmission rate of the Data frame is equal to or greater than 2 Mbps. However, in most of the rate adaptation algorithms, they simply assume that all the control frames are always transmitted at the lowest basic rate (i.e., 1 Mbps in IEEE 802.11b). As a result, the bandwidth wastage by the control frames is considerable and fixed irrespective of the length and the transmission rate of the Data frame. In the following, we develop a simple analytical model to show how considerable this wastage is.

6.2.2 Maximum Throughput of IEEE 802.11

If we want to consider the maximum throughput of IEEE 802.11 under the saturation condition, as done in Chapter 2, we need to find the probability of collision and the probability that the medium is idle, which require a detailed analysis. However, since our main objective here is to show the fixed overheads introduced by the physical and MAC layers, we develop a simple model under the assumptions that only one active single-hop flow is present in the network and that no wireless errors occur. Clearly, the throughput obtained under these assumptions is the

¹This is also to ensure the coexistence with the legacy IEEE 802.11 products that only support 1 and 2 Mbps.

maximum that IEEE 802.11 can achieve. Under the assumptions, the average time required to transmit a packet is:

$$T = DIFS + (CW_{min}/2)\sigma + 3SIFS + T_{RTS} + T_{CTS} + T_{Data} + T_{ACK} \quad (6.1)$$

where σ represents the slot time and is specified by the physical layer. Clearly, $(CW_{min}/2)\sigma$ is the average back-off time. T_{RTS} , T_{CTS} , T_{Data} , and T_{ACK} represent the transmission time of the corresponding physical layer frames. For a specific physical layer frame, the transmission time T_{frame} is:

$$T_{frame} = L_{frame}/R_{frame} + PHY_{hdr} \quad (6.2)$$

where L_{frame} and R_{frame} are the length of the MAC layer frame and the transmission rate chosen by the MAC layer, respectively. The PHY_{hdr} is the physical layer overhead in units of time. Since $DIFS$, CW_{min} , σ , $SIFS$, and PHY_{hdr} are fixed for a specific physical layer, we denote the sum of the fields containing them as C_{phy} . That is,

$$C_{phy} = DIFS + (CW_{min}/2)\sigma + 3SIFS + 4PHY_{hdr} \quad (6.3)$$

Since the lengths of the RTS, CTS, ACK, and the MAC layer header of Data frame are fixed, i.e., 20, 14, 14, and 28 bytes, respectively, Equation (6.1) becomes,

$$T = C_{phy} + 8(20/R_{RTS} + 14/R_{CTS} + (L + 28)/R_{Data} + 14/R_{ACK}) \quad (6.4)$$

where L is the length of the data packet handed to the MAC layer from the upper layer, while R_{RTS} , R_{CTS} , R_{Data} and R_{ACK} are the transmission rates (in terms of Mbps) chosen by the MAC layer for the corresponding frames. Therefore, the maximum throughput (in terms of Mbps) is as follows:

$$Th = \frac{L \times 8}{T} \quad (6.5)$$

Figure 6.3 (in Section 6.4) presents the numerical results corresponding to the IEEE 802.11b, where the R_{RTS} and R_{CTS} are equal to 1 Mbps, while the R_{ACK} is equal to 1 or 2 Mbps determined by R_{Data} as discussed in Section 6.2.1. The C_{phy} correspondingly is 1168 μs . The results are obtained by varying the packet length L and the transmission rate of the Data frame

R_{Data} . It is easy to see that the throughput increases with L or R_{Data} . Moreover, the overhead is considerable. For example, when L is 64 bytes and R_{Data} is 11 Mbps, the throughput is very small, about 0.33 Mbps.

To reduce the overhead, several ways are possible as known from equations (6.3) and (6.4). First, as done in [10], the contention window can be tuned based on the contention level (e.g., number of active nodes) in the network. The tuning method reduces the time that the medium is idle when the contention level is low, and reduces the probability of collisions when the contention level is high. The second method is to reduce the physical layer overheads (e.g., preamble) as done in [61]. Another method is to reduce the usage of RTS/CTS as much as possible. For example, whenever a node gets control of the medium, it transmits several data packets consecutively (i.e., a burst) in which the RTS/CTS is transmitted only once as done in [52]. In contrast, in this chapter, we aim to transmit the control frames at a higher rate to reduce the overheads. In fact, as shown in the analytical and simulation results presented later, the throughput will be improved to a great extent by using this method.

6.3 Full Auto Rate (FAR) Algorithm

In this section, we first discuss the VCS issues in the multi-rate scenario, and introduce a modified virtual carrier sensing (MVCS) mechanism assuming that the control frames are always transmitted at the low basic rate. Then, we show that the control frames can also be transmitted at a higher rate with the help of EIFS deferment. Finally, we present our FAR, and discuss how the MVCS should be refined in the FAR where the rates of all the frames are adapted.

6.3.1 VCS Issues in Multi-rate IEEE 802.11

As mentioned in Chapter 1, for the operation of VCS, every frame should contain a *duration field*, which stores the time duration that the medium is reserved for *all* the remaining frames in the on-going sequence. The duration values carried in the RTS, CTS, Data, and ACK frame are represented by D_{RTS} , D_{CTS} , D_{Data} , and D_{ACK} , respectively. Clearly, to calculate the duration value for a frame, the transmission rates and the lengths of all the remaining frames in the sequence must be known a priori. For example, to calculate the duration value carried in the RTS frame (i.e., D_{RTS}), the sender must know the transmission rates and the lengths of

CTS, Data, and ACK frames. As the lengths of all the frames are known at the sender, and if the transmission rate of the RTS/CTS is assumed to be 1 Mbps, so for the VCS to operate properly, only the transmission rate of the Data frame has to be known before the transmission of the RTS. In a sender-based rate adaption algorithms, this rate is certainly known. However, in the receiver-based rate adaptation algorithms such as RBAR [23] and OAR [52], since the actual transmission rate of the Data frame is determined by the receiver after it has received the RTS frame, the sender has to calculate the D_{RTS} by assuming a transmission rate for the Data frame. Whenever the rate chosen by the receiver is different from the one assumed by the sender, the reservation made by D_{RTS} is inaccurate.

The reservation may be corrected by D_{CTS} or D_{Data} , which is always precise if the rate of the control frames follows the rules described in Section 6.2.1. However, two problems arise. The first one is that since the Data frame may not be transmitted at the basic rate, it may not be decoded correctly by the overhearing nodes and thus the reservation made according to D_{RTS} cannot be corrected by D_{Data} . The second problem occurs as follows. When the actual rate of the Data frame is larger than the rate assumed by the sender, the remaining NAV value previously updated according to D_{RTS} is larger than the new duration value contained in the CTS or Data frame. As mentioned in Chapter 1, to ensure the proper operation of the VCS in a multi-hop network, IEEE 802.11 does not allow reduction in NAV. Therefore, the reservation made according to D_{RTS} also cannot be corrected in this case.

To cope with the first problem, the RBAR [23], which is the only one that has considered the above problems, defines a sub-header, called the Reservation SubHeader (RSH), to replace the original header of the Data frame. The RSH includes a separate Frame Check Sequence (FCS) and should be transmitted at one of the basic rates to ensure all the overhearing nodes receive it correctly. Moreover, the physical layer frame needs to be modified to include fields indicating the rate of the RSH header and the rate of the actual Data part, and thus the physical layer is required to switch transmission rates twice during the transmission of the payload. To cope with the second problem, the VCS is modified to allow the NAV value to be reduced by the frames belonging to the *same* frame exchange sequence. To achieve this, every node has to maintain a history of the NAV updates. Obviously, the RBAR designed as above greatly

increases the complexity as well as overheads, and thus it may not be practically useful as pointed out in [48].

The above discussion assumes that the rate of the control frames follows the rules discussed in Section 6.2.1. It is obvious that the situation becomes more complex when the transmission rate of the control frames is also made variable, which is one of the objectives in this chapter. In fact, in such a scenario, even the sender-based algorithms need to cope with the problems caused by the inaccurate reservation. Therefore, it is crucial to devise a new VCS mechanism for the multi-rate IEEE 802.11.

6.3.2 Modified Virtual Carrier Sensing (MVCS)

From the previous subsection, we know that the main problem in the current VCS for multi-rate is that it needs to know the transmission rate of *all* the remaining frames. However, we must realize that the main purpose of the VCS, a complementary component of the physical carrier sensing, is to allow the next frame in the ongoing sequence to go through. Therefore, instead of reserving the medium for *all* the remaining frames in the sequence, we propose to reserve the medium only for the *immediate next* frame². We call it Modified Virtual Carrier Sensing (MVCS).

The MVCS can operate properly with the help of the physical carrier sensing. We show this using the scenario present in Figure 6.1. If a node (e.g., node C) is within the range of the sender as well as the receiver, the node can defer properly by using only the physical CS mechanism. Now let us consider the nodes that are not within the range of both the sender and the receiver. Figure 6.2 presents the main process of the MVCS at these nodes. For example, node A is within the range of the sender but not within the range of the receiver. When node A overhears an RTS frame, under the MVCS, it defers by a duration to allow the CTS to pass through. Then, when the *Src* is transmitting the Data frame, the physical CS at node A makes itself to defer. After node *Src* has transmitted the Data frame, node A defers for a duration to allow the transmission of the ACK frame. Therefore, under MVCS, the nodes within the range of the sender but not of the receiver can defer properly. Note that it is not possible for node

²In fact, in the ECS presented in Chapter 5, we have used a similar concept to determine the value of the EIFS.

A to initiate its transmission during the SIFS interval at the *Src* just before transmitting the Data frame. The reason is, after node A has deferred by D_{RTS} it will further defer by a DIFS, which is larger than SIFS. Similarly, the nodes (e.g., node B) that are within the range of the receiver but not of the sender, can defer properly with the duration value carried in the CTS and ACK frames. Therefore, all the nodes that potentially interfere with the ongoing frame exchange sequence can defer properly.

Figure 6.1: Scenario Showing the Range of Interference

Figure 6.2: Modified Virtual Carrier Sensing (MVCS)

So far, we have assumed that all the frames are overheard clearly by the interfering nodes, which may not be true in a practical scenario. For example, when the sensing range is greater than the transmission range, some nodes can detect the carrier of a frame but cannot interpret the contents. However, with the help of the EIFS deferment, all the nodes defer properly as discussed later.

