M.Sc. Engg. Thesis

Optimal Contention Window Setting based On Packet Size Distribution for Error Prone Channel

by
Farah Tanjeem

Submitted to

Department of Computer Science and Engineering

In partial fulfillment of the requirements for the degree of
Master of Science in Computer Science and Engineering

Bangladesh University of Engineering and Technology (BUET)
Dhaka 1000

September 2014

The thesis titled "**Optimal Contention Window Setting based on Packet Size Distribution for Error Prone Channel**," submitted by Farah Tanjeem, Roll No. 100705062, Session October 2007, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on September 17, 2014.

Board of Examiners

1. ______________________	
Dr. Md. Yusuf Sarwar Uddin	Chairman
Assistant Professor	(Supervisor)
Department of CSE	
BUET, Dhaka 1000	
2. ______________________	
Dr. Mohammad Mahfuzul Islam	Member
Professor & Head	(Ex-officio)
Department of CSE	
BUET, Dhaka 1000	
3. ______________________	
Dr. A. K. M. Ashikur Rahman	Member
Professor	
Department of CSE	
BUET, Dhaka 1000	
4. ______________________	
Dr. Rajesh Palit	Member
Assistant Professor	(External)
Department of CSE	
North South University, Dhaka	

Candidate's Declaration

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Farah Tanjeem

Candidate

Contents

Acknowledgements **x**

1 Introduction **1**

1.1 Wireless Ad Hoc Network 1

1.2 Research Problems/Issues in Wireless Ad Hoc Network 3

1.3 Motivation for work and Problem Statement 6

1.4 Scope of the Work 8

1.5 Contribution of the Thesis 9

1.6 Overview of the Thesis 9

2 Preliminaries **11**

2.1 Types of Wireless Network 12

2.2 Applications of Wireless Ad Hoc Network 14

2.3 How do Ad Hoc Networks Differ? 16

2.4 IEEE 802 Networking standard 18

2.4.1 Physical Layer 19

2.4.2 Medium Access Control (MAC) Sub Layer 19

2.4.2.1 Classification of MAC Protocols 19

2.4.2.2 Access Mechanism for MAC 21

2.4.2.3 Timing Intervals 22

2.4.2.4 Modes of Operation 24

2.5 IEEE 802.11 Parameters 27

2.5.1 Contention Window Size 28

2.5.2 Retransmission and other Counters 29

2.5.3 RTS Threshold 30

2.5.4 Packet Size 31

2.6 Packet Loss Modeling 31

2.6.1 Gilbert-Elliott: The Classical 2-State Markov Model for Error Processes 32

2.6.2 Markov Model for Error analysis 33

2.6.3 Other Error Models 34

2.7 Different Approaches to change BEB in Previous Researches 35

3 Modified MAC Protocol 39

3.1 Impact of Packet Size and Contention Window on Error Prone channel . . 39

3.2 Basic Idea for Adjustment of CW Based on Packet Sizes 42

3.3 Probabilistic Analysis of Uniform Binning 44

3.4 Modified MAC Protocol 49

3.4.1 Packet Labeling 49

3.4.1.1 Learning Phase 49

3.4.1.2 Packet Allocation in Bins 50

3.4.2 Contention Window Adjustment Based on Labels 51

3.4.2.1 Why Smaller Packets to Smaller CW Sub-Range? 52

3.5 Disadvantage of Using Fixed Packet Sizes for Labeling 54

3.6 Proposed MAC Algorithm 54

4 Simulation Results 58

4.1 Simulation Setting 58

4.1.1 Simulation Environment and Parameters 59

4.1.2 Implementation of Error Model in NS-2 62

4.1.3 Random Packet Generation 65

4.2 Simulation Technique 66

4.2.1 Scenario Generation for Random Topology 67

4.2.2 Traffic Generation for Random Topology 68

4.3 Performance Metrics 69

4.4 Performance Evaluation 70

4.4.1 Impact of Error Prone channel on IEEE 802.11 MAC 70

4.4.1.1 Packet Delivery Fraction 70

4.4.1.2 Throughput 71

4.4.1.3 Average E2E Delay 72

4.4.1.4 Impact of Packet Size On IEEE 802.11 MAC Protocol . . 73

4.4.1.5 Impact of CWMin on 802.11 MAC Protocol 74

4.4.2 Performance Evaluation of Modified IEEE 802.11 MAC Protocol . . 76

4.4.2.1 E2E Delay Variation 77

4.4.2.2 Packet delivery Fraction (PDF) Variation 78

4.4.2.3 Throughput Variation 79

4.4.2.4 Retransmission Count 80

4.4.3 Impact of Varied Schemes in Modified Algorithm 80

4.4.3.1 Selection of Packet Distribution Scheme 81

4.4.3.2 Random Agnostic Binning 82
4.4.3.3 Arbitrary Boundary Setting for Bins 82
4.4.3.4 Reverse CW Assignment Scheme 83
4.4.3.5 Selection of Different Number of Bins/Classes 84
4.4.3.6 Learning Time Adjustment of Window Timer 85
4.5 Summary 85

5 Conclusion 87

5.1 Major Contributions 87
5.2 Future Directions of Further research 88

6 Appendix 93

6.1 What is NS-2 93
6.2 Installation 96
6.3 Set Path 96
6.4 Scenario Generation 96
6.5 CBR Traffic Generation 97
6.6 Writing Simulation Generation File 98
6.7 Trace Generation 99
6.8 Trace Analysis 99

List of Figures

1.1 A typical ad hoc network. 2
1.2 Different Topology in Wireless Ad Hoc Network 7
2.1 Intelligent Public Transportation System 16
2.2 MAC Layer in IEEE Standard 18
2.3 MAC Layer in IEEE Standard 20
2.4 Basic Access Mechanism 22
2.5 RTS/CTS Access Mechanism in DCF 23
2.6 Basic Access mechanism in DCF 24
2.7 DCF backoff procedure with multiple stations deferring and go into random backoff. 25
2.8 Exponential Increase of CW 27
2.9 The Gilbert-Elliott model generating a 2-state Markov modulated failure process 32
2.10 Loss Model with Infinite No of states 33
2.11 Two-state continuous time chain 33
2.12 Temporal four-state Semi-Markovian based model 35
3.1 Packet distribution in Bins for selection of CW sub-range 43

3.2 Uniform Binning for selection of CW Sub-range 48
3.3 Proposed Methodology 49

4.1 5X5Grid Topology 59
4.2 Random Topology 60
4.3 Structure of Mobile Node in NS2 64
4.4 Beta Distribution 66
4.5 PDF Variation in IEEE 802.11 MAC for Different Error Rates 71
4.6 Throughput Variation in IEEE 802.11 MAC for Different Error Rates . . . 72
4.7 E2E Delay Variation in IEEE 802.11 MAC for Different Error Rates 72
4.8 E2E delay Variation for Different Max Packet Size 73
4.9 E2E delay Variation for CWMin 74
4.10 E2E Delay Variation with Original IEEE 802.11 MAC for Different Topology 77
4.11 E2E delay Variation for Selection of Different No of Bins 78
4.12 PDF Variation for Different Topology 78
4.13 PDF Variation for Selection of Different No of Bins 79
4.14 Throughput Variation for Different Topology 79
4.15 Throughput Variation for Selection of Different No of Bins 80
4.16 E2E Delay Variation for different Packet Distribution Schemes 81
4.17 E2E delay Variation for Random Agnostic Binning 82
4.18 E2E delay Variation with Arbitrary boundary setting for bins 83
4.19 E2E Delay Variation for Reverse CW Assignment 83
4.20 E2E delay Variation with Different No of classes 84
4.21 E2E Delay Variation for different Window Timer Interval 85

List of Tables

2.1 Types of Wireless Networks. 13

2.2 IEEE 802.11 Parameters 28

2.3 Effect of Contention Window Size. 29

2.4 Effect of Setting RTS Threshold 30

2.5 Effect of Packet Size 31

2.6 Research issues considering BEB Modification via CW Change scheme. . . 37

2.7 Research issues considering Error Prone Channel. 38

4.1 Simulation Setting-1 61

4.2 Simulation Setting-2 62

4.3 Additional Simulation parameters for modified algorithm. 76

4.4 Summary of comparison between Modified and Basic MAC. 86

Acknowledgments

All praises due to Allah, the most benevolent and merciful.

Foremost, I would like to express my heart-felt gratitude to my supervisor, Dr. Md. Yusuf Sarwar Uddin for his constant supervision of this work. He helped me a lot in every aspect of this work and guided me with proper directions whenever I sought one. His patient hearing of my ideas, critical analysis of my observations, enthusiasm, immense knowledge and detecting flaws (and amending thereby) in my thinking and writing have made this thesis a success.

I would also want to thank the members of my thesis committee for their valuable suggestions. I thank Professor Dr. Mohammad Mahfuzul Islam, Dr. A. K. M. Ashikur Rahman and specially the external member Professor Dr. Rajesh Palit.

Besides my supervisor and board members, i also want to thank my friends who were a great source of support and encouragement.

Finally, I would like to mention that i will remain ever grateful to my beloved parents and my grandmother who always exist as a single source of inspiration behind every success of mine I have ever made. I would also like to thank my husband Md. Mahadi Rahman for his consistent support, patience and understanding during this work in my everyday living.

Abstract

Ad Hoc Network is a decentralized type of network where wireless devices are allowed to discover each other and communicate in peer to peer fashion without involving central access points. In most ad hoc networks, nodes compete for access to shared wireless medium, often resulting in collision (interference). IEEE 802.11, a well-known standard uses medium access control (MAC) protocol to support delivery of radio data packets for both ad hoc networks and infrastructure based network. But Designing a Medium Access Control (MAC) protocol for ad hoc wireless networks is challenging, particularly when the protocol needs to achieve optimal performance both in terms of throughput and end to end delay to deliver a packet. Error prone channel has a significant impact on unsuccessful transmission probability which is often ignored by previous researches. Standard DCF (Distributed Coordination Function) operation of IEEE 802.11 enacted by binary exponential back-off cannot differentiate collision from corruption and increases back-off time through larger contention window (CW) upon a failure. This leads to increased delay in error prone network when nodes are not contending at all. Since packet corruption depends on bit error rate (BER) and length of packets, packet size can have significant impact on the throughput in error-prone environment. In this paper, we analyze effect of packet size in determining optimal CW to improve throughput and efficiency for error prone networks. We propose a dynamic learning based scheme to adaptively select CW confined within a range for different packet distribution. To validate our scheme extensive simulations have been done and simulation results show significant improvement in E2E delay performance.

Chapter 1

Introduction

Wireless communications is one of the most vibrant areas in the communications field nowadays. There has been an immense increase in demand for wireless connectivity, driven so far mainly by cellular telephony but expected to be soon eclipsed by wireless data applications. People dwelling areas like homes, offices, airports or stations can be equipped with needed infrastructure if required, however there are some cases where infrastructure-less, multi hop wireless communication is desired. Wireless telecommunications networks are generally implemented with some type of remote information transmission system that uses electromagnetic waves, such as radio waves, for the carrier and this implementation usually takes place at the physical level of the network. In recent years multi hop wireless ad hoc and sensor networks are getting popular in wireless technology world and have been (and still being) studied extensively. These networks are composed of a collection of hosts operating in a self-organized and decentralized manner, which might communicate together using a radio interface.