Clearly, the MVCS reduces the overheads as well as the complexity in the receiver-based rate adaptation algorithms. For example, the Reservation SubHeader (RSH), the maintenance of NAV update history, and the modifications to the physical layer are not required any more.

Moreover, since the MVCS modifies the VCS of IEEE 802.11 slightly only, it can be easily incorporated into the receiver-based auto-rate algorithms (e.g., RBAR). More importantly, the MVCS forms the basis to adapt the transmission rates of the control frames.

6.3.3 Rate Adaptation of Control Frames

As discussed before, IEEE 802.11 specifies that all the control frames (i.e., RTS, CTS and ACK) should be transmitted at one of the basic rates to ensure that all the nodes in the interference range can interpret them. Moreover, it can be easily verified by checking the frame format defined in the standards that in an ad-hoc network the only useful information contained in the RTS/CTS/Data/ACK for the *overhearing* nodes is the duration value. On the other hand, as in the discussion of the MVCS, to prevent potential interference from other nodes, a frame needs to reserve the medium only for the next frame in the sequence. Therefore, if somehow the overhearing nodes can defer properly to allow the next frame pass through even when they cannot interpret the contents of the overheard frames, the control frames can also be transmitted at a rate higher than the basic rates. This can be achieved with the help of the EIFS deferment as explained in the following. Since the EIFS is equal to SIFS + TxTime (ACK) + DIFS (i.e., $364 \mu\text{s}$ in DSSS) [24], it is long enough for the complete transmission of any of the control frames at any rate. For example, when a node detects an erroneous frame corresponding to the RTS, it defers by an EIFS value, which is large enough to allow the CTS to go through. This is also true when a node detects an erroneous frame corresponding to a Data or ACK. Therefore, we propose that the RTS and ACK frames should be always transmitted at the highest attainable rate under the given channel conditions. However, when the detected erroneous frame corresponds to a CTS frame, since the transmission time required by the Data frame may be greater than the EIFS, the overhearing nodes, relying solely on the EIFS deferment, may not defer properly. Therefore, in such a case, the CTS should be transmitted at the lowest basic rate to ensure that the CTS could be clearly understood even by the overhearing nodes. In contrast, when the transmission time required by the Data frame is smaller than the EIFS, the CTS should be transmitted at the highest attainable rate. Note that with the increase of the maximum data rate in the emerging standards (e.g., 54 Mbps in IEEE 802.11a and g), the chance of the transmission time of the Data frame being smaller than the EIFS increases.

6.3.4 Full Auto Rate (FAR))

Now we present our Full Auto Rate (FAR) algorithm, which integrates the above two proposals, i.e., MVCS and rate adaptation of control frames. Table 6.1 lists the main operations of the FAR. We first discuss the rate adaptation of the frames (see second column of the table). Our FAR combines the sender- and receiver-based methods proposed in the literature. Since within a specific frame exchange sequence the RTS and CTS are the first frames transmitted in the corresponding directions, the rate selection has to be done at the sender side of these frames. On the other hand, since the RTS/CTS frames can probe the channel conditions in the two directions, the rate selection for the Data/ACK frames can be done at the receiver side of these two frames. Specifically, the transmission rate of the RTS is chosen as follows. Every node maintains a cache variable, which indicates the rate that should be used for the RTS. The initial value of this cache variable is set to the lowest basic rate (i.e., 1 Mbps). Whenever an attempt of transmitting an RTS or a Data frame is successful, the variable is updated with the rate being used for this successful transmission. Otherwise, whenever the attempt fails, the variable is reset to the lowest basic rate. The rate selection for the CTS frame is a little bit more complex. If the transmission time of the Data frame (which can be computed at the receiver after it has chosen a rate for the Data frame) is greater than the EIFS value, the CTS frame is transmitted at the lowest basic rate to ensure that it is understood by all the nodes in the interference range. Otherwise, the CTS is transmitted at a rate that is determined by using a similar method as the rate selection for the RTS frame. Obviously, for the RTS/CTS frames, one can adopt any other sender-based algorithm (e.g., [34]) to decide the rate. Now we discuss how the transmission rates of the Data/ACK frames are decided by the receiving side of these frames. Specifically, based on the SNR of the received RTS, the receiver suggests a transmission rate for the Data frame and piggyback this rate in the CTS as done in the RBAR. Similarly, the receiving side of the CTS frame chooses a rate for the ACK frame based on the SNR of the CTS and piggyback the rate in the Data frame.

Now we discuss how the MVCS proposed before should be refined to operate in our FAR (see third and fourth columns of Table 6.1). In the FAR, before the transmission of the RTS frame, the sending node does not know the exact transmission rate of the CTS frame in the

sequence. Therefore, the medium should be reserved as if the CTS was always transmitted at the lowest basic rate. Moreover, as the length of CTS and the lowest basic rate are fixed values, the corresponding duration is also fixed, which can be easily calculated at every node overhearing RTS. Therefore, instead of having the duration value in the RTS frame, it carries the length of the Data frame in bytes, which is needed by the receiver to calculate the D_{CTS} . On the other hand, before the transmission of the CTS and Data frames, the corresponding sending node always knows the exact transmission rate of the immediate next frame in the sequence. Therefore, the CTS and Data frames should simply carry the duration in microseconds, which is long enough for the next frame in the sequence to pass through. Note that the duration value in the ACK frame is zero. Also note that the rate chosen for the Data and ACK frames can be easily derived from the duration value carried in the CTS and Data frames, respectively. Therefore, it is unnecessary to add an additional field to store the rate being chosen. As for the overhearing nodes, when they overhear an RTS frame, they defer by a fixed duration that is long enough for the CTS to be transmitted at the lowest basic rate. Otherwise, when they overhear any other types of frame, they simply defer by a duration advertised in the overheard frame. Note that the deferment when overhearing an RTS may not be precise since the CTS may be transmitted at a higher rate. However, the deviation is very small.

Table 6.1: Main Operation in Full Auto Rate

Frame	Transmission Rate	Duration Field	Deferment Time (overhearing nodes)
RTS	Cached rate	Data length (in bytes)	$TxTime(CTS, \text{Basic-rate})$
CTS	Cached rate or Lowest basic-rate	Data length / Chosen rate	Duration value
Data	Piggybacked in CTS	ACK length / Chosen rate	Duration value
ACK	Piggybacked in Data	0	Duration value

6.4 Analytical Modelling

In this section, based on the model present in Section 6.2, we analytically show the advantages of FAR. As clearly shown in [23], in wireless ad hoc networks, a receiver-based algorithm has much better performance than a sender-based algorithm (e.g., [34]). Therefore, here we only

compare the maximum throughputs of three schemes: RBAR [23], RBAR with MVCS, and FAR. Also, we do not consider the performance of the OAR [52] since it is mainly based on the RBAR.

RBAR with MVCS versus RBAR: When the original VCS is used, the RBAR transmits the RSH sub-header of the Data frame at the same rate as that of RTS (i.e., 1 Mbps). Moreover, RSH sub-header needs a separate four-bytes FCS. On the contrary, with MVCS all these are unnecessary. Therefore, using Equation (6.5), the ratio between the maximum throughputs of the RBAR with MVCS and the RBAR is:

$$r = \frac{C_{phy} + 8(48/R_{RTS} + 14/R_{CTS} + (L+4)/R_{Data} + 14/R_{ACK})}{C_{phy} + 8(20/R_{RTS} + 14/R_{CTS} + (L+28)/R_{Data} + 14/R_{ACK})} \quad (6.6)$$

FAR versus RBAR with MVCS: Under FAR, we assume that all the control frames are transmitted at the same rate as the Data frame, therefore the ratio is:

$$r = \frac{C_{phy} + 8(20/R_{RTS} + 14/R_{CTS} + (L+28)/R_{Data} + 14/R_{ACK})}{C_{phy} + 8(20/R_{Data} + 14/R_{Data} + (L+28)/R_{Data} + 14/R_{Data})} \quad (6.7)$$

FAR versus RBAR: Similarly, the ratio between the maximum throughputs under FAR and RBAR is as follows:

$$r = \frac{C_{phy} + 8(48/R_{RTS} + 14/R_{CTS} + (L+4)/R_{Data} + 14/R_{ACK})}{C_{phy} + 8(20/R_{Data} + 14/R_{Data} + (L+28)/R_{Data} + 14/R_{Data})} \quad (6.8)$$

Figures 6.4, 6.5, and 6.6 present the results computed from the equations (6.6), (6.7), and (6.8), respectively. Note that in Figure 6.5 when the R_{data} is equal to 1 Mbps, the ratio is always one, which is not shown in the figure. From these results, we see that when all the other parameters are fixed, as R_{data} increase, the throughput ratio also increases. In fact, one can formally prove this by showing that the first-order derivative with respect to R_{data} is always greater than zero in equations (6.6), (6.7), and (6.8). On the other hand, the first-order derivative with respect to C_{phy} is always smaller than zero, implying that a smaller C_{phy} leads to a higher ratio. The above two properties of the ratios are crucial because the C_{phy} becomes smaller while the R_{data} becomes larger as IEEE 802.11 standards evolve. For example, in IEEE 802.11a [25], C_{phy} is reduced to 226.5 μs , while R_{data} can reach 54 Mbps. The first-order derivative with respect to L is always smaller than zero, implying that a larger L leads to a smaller ratio. However, when the L increases, the throughput also increases as

Figure 6.3: Maximum Throughput in IEEE 802.11b

Figure 6.4: Throughput Ratio between RBAR with MVCS and RBAR

Figure 6.5: Throughput Ratio between FAR and RBAR with MVCS

Figure 6.6: Throughput Ratio between FAR and RBAR

shown in Figure 6.3. Therefore, though the ratio is decreasing with the increase in L , the absolute throughput improvement (in terms of Mbps) is higher, which is also observed from the simulation results in Section 6.5.

6.5 Simulation Results

The simulations were performed under the NS-2 with CMU wireless extensions [15]. As in [52], we use the fading model with Ricean probability density that has been incorporated into the NS-2 by the authors of [47]. The optimum data rate for a given SNR is selected according to [52].

In the simulation, seven different algorithms are considered. Four of them are run at a fixed transmission rate of 1, 2, 5.5, or 11 Mbps as supported in IEEE 802.11b [26]. The other three are rate adaptation algorithms: RBAR, RBAR with MVCS, and FAR. Again, here we do not

compare the performance of FAR with the sender-based algorithms (e.g., [34]) and OAR [52] due to the reasons discussed at the beginning of Section 6.4. For each of the algorithms, we conduct the simulation with five different packet lengths, i.e., 64, 128, 256, 512, 1024 bytes. For each of the flows, the source rate is large enough to occupy the whole bandwidth. In the single-hop scenario, the static routing is used while in the multi-hop scenario the Ad-hoc On-demand Distance Vector (AODV) [46] is used as the routing protocol. In the simulation, the random waypoint mobility model [9] is employed, where the pause time is zero and the maximum speed of mobility is 5 meters/s. The transmission range as well as the sensing range is 250 meters.

6.5.1 Single Flow within Single-hop

In this simulation, there are only two nodes, which are continuously moving within a 200×200 meter arena. Figure 6.7 presents the average throughputs. Clearly, the algorithms with rate adaptation have better performance than all the algorithms with fixed rate. Within the fixed rate algorithms, the one at the rate 5.5 Mbps shows the best performance, implying that the average distance of the two nodes in this scenario is close to the optimal distance under which the rate of 5.5 Mbps achieves the best performance. Moreover, for all the fixed rate algorithms, the throughputs conform to the upper bound derived in Section 6.2.

Figure 6.7: Throughput under Single-flow within Single-hop

Among the adaptive algorithms, as expected, the FAR shows the best performance while the RBAR with MVCS is the second best. Moreover, the longer the packet length is, the more absolute gain is achieved by the FAR over the RBAR. For example, when the packet length

is 1024 bytes, the throughput improves by 0.5 Mbps whereas the improvement is about 0.08 Mbps when the packet length is 64 bytes. The reason is that for a packet of small length, the transmission time reduced by the higher rate is very small compared to the fixed overheads. However, the results are somewhat different from the throughput ratio viewpoint. Figure 6.8 presents the throughput ratio among the adaptive rate algorithms. The larger the packet length, the smaller is the ratio. This has also been reflected by the analytical results in Section 6.4. Note that the ratio obtained from the simulation is smaller than that from the analytical results at the rate of 5.5 Mbps. The reason is, in the analysis the rate of RTS/CTS under FAR is assumed equal to the rate of the Data frame, which is not always true in practice.

Figure 6.8: Throughput Ratio under Adaptive Rate Algorithms

6.5.2 Multiple Flows within Single-hop

In this simulation, there are ten nodes with five flows and all the nodes are continuously moving within a 200×200 meter arena. Figure 6.9 presents the aggregate average throughput of all the flows. In general, the performance among the seven algorithms follows the same trend as in the previous scenario except for the following two observations: (i) among the fixed rate algorithm, it is the algorithm with 11 Mbps that shows the best performance rather than the one with 5.5 Mbps; and (ii) the throughput under RBAR is very close to the fixed rate algorithm with 11 Mbps. The reason for the first observation is as follows. Under the fixed rate algorithm with 11 Mbps, all the transmissions experiencing a SNR that is not large enough to support 11 Mbps will fail, making the flows continuously back off and defer. Moreover, since in this scenario there

are 10 nodes within a small area, it is very likely that at least one of the flows can support the 11 Mbps. Therefore, most of the time the medium will be occupied by the flows with 11 Mbps, explaining why the 11 Mbps gets the best performance in this scenario. On the other hand, in the previous scenario in which only two nodes are present, the nodes may not communicate at 11 Mbps during most of the time, explaining why the 5.5 Mbps gets the best performance. Now we explain the second observation. Compared to the fixed rate algorithm with 11 Mbps, the RBAR will allocate more time to the flows that can only support rates lower than 11 Mbps, reducing the aggregate throughput of RBAR. On the other hand, the time wasted by the flows that cannot support 11 Mbps in the fixed rate algorithm is greatly reduced by the RBAR since it can adapt to the channel conditions. Therefore, the above two effects cancel out each other, and finally result in similar performance. However, the RBAR is better as it is fairer than the fixed 11 Mbps algorithm.

Figure 6.9: Throughput under Multiple-flows within Single-hop

6.5.3 Multiple Flows within Multi-hop

In this scenario, there are twenty nodes with five flows and all the nodes are moving within a 1000×500 meter arena. Figure 6.10 presents the aggregate average throughput of all the flows. The performance of the adaptive rate algorithms follows the same trend as in the previous examples, i.e., the FAR is the best, followed by RBAR with MVCS, and then by RBAR. However, in terms of the aggregate throughput, the adaptive algorithms do not show any distinct advantage compared to some of the fixed rate algorithms. For example, the fixed rate

algorithm with 5.5 Mbps always shows the best performance, while RBAR is inferior to most of the fixed rate algorithms except the one with 1 Mbps. Through detailed analysis of the trace records, we found that the interaction between MAC layer and the routing protocol plays a major rule in determining the performance. Under the fixed rate, whether or not a multi-hop flow can start depends on whether that flow can find a route from the source to the destination with all the hops having that fixed rate. When the fixed rate is very high (e.g., 11 Mbps), it is very likely that sometimes no flows in the network can find a route. On the other hand, when the fixed rate is very low (e.g., 1 or 2 Mbps), all the flows may be active but the medium is used at a very low rate. This explains why the fixed rate algorithm with 5.5 Mbps shows the best performance. Under the adaptive rate algorithm, it is very likely that all the flows can find a route since the nodes can adjust the rate based on the channel conditions. Therefore, though the aggregate throughputs of the adaptive algorithms may be smaller than that under the fixed rate algorithms, the bandwidth is more evenly distributed among the flows.

Figure 6.10: Throughput under Multiple-flows within Multi-hop

6.5.4 Lessons Learnt from the Simulation

Three important conclusions are obtained: (i) the FAR improves the performance of IEEE 802.11 in all the scenarios; (ii) it is not enough to compare the performance only in terms of throughput in the scenarios with multiple flows. Fairness should also be considered; (iii) In the multi-hop networks, the interaction between the routing protocol and MAC protocol greatly affects the overall performance. Therefore, it is of great importance to do cross-layer optimization in wireless ad hoc networks.

6.6 Conclusions

In this chapter, we have introduced a new rate adaptation algorithm, called Full Auto Rate (FAR). In contrast to the published schemes, which only consider the rate adaptation of the Data frame while assuming that the control frames are always transmitted at a low basic rate, our FAR aims to transmit all the frames at the highest attainable rate under the given channel conditions. Moreover, while the schemes in the literature are either sender- or receiver-based, our FAR is a combination of the two methods, i.e., rate adaptations of RTS/CTS are sender-based and those for Data/ACK are receiver-based. We have also thoroughly addressed the issues involved in the virtual carrier sensing (VCS) in multi-rate ad-hoc networks, and proposed a modified virtual carrier sensing (MVCS) mechanism. To study the performance of FAR, both analytical and simulation methods are used, which show that the FAR greatly improves the performance of IEEE 802.11. However, our results also show that the current routing protocols cannot actually make best use of the multi-rate capability provided by the MAC layer. Therefore, further improvements are possible by refining the routing protocols.

Chapter 7

Conclusions and Future Work

In this chapter, we summarize our main contributions and present the future work.

7.1 Main Research Contributions

The unique characteristics of wireless ad-hoc networks make the design and performance evaluation of the MAC protocol very challenging. In this thesis, we have mainly focused on the performance evaluation and performance improvement of IEEE 802.11 MAC protocol in wireless ad-hoc networks. In the performance evaluation part, we have developed novel analytical models for the single-hop scenario and the hidden-terminal scenario. In the performance improvement part, under simplified assumptions, we have developed two fully distributed algorithms (i.e., FMAC/CSR and ATC) to achieve fairness in wireless ad-hoc networks. Then, we have proposed two modifications (i.e., ECS and FAR) in the IEEE 802.11 to improve its performance under more practical assumptions. Our main research contributions are as follows:

Analytical model for the single-hop scenario: In this part, we have provided a simple analytical model to evaluate the throughput, delay, and fairness of IEEE 802.11 in the single-hop ad-hoc networks. Using the proposed model, we have carried out an extensive and insightful performance evaluation of IEEE 802.11 under different system parameters. Our results show that the throughput, delay, and fairness are quite sensitive to the chosen system parameters, and thus dynamic tuning of the system parameters is of great interest to achieve optimal performance in IEEE 802.11. Moreover, the optimal values of the parameters are inconsistent with each other for different performance metrics, implying that some tradeoff among the metrics must be made in the dynamic tuning. Finally, our results also show that the previous models have largely overestimated the throughput as well as the packet delay.

In contrast to the published models that have focused either on throughput or on delay, our model is unique as it obtains the throughput, delay, and fairness all together. Moreover, while most of the published models adopt the simplified infinite retrials assumption, our model considers the fact that a packet is either transmitted successfully or dropped after a finite number of retrials. In light of this, we derive five kinds of delay and then establish the relationship among these delays. We also derive the error introduced by the infinite retrials assumption. Such a detailed delay analysis is not available in the literature.

Analytical model for the hidden-terminal scenario: In this part, we have developed an embedded-markov model to explain and predict the short-term unfairness of IEEE 802.11 in the hidden-terminal scenario. Using the concepts of “expected state holding time” and “first passage time”, we have analytically answered the following two questions: (i) once a node gets control of the medium, what is the average number of packets this node can transmit consecutively without collision; and (ii) once a node loses its control of the medium, what is the average time the node has to wait before it gets control of the medium again. The first question reflects on how long a node can capture the medium, whereas the second question reflects on how long a node may be starved. The analytical results are validated using simulation method. Also, from the results, we have obtained several important observations about the fairness in IEEE 802.11. For example, we found that the resetting of the contention window improves the short-term fairness.

In contrast to the published works, the above model is unique as it is the first model that has evaluated the short-term unfairness of IEEE 802.11 within a hidden-terminal scenario.