1.1 Wireless Ad Hoc Network

"Ad Hoc" is actually a Latin phrase that means "for this purpose." It is often used to describe solutions that are developed on-the-fly for a specific purpose. Ad Hoc Networks are organized spontaneously by nodes wishing to communicate without infrastructure support. Nodes can freely leave and enter the network at any time. An ad hoc network is a (possibly mobile) collection of communications devices (nodes) that wish to communicate, but have no fixed infra-structure available, and have no pre-determined organization of available links. Individual nodes dynamically discover other nodes to communicate directly. A key assumption is that not all nodes can directly communicate with each other, so nodes are required to relay packets on behalf of other nodes in order to deliver data across the network. A typical ad hoc network has been shown in Figure 1.1.

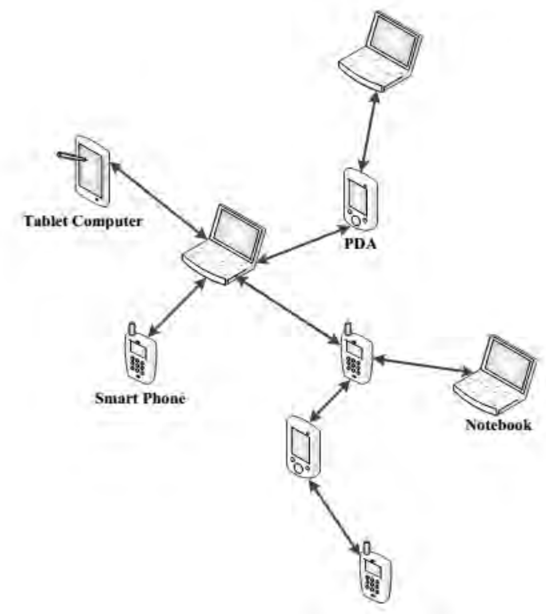


Figure 1.1: A typical ad hoc network.

Minimal configuration and quick deployment make ad-hoc networks suitable for emergency situations like natural disasters or military conflicts. Presence of dynamic routing protocol will enable ad hoc networks to be formed more quickly.

Wireless Ad Hoc Networks can be further classified based on their applications:

1. *Mobile Ad Hoc Networks (MANET)*: MANETs have important feature that nodes are mobile and can change their position with the passage of time.

2. *Wireless Sensor Networks:* Networks of special purpose sensor nodes which normally have limited computing functionality. These wireless networks consist of distributed

autonomous nodes using sensors to monitor environmental conditions, such as temperature, sound or motion at different places.

However, most of them share the common feature of limited energy resources and capability to communicate using one or more wireless technologies. A brief overview of wireless ad hoc network is presented in section 2.1.

1.2 Research Problems/Issues in Wireless Ad Hoc Network

The "new world" of Ad Hoc Network technology has been visited by now, there is no completely unexplored territories in this field. Active research has produced a wide range of proposals, but there are some problems that are yet to be solved. We start from the issues recognized in Wireless Ad Hoc networks.

1. *Bandwidth Efficiency:* Bandwidth efficiency can be defined as ratio of bandwidth used for actual data transmission to the total available bandwidth. Radio spectrum is a limited resource and shared by all nodes in the range of its transmitters. With the increasing demand, capacity crunch is more and more likely to happen. Medium access control protocol should be designed efficiently and in a manner .

2. *Media Access:* Unlike cellular networks, there is lack of centralized control and global synchronization in ad hoc wireless networks. Hence, TDMA and FDMA schemes are not suitable. In ad hoc wireless networks, same media is shared by multiple mobile ad hoc nodes and nodes cannot rely on a centralized coordinator. Access to the common channel must be made in a distributed fashion, through the presence of a MAC (Media Access Control) protocol so that all nodes get a fair share of available bandwidth. Nodes must contend for access to the channel through this protocol while at the same time avoiding possible collisions with neighboring nodes.

3. *Error Prone Shared Radio Channel*: The radio channel is a broadcast medium by

nature. During propagation through the wireless medium, the radio waves suffer from several impairments such as attenuation, multipath propagation and interference. Error rates are significantly higher in wireless medium. The bit error rate is the percentage of bits that have errors relative to the total number of bits received in a transmission. Typical Bit Error rates (fraction of bits that are received in error) are of the order of 10^{-4} in a wireless channel as against 10^{-9} in fiber optic cables. High BER value effects the limited energy resource of a wireless network leading to packet loss.

4. *Routing:* The presence of mobility implies that links make and break often and in an in-deterministic fashion. Classical distributed Bellman-Ford routing algorithm is used to maintain and update routing information in a packet radio network. Ad hoc mobile networks are different from packet radio networks as nodes can move more freely, resulting in a dynamically changing topology. Existing distance-vector and link-state-based routing protocols are unable to catch up with such frequent link changes in ad hoc wireless networks, resulting in poor route convergence and very low communication throughput. Hence, new routing protocols are needed.

5. *Scalability:* Ad Hoc networks suffer, by nature from the scalability problems in capacity. Scalability is an important research topic for the future, not only because of its necessity, but also because of the applicability of same ideas in the Internet. In a non-cooperative network, where Omni-directional antennas are being used, throughput per node decreases at a rate $\frac{1}{\sqrt{N}}$, where N is the number of nodes [7]. That is, in a network with 100 nodes, a single device gets approximately one tenth of the theoretical data rate of the network interface card at maximum.

6. *Quality of Service:* Voice, live video and file transfer, all have very differing requirements what comes to delay, jitter, bandwidth, packet loss probability etc. Quality of Service (QoS) is being developed to meet the emerging requirements. However, the lack of fixed infrastructure in ad hoc networks makes the QoS appear even more challenging problem than ever before. MAC protocol that supports real time traffic must support some kind of resource reservation mechanism that takes mobility of nodes

into consideration. algorithms and protocols regarding QoS robustness,routing policies required to be researched in the future.

7. *Security:* Ad hoc networks are particularly prone to malicious behavior. Lack of any centralized network management or certification authority makes these dynamically changing wireless structures very vulnerable to eavesdropping, interference etc. Information sent in an ad hoc route can be protected in some way but since multiple nodes are involved, the relaying of packets has to be authenticated by recognizing the originator of the packet and the flow ID or label.

8. *Power Control and Energy Efficiency:* Most existing network protocols do not consider power consumption an issue since they assume the presence of static hosts and routers, which are powered by mains. For ad hoc mobile networks, mobile devices must perform both the role of an end system (where the user interacts and where user applications are executed) and that of an intermediate system (packet forwarding). Hence, forwarding packets on the behalf of others will consume power, and this can be quite significant for nodes in an ad hoc wireless network. However, mobile devices today are mostly operated by batteries. Power-aware networks are currently being extremely popular within the ad hoc networking research. There are two research topics which are partially similar: the maximization of lifetime of a single battery and the maximization of lifetime of whole network.

9. *Synchronization:* Synchronization is necessary for bandwidth reservation. Exchange of control packets may be required for achieving time synchronization among nodes. The MAC protocol must take into consideration the synchronization between nodes in the network.

10. *Compatibility with other technologies and applications:* The interoperability among different WLANs (wired or wireless) is important for efficient communication between hosts operating with different technologies and to provide a seamless communication across the WANs.

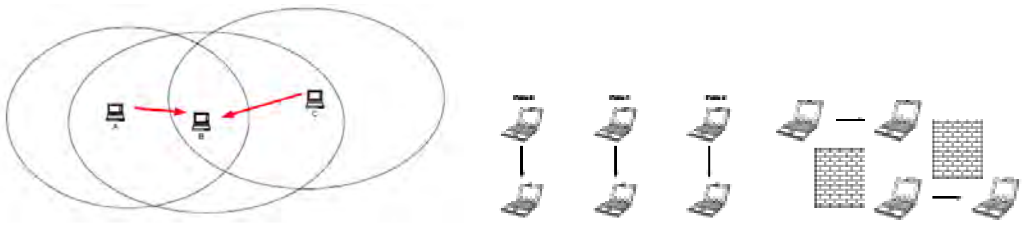
11. *Topology related issues:* Network topology is dynamic. The presence of mobility,

hidden terminals, and exposed nodes problems must be accounted for designing MAC protocols for ad hoc wireless network.

- *Hidden Node Problem:* The hidden node problem occurs in some type of network when a node is visible from a wireless access point (AP), but not from other nodes communicating with that AP. When two independent transmitters try to send packet to the same receiver, collision due to concurrent transmission leads to throughput degradation. It necessitates the retransmission of the packets. The scenario is depicted in Figure 1.2a.
- *Exposed Terminal Problem*: When two receivers are out of range of each other, yet the two transmitters in the middle are in range of each other, exposed terminal problem occurs. In this case a node is prevented from sending packets to other nodes due to a neighboring transmitter and causes throughput degradation.
- *3 pair's scenario:* A type of network (in Figure 1.2b)topology where unfairness arises as the central pair can't access the medium due to asymmetric contention. Here collision issue is less, but low probability for central node to access the medium causes degraded throughput.
- *Asymmetric Hidden Terminal Problem:* The transmission of the upper node always collides and the transmission of the lower node always succeeds in Figure 1.2c. This asymmetry leads to a long term unfairness between the two transmitters when 802.11 is used and RTS/CTS mechanism does not solve the problem.

1.3 Motivation for work and Problem Statement

MAC protocol supplies the functionality required to provide a reliable delivery mechanism for user data over noisy, unreliable wireless media. Designing a Medium Access Control (MAC) protocol for ad hoc wireless networks is challenging due to error-prone shared



(a) Hidden Terminal (b) 3 pair's scenario (c) Asymmetric Hidden Terminal

Figure 1.2: Different Topology in Wireless Ad Hoc Network

radio channel and limited bandwidth. Many research works assume packet loss in wireless channel is mainly due to collision, arising from simultaneous packet transmission attempts of multiple nodes causing packet loss for all parties involved and wireless channel is error-free [4]. However, another significant cause of packet loss known as packet *corruption* is mainly due to bit errors in noisy channel, which is often ignored by many researchers. Wireless links can be noisy and therefore corruption can be caused due to reasons such as variation of signal strength, surrounding atmosphere and presence of certain objects, devices and occlusions.

But standard DCF (Distributed Coordination Function) operation of IEEE 802.11 enacted by binary exponential back-off cannot differentiate collision from corruption. DCF assumes packet loss is mainly due to collision and therefore takes only collision resolution mechanism. IEEE 802.11 specification requires all nodes to choose a random backoff interval between 0 to CW and sets forth a (time) separation between multiple nodes accessing the channel by (appropriately) adjusting contention window (CW). Contention Window prevents multiple nodes from sending packets in the channel at the same time, but does not help when the channel is inherently noisy. In noisy environment, this measure indeed hurts by delaying unnecessarily some nodes when those are actually not contending at all. So in a non-ideal channel for non-contending nodes impact of optimal CW is less. This leads to an interesting problem of how to adjust CW for packet loss due to both collision and corruption. Since packet corruption depends on bit error rate (BER) and length of packets, packet size can have significant impact on the performance in error-prone environment as considered by authors in paper ([2], [3]).

The idea of optimal frame length that improves energy efficiency for a general MAC protocol is presented in [15],[16]. Although it provides insight on the optimization of MAC layer under error-prone environment, but it does not consider characteristics of 802.11 DCF like exponential backoff and the contention mechanism. So we can not directly apply this mechanism to 802.11 DCF. Optimal contention window alone is not an effective scheme under bad channel conditions. It is shown [3] that under error-prone channel environment, optimal packet size can have more significant impact on energy efficiency than optimal contention window, and combining both optimal contention window and optimal packet size can achieve the maximum optimization. We find that there is much space for research on tuning CW for error prone channel based on packet size. After a long research and experiments we have finally devised a method of adaptively tuning CW with the help of current packet size distribution of the network under different error models and error rates. Here the distribution means size of packets currently flowing through the network.

1.4 Scope of the Work

The main task of this thesis is to modify existing IEEE 802.11 MAC Protocol for error prone wireless Ad Hoc channel with a view to increase performance and efficiency. We have considered error/corruption as a reason of packet transmission failure along with collision for error prone channel. Effect of packet error on collision probability can have a significant impact on efficiency of wireless channel. As corruption depends on bit error rate (BER) and packet length, modifying CW setting scheme based on packet size can have a positive impact in performance gain of wireless channel. We realize the factor and modify existing IEEE 802.11 DCF protocol based on analysis from different packet size distribution. We make simulation based experiments of our scheme using NS2 and describe our observations in chapter 4. We evaluate the performance of our technique by varying various parameters (No of nodes, Error rate, random packet generation using two different distribution technique). We also compare our approach with original IEEE 802.11 Mac protocol in terms of through, end to end delay and retransmission count.

Experiments reveal that in all performance metrics except throughput, our approach outperforms original MAC protocol with a margin, thus placing modified scheme in a dominating position over its counterpart. All observations are made based on simulations, in an artificial replication of real world with the simulator and the results are obviously subject to deviation.

1.5 Contribution of the Thesis

We want to devise a modified MAC protocol suitable for error prone wireless ad hoc network. In our proposed scheme, we keep main principle of the MAC Protocol of IEEE 802.11 DCF unchanged. The only difference between modified and original algorithm is the conditions that determine the contention window(CW) size to use. We have made selection of CW dependent on packet size. IEEE 802.11 specification requires all nodes to choose a random back off interval between 0 and CW (contention window), and wait for the chosen number of slot times before trying to access the channel. In our modified protocol, CW has uniformly partitioned with different equal backoff sub-ranges. We have divided number of contending nodes want to access the medium into uniform groups and we assign a specific backoff sub-range to a certain group based on current packet size. Contrary to BEB algorithm, our protocol dynamically selects contention window sub-range instead of whole selection range based on packet size distribution in the network. In contrast with original MAC protocol, our scheme can accomodate packets with less retransmissions and at the same time ensures less delay. In the simulation study, as described in chaper 4, we find that average E2E delay is far reduced in modified MAC protocol than the original one and it saves energy in packet transmission.

1.6 Overview of the Thesis

The rest of the chapters are organized as follows. Chapter 2 gives a preliminary description of some protocols and concept related to error prone wireless ad hoc network that may be helpful to understand the context of the thesis. Chapter 3, the main chapter of this dissertation, illustrates our proposed modified algorithm. Chapter 4 contains the simulation results of our scheme and a comparative study against original MAC DCF protocol in several performance metrics. Chapter 5 draws the conclusion mentioning the key contributions of this thesis followed by some future directions in further research in this field.

Chapter 2

Preliminaries

Wireless networks are computer networks that use radio frequency channels as their physical medium for communication. Every node within the network broadcasts information which can be received by all nodes within its direct transmission range. Since nodes transmit and receive over the air, they need not be physically connected. Hence, such networks provide data connectivity along with user mobility.

The world's initial wireless radio communication was invented by Guglielmo Marconi in 1987. In 1901, he successfully demonstrated his wireless telegraph system to the world by transmitting radio signals across the Atlantic ocean from England to America, covering more than 1700 miles. This signaled the start of the radio communication era.

Wireless communications is one of the quickest growing industries in the world. The wireless communications industry has many segments such as cellular telecommunication, wireless LAN, and satellite based communication networks. But the major portion of growth in wireless industry has been due to cellular networks. The early 1980s saw the commercial deployments of the world's first mobile cellular networks. The First-generation (1G) cellular networks used analog signal technology. Second-generation (2G) cellular systems used digital transmissions such as TDMA and CDMA. The 1G and 2G systems used primarily for voice communication. 2.5G is something related to the general packet radio service (GPRS). The Third generation (3G) systems are expected to provide services such as enhanced multimedia, higher bandwidth and roaming capability throughout the world. The 4G services are expected to provide additional enhancements within the services provided by 3G. The deployment of 3G and 4G services would truly transform the world into global village.

In the early 1990s a spate of new developments signaled a new phase in ad hoc networking. The IEEE 802.11 subcommittee adopted the term "ad hoc networks." The concept of commercial (non-military) ad hoc networking had arrived and interest grew. Spurred by the growing interest in ad hoc networking, a number of activities and commercial standards evolved within the mid to late '90s. Within the IETF, the Mobile Ad Hoc Networking (MANET) working group was born, and sought to standardize routing protocols for ad hoc networks. The 802.11 subcommittee standardized a medium access protocol that is based on collision avoidance and tolerated hidden terminals. HIPERLAN and Bluetooth were some other standards that addressed and benefited ad hoc networking.

Yet, research in the area of ad hoc networking is getting much attention from academia, industry, and government. Since these networks pose many complex issues, there are many open problems for research and opportunities for creating vital contributions.

2.1 Types of Wireless Network

Networks can be classified based on topology or on the types of protocols they support. A short description of different types of wireless network has been defined in table 2.1

The point of the classification is not to partition each technology into a separate class, but to highlight the high-level differences. Wireless networks can broadly be categorized as under:

1. *Wireless Personal Area Networks (WPAN):*Interconnect devices within a relatively small area which is generally within a person's range. For example, both Bluetooth (IEEE 802.15) and invisible infrared (IR) provides a WPAN for interconnecting a headset to a laptop. These will allow the connectivity of personal devices within an area of about 30 feet. However, IR requires a direct line of site and range is also

Table 2.1: Types of Wireless Networks.

Type	*Range*	*Applications*	*Standards*
Personal Area Network	Within Reach Of A Person	Cable Replacement For Peripherals	IEEE 802.15, Bluetooth, ZigBee, NFC
Local Area Network	Within A Building Or Campus	Wireless Extension Of Wired Network	IEEE 802.11, WiFi, HiperLAN
Metropolitan Area Network	Within A City	Wireless Inter-network Connectivity	IEEE 802.16, WiMAX
Wide Area Network	Worldwide	Wireless Network Access	Cellular (2G, 2.5G, 3G, UMTS, LTE Etc.)

less.

2. *Wireless Local Area Networks (WLAN):* WLAN links two or more devices over a short distance, usually providing a connection through an access point for internet access. Products using IEEE 802.11 WLAN standard are marketed under Wi-Fi brand name. Wi-Fi is the most popular wireless communication protocol for local area networks. WLANs allow users in a local area, such as a university campus or library, to form a network or gain access to the internet. Bluetooth is another wireless protocol commonly used in cellular phones and computer peripherals for short range network communication.

3. *Wireless Mesh Network:* Made up of radio nodes organized in mesh topology.

4. *Wireless MAN (Metropolitan Area Networks):* Type of wireless network that connects several wireless LANs. WiMAX is a type of Wireless MAN and is described by IEEE 802.16 standard.

5. *Wireless WAN (Wide Area Networks):* Wireless networks that typically cover large

areas, such as between neighboring towns and cities, or city and suburb via multiple satellite systems or antenna sites looked after by an ISP.

6. *Cellular Network (Mobile Network):* Radio network distributed over land areas called cells, each served by at least one fixed-location transceiver base station. When joined together these cells provide radio coverage over a wide geographic area.

7. *Global Area network (GAN):* Network used for supporting mobile across an arbitrary no of wireless LANs, satellite coverage areas etc.

8. *Space Network:* Networks used for communication between spacecraft, usually in vicinity of the Earth. NASA's space network is an example of this type of network.

2.2 Applications of Wireless Ad Hoc Network

Ad Hoc Networks are suited for use in situations where an infrastructure is unavailable, not trusted or to deploy one is not cost effective. Infrastructure may not be practical for short-range radios like Bluetooth (range ~ 10m).

Commercial Application

When it comes to instantaneous network deployment possible areas could be:

1. *Business Environment:* Suited in places where the need for collaborative computing might be more important outside the office environment than inside.

2. *Game Theory*: Originates from economics and has been applied in various fields. Ad hoc networks rely on the cooperation of participating nodes to route data between source and destination pairs that are outside each other's communication range. Such data forwarding consumes valuable (and scarce) battery power. This tension between cooperation and cost invites a game-theoretic study, where each node must

strategically decide the degree to which it must volunteer its resources for the common good of the network. Game theory deals with multi-person decision making, in which each decision maker tries to maximize his utility.

Military Applications

Some of the essential requirements of a combat operations include network deploy ability, security, end to end IP, high mobile connectivity and anti-jamming mechanisms.

1. In most of the cases, military operations are usually spontaneous i.e. with little or no fixed network infrastructure. These operations require a communication solution, which is spontaneous too. In other words, the soldiers should be able to form a network when and where it is needed.

2. In comparison with geographical positioning systems, mobile ad hoc networks can support built in geographical location by using an extremely accurate form of triangulation. Readings are faster than geographical positioning systems.

3. Mobile ad hoc networks also allow devices to transmit at a lower output power to the neighbors, which benefits the overall network by lowering the chance of detection and by increasing battery power.

Intelligent Transportation System

Mobile ad hoc networks can be used for intelligent transportation systems. This idea refers to systems that add information and communication technology to transport infrastructure and vehicles. The aim of these systems is to manage factors like shipment, routes, improve safety, traffic signal control, reduce transportation times and fuel consumption.

We can exemplify the advantage of an intelligent transportation system on a citywide real time travel information system in buses shown in Figure 2.1 on the following page

A multitude of public transportation buses can be equipped with the system named above. The system is a city wide communication network based on mesh technology.

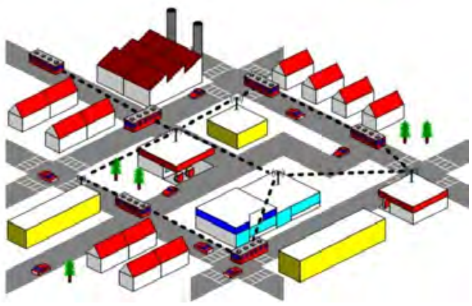


Figure 2.1: Intelligent Public Transportation System

Its purpose is to offer real time travel information for the passengers of the buses. Vehicles could operate as a mobile ad hoc network in which a single vehicle could detect traffic events and initiate a broadcast to other vehicles.

Educational Classrooms

Simple installation of a communication infrastructure is required to create an interactive classroom on demand.

Emergency Operations for Public Safety

Mobile Ad Hoc Networks can also be implemented to reinforce the work of police, fire departments and emergency services. The demand for this kind of services combined with a high bandwidth network is increasing. This type of services requires high mobility, flexibility and reliability. In disaster areas or rescue operations it is very useful to be able to arrange a wireless ad hoc network without being dependent on a fixed infrastructure.