Fairness Schemes (FMAC/CSR and ATC): In this part, we have systematically identified the reasons of unfairness in IEEE 802.11. We have also proposed a general framework to achieve fairness in IEEE 802.11. Following the framework, we have proposed two specific fairness schemes: Fair MAC with Cooperation between Sender and Receiver (FMAC/CSR), and Adaptive Transmission Control (ATC). The basic idea of these schemes is that, based on the information at the senders as well as at the receivers, the flows detect the occurrence of unfairness in a dynamic and distributed manner, and then they contend for the medium with different priorities in order to compensate for their past usages of the shared medium. FMAC/CSR and

ATC differ from each other in the methods of compensation, and in the role that the receiver plays. Extensive simulation results show that the proposed schemes substantially improve the MAC fairness without unduly degrading the throughput.

While most of the published fairness schemes are sender-based, meaning that a sender (of a flow) contends for the shared medium based on its own understanding of the contention on the medium, our proposed schemes utilize the information available at the sender as well as at the receiver.

Enhanced Carrier Sensing (ECS): In the current IEEE 802.11 standards, whenever a node detects an erroneous frame on the medium, it defers the transmission by a fixed duration (i.e., Extended Inter-Frame Space, EIFS). We have shown that this duration is sometimes smaller and sometimes larger than the desired period by which the transmission should be deferred, and it is the main cause of the unfairness and throughput degradation. We refer to this as imprecise-EIFS problem. In order to combat this problem, we have proposed an enhanced carrier sensing (ECS) mechanism, which intelligently distinguishes among the erroneous frames based on their lengths and defers the transmission accordingly. Simulation results show that the ECS mechanism improves the throughput and fairness substantially. Based on the ECS, we have also studied the interaction between the MAC and routing mechanisms, and obtained several useful observations about the interaction.

The issue of imprecise-EIFS in the carrier sensing has not been raised in the literature, and thus ECS is original contribution of this research-work.

Full Auto Rate (FAR): The multi-rate capability provided by the various physical layers in IEEE 802.11 requires that the MAC layer adapt the transmission rate according to the channel conditions. Recently, several rate-adaptation algorithms have been proposed. However, all the schemes consider the rate adaptation for the Data frames only, while assuming that the control frames are always transmitted at a low basic rate, leading to considerable overhead. To reduce the overhead, we have proposed a new rate-adaptation algorithm, called Full Auto Rate (FAR), which is able to adapt the transmission rate of all the frames (both control and Data frames). Moreover, in order to cope with the issues involved in the virtual carrier sensing (VCS), we have proposed a modified virtual carrier sensing (MVCS) mechanism. Both analytical and simulation results show that the FAR greatly improves the performance of IEEE 802.11.

7.2 Future Work

7.2.1 Analytical Models for Multi-hop Networks

As discussed in Chapter 1, from the viewpoint of the MAC layer, the network topologies in wireless ad-hoc networks can be classified into three categories: single-hop, multi-hop without spatial-reuse, and multi-hop with spatial-reuse.

In this thesis, we have proposed a general analytical model for the single-hop network scenario. Among the multi-hop without spatial-reuse network scenarios, we have presented an embedded-markov model for the hidden-terminal scenario. However, the model is quite specific for the simple 3-nodes hidden-terminal case, though the modelling process using the expected state holding time and the first passage time is quite general for the study of the short-term behavior. Therefore, extending the model for a scenario with a larger number of hidden-terminals is of interest. Moreover, developing analytical models for other multi-hop without spatial-reuse network scenarios is of interest. These scenarios have been systematically identified in Chapter 4. Of course, if we are not confined to the MAC layer, the analytical models for multi-hop networks with spatial-reuse are also of great interest. This is especially true for fairness modelling, which has not been addressed in the literature.

7.2.2 Fairness Scheduling in Multi-hop Networks

In this thesis, we have proposed two novel fairness schemes, FMAC/CSR and ATC. The ATC greatly improves the fairness but still cannot achieve fairness in some of the scenarios where spatial-reuse is not possible. In contrast, FMAC/CSR is able to achieve fairness in all such scenarios. FMAC/CSR is designed under the assumption that all the flows in the network are contending with each other. However, in a multi-hop network with spatial-reuse, it is common that two flows are not contending with each other (i.e., they can transmit simultaneously) but a flow has a global effect in the whole network. Therefore, to achieve fairness when spatial-reuse is possible, our FMAC/CSR needs further investigation. Firstly, we should refine the fairness model (defined in Section 4.3.1) such that it can reflect not only the local contention but also the global constraints in achieving fairness. Secondly, to practically implement the fairness model, an algorithm that should reside at the sub-network or higher layer is needed to collect

the status information of the whole network (which is required to determine the fair share) and then to propagate the scheduling decision (i.e., fair share for each flow) to the whole network. Once the fair share of a flow is known, our FMAC/CSR that exploits local cooperation can be adopted to achieve fairness in every neighborhood.

7.2.3 Integration of FMAC/CSR, ECS, and FAR

In this thesis, we have proposed FMAC/CSR, ECS, and FAR. FMAC/CSR addresses the unfairness issues due to concealed information problem, imprecise collision detection, and deficiency of the BEB algorithm, while the ECS addresses the unfairness and throughput degradation due to the deficiency in the carrier sensing mechanism. On the other hand, the FAR focuses on adapting transmission rate to improve the performance of IEEE 802.11. These schemes highlight different issues in the IEEE 802.11, and provide solution accordingly under a different set of assumptions. For example, we have assumed a fixed modulation rate in ECS, while in the FAR we have assumed $SR = TR$. Therefore, it would be challenging to integrate these schemes together for a practical wireless ad-hoc network.

7.2.4 Cross Layer Optimization between MAC Layer and Routing Protocols

In the performance evaluation of ECS, we have found that the overall system performance greatly depends on the interaction of MAC and routing mechanisms. Similarly, in the performance evaluation of FAR, we have found that the current routing protocols could not actually utilize the multi-rate capability of a MAC protocol. Therefore, it is of great importance to do the cross-layer optimization in wireless ad-hoc networks.

7.2.5 MAC Protocols in Sensor Networks

Recently, the wireless sensor networks have attracted considerable research interest. In general, the wireless sensor network belongs to the wireless ad-hoc networks. However, due to the unique application scenario, it has its own distinct characteristics compared to a general-purpose ad-hoc network. For example, the traffic in a sensor network normally converges to some infrastructure-based stations. Moreover, the processing capability of a node in a sensor network is much less than that in a general-purpose ad-hoc network. Also, the communication requirement

may also be different, for example, it may require real-time transmission as a sensor network normally is involved in some critical tasks like temperature detection and control. All these new characteristics propose new challenges in the designing of the MAC protocol, and therefore the MAC protocol design in sensor networks is of great interest to the research community.

References

- [1] N. Abramson. The ALOHA System, Computer Communication Networks. Prentice Hall, 1973.
- [2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: a Survey. *Elsevier Computer Networks*, 38(4):393–422, 2002.
- [3] G. Anastasi, E. Borgia, M. Conti, and E. Gregori. IEEE 802.11 Ad Hoc Networks: Performance Measurements. In *The 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)*, pages 758–763, 2003.
- [4] D. Bertsekas and R. Gallager. Data Networks. Prentice-Hall International Inc, 1992.
- [5] V. Bharghavan. Performance Evaluation of Algorithms for Wireless Medium Access. In *IEEE International Computer Performance and Dependability Symposium (IPDS'98)*, pages 86–95, 1998.
- [6] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a Media Access Protocol for Wireless LANs. In *ACM Sigcom*, pages 212–225, 1994.
- [7] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Area in Communications*, 18(3):535–547, 2000.
- [8] G. Bianchi and I. Tinnirello. Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 Network. In *IEEE Infocom*, pages 844–852, 2003.
- [9] J. Broch, D. A. Maltz, D. B. Johnson, and J. Jetcheva Y. Hu. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *ACM Mobicom*, pages 85–97, 1998.
- [10] F. Cal, M. Conti, and E. Gregori. Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit. *IEEE/ACM Transactions on Networks*, 8(6):785–799, 2000.

- [11] M. M. Carvalho and J. J. Garcia-Luna-Aceves. Delay Analysis of IEEE 802.11 in Single-hop Networks. In *IEEE ICNP*, pages 146–155, 2003.
- [12] A. Chandra, V. Gummalla, and J.O. Limb. Wireless Medium Access Control Protocols. *IEEE Communications Surveys*, pages 2–15, Second Quater, 2000.
- [13] P. Chevillat, J.Jelitto, A. Noll Barrerto, and H.L. Truong. A Dynamic Link Adaptation Algorithm for IEEE 802.11a Wireless LANs. In *IEEE ICC*, pages 1141–1145, 2003.
- [14] R.R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya. Using Directional Antennas for Medium Access Control in Ad Hoc Networks. In *ACM Mobicom*, pages 59–70, 2002.
- [15] CMU Monarch Group. CMU Monarch Extensions to NS, <http://www.monarch.cs.cmu.edu/>.
- [16] D.D. Falconer, F. Adachi, and B. Gudmundson. Time Division Multiple Access Methods for Wireless Personal Communications. *IEEE Communications Magazine*, 33(1):50–57, 1995.
- [17] Z. Fang and B. Bensaou. Fair Bandwidth Sharing Algorithms based on Game Theory Framework for Wireless Ad-hoc Networks. In *IEEE Infocom*, pages 1284–1295, 2004.
- [18] Z. Fang, B. Bensaou, and Y. Wang. Performance Evaluation of a Fair Back-off Algorithm for IEEE 802.11 DFWMAC. In *ACM Mobihoc*, pages 48–57, 2002.
- [19] R.G. Gallager. Discrete Stochastic Processes. Kluwer Academic Publishers, 1996.
- [20] J. J. Garcia-Luna-Aceves and Chane L. Fullmer. Floor Acquisition Multiple Access (FAMA) in Single-channel Wireless Networks. *ACM Mobile Networks and Applications*, 4(3):157–174, 1999.
- [21] S. Golestani. A Self-clocked Fair Queueing Scheme for Broadband Applications. In *IEEE Infocom*, pages 636–646, 1994.
- [22] Z.J. Haas and J. Deng. Dual Busy Tone Multiple Access (DBTMA)-A Multiple Access Control Scheme for Ad Hoc Networks. *IEEE Transaction on Communications*, 50(6):975–985, 2002.
- [23] G. Holland, N. H. Vaidya, and P. Bahl. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *ACM Mobicom*, pages 236–251, 2001.