2.3 How do Ad Hoc Networks Differ?

In wired networks, signals are transmitted over guided medium, whose system impulse response is relatively stable and isolated from the environment. However, in wireless networks, signals are transmitted over wireless channels in open space, and suffer time-varying environmental interference. Compared to wired links, the wireless channels exhibit many different forms of channel impairments such as multipath fading and carrier frequency/phase noises. In a wireless network, all participating computers potentially communicate with each other directly. Wireless networks therefore have only two topologies: infrastructure based and infrastructure less (ad hoc).

The infrastructure based wireless network topology is a hub and spoke topology, also known as point to multi-point or one to many topology. There exists a single central wireless access point (WAP) for this topology. The WAP acts as hub in the network, with all the other computers (or spokes) connecting to it.

An ad hoc wireless network topology is a many to many topology. There is no central access point in this network structure. These are essentially mesh networks. A mobile node may communicate with each other directly or indirectly. If it is an indirect communication, a multi-hop scenario occurs, where the packets originated from the source host are relayed by several intermediate mobile nodes before reaching the destination. The vision of ad hoc network is wireless internet, where users can move anywhere anytime and still remain connected with the rest of the world. The Mobile Ad Hoc Network is characterized by energy constrained nodes, bandwidth constrained links and dynamic topology. In real-time applications, such as audio, video, and real-time data, the need for Quality of Service (QoS) in terms of delay, bandwidth, and packet loss is becoming important. Providing QoS in ad hoc network is a challenging task because of dynamic nature of network topology. Hence it is important to have a dynamic routing protocol with fast re-routing capability, which also provides stable route during the life-time of the flows.

The ad hoc topology has the advantage of not requiring a central access point or WAP. Infrastructure based networks, on the other hand, require the extra equipment of a central WAP. However, this brings the advantage of higher speeds and stronger security to such networks. If many devices are connected to the ad-hoc network, there will be more wireless interference. If a device is out of range of another device it wants to connect, it will pass the data through other devices on the way. Passing the data through several computers is just slower than passing it through a single access point. Ad hoc networks don't scale well.

Each of these topologies has its own advantages and disadvantages and one may serve one better than the other. The infrastructure based topology is typically used for permanent networks, while the ad hoc topology is used for temporary networks.

2.4 IEEE 802 Networking standard

The institute of Electrical and electronics standard (IEEE) has defined several standards for LANs. Such standards collectively come under IEEE 802 standard. This standard deals with data link layer and physical layer of OSI model. It defines rule for cabling, signaling and media access control, which assure interoperability between network products manufactured by different vendors. IEEE 802.11 is one of the most popular standards for wireless networks. The objective of this standard is to provide wireless connectivity to wireless devices/nodes that require rapid deployment, which may be portable or mounted on moving devices. The 802.11 standard was first published in 1997.

802.11 operate two MAC function: The Point Coordination Function (PCF) and the Distributed Coordination Function (DCF). The PCF is responsible for time-bounded service and the DCF is of asynchronous data service. Figure 2.2 shows the MAC layer of an IEEE standard.

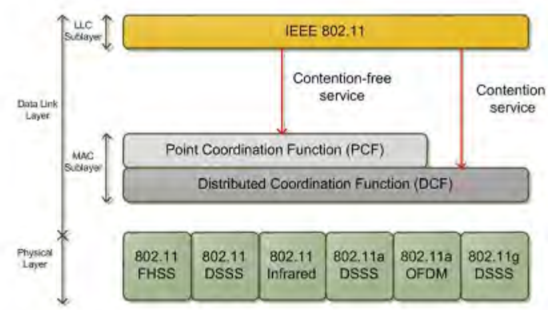


Figure 2.2: MAC Layer in IEEE Standard

2.4.1 Physical Layer

The basic 802.11 standard supports three different physical layers. Two of them, Frequency hopping spread spectrum (FHSS) and Direct Sequence spread spectrum (DSSS) are based on radio transmissions and the third is based on infrared. The physical layer is subdivided conceptually into two parts- physical medium dependent sub layer (PMD) and physical layer convergence protocol (PLCP). PMD handles encoding, decoding and modulation of signals. The PLCP abstracts the functionality that physical layer has to offer to the MAC layer.

The physical layer provides mechanism for sensing the wireless channel and determining whether or not it is idle. This mechanism is called clear channel assessment (CCA). CCA is generated based on sensing of air interface either by sensing the detected bits in the air or by checking the received signal strength (RSS) of the carrier against a threshold.

2.4.2 Medium Access Control (MAC) Sub Layer

MAC protocol supplies the functionality required to provide a reliable delivery mechanism for user data over noisy, unreliable wireless media. MAC provides:

- Reliable data delivery
- Fairly control access to the shared wireless medium.
- Protect the data that it delivers.

2.4.2.1 Classification of MAC Protocols

MAC protocols can be categorized as Contention-free and Contention-based, as shown in Figure 2.3.

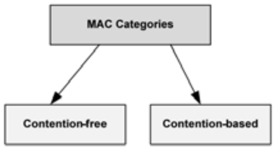


Figure 2.3: MAC Layer in IEEE Standard

Contention Free Protocols

The contention free protocols as in [21] are more efficient than those of the contention based. These schemes are generally based on TDMA, FDMA or CDMA that utilizes the synchronization technique and the channel access mechanism of the physical layer, where structure of the network is spatially divided into slots or cells. These protocols work well for multimedia traffic and are more applicable for static networks with centralized control. However, these schemes are more complex, require centralized control, use multiple channels simultaneously, specialized sensor hardware and there is a dependency on physical layer. Therefore the focus is mainly on the contention based and transport layer schemes, where wireless sensor networks(WSN) need to cope with congestion, fairness and packet loss.

Contention Based Protocols

Most of the proposed contention based protocols use Carrier Sense Multiple Access (CSMA) scheme, where for a station (STA) to transmit, it must sense the medium to determine if another station is transmitting. If the medium is busy, the STA will defer until the end of the current transmission. After deferral or just before attempting to transmit again, the STA shall select a random back-off interval and shall decrement the back-off interval counter while the medium is idle.

The CSMA/CA protocol is designed to reduce collision between multiple stations accessing the medium. However CSMA/CA tends to suffer from hidden and exposed node problems. To resolve hidden terminal problem, the transmitting and receiving STA exchange short control frames (RTS and CTS frames) after determining that the medium is idle and after any deferrals or back-offs, prior to data transmission.

2.4.2.2 Access Mechanism for MAC

Basic Access Mechanism

The basic access mechanism is carrier sense multiple access with collision avoidance (CSMA/CA) with binary exponential backoff similar to IEEE 802.3, with some significant exceptions. CSMA/CA is a "listen before talk" (LBT) access mechanism. Because of the nature of radio environment, it is very difficult for a transmitting node to detect packet collisions in the network. Hence, carrier sense multiple access with collision detection (CSMA/CD) is not preferred in wireless LANs.

When there is a transmission in the medium, the station will not begin its own transmission. This is the CSMA portion of the access mechanism. If there is a collision and the transmission gets corrupted, the operation of the access mechanism works to ensure the correct reception of the information transmitted on the wireless medium.

As IEEE 802.11 implements this access mechanism, when a station listens to the medium before beginning its own transmission and detects an existing transmission in progress, the listening station enters a wait period determined by the binary exponential backoff (BEB) algorithm. CSMA is very effective when the medium does not have high traffic, so that medium transmit with minimum delay. Stations transmitting at the same time result in collision as the protocol initially is designed for single channel transmission. Finite state machine of this mechanism has been shown in Figure 2.4.

Detailed DCF modes of operation have been described in section 2.4.2.4.

RTS/CTS Handshaking Scheme

RTS/CTS mechanism is used to avoid well-known hidden terminal problem of CSMA based MAC protocols.

1. A node wishing to send data initiates the process by sending a Request to Send

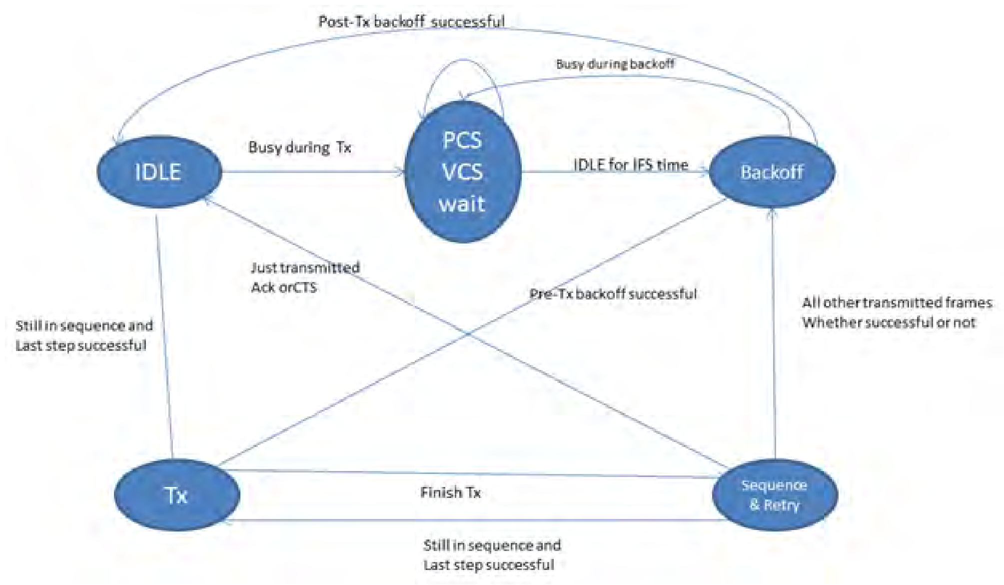


Figure 2.4: Basic Access Mechanism

frame (RTS).

2. The destination node replies with a Clear to send frame (CTS).

3. Any other node receiving the RTS or CTS frame should refrain from sending data for a given time (solving the hidden node problem).

4. The amount of time the node should wait before trying to get access to the medium is included in both the RTS and the CTS frame.

5. RTS Threshold value determines when RTS/CTS handshaking will be triggered. If packet size the node wants to transmit is larger than the threshold, RTS/CTS handshake gets triggered. If the packet size is equal to or less than the threshold, the data frame gets sent immediately. Detailed mechanism has been described in Figure. 2.5

2.4.2.3 Timing Intervals

Inter Frame spacing (IFS) refers to the time interval between the transmission of two successive frames by any station. There are four timing intervals as depicted in Figure

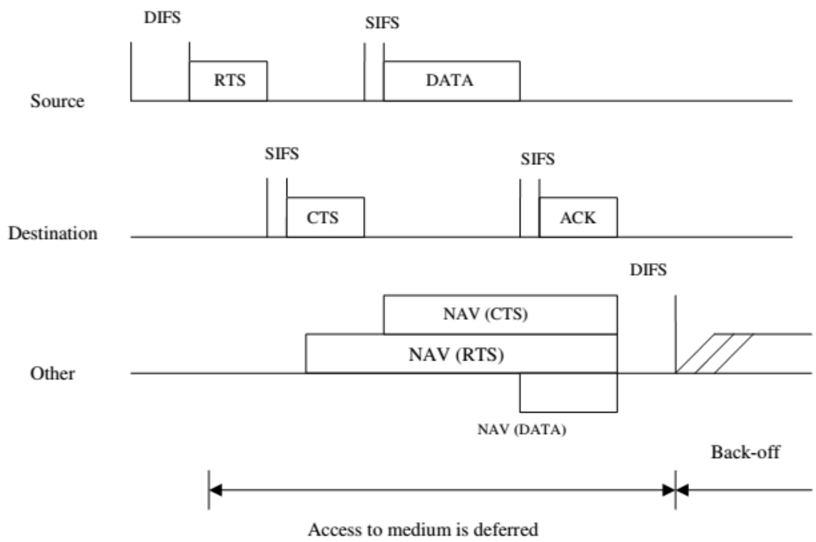


Figure 2.5: RTS/CTS Access Mechanism in DCF

2.6. They denote priority levels of access to the medium.

1. Short inter-frame space (SIFS): The SIFS is the shortest of all the IFS's and denotes highest priority to access the medium. It is defined for short control messages such as acknowledgment for data packets and polling responses.

2. Priority inter-frame space (PIFS): The PIFS is equal to SIFS plus one slot time. This is used for real time services.

3. Distributed inter-frame space (DIFS): DIFS is used by the stations that are operating under the DCF mode to transmit packets. This is for asynchronous data transfer within the contention period. DIFS is equal to SIFS plus two slot times

4. Extended inter-frame space (EIFS): The EIFS is the longest of all the IFS's and denotes the least priority to access the medium. EIFS is used for resynchronization whenever physical layer detects incorrect MAC frame reception.

The exact values of the IFS are obtained from the attributes specified in the physical layer management information base PHYMIB and are independent of station bit rate.

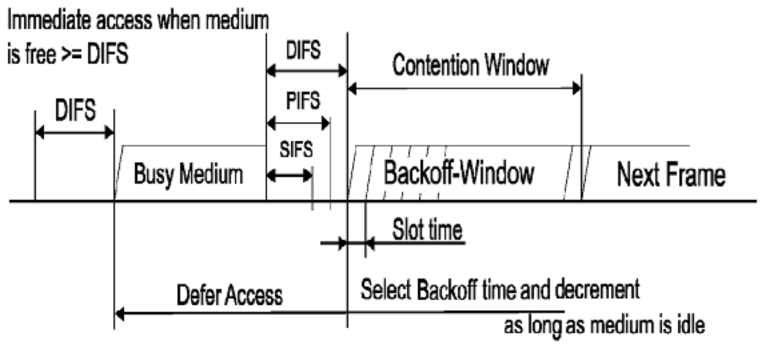


Figure 2.6: Basic Access mechanism in DCF

2.4.2.4 Modes of Operation

PCF Operation

Point coordination function (PCF) is a Media Access Control (MAC) technique used in IEEE 802.11. It resides in a point coordinator known as Access Point (AP), to coordinate the communication within the network. The AP waits for PIFS duration rather than DIFS duration to grasp the channel. PIFS is less than DIFS duration and hence the point coordinator always has the priority to access the channel.

DCF Operation

The basic 802.11 MAC protocol is the DCF based on CSMA. Stations deliver MAC Service Data Units (MSDUs), after detecting that there is no other transmission in progress on the channel. However, if two stations detect the channel as free at the same time, collision occurs.

The 802.11 defines a Collision Avoidance (CA) mechanism to reduce the probability of such collisions. Figure 2.7 represents the details of DCF operation.

1. When MAC receives a request to transmit a frame, a check is made of the physical and virtual carrier sense mechanisms. Physical Carrier Sensing (PCS) notifies the MAC layer if there is a transmission going on and VCS (Virtual Carrier Sensing) is NAV procedure, If NAV is set to a number, station waits until it resets to zero.

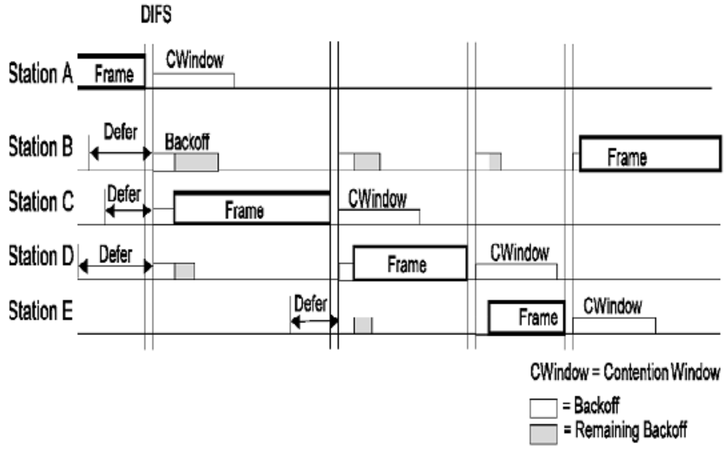


Figure 2.7: DCF backoff procedure with multiple stations deferring and go into random backoff.

2. Before starting a transmission a station has to keep sensing the channel for an additional random time after detecting the channel as being idle for a minimum duration called DIFS, which is 34µs for the 802.11a PHY. Only if the channel remains idle for this additional random time period, the station is allowed to initiate its transmission.

3. If the medium is in use during the DIFS interval, the MAC will select a backoff, in which the station defers channel access by a random amount of time chosen within a 0 to contention window (CW) and increment the retry counter. The value of CW can vary between CWMin and CWMax. The time intervals are all integral multiples of slot times, which are chosen judiciously using propagation delay, delay in transmitter and other physical layer dependent parameters.

4. The MAC will decrement the backoff value each time the medium is detected to be idle for an interval of one slot time. As soon as the backoff counter reaches zero and expires, the station can access the medium.

5. During the back-off process, if a node detects a busy channel, it freezes the backoff counter and process is resumed once the channel becomes idle for a period of DIFS.

6. If there is a collision, the contention window is doubled; a new backoff interval is selected.

Random Backoff Time

The binary exponential backoff mechanism chooses a random number which represents the amount of time that must elapse while there are not any transmissions. The random number resulting from this algorithm is uniformly distributed in a range, called the contention window (CW), the size of which doubles with every attempt to transmit that is deferred, until a maximum size is reached for the range. The backoff procedure has been described in Figure 2.8.

BEB algorithm adjusts CW by indirectly estimating the traffic load in communication channel at individual nodes. This estimation of traffic load is done by counting the number of consecutive collisions involving the same packet transmitted by the node. After k consecutive collisions, a random number between 0 to $2^k - 1$ (Where 2^k is the current size of the contention window) is chosen, and the node remains idle, not attempting to transmit, for that number of slots. For example, if two nodes collide, after the first collision, each node waits for 0 or 1 slot time before attempting to transmit again.

The value of CW_i is equal to

$CW_i = 2^i * CWMin$, here i depends on number of failed transmissions of a packet.

If packet collision occurs, CW_i is doubled up to a maximum value,

$CW_m = CWMax = 2^m * CWMin$, Where $m = log_2(\frac{CWMax}{CWMin})$, identifies number of backoff stages.

If they collide again (second collision), each node picks 0, 1, 2 or 3 slots and waits for that number of slots. Thus this randomization interval grows exponentially after each collision.

The backoff procedure will be invoked when a station is ready to transfer a frame and finding the medium busy as by the indication of the physical or virtual CS mechanism.

The backoff procedure will also be invoked when a transmitting station infers a failed

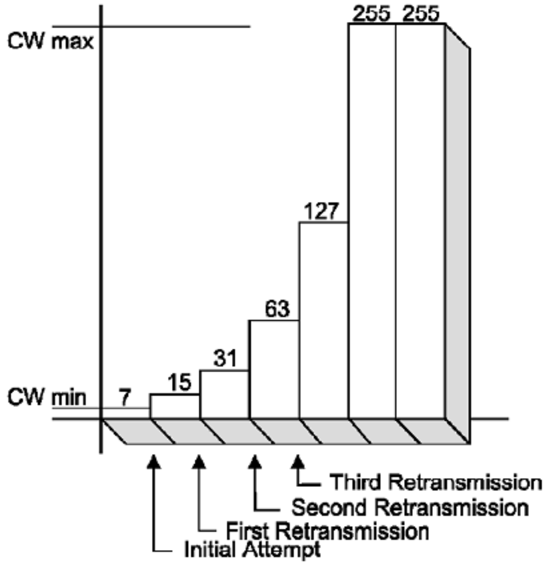


Figure 2.8: Exponential Increase of CW

transmission. The station will set its backoff timer to a random backoff following a DIFS period during which the medium is determined to be idle. The station performing the backoff procedure will use the CS mechanism to determine any activities during the backoff slot. If there is no activity indicated the backoff procedure will decrement its backoff time by a SlotTime.

2.5 IEEE 802.11 Parameters

The MAC attributes tune the performance of MAC protocol, monitor the performance and provide identification of MAC implementation.

Standard parameter values for IEEE 802.11 have been presented in table 2.2.

Parameter	802.11(FHSS)	802.11(DSSS)	802.11b	802.11a
Tslot	50 µs	20 µs	20 µs	9 µs
SIFS	28 µs	10 µs	10 µs	16 µs
DIFS	SIFS+ (2 X Tslot) µs			
Operating Frequency	2.4 GHz	2.4 GHz	2.4 GHz	5 GHz
Maximum Data Rate	2 Mbps	2 Mbps	11 Mbps	54 Mbps
CW_{Min}	15	31	31	15
CW_{Max}	1023	1023	1023	1023

Table 2.2: IEEE 802.11 Parameters

Below mentioned parameters play very important role in the operation of an error prone network. In the original standard, fixed values are used for these parameters. But performance of a network can be improved in many ways by tuning these parameters in optimum values. Effect of some parameters in both basic and error prone channel has been described.

2.5.1 Contention Window Size

The size of contention window (CW) is an important parameter. As we know, Backoff is performed for R slots. R is a randomly chosen integer in the interval $[0, CW]$. If the CW size is small in size, then there is high probability of packet collision. On the other hand, if the size of CW is very large, there will be some unnecessary delay because of large backoff values.

The initial contention window is set to a random value between (0, CWMin) and each time a collision occurs, CW doubles its size up to a maximum of CWMax. So at high load, CW size is high and therefore the resolution power of the system is high. At low loads, small CW ensures low access delay. Effect of selection of CWMin in error prone and ideal environment have been described in Table 2.3.

	Ideal Channel		Error Prone Channel	
	Basic Scheme	RTS/CTS	Basic Scheme	RTS/CTS
Contention window	Impacts more. Choosing a larger one leads to large backoff interval and larger overhead. Choosing a small one leads to a large number of collisions.	Collision cost is much smaller, impact of CW is less than basic mode.	Less Impact as nodes are not actually contending. Larger CW induces longer delays.	

Table 2.3: Effect of Contention Window Size.

2.5.2 Retransmission and other Counters

Acknowledgments (ACK) must be sent for data packets in order to ensure their correct delivery. For unicast packets, the receiver accesses the medium after waiting for a SIFS and sends an ACK. Other stations have to wait for DIFS plus their backoff time. ACK ensures the correct reception of the MAC layer frame by using cyclic redundancy checksum (CRC) technique. If no ACK is received by the sender, then a retransmission takes place. The number of retransmission is limited, and failure is reported to the higher layer after the retransmission count exceeds the limit.