- [24] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
- [25] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer (PHY) in the 5 GHz Band, 1999.
- [26] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer (PHY) Extension in the 2.4 GHz Band, 1999.
- [27] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher Data Rate Extension in the 2.4 GHz Band, 2003.
- [28] IETF. Mobile Ad Hoc Network Working Group, <http://www.ietf.org/html.charters/manet-charter.html>.
- [29] R. Jain, D. Chiu, and W. Hawe. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System. Technical report, DEC, 1984.
- [30] J. So and N. H. Vaidya. A Multi-Channel MAC Protocol for Ad Hoc Wireless Networks. Technical report, UIUC, 2003.
- [31] J. Jubin and J.D. Tornow. The DARPA Packet Radio Network Protocols. *Proceedings of the IEEE*, 75(1):21–32, 1987.
- [32] E.S. Jung and N. Vaidya. A Power Control MAC Protocol for Ad Hoc Networks. In *ACM Mobicom*, pages 36–47, 2002.
- [33] R.E. Kahn. Advances in Packet Radio Technology. *Proceedings of the IEEE*, 66(11):1468–1496, 1978.
- [34] A. Kamerman and L. Monteban. WaveLAN-II: A High-performance Wireless LAN for the Unlicensed Band. *Bell Labs Technical Journal*, pages 118–133, Summer, 1997.
- [35] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed Multi-hop Scheduling and Medium Access with Delay and Throughput Constraints. In *ACM Mobihoc*, pages 200–209, 2001.
- [36] V. Kanodia, A. Sabharwal, B. Sadeghi, and E. Knightly. Ordered Packet Scheduling in Wireless Ad Hoc Networks: Mechanisms and Performance Analysis. In *ACM Mobihoc*, pages 58–70, 2002.

- [37] P. Karn. MACA - a New Channel Access Method for Packet Radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–140, 1990.
- [38] L. Kleinrock and F.A. Tobagi. Packet Switching in Radio Channels: Part I -Carrier Sense Multiple-Access Modes and their Throughput-delay Characteristics. *IEEE Transaction on Communications*, 23(12):1400–1416, 1975.
- [39] R. Kohno, R. Meidan, and L. B. Milstein. Spread Spectrum Access Methods for Wireless Communications. *IEEE Communications Magazine*, 33(1):58–67, 1995.
- [40] C.E. Koksal, H. Kassab, and H. Balakrishnan. An Analysis of Short-Term Fairness in Wireless Media Access Protocols. In *ACM SIGMETRICS*, pages 118–119, 2000.
- [41] H. Lou, P. Medvedev, J. Cheng, and S. Lu. A Self-Coordinating Approach to Distributed Fair Queueing in Ad Hoc Wireless Networks. In *IEEE Infocom*, pages 1370–1379, 2001.
- [42] H. Luo, S. Lu, and V. Bharghavan. A New Model for Packet Scheduling in Multihop Wireless Networks. In *ACM Mobicom*, pages 76–86, 2000.
- [43] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving MAC Layer Fairness in Wireless Packet Networks. In *ACM Mobicom*, pages 87–98, 2000.
- [44] T. Nandagopal, S. Lu, and V. Bharghavan. A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms. In *ACM Mobicom*, pages 132–142, 1999.
- [45] J. Pavon and S. Choi. Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement. In *IEEE ICC*, pages 1108–1113, 2003.
- [46] C.E. Perkins and E.M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.
- [47] R. Punnoose, P. Nikitin, and D. Stancil. Efficient Simulation of Ricean Fading within a Packet Simulator. In *IEEE Vehicular Technology Conference*, pages 764–767, 2000.
- [48] D. Qiao, S. Choi, and K.G. Shin. Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs. *IEEE Transaction on Mobile Computing*, 1(4):278–292, 2002.
- [49] D.J. Qiao, S. Chio, A. Jain, and K. G. Shin. MiSer: An Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h. In *ACM Mobicom*, pages 161–175, 2003.

- [50] B. Ramamurthi, D.J. Goodman, and A. Saleh. Perfect Capture for Local Radio Communications. *IEEE Journal on Selected Area in Communications*, 5(5):806–813, 1987.
- [51] T.S. Rappaport. Wireless Communications. Prentice-Hall, 1996.
- [52] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic Media Access for Multirate Ad Hoc Networks. In *ACM Mobicom*, pages 24–35, 2002.
- [53] Y. C. Tay and K. C. Chua. A Capacity Analysis for the IEEE 802.11 MAC Protocol. *ACM Wireless Networks*, 7(2):159–171, 2001.
- [54] O. Tickoo and B. Sikdar. Queueing Analysis and Delay Mitigation in IEEE 802.11 Random Access MAC based Wireless Networks. In *IEEE Infocom*, pages 1404–1413, 2004.
- [55] K. To and Y.Z. Chen. A-DWOP: an Extension to DWOP for Multi-hop Wireless Ad Hoc Networks. Technical report. <http://www.owl.net.rice.edu/takhoa/courses/537/>.
- [56] N. Vaidya, P. Bahl, and S. Gupta. Distributed Fair Scheduling in a Wireless LAN. In *ACM Mobicom*, pages 167–178, 2000.
- [57] Y. Wang and J. J. Garcia-Luna-Aceves. Modeling of Collision Avoidance Protocols in Single-Channel Multihop Wireless Networks. *ACM Wireless Networks*, 10(5):495–506, 2004.
- [58] H.T. Wu, Y. Peng, K.P. Long, S.D. Cheng, and J. Ma. Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement. In *IEEE Infocom*, pages 599–607, 2002.
- [59] K. Xu, M. Gerla, and S. Bae. How Effective is the IEEE 802.11 RTS/CTS Handshke in Ad Hoc Networks? In *IEEE Globecom*, pages 72–76, 2002.
- [60] S. Xu and T. Saadawi. Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks? *IEEE Communications Magazine*, 39(6):130–137, 2001.
- [61] Ka Gavin Yeung. Packet Concatenation (PAC) for Rate Adaptive Mobile Ad Hoc Networks. Technical report, 2003. <http://www.cs.ucla.edu/classes/fall02/cs218/11/uploads/MACFinalReportPAC218Gerla.pdf>.

Appendix A

Adaptive Transmission Control (ATC)

In this appendix, based on the fairness framework proposed in Chapter 4, we present another distributed algorithm, called ATC.

A.1 Introduction

In Chapter 4, we have proposed a novel fairness scheme, called FMAC/CSR, and shown that it is able to achieve fairness in all the scenarios where spatial-reuse is not possible. However, the FMAC/CSR needs major modifications to the current IEEE 802.11 standards. For example, in the sender-based differentiated access mechanism, the freezing mechanism of IEEE 802.11 has not been used any more in FMAC/CSR, and in the aggressive-notification mechanism a new frame has been introduced. In light of this, here we aim to propose another fairness scheme that requires less modifications to the IEEE 802.11 standards.

As mentioned in Chapter 4, the basic idea of FMAC/CSR is as follows. Based on the estimated information, the flows dynamically detect the unfairness, and then enter a contention mode (i.e., aggressive, normal, or restrictive) to compensate for their past usages of the shared medium. To recall, the information obtained from the estimation algorithm include: the number of active flows n in the contention range, the actual share (w_x) of the flow x , the contention mode that the flow should enter, and the corresponding degree of aggressiveness/restrictiveness (represented by N_a and N_r , respectively).

Following the similar reasoning as in FMAC/CSR, we propose the Adaptive Transmission Control (ATC) algorithm. Specifically, the contention window (CW) at the sender side is tuned

according to the mode it enters as well as the degree of aggressiveness/restrictiveness. However, due to the freezing mechanism used in IEEE 802.11, the CW tuning mechanism does not affect the nodes that do not generate a new back-off counter when they contend for the medium. Therefore, we propose early-reset mechanism, which stops the back-off counter and resets the CW whenever a node enters the aggressive mode while having a non-zero back-off counter. Moreover, to cope with the concealed information problem, we also incorporate the restrictive-notification mechanism in ATC, which exploits the information available at the receivers only, and helps the senders make more rational decisions.

The remainder of this appendix is organized as follows. In Section A.2, we present the ATC algorithm. The simulation results are presented in Section A.3, and the appendix is concluded in Section A.4.

A.2 Adaptive Transmission Control (ATC)

A.2.1 Sender Side Mechanism: CW Tuning

In IEEE 802.11, whenever the medium becomes idle (after a collision or after a successful transmission of a packet), a sender may contend for the medium by generating a new back-off counter, or directly using the *frozen* back-off counter that has been generated previously. In this subsection, we present the CW tuning mechanism for the senders that will generate a new back-off counter when contending for the medium. Specifically, when a sender generates the back-off counter, rather than using the CW *directly*, we propose that the sender should use a scaled value $S \times CW$, where S is the scaling factor determined by the flow's contention mode. The S can be simply set to unity when the flow is entering the normal mode. When the flow enters the aggressive mode, S should be smaller than one. On the other hand, when a flow enters the restrictive mode, the S should be greater than one. Moreover, S should also differentiate between the flows that have different N_a or N_r . Therefore, S can be set as follows,

$$S = \begin{cases} 1 & \text{normal mode} \\ 2 \times N_r & \text{restrictive mode} \\ 1/(2 \times N_a) & \text{aggressive mode} \end{cases} . \quad (\text{A.1})$$

Comparing the above CW Tuning mechanism with the sender-based differentiated access mechanism of FMAC/CSR (presented in Section 4.5.1), we note two things. The first one

is as follows. As mentioned before, to compensate the past usage of the medium, there are two different approaches in general, that are, *deterministic* and *probabilistic* approaches. In FMAC/CSR, we have adopted a hybrid of the above two approaches. In contrast, here we have used a purely *probabilistic* approach. The second thing to be noted is that the freezing mechanism of IEEE 802.11 is retained in the ATC but not in FMAC/CSR, and that is why the early-reset mechanism discussed below is required for the ATC.