Each frame is associated with a retry counter based on frame size as compared to RTS/CTS threshold. Fragments are given a maximum lifetime by MAC before discarding them.

Short Retry Count:

Number of retransmission attempts for frames shorter than RTS threshold before the frame is abandoned and a failure is indicated to higher layer protocols. It is used in four way handshaking mechanism.

Long Retry Count:

Number of retransmission attempts for frames longer than RTS threshold before the frame is abandoned and a failure is indicated to higher layer protocols. It is used in basic scheme.

Failed Count:

Failed count is a counter that tracks the number of frame transmissions that are abandoned because they have exceeded either the short or long retry count.

ACK Failure Count:

ACK Failure count is a counter that tracks the number of times a data or management frame is sent to an individual address and does not result in the reception of an ACK frame from the destination.

2.5.3 RTS Threshold

This parameter controls the transmission of RTS control frames prior to data and management frames.

Using a small value causes RTS packets to be sent more often, consuming more of the available bandwidth, therefore reducing the apparent throughput of the network packet. However, the more RTS packets that are sent, the quicker the system can recover from interference or collisions in a heavily loaded network, or a wireless network with much electromagnetic interference. Impact of setting RTS Threshold in error prone and ideal environment have been depicted in Table 2.4.

Parameter name	Effect when decreased	Effect when increased
RTS Threshold	Greater effective throughput if there are a large number of hidden node situations	An improvement will be realized only if there is no interference.

Table 2.4: Effect of Setting RTS Threshold

2.5.4 Packet Size

Packet size can have a significant impact on error prone environment rather than ideal environment. Impact of packet size in error prone and ideal environment have been described in Table 2.5.

	Ideal Channel		Error Prone Channel	
	Basic Scheme	RTS/CTS	Basic Scheme	RTS/CTS
Packet Size	Smaller packet preferred. If payload small probability of collision comparatively low.	Longer packet preferred to reduce overhead. If payload large, probability of probability of collision high, so RTS/CTS beneficial.	Greater impact as packet error rate depends on BER (Bit Error Rate) and packet Length.	

Table 2.5: Effect of Packet Size

2.6 Packet Loss Modeling

To model packet loss means to develop a parametrized model that accounts for errors due to wireless effects such as noise, fading, shadowing and interference. Communication channel can be classified into two categories: memory-less channels and channels with memory [8]. Space or satellite channels are examples of memory-less channel. If errors are random (due to noise) and do not depend on state or time, this model is known as finite state memory less channel. As a result, errors appear randomly on memory-less channels, and therefore these channels are often referred as random-error channels.

Many real communication channels have memory. Errors on these channels tend to occur in clusters or bursts [8]. A Markov model is generally used when the occurrence of errors is correlated in time or depends on state information. This model goes back to the early work of Gilbert- Elliott who used a two state Markov chain where one state corresponds to a noisy channel and the other state to a noise free channel.

2.6.1 Gilbert-Elliott: The Classical 2-State Markov Model for Error Processes

Gilbert-Elliot is probably the simplest model which is widely used for describing error patterns in transmission channels. This model denotes a *good* (G) and *bad* (B) state. Each of them may generate errors as independent events at a state dependent error rate, $1-k$ in the good and $1-h$ in the bad state, respectively. The model is shown in Figure 2.9. For applications in data loss processes, we interpret an event as arrival of a packet and an error as a packet loss. The transition matrix A is given by the two transitions.

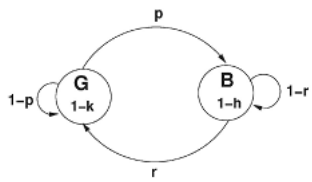


Figure 2.9: The Gilbert-Elliott model generating a 2-state Markov modulated failure process

Gilbert model memorizes only the previous state, thus the probability, that the next packet will be lost is dependent only on the previous state.

$p = P\{q_t = B|q_{t-1} = G\};$

$r = P\{q_t = G|q_{t-1} = B\};$

$$A = \begin{bmatrix} (1-p) & p \\ r & 1-r \end{bmatrix}$$

Where, q_t denotes the state at time t.

We can build loss-model with infinite number of states (m is infinite value in Figure. 2.10). Such model gives us opportunity to model packet loss probabilities in dependence on burst lengths (several consecutively lost packets). If packet is correctly received, then the state returns to $X = 0$ [9].

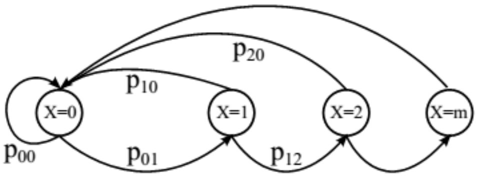


Figure 2.10: Loss Model with Infinite No of states

In order to model packet losses, it is sufficient to use models with limited number of states. Three the most commonly used models with limited number of states are: the k-th order Markov chain model, 2-state Markov chain model and the Bernoulli loss Model.

2.6.2 Markov Model for Error analysis

One of the most used models for time varying wireless link errors is the Two State Markov Model. The Markov Error model has two states: error state and error free state, each having its own distribution. When a channel is in error state, any packets sent would be either lost or corrupted. In error free state, all packets are successfully transmitted over the wireless link. Each user has its own Markov Model that is some users may be experiencing an error state at a given time interval, where others may have error free transmission. This effect is a result of location dependence of errors as well as mobility of users. In the Markov Model the length of stay in each state can be expressed in terms of transitional probabilities

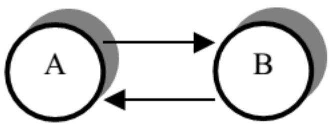


Figure 2.11: Two-state continuous time chain

A versatile error model uses a two-state continuous-time Markov chain. As depicted in Figure 2.11, the model alternates between two states (A and B) which can have different dropping rates.

One possible configuration is to have state A set with 0% drop rate corresponding to "good" periods where the network does not corrupt packets, and state B set with a non-zero drop rate corresponding to "bad" periods when packets are dropped in the network. The model uses exponential distributions to determine the sojourn times in each state. It can be configured as a Semi-Markovian model in which the sojourn time in a state are sampled from a preselected but arbitrary distribution.

2.6.3 Other Error Models

Rate-based Error Models: This model drops a specified proportion of segments. Trivial rate-based error models drop a predetermined set of segments. For example, the error model drops the 2nd, 12th, 19th, and so on, incoming segments. Such an error model does not simulate realistic situations.

Temporal Error Models: Temporal error models capture the effect of the channel going "bad" for a specified proportion of time, during which transmitted segments will be lost only if they happen to be in transit at that time. The two-state model of Figure 2.11 can be configured as time-based. Here, instead of assigning a drop rate to each state, we designate one as a "good" state where no segments are dropped and the other as "bad" where all segments are dropped. This temporal error model, of course, is simply an On/Off model. In order to have a comparable temporal model we need to allow partial dropping in a state as well.

Temporal Four-State Semi-Markovian Based Model: The model has two main states, A and B. Each state has two sub-states ON and OFF. Figure 2.12 illustrates a temporal four-state Semi-Markovian based model. While the model is in an OFF sub-state (A OFF or B OFF) no incoming packets are dropped (dropping OFF), whereas during the ON sub-states all incoming packets are dropped. Clearly this model is more versatile than the time-based two-state model of Figure 2.11, and allows partial dropping in each main state.

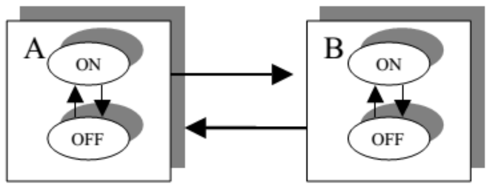


Figure 2.12: Temporal four-state Semi-Markovian based model

2.7 Different Approaches to change BEB in Previous Researches

MAC protocols can be modified based on the backoff mechanism, which in turn uses the contention window to modify the behavior of the protocol. The use of the contention window can be a modification of the window size or of the way of increasing/decreasing it. The protocols that belong to this category are mainly based on the DCF of IEEE 802.11 for the medium access method. As we are using Backoff based scheme to modify IEEE 802.11 MAC protocol behavior in error prone channel, we only cite works that focus on CW modification schemes that can be related with ad hoc wireless error prone channel.

Through BEB of Traditional IEEE 802.11: There are some approaches where just after a successful transmission, the contention window is decreased, and after a collision, the contention window is increased. Surprisingly, in the literature, most of these algorithms are mainly designed for single-hop networks, and very few of them are tested in ad hoc networks conditions. MILD (Multiplicative increase Linear Decrease) and DIDD (Double Increase double decrease) scheme, which are mainly based on IEEE 802.11 change contention window after every transmission trial.

Through Local Information Based CW Control: The protocols of this type modify the contention window based on some computations made by the individual station. These computations use some information gathered by each station from its neighborhood. For example, in MBFAIR [12], the algorithm computes and modifies the contention window size accordingly to ensure fair share. The fair share is computed based on some information received from the two-hop neighborhood. These algorithms mainly focus on fairness issues. But when the number of stations is not known it is difficult for the protocol to have an optimal behavior.

Through History based CW control: In this approach, backoff algorithm optimizes *CW* size by considering history of packet loss. For history based *CW* control scheme backoff range is divided into several small backoff sub-ranges. In the proposed scheme, several network levels are introduced, based on an introduced channel state vector that keeps network history. After successful transmissions and collisions, network nodes change their *CW* based on their network levels. CS(Channel State) Vector updated as per selected sub-range upon each transmission trail. History Based Contention window control (HBCWC), Dynamic, deterministic contention window control (DDCWA) and Determinist Contention window (DCWA) algorithm also fall in this category. Extra memory space required and additional computation required to be done for keeping and maintaining channel state array.

Interval Based Fixed CW Setting Using Local Information: This type of solution relies on local information (gathered during a given time interval). Successful transmission and collisions undergone by each station has been calculated without taking advantage of carrier sensing mechanism. Contrary to BEB algorithm, modified protocol has only two distinct contention window sizes for each interval and it uses the same contention window for all the packets it has to send in a given time interval. At the end of this interval, some computations are done to choose the contention window for the next interval. *CW* size is not modified after each transmission, in order to reduce oscillation between two states. These algorithms mainly focus on fairness issue and gives better performance in unloaded networks.

Table 2.6 and 2.7 list some previous research issues.

There are some research works, which concentrate on performance and energy efficiency issues of wireless error prone environment. Some researchers developed an analytical model with Markov chain model to identify effects of packet size and CW in error

Table 2.6: Research issues considering BEB Modification via CW Change scheme.

Scheme	*Research Focus*	*Citations*
Traditional BEB like scheme	MILD, DIDD schemes for changing CW after a collision or success.	[10], [11]
Through local information	Computations done by each station with local information gathered from neighborhood.	[12], [13], [14]
History Based CW control	Slowly modify CW value based on history of packet loss. Instead of just doubling the upper bound of CW, DCWA and HBCWC algorithms increase both backoff range bounds (Upper and lower) and selects CW within a specific sub range.	[5], [20], [17], [19], [18]
Interval based Fixed CW setting	Two distinct contention window sizes selected for each interval based on local information like collisions and successful transmissions undergone by the station.	[1]

prone channel, where some works [4] provide saturation throughput analysis to show that channel errors have significant impact on system performance. Although packet transmission errors are considered in some papers, the impact of packet errors on collision probability and packet sending probability has been ignored as described by authors in paper [3]. Most of the papers limit their work on the throughput and delay under saturated traffic condition. Very few solutions improve fairness and efficiency at the same time. Impact of optimal packet size and effects of contention window and packet size on the energy efficiency of wireless local area network have been analyzed with the help of Markov model under saturated traffic condition by authors in [2], [3]. But simulation has been done only for single hop network for both the mentioned papers.

Table 2.7: Research issues considering Error Prone Channel.

Scheme	*Research Focus*	*Citations*
Markov Model Based schemes	Analytical model developed with Markov chain model to identify effects of packet size and CW in error prone channel.	[2], [3]
Throughput analysis	Saturation throughput Analysis to show that channel errors have significant impact on system performance.	[4], [22]
Performance Improvement schemes	On Improving performance under congested and error prone environment through DIDD (Double Increment Double Decrement) scheme.	[6]

Chapter 3

Modified MAC Protocol

In this chapter we discuss our proposed contention window setting algorithm based on packet size distribution for wireless ad hoc error prone networks. The modified scheme proposed here dynamically tunes contention window range based on adaptive packet labeling. Here packet size plays a vital role in addition with contention window for maximizing performance in error prone wireless network.

Before going through the details of our proposed algorithm, we concentrate on two important parameters contention window and packet size relevant to wireless error prone channel.

3.1 Impact of Packet Size and Contention Window on Error Prone channel

As we know, collision is one of the main reason for delivering varying service rates due to random access when multiple nodes transmit at the same time. When more than one node simultaneously finishes backoff according to a binary exponential backoff mechanism collision occurs. On the other hand, when two independent transmitters try to send packet to the same receiver, collision due to concurrent transmission leads to throughput degradation which necessitates the retransmission of the packets. RTS/CTS handshaking is proposed to address these issues, but additional delay and protocol overhead makes RTS/CTS mechanism particularly inappropriate for delay-sensitive real-time services such as video [24].

Another significant cause of performance degradation and its variability is packet *corruption* in IEEE 802.11 network, which is often ignored by many researchers. Packet error typically happens due to non-ideal channel conditions. During propagation through wireless medium, radio waves suffer from several impairments such as attenuation, multi-path fading and interference. In mobile wireless networks, path loss, fading and interference cause variations in the received SNR (signal to noise ratio), which influences the bit error rate. Bit error rate can be considered as an approximate estimate of bit error probability. A packet is declared incorrect if at least one bit is erroneous. Packet error rate (PER) is determined by the packet length and bit error rate (BER).

The upper bound of the PER can be expressed as:

$PER = 1\text{-}(1 - BER)^L$

Where L= $Data\, packet\, length\, in\, bytes = Physical\ Layer\ header{+}Mac\ Layerheader{+} Packet\ header$

For small error probabilities, this is approximately, $PER \approx BER \times L$

IEEE 802.11 specification requires all nodes to choose a random backoff interval between 0 and CW (contention window) dynamically controlled by Binary Exponential Backoff (BEB) algorithm, and wait for that chosen number of slot times before trying to access the channel for collision avoidance. Initially, CW is set to CWMin (minimum contention window size). However, when there is a collision, the contention window size is doubled, until a maximum value CWMax. This technique of randomization and scaling the contention window size is used to reduce collisions by preventing multiple nodes from sending packets in the channel at the same time. For ideal channel selection of optimal CW impacts more because choosing a larger one leads to large backoff interval and larger overhead before each packet transmission. But a smaller CW will cause more collisions. A larger packet size will cause each station to wait longer before its next chance for channel access, but increases throughput by reducing protocol overhead in ideal channel.

The scheme of scaling and randomization of CW doesn't facilitate, when the channel is inherently noisy because impact of packet error on performance metrics of IEEE 802.11 protocol is different from that of collisions. Error prone network causes packet retransmission at sender station and reception error at receiver station. But for both collision and corruption, only collision resolution measures are taken in original DCF. In both situations, the sender will not be able to receive acknowledgment from the receiver, it will increase its backoff counter and re-transmit, until the number of retries is met or the packet is transmitted successfully. We know that packet retransmissions incur delay and packet loss at sender station. Moreover, it introduces unnecessarily delays by setting additional time separation for some nodes when these are actually not contending at all. So in a non-ideal channel for non-contending nodes impact of optimal CW is less. But at the same time selection of larger CW induces longer delays.

As packet error rate linearly depends packet length, packet size can have greater impact on error prone channel. It is shown that under error-prone environment, a trade-off exists between the desire to reduce the ratio of overhead in the data packet by adopting a larger packet size, and the need to reduce the packet error rate by using a smaller packet length [3]. It is also observed that, in error-prone environments, optimal packet size has more significant improvement on the performance than optimal contention window [2] and combing both optimal contention window and optimal packet size can achieve the maximum optimization.

We find that there is much space for research on tuning contention window(CW) for error prone channels based on sizes of packets. We,in this paper devise a method of adaptively tuning CW with the help of packet size distribution to achieve better performance under different error rates of error model.

3.2 Basic Idea for Adjustment of CW Based on Packet Sizes

In this section, we discuss basic idea of our proposed contention window setting scheme based on packet size distribution for ad hoc wireless error prone network.

To devise a modified MAC scheme for error prone environment, we propose a dynamic contention window range control scheme. We have made selection of appropriate CW dependent on packet size.

As we know, delivery delay of packets in IEEE 802.11 DCF depends on two things, the size of the contention window and no of retransmissions. Retransmission depends on collision probability, which in turn depends on how many nodes are contending for the channel. It is already known that collision probability increases with increasing number of nodes. In classical MAC DCF protocol, all nodes who want to access media have to contend and choose a random back off interval between 0 and CW (contention window), and wait for the chosen number of slot times before trying to access the channel. So, in our approach, we partition packets among different collision sub-domains depending on their sizes and assign them different sub-range from the entire contention window range, namely 0-CW.

To partition packets into groups based on their sizes, we introduce the notion of *labels* and *bins*. We divide packets flowing through the network into a set of bins based on their respective sizes. We consider 4 bins for packets each with their own start and end boundary. Packets having a certain size, i.e., higher than the low boundary of a certain bin but smaller than the high boundary of the same bin, belong to that particular bin. *label* is used to classify packets based on which bin they reside. High boundary of a *bin* is defined by *label*. In our proposed scheme, we consider 4 bins and we mark the boundary sizes by LOW, MEDIUM and HIGH respectively (from lower size packets to higher size packets).

If f_1 fraction of packets come with size higher than 0 and lower than label LOW, than they will belong to bin1. f_2, $f_3$and f_4 are the fraction of packets that reside in bin2,

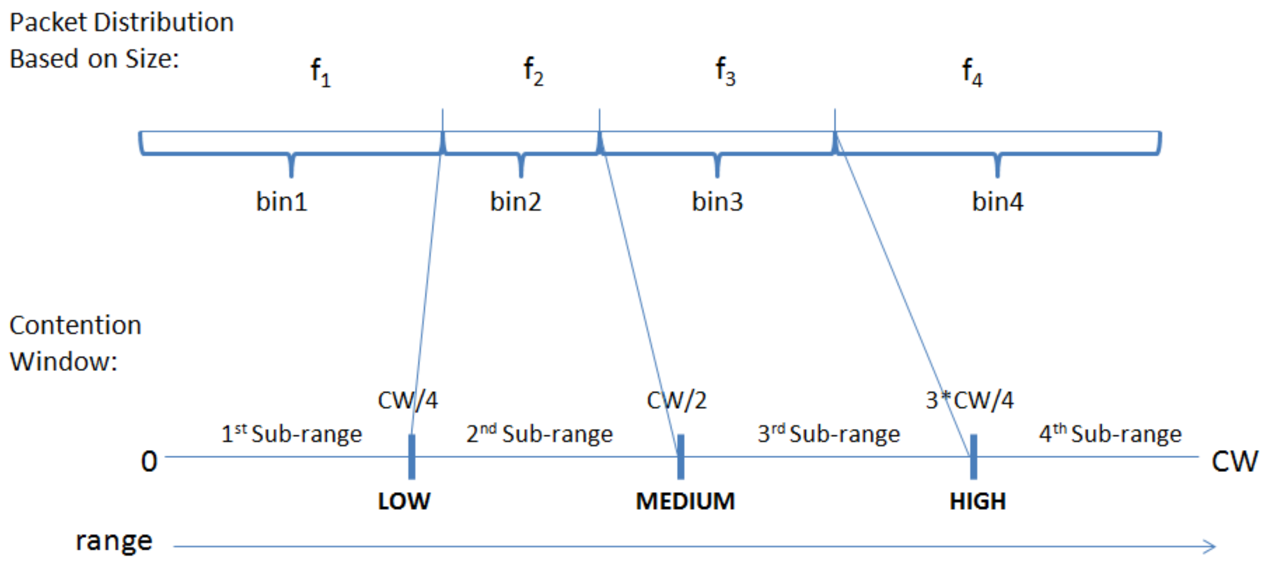


Figure 3.1: Packet distribution in Bins for selection of CW sub-range

bin3 and bin4 as depicted in Figure 3.1. To capture the ad hoc nature of network, we adaptively set the value of packet labels based on network traffic in different intervals.

Now to create different contention domain for the packets belonging to different bins, we divide total CW range into several sub-ranges and each sub-range is assigned to each bins. That is, packets residing in a certain bin choose their next CW sampled from its respective CW sub-range. Since lower sized packets (packets from bin1) are usually control packets and should be tried immediately, they are assigned lower CW sub-range. Higher sized packets are assigned progressively higher CW sub-ranges. Every time when retransmission happens and CW gets doubled (range becomes twice as much as its earlier value), the respective sub-ranges for respective bins hold. So under error prone channel, every packet is being retransmitted until its successful reception but with a decreased contention probability as no of contending nodes for a specific bin is less compared to legacy DCF.

Now the question remains how these boundary sizes for binning and CW subranges are determined. We used the following simple strategy:

Packets belonging to a certain bin take a CW subrange in proportion to the number of packets reside in that bin.

For example, if bin1 contains 30% packets generated in the network, it takes first 30% of the whole contention window, if bin2 contains 40% packets it takes the next 40% contention windows. Since total portions of packets across all bins would be 100%, CW sub-range would up-to 1, thus taking up the whole range. Admittedly, packets in bin1 experiences less contention wait time than packets belonging in bin4, as they take sample from smaller contention window range. We cannot, however, put *all* packets into bin1 because in that bin1 packets would take entire CW range thus nullifying their very advantage of being in bin1. Now the question is what fraction of packets each bin should contain, or put it in a different way, how to determine the bin boundaries, so that the overall packet delivery improves, say packet delivery delay is reduced. In the following section 3.3, we attempt to establish this based on some assumption for error-prone channels.