A.2.2 Sender Side Mechanism: Early Reset

The CW tuning mechanism only applies to the senders that generate a new back-off counter while contending for the medium. However, if a sender already has a non-zero back-off counter, the scaling factor S does not have any effect since the sender will *not* generate a new back-off counter from the $S \times CW$. This leads to a problem if the flow is supposed to enter the *aggressive* mode. Consider that a sender generates a large back-off counter, and waits for the back-off counter to expire. During the waiting period, due to the freezing mechanism, other flows may transmit more than $(n-1)$ packets, and therefore, the flow should enter the aggressive mode. However, the back-off counter may still be greater than zero and therefore it has to wait further, which potentially keeps the medium *idle* as other flows are in the restrictive mode.

To solve the above problem, we introduce *early reset mechanism*. In particular, whenever a flow is supposed to enter the aggressive mode from the *normal* but its sender still has a non-zero back-off counter, the sender should stop the current back-off counter, reset the contention window (CW), and then generate a new back-off counter from the $S \times CW$ window. Note that the early reset mechanism is not used if the node is already in the aggressive mode.

A.2.3 Receiver Side Mechanism: Restrictive Notification

As in FMAC/CSR, whenever a receiver finds (based on its own understanding) that the flow should enter the restrictive mode, it piggybacks a *restrictive-notification* in the ACK frame to the sender to slow down the sender. Whenever the sender receives such a notification, in addition to the back-off process as defined in IEEE 802.11, the sender defers by a time equal to $(N_r + 1) \times T_{succ}$ where T_{succ} is the time needed to successfully transmit a *packet* as defined

in Equation (2.10). Note that the sender should ignore the notification if it is already in the restrictive mode.

As mentioned in the introduction section of this appendix, our main objective here is to design a simple fairness scheme that requires minor modifications to IEEE 802.11. Therefore, in ATC, we do not incorporate the aggressive-notification mechanism of FMAC/CSR since that requires the introduction of an additional frame. However, without using aggressive-notification mechanism, the fairness delivered by ATC may not be very favorable in certain scenarios as shown in the simulation results. This is the cost we have to pay for the simplicity of the ATC.

Summary of FMAC/CSR and ATC: Table A.1 compares the main operations of FMAC/CSR and ATC. Clearly, FMAC/CSR and ATC differ from each other very much in the operations that the sender as well as the receiver carries out in the three contention modes.

Table A.1: Comparison between FMAC/CSR and ATC

Mode	Side	FMAC/CSR	ATC
Aggressive	Sender	<i>Back off:</i> [0, max(n, 2n-N _a)]	Early Reset (if having an old backoff value) <i>Back off:</i> [0, CW/(2 × N _a)]
	Receiver	<i>First Back off:</i> [2n, max(3n, 4n-N _a)] <i>Then send</i> Aggressive-notification	NA
Normal	Sender	<i>Back off:</i> [4n, CW]	<i>Back off:</i> [0, CW]
	Receiver	NA	NA
Restrictive	Sender	<i>First Defer:</i> TxTime(pkt) × (N _r + 1) <i>Then Back off:</i> [4n, N _r × CW]	<i>First Defer</i> (if received a restrictive notification): TxTime(pkt) × (N _r + 1) <i>Then Back off:</i> [0, 2N _r × CW]
	Receiver	<i>Piggyback:</i> Restrictive-notification	<i>Piggyback:</i> Restrictive-notification

A.3 Simulation Results

In this section, we present the simulation results to compare the proposed adaptive transmission control (ATC) scheme with IEEE 802.11. The simulation environment is the same as for the performance evaluation of FMAC/CSR. To study the improvements contributed by various mechanisms proposed before, we divide the ATC into three different categories where each has different *extent* of the mechanisms as shown in Table A.2.

Table A.2: Three Categories of ATC

Scheme	Mechanisms
ATC-1	Estimation, CW Tuning
ATC-2	ATC-1 plus Early Reset
ATC-3	ATC-2 plus Restrictive Notification

A.3.1 Hidden-terminal Topology (Figure 4.4)

The Jain's index for the hidden-terminal topology is presented in Figure A.1. Compared to IEEE 802.11, ATC-1 greatly improves the fairness as shown in the figure. The ATC-2, which incorporates the early reset mechanism, improves the fairness *further*. However, in this topology, the ATC-3, which further incorporates the receiver coordination mechanism, does not show any advantage in comparison to the ATC-2.

Figure A.1: Fairness Index in Hidden-terminal Scenario

While the ATC greatly improves the fairness, from the aggregate throughputs presented in Table A.3, we see that the throughput degrades by about 10%. However, the throughput degradation under the scheme in [36] is typically more than 25%. In contrast, as shown in Chapter 4, FMAC/CSR in this scenario even improves the throughput of IEEE 802.11. As mentioned, the ATC adopts a fully probabilistic approach, while the scheme in [36] adopts a fully deterministic approach. In contrast, FMAC/CSR adopts a hybrid approach. This example shows clearly the pros and cons of the different compensation approaches.

Also note that the aggregate throughput under ATC-2 is slightly better than that under the ATC-1, which is attributed to the early-reset mechanism, which facilitates the distributed cooperation among the contending nodes.

Table A.3: Throughput under Hidden-terminal Topology

Throughput (Mbps)		IEEE 802.11	ATC-1	ATC-2	ATC-3
Average	S_A to R_A	0.678	0.609	0.615	0.615
	S_B to R_B	0.676	0.604	0.613	0.613
Aggregate		1.354	1.213	1.228	1.228

A.3.2 Scenario-1 of Figure 4.2

In this subsection, we present the results assuming that the *Large-Col-EIFS* mechanism discussed in Chapter 4 is included. The Jain's index is presented in Figure A.2 and the aggregate throughput in Table A.4. As for fairness, IEEE 802.11 exhibits substantial short-term unfairness. Moreover, the ATC-1 and ATC-2 do not improve the fairness. On the contrary, the ATC-3, which incorporates the restrictive-notification mechanism, greatly improves the fairness. However, under the ATC-3, the unfairness remains and the aggregate throughput degrades by about 25% since we have not included the aggressive-notification mechanism in ATC.

Figure A.2: Fairness Index in Topology of Figure 4.2

Table A.4: Throughput under the Topology of Figure 4.2

Throughput (Mbps)	IEEE 802.11	ATC-1	ATC-2	ATC-3
Aggregate	1.405	1.405	1.405	1.030

A.3.3 High-contention Topology (Figure 4.6)

In this subsection, we present the results for the high-contention scenario. The Jain's index is displayed in Figure A.3. Again, the ATC-1 greatly improves the short-term fairness compared

to IEEE 802.11, and the ATC-2 improves it *further*. The receiver-side mechanism again does not play any role in this topology.

Table A.5 shows that the aggregate throughput degradation under ATC-1 is negligible compared to IEEE 802.11. The ATC-2 even slightly increases the aggregate throughput, which is again an attribute of the early reset mechanism that facilitates distributed cooperation.

Similar fairness and throughput benefits were observed when the number of contending nodes was *larger* than five but the nodes were within the one hop distance.

Figure A.3: Fairness Index in High-Contention Topology

Table A.5: Throughput in High-contention Topology

Throughput (M bps)	IEEE 802.11	ATC-1	ATC-2	ATC-3
Aggregate	1.410	1.408	1.416	1.416

A.3.4 Asymmetrical Information Topology (Figure 4.3)

In the above, we have presented the performance of ATC in the scenarios where IEEE 802.11 exhibits short-term unfairness though the long-term fairness is there. We found that ATC can achieve the short-term fairness in these scenarios. Now, we discuss the scenarios where IEEE 802.11 exhibits the long-term unfairness.

Table A.6 presents the throughput for the asymmetrical information topology. We found that the ATC-1 improves the fairness but still cannot achieve the long-term fairness. The ATC-2, which incorporates the early-reset mechanism, even worsens the fairness compared to ATC-1. The ATC-3, which further incorporates the restrictive-notification mechanism, does not improve the fairness compared to ATC-2. The reason why ATC-1 cannot achieve long-term

fairness is similar to that why FMAC/CSR-1 cannot achieve long-term fairness. However, the long-term unfairness under ATC-1 is more than that under FMAC/CSR-1. This is due to the fact that ATC-1 uses a fully probabilistic approach for the compensation while FMAC/CSR-1 uses a hybrid approach, which is more deterministic. Now, we explain why the early-reset mechanism in ATC-2 worsens the fairness compared to ATC-1. Clearly, the early-reset mechanism at node S_A will not be active because S_A always operates as if flow A were in the normal mode. On the other hand, whenever flow B should be in the aggressive mode, node S_B may not actually enter the aggressive mode if the early-reset is not used as is the case in ATC-1. On the contrary, with the early-reset mechanism of ATC-2, node S_B is very likely to enter the aggressive mode, and thus flow B under ATC-2 gets more bandwidth than that under ATC-1. This explains why ATC-2 is more unfair than ATC-1. Moreover, the restrictive-notification mechanism does not play any role at node R_A . Similarly, R_B always operates as if flow B were in the normal mode, explaining why the ATC-3 has the same performance as ATC-2.

Similarly, ATC cannot achieve long-term fairness in the other scenarios where IEEE 802.11 exhibits long-term unfairness. Such scenarios include those shown in figures 4.5 and 4.16. However, here we do not present the performance results of these scenarios.

Table A.6: Throughput in Asymmetrical Information Topology

Throughput (Mbps)		IEEE 802.11	ATC-1	ATC-2	ATC-3
Average	S_A to R_A	0.073	0.400	0.295	0.295
	S_B to R_B	1.345	0.908	1.140	1.140

A.4 Conclusions

In this appendix, we have proposed the adaptive transmission control (ATC) algorithm, which is based on the dynamic estimation of the medium state and incorporates a suit of mechanisms, such as CW tuning, early-reset, and restrictive notification. The simulation results show that the ATC greatly improve the fairness. Compared to FMAC/CSR, the ATC is simpler and need less modifications to IEEE 802.11. However, in certain scenarios, the performance delivered by ATC is worse than that delivered by FMAC/CSR, exhibiting the tradeoff between simplicity and fairness.

Appendix B

Average Throughput Computation in Scenario-10

In the scenario exhibited in Figure 5.1, using the simulation, the flow from node A to B gets about 0.25 Mbps while the flow from B to C gets about 1.15 Mbps, therefore the proportions are as follows:

$$\begin{aligned}\text{Proportion}(\text{flow from A to B}) &= 0.25/1.4 \approx 0.18 \\ \text{Proportion}(\text{flow from B to C}) &= 1.15/1.4 \approx 0.82.\end{aligned}$$

In this appendix, using an analytical model we explain why there is such a large difference in the throughputs of the two flows. Since nodes A and B know the state of the medium before initiating a RTS, the collision probability is very small and the contention window (CW) is very unlikely to be greater than CW_{min} (i.e., 31). Neglecting the collision probability, the system can be modelled as a two-states Markov chain, which is illustrated in Figure B.1.

Figure B.1: Two-states Markov Chain for Scenario-10

The state $S(A, B)$ represents that the FES(A, B) is in progress while the state $S(B, C)$ corresponds to the FES(B, C). To find the state probabilities of $S(A, B)$ and $S(B, C)$, we only need to know the transition probabilities, which must satisfy the following conditions:

$$\begin{aligned}Pr(S(B, C)|S(A, B)) &= 1 - Pr(S(A, B)|S(A, B)) \\ Pr(S(B, C)|S(B, C)) &= 1 - Pr(S(A, B)|S(B, C)).\end{aligned}$$

Therefore, we need only compute two transition probabilities, i.e., $Pr(S(A, B)|S(A, B))$ and $Pr(S(A, B)|S(B, C))$.

Let us first compute $Pr(S(A, B)|S(A, B))$. From Figure 5.3 (a) we notice that after the completion of FES(A, B), nodes A and B will first defer their transmission by DIFS. Then, node A generates a *new* random back-off timer while node B *does not*, since it already has a non-zero back-off timer that was generated in the previous competition round. We call this nonzero back-off timer as the *remaining* back-off timer. If the newly generated back-off timers at node A is denoted by I while the *remaining* back-off timer at node B is J , then the condition under which node A will get access to the medium *again* after the completion of the FES(A, B) is that, the I is smaller than the J . Therefore,

$$Pr(S(A, B)|S(A, B)) = Pr(I < J) \quad (B.1)$$

where I observes the uniform distribution over the range $[0, 31]$, but it is not trivial to get the distribution of J . If we denote the random back-off timers generated in the *previous* round at nodes A and B as X and Y respectively, then J is equal to $(Y - X)$ and the distribution can be derived as follows:

$$Pr(J = j) = \frac{2 \times (32 - j)}{32 \times 31} \quad j \in [1, 31]. \quad (B.2)$$

Using the conditional probability, the Equation (B.1) becomes,

$$Pr(S(A, B)|S(A, B)) = \sum_{j=1}^{31} Pr((0 \leq I < j)|(J = j)) \times Pr(J = j) .$$

Since I and J are *independent* to each other, the above formula becomes,

$$Pr(S(A, B)|S(A, B)) = \sum_{j=1}^{31} Pr(0 \leq I < j) \times Pr(J = j).$$

Therefore, the $Pr(S(A, B)|S(A, B))$ is computed as follows:

$$Pr(S(A, B)|S(A, B)) = \sum_{j=1}^{31} j/32 \times Pr(J = j) = 0.3438.$$

Let us now calculate the $Pr(S(A, B)|S(B, C))$. After the completion of FES(B, C), node B first defers by DIFS and then generates a *new* random back-off timer. On the other hand, since node A suffers from the *large-EIFS* problem (see Figure 5.3 (b)), it will first defer by EIFS rather than by DIFS and then resume to count down the frozen back-off timer. The only

condition under which node A will get access to the medium after the completion of FES(B, C) is that, the J is larger than $(EIFS - DIFS)$ and the I is smaller than $(J - (EIFS - DIFS))$. Since the back-off timer is *integral* multiple of slot-time (i.e., $20 \mu s$ in the DSSS [24]), the $(EIFS - DIFS)$, which is equal to $314 \mu s$, should be rounded to 16 slots. Therefore,

$$Pr(S(A, B)|S(B, C)) = Pr((J > 16) \text{ and } (I < J - 16)).$$

Using conditional probability, it becomes,

$$Pr(S(A, B)|S(B, C)) = \sum_{j=17}^{31} Pr((0 \leq I < j - 16)|(J = j)) \times Pr(J = j).$$

Contrary to the situation of computing $Pr(S(A, B)|S(A, B))$, in this case J observes the uniform distribution while I observes the distribution indicated by Equation (B.2). Therefore, the $Pr(S(A, B)|S(B, C))$ is computed as follows:

$$Pr(S(A, B)|S(B, C)) = \sum_{j=17}^{31} Pr(0 \leq I < j - 16) \times Pr(J = j) = 0.1764.$$

Therefore, after that a FES(B, C) is completed, the probability that node A can get control of the medium is very small (i.e., 0.1764), showing the unfairness caused by the large-EIFS problem at node A.

Once we have obtained all the transition probabilities, it is easy to get the state probabilities as follows:

$$\begin{aligned} Pr(S(A, B)) &= 0.21 \\ Pr(S(B, C)) &= 0.79. \end{aligned}$$

We notice that the analytical results are quite *close* to the simulation results, and thus verify our simulation.

Appendix C

MAC Performance and its Interaction with Routing Mechanism in Scenario-11

In Chapter 5, under the assumptions of static routing and without capture capability, we have shown the performance of the scenario in Figure 5.4. In this appendix, we give an extensive performance study of the above scenario under varying conditions. Specifically, using static routing, we first illustrate how hidden-terminal problem, capture, and imprecise-EIFS problem affect the performance of IEEE 802.11. Then, replacing the static routing with AODV and DSDV, we study the interaction between the MAC performance and the routing mechanisms.

The rest of the appendix is organized as follows. In Section C.1, we present the factors affecting MAC performance: hidden-terminal, imprecise-EIFS, and capture. In Section C.2, we study the MAC performance. The interaction between the MAC protocol and routing mechanism is studied in Section C.3, and the appendix is concluded in Section C.4.

C.1 Factors Affecting MAC Performance

In this section, we consider the scenario shown in Figure C.1 (which is the same to Figure 5.4) as an example to illustrate the three factors that affects MAC performance: capture, hidden terminal, and imprecise-EIFS.

Figure C.1: 4-nodes with Two Single-hop Flows

C.1.1 Capture Effect

Capture is said to occur when a receiver can clearly receive one of the multiple simultaneous transmissions present around the receiver. In Chapter 1, we have already introduced the conditions under which a frame can be captured. Consider the scenario shown in Figure C.1, when nodes A and D transmit simultaneously. At node B, the signal strength received from A is much higher than that from D. Therefore, if node A's transmission arrived at node B earlier than that by node D, node B can correctly receive node A's transmission even in the presence of transmission from D.

Intuitively, the capture improves the system throughput, but it results in unfair sharing of bandwidth with preference given to nodes, whose packets have higher SNR.

C.1.2 Hidden Terminal Problem

In Chapter 1, we present the scenario (Figure 1.1), which is commonly used to illustrate the hidden-terminal problem. In fact, there we have made an implicit assumption, i.e., the sensing Range (SR) is equal to the transmission Range (TR). Now, consider the situation that SR is two times of TR, and that no physical obstacles lay between nodes A and C. When node A starts to transmit to node B, node C senses this transmission though it cannot understand the content of the transmission. Therefore, the carrier sensing (CS) at node C indicates a busy medium and thus node C will not initiate any transmission. It seems that the hidden-terminal problem does not arise any more in such case.

Now, we consider the scenario in Figure C.1. When node A starts to transmit to node B, since node D is out of the SR of node A, it cannot detect the transmission. Therefore, node D may also start to transmit to node C, resulting in collision or capture. In such case, nodes A and D are hidden terminal of each other. Therefore, the hidden-terminal problem still exists when SR is assumed to be two times of TR, but the hidden terminals are three hops away from each other.

C.1.3 Imprecise-EIFS Problem

As indicated in Figure 5.5, when nodes C and D are deferring while FES(A, B) is in progress, node D suffers from the small-EIFS problem after node B sends a CTS to A. Moreover, after

node B sends an ACK back to node A, node D suffers from the large-EIFS problem. The same problems occur at node A when a FES(C, D) is in progress.

In the following sections, we carry out an extensive study for the above scenario. Specifically, under a given routing protocol, we conduct simulations under four different scenarios, called S1 to S4, which are described in Table C.1.

Table C.1: Four Different Scenarios

Scenario	IEEE 802.11/ECS	Allow capture
S1	IEEE 802.11	no
S2	IEEE 802.11	yes
S3	ECS	no
S4	ECS	yes

C.2 MAC Performance under Static Routing

In this section, the simulation results are presented under the static routing. For convenience, we refer to the scenarios as static-S1 to static-S4.

C.2.1 Throughput Comparison

Table C.2 presents the average throughputs under the four scenarios. From the table, we see that the throughput of each flow under static-S1 is much smaller than that under the remaining three scenarios, while the throughputs under the other three scenarios are very close to each other.

Table C.2: Average Throughputs under Static Routing

Scenario	Flow	Average Throughput(M bps)
static-S1	A to B	0.314
	D to C	0.307
static-S2	A to B	0.683
	D to C	0.641
static-S3	A to B	0.662
	D to C	0.672
static-S4	A to B	0.665
	D to C	0.670

The reasons why the throughput in static-S1 degrades so much have been explained in Chapter 5. Specifically, when node A is transmitting a Data frame to node B, as node D suffers

from the small-EIFS problem, node D may transmit its RTS. The two transmissions result in a collision at node B and thus node B discards the frames. Likewise, due to symmetry in the topology, node C is also likely to drop the Data frame due to the interference from node A. This explains why the throughputs under static-S1 degrade a lot.

The dropping of the data frame at node B can be avoided in two ways. One is to prevent node D from sending out the RTS when node A is transmitting its data frame. The ECS achieves this by preventing the small-EIFS problem and thus the throughputs under S3 and S4 do not degrade. Another way to avoid frame dropping is to allow the capture of the data frames as being done in the static-S2. In this scenario, since node A's data is transmitted before node D's RTSs and the SNR of the data frame received at node B is greater than the threshold, the data frame can be captured by node B. Therefore, the throughputs in S2 also do not degrade.

When ECS is used without capture (scenario S3), we achieve throughput about the same as that in the case of S2, which highlights the benefit of the ECS. In static-S4, the capture is used with ECS, but the throughput does not increase compared to that under static-S3. The reason is that in the S4, as explained later, the capture occurs only of the RTS frames and this kind of capture does not enhance the performance.

C.2.2 Fairness Comparison

Figure C.2 presents the Jain's index. We notice that all the scenarios exhibit short-term unfairness to a certain extent. Since the common problem existing in all the four scenarios is the hidden-terminal problem, it is clear that the hidden-terminal problem results in short-term unfairness as observed in Chapter 3. Interestingly, though the static-S1 attains the worst bandwidth utilization, it achieves the best fairness. Compared to the static-S2, which shows the worst fairness, the ECS (S3 and S4) improves the fairness while the throughput does not degrade. Moreover, capture does not play much role in the fairness under ECS as S3 and S4 almost show the same fairness.

Before analyzing the results of Jain's index, let us first consider the situations under which the frames sent by nodes A and D overlap. One such situation is, node D (or A) starts transmitting its RTS frame even though node A (or D) is transmitting a data frame. This situation occurs whenever the imprecise-EIFS problem exists. As discussed in the previous

subsection, when capture is allowed in such situations, the data frame can be received correctly. Otherwise, both the frames are dropped. Another overlapping situation, which arises due to the hidden-terminal problem, is that both the nodes transmit their RTSs. In particular, since nodes A and D are hidden from each other, both of them may transmit their RTSs during a certain interval before any one of them gets a CTS. Consider the situation when RTS of A arrives at node B earlier than the RTS sent by node D, and therefore, node B captures the RTS of A if the capture is allowed. However, after completely receiving the captured RTS, the carrier sensing at node B indicates a busy medium since node D is still transmitting its RTS. This forbids node B from sending a CTS back to node A, and thus the four-way handshake initiated by node A fails even though its RTS is captured. Therefore, both the RTS frames are dropped irrespective of whether the capture is allowed. Note that, if a data frame is captured, the four-way handshake can be successful because the node capturing the data frame always sends back an ACK without considering whether the medium is busy [24]. In fact according to [24], in addition to the data frame, the capture of an ACK frame or a broadcast frame is also effective since these frames are the last ones in the sequence. The capture of other frames, such as RTS, or CTS, is not effective in the handshake because ultimately they will be discarded. This implies that a capture is effective only if the receiver processes the captured frame.

Now, we analyze the results of the Jain's index shown in Figure C.2. In static-S1 and S2, the imprecise-EIFS problem leads to the overlapping between the Data and RTS frames. When the capture is not allowed (static-S1), both the frames are dropped and then the CWs at nodes A and D are increased. On the other hand, if capture is allowed (static-S2), the Data frame is captured. Then, the node that sends the Data frame get san ACK and resets its CW. On the

Figure C.2: Fairness Comparison under Static Routing

contrary, the node that sends the RTS frame retransmits its RTS in vain, leading to a larger CW. This obviously results in unfairness and explains why the static-S2 shows worse fairness than that under static-S1. In scenarios static-S3 and S4, the overlapping of frames occurs only when both nodes A and D transmit their RTSs, and the difference between these two scenarios is that the capture is not allowed in S3. However, a capture of a RTS does not lead to any advantage as explained above, and as a result, the two scenarios show similar fairness. Moreover, they show better fairness than that under static-S2 as the ECS solves the imprecise-EIFS problem, which is a major reason of unfairness. The presence of hidden-terminal, imprecise-EIFS, and absence of capture each plays an important role in determining the fairness of static-S1, and the combined effect of these makes S1 the fairest among the four scenarios.

From the above study, we deduce that: *(i)* hidden-terminal problem results in substantial short-term unfairness, *(ii)* capture, if effective, increases the throughput at the cost of fairness. On the other hand, if the capture is not effective in the handshake, it does not have any effects on the fairness as well as the throughput, and *(iii)* the imprecise-EIFS problem results in throughput degradation and unfairness, and the ECS improves the fairness without degrading the throughput performance.

C.3 Interaction between MAC Performance and Routing Mechanisms

To study the effects of the interaction, we replace static routing with AODV and DSDV, one at a time.

C.3.1 Throughput Comparison

Table C.3 presents the average throughputs under different routing protocols. We notice that the average throughputs under DSDV are almost the same as those under the static routing. On the other hand, under the AODV-S1 to S3, the flow from node D to C completely starves while the other flow gets the entire bandwidth. This is because the AODV at node D cannot even find a route to node C as explained later. Note that the results presented here are obtained under the situation where the flow from node A to B starts first. However, if we start the flow from D to C first, exactly opposite results are obtained, i.e., the flow from node A to B starves.

Table C.3: Throughputs under Different Routing Protocols

Scenario	Flow	Average Throughput(M bps)		
		Static	DSDV	AODV
S1	A to B	0.314	0.314	1.401
	D to C	0.307	0.308	0
S2	A to B	0.683	0.684	1.402
	D to C	0.641	0.669	0
S3	A to B	0.662	0.675	1.402
	D to C	0.672	0.659	0
S4	A to B	0.665	0.650	0.660
	D to C	0.670	0.685	0.678

In order to explain the flow's starving, we first describe the route discovery process of AODV [46]. The AODV uses two kind of messages for the route discovery: Route Request (RREQ) and Route Reply (RREP). When a node, say node S , needs a route to some destination node, say node R , node S first broadcasts a RREQ and then waits for a RREP. When a node receives the RREQ and has a fresh enough route to node R , the node generates a RREP message and then unicasts it to the next hop toward the originator of the RREQ (i.e., node S). If a RREP is not received by the time RREQ_TIMEOUT expires, to find the route, node S will broadcast another RREQ, up to a maximum of RREQ_RETRIES times. To reduce congestion in the network, the value of RREQ_TIMEOUT follows a binary exponential back-off mechanism. In our simulation, the initial value of RREQ_TIMEOUT is 1.8 second while the RREQ_RETRIES is set to 3.

Now, we explain why the AODV's route discovery process at node D always fails in the S1 scenario. Whenever an RREQ is handed to the MAC layer at node D, this packet will be transmitted immediately by the MAC protocol if the medium is determined to be idle. On the other hand, if the medium is determined busy, the MAC protocol backs off and defers the transmission of the RREQ. Figure C.3 demonstrates when node D determines the medium to be idle or busy while FES(A, B) is in progress. When the medium is determined to be idle, the MAC layer at node D decrements its back-off timer and transmits the RREQ when the timer becomes zero. We notice from the diagram that the RREQ can be successfully transmitted only if it is sent in a very small crucial interval of duration ν . At other instances, however, the carrier sensing at node D may indicate the medium to be idle though really it is not idle,

and thus the transmission of the RREQ results in a collision. This has been shown as collision in the figure. If we denote the sum by λ of all idle durations determined by node D within a single FES, then the ratio ν/λ reflects the probability that node D can successfully transmit the RREQ. In the AODV-S1, the ν and λ are as follows:

$$\nu = y - (EIFS - DIFS) - TxTime(RREQ), \quad (C.1)$$

$$\begin{aligned} \lambda = & TxTime(RTS) + SIFS + TxTime(Data) + SIFS \\ & - (EIFS - SIFS) + y - (EIFS - DIFS) \end{aligned} \quad (C.2)$$

where y is the back-off time at node A. As ν depends on y , which in turn is determined by the CW value, we first consider the CW value at node A. One should note that the RREQ_TIMEOUT between two consecutive RREQs sent by node D is normally long enough for nodes A and B to exchange several FESs successfully, leading to the reset of the CW at node A. Therefore, whenever the RREQ is transmitted by node D, the CW at node A is always equal to CW_{min} . Since the maximum value of y is equal to CW_{min} (i.e., 31), which is smaller than the transmission time (i.e., 40 slots in the case) of RREQ, the ν in Equation (C.1) is equal to zero as ν cannot be negative. Therefore, ν/λ is zero. In other words, whenever node D transmits the RREQ, it always collides with a frame (either RTS or data) sent by A. This explains why the AODV at node D cannot start in the S1 scenario. Similarly, we can show why the AODV cannot start in S2 and S3 scenarios. However, in scenario S4, ν/λ is significant enough to allow the AODV to start at node D.

In fact, what we have analyzed above applies not only to the AODV, but also to all the on-demand routing protocols. To verify this, we carried out simulation using DSR. The results show that the DSR at node D in S1, S2 and S3 does not start.

Figure C.3: Medium State Determined by Node D when FES(A, B) is in Progress

C.3.2 Fairness Comparison

Similar to the throughput, the Jain's index under DSDV is also almost the same as that under the static routing. It is easy to see that the Jain's indices under AODV-S1, -S2 and -S3 are always equal to 0.5 as absolute unfairness occurs in these three scenarios. Performance for scenario S4 under all three routing protocols is compared in Figure C.4.

Since the DSDV is very similar to the static routing except that the DSDV periodically exchanges route messages, they show similar fairness. We notice that the AODV-S4 shows worse fairness in comparison to other two. The reason is as follows. Whenever the MAC layer at a node drops a packet after several unsuccessful attempts, the AODV views this as an indication of the link breakage, and thus initiates a fresh route discovery process that may take a long time during which the other node gets the entire bandwidth of the medium. This situation may occur very frequently due to the hidden-terminal problem. On the contrary, DSDV and static routing do not view the dropping of a packet as an indication of link breakage, and therefore they do not initiate the route discovery process. Therefore, AODV shows worse fairness than that under the static routing or DSDV.

Figure C.4: Fairness Comparison under S4

From the above discussion, we can deduce that: (i) the short-term MAC unfairness may lead to the stalling of the on-demand routing protocols (e.g., AODV and DSR), and (ii) the routing mechanisms (e.g., route discovery and link breakage detection) also affect the system fairness.

C.4 Conclusions

From this appendix, the main lessons learnt are: (i) we should distinguish between the *MAC fairness* and the *overall system fairness*, since under identical MAC conditions different overall fairness performance may be obtained if different routing protocols are used, (ii) in studying the *MAC fairness*, we should isolate the factors that are not directly related to the MAC layer, and therefore, the study is to be carried out under the static routing, CBR backlogged traffic, static topology etc, and (iii) to study the *overall system fairness*, in addition to the study of MAC fairness, we should also consider many other factors such as the routing mechanism, traffic pattern, mobility etc.