After successful transmissions and collisions, network nodes change their CW based on their packet size. The size of CW initiates with a minimum size (CWMin) and doubles with every attempt to transmit that is deferred, until a maximum size (CWMax) is reached for the range as defined in standard IEEE 802.11 DCF. Main steps of our proposed scheme is mentioned below:

- Labeling packets into groups based on packet size distribution.
- Bins are determined based on some percentile of packet size distribution.
- Identifying boundary size of CW sub-range for each group.
- CW adjustment for each packet belongs to a specific bin.

3.3 Probabilistic Analysis of Uniform Binning

It has been argued that packet size significantly affects collision and corruption, contention window can be adjusted based on packet sizes. As we described, packets belonging to a certain bin takes a certain portion of contention window sub-range when a scheduling the packets. In that, each bin maps to a certain CW sub-range. It is therefore important to see what fraction of packets should fall into bins that would lead of improved performance, say reduced overall delay.

We try to estimate expected delivery delay of packets as a function of fractions of packets that fall into different bins. Let f_1, f_2, f_3, and f_4 be the fractions of packets belonging to bin1, bin2, bin3 and bin4 respectively. While we analytically deduce the optimal values for the mentioned f's, we make the following assumptions:

- Collision probability remains constant across bins. That is, collision chances among packets remain unchanged as we do binning, and even in successive retransmission attempts.
- Probability of packet corruption due to bit error rate depends linearly on packet sizes.
- Packet of different sizes are equally likely.
- Packet delivery delay mainly depends of contention wait times and the number of retransmissions suffered by packets (due to either collision or corruption).
- We have considered the analogy, smaller delay for smaller packets and longer delay for larger packets.

Unsuccessful transmission occurs when more than one user simultaneously transmits packets that collide with each other or unsatisfactory channel conditions corrupt the packet at the receiver even if the packet contends successfully. We are considering packet corruption probability as P_c and packet collision probability as P_{col}. While collision probabilities P_{col}'s remain the same across four bins, corruption probabilities P_c's vary across bins as different bins contain different sized packets. So let P_{c_1}, P_{c_2}, P_{c_3} and P_{c_4} denote the corruption probability of packets that reside in bin1, bin2, bin3 and bin4 respectively.

Let s_{max} denote the size of the largest packet in the network. Since packets are assumed to be equally likely and bin1 contains f_1 fractions of packets, packets smaller than $f_1 \times s_{max}$ belongs to bin1. Similarly, packets in bin2 are smaller than $(f_1 + f_2) \times s_{max}$.

Since as per assumption corruption probability linearly depends of packet lengths, we can write the following:

$$P_{c_1} \propto f_1 \times s_{max} \tag{3.1}$$

$$P_{c_2} \propto (f_1 + f_2) \times s_{max} \tag{3.2}$$

$$P_{c_3} \propto (f_1 + f_2 + f_3) \times s_{max} \tag{3.3}$$

$$P_{c_4} \propto (f_1 + f_2 + f_3 + f_4) \times s_{max} \tag{3.4}$$

$$P_{c_4} \propto s_{max}, \text{ as } f_1 + f_2 + f_3 + f_4 = 1 \tag{3.5}$$

The above equations can be rephrased as the following for some suitable constant γ:

$P_{c_1} = f_1 \times \gamma$

$P_{c_2} = (f_1 + f_2) \times \gamma$

$P_{c_3} = (f_1 + f_2 + f_3) \times \gamma$

$P_{c_4} = \gamma$

Since retransmission happens when collision or corruption occurs and these two events are independent, the retransmission probability per bin can be obtained as follows:

$$P_{Ret_1} = 1 - (1 - P_{c_1})(1 - P_{col}) \tag{3.6}$$

$$P_{Ret_2} = 1 - (1 - P_{c_2})(1 - P_{col}) \tag{3.7}$$

$$P_{Ret_3} = 1 - (1 - P_{c_3})(1 - P_{col}) \tag{3.8}$$

$$P_{Ret_4} = 1 - (1 - P_{c_4})(1 - P_{col}) \tag{3.9}$$

The number of retransmissions per packet can be modeled as geometric random variable. Using $X \sim geom(p)$ gives $E[X] = \frac{1}{p}$, we get the expected number of retransmissions for bin1 packets as:

$E[Ret_1] = \frac{1}{\{(1-P_{Ret_1})\}}$

Putting values of P_{Ret_1} and P_{c1} in the above equation we can get below mentioned equation,

$$\begin{aligned}
E[Ret_1] &= \frac{1}{(1-P_{c_1})(1-P_{col})} \\
E[Ret_1] &= \frac{1}{(1-(f_1 \times \gamma))(1-P_{col})}
\end{aligned}$$

Each time a transmission or retransmission happens, packet residing a certain bin chooses a sample CW (randomly pick one from its respective CW sub-range mapped to its bin) and waits that amount of time to access the channel. The amount of wait time thus depends on the (high) boundary values of CW sub-ranges. For example, wait times for bin1 packets are proportional to $f_1 \times CWMax$, for bin2 are $(f_1 + f_2) \times CWMax$, and so on.

Let delivery delay of bin1, bin2, bin3 and bin4 packets be D_1, D_2, D_3 and D_4. If we put values of P_c's, we get below mentioned equations for expected delay for packets in each bin. For bin1, we get:

$$E[D_1] = \frac{f_1 \times CWMax}{(1-f_1 \times \gamma)(1-P_{col})} \tag{3.10}$$

Replacing all constants with C, we get:

$$\begin{aligned}
E[D_1] &= \frac{f_1}{(1-(f_1 \times \gamma))} \times C && (3.11) \\
E[D_2] &= \frac{(f_1+f_2)}{(1-((f_1+f_2) \times \gamma))} \times C && (3.12) \\
E[D_3] &= \frac{(f_1+f_2+f_3)}{(1-((f_1+f_2+f_3) \times \gamma))} \times C && (3.13) \\
E[D_4] &= \frac{1}{(1-\gamma)} \times C && (3.14)
\end{aligned}$$

Hence, the overall expected delay can be denoted as $E[D]$, which is the summation of delays in each bin weighted by their respective fractions.

$$\begin{aligned}
E[D] &= \frac{f_1 f_1 C}{(1-(f_1 * \gamma))} + \frac{(f_1+f_2) f_2 C}{(1-((f_1+f_2)\gamma))} + \\
&\quad \frac{(f_1+f_2+f_3) f_3 C}{(1-((f_1+f_2+f_3)\gamma))} + \frac{f_4 C}{(1-\gamma)} && (3.15)
\end{aligned}$$

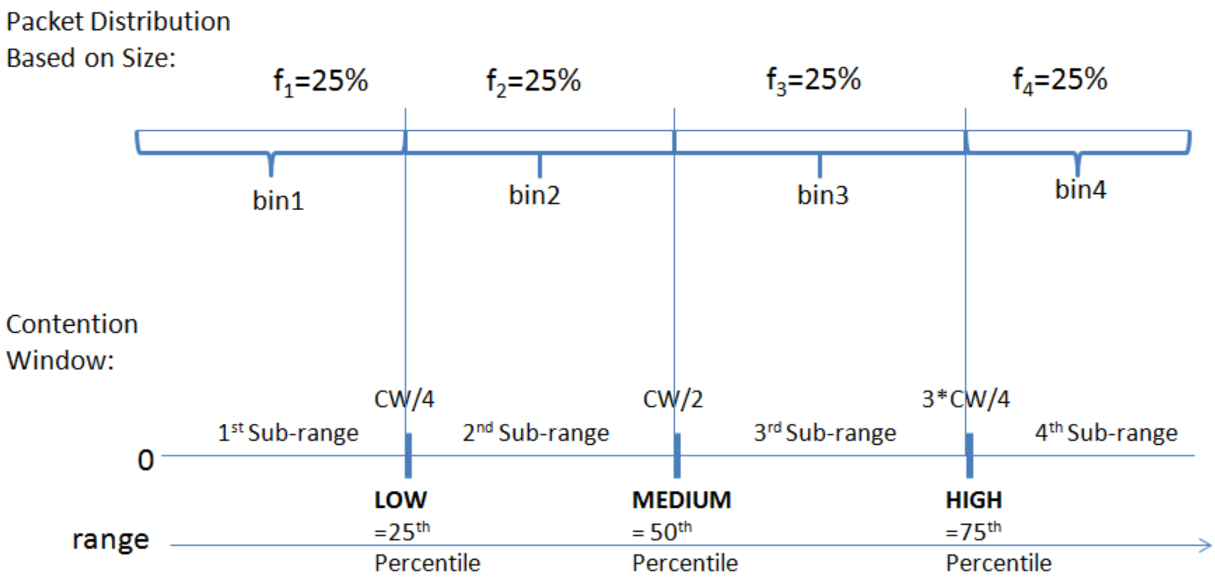


Figure 3.2: Uniform Binning for selection of CW Sub-range

We want to minimize $E[D]$ for certain values of f's. Using an optimization solver (`www.ampl.com`), we obtain the following assignments:

$$f_1 = 0.250009$$
$$f_2 = 0.250003$$
$$f_3 = 0.249997$$
$$f_4 = 0.249991$$

For simplicity, we consider 25% packets in each bin. That means, each bin should contain equal 25% of packets. Therefore, CW range is also equally partitioned to 4 sub-ranges, each of size $CW/4$. This is shown in Figure 3.2. We have considered 4 bins and for simplicity and based on our proposed strategy take a CW sub-range in proportion to the number of packets reside in that bin. But the whole analogy can be extended for n bins. However, we have shown simulation results for 8 bins with each bin containing almost equal to 12.5% of packets and CW range is also equally partitioned to 8 sub-ranges. Although we assumed that packets are of uniform sizes, the same results follow for skewed distribution also (as we see in experiment section).

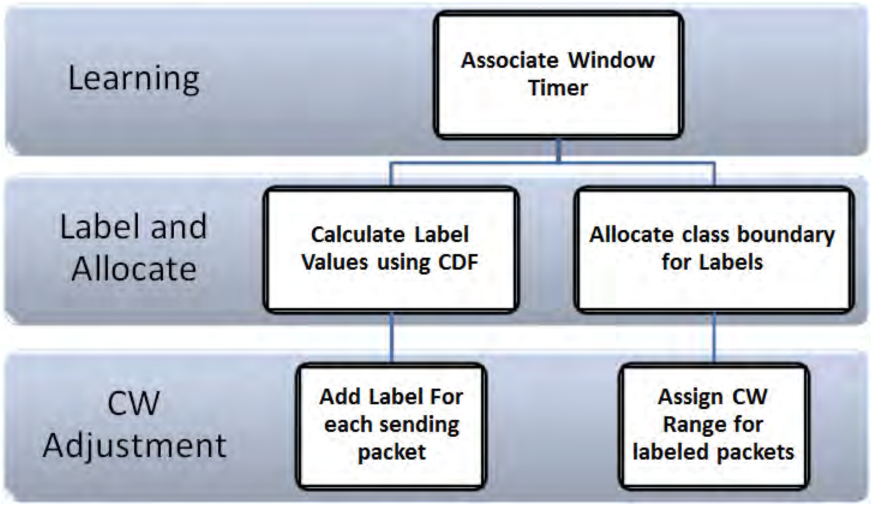


Figure 3.3: Proposed Methodology

3.4 Modified MAC Protocol

In our methodology, we keep main principle of the MAC Protocol of IEEE 802.11 DCF unchanged. The only difference between modified and original algorithm are the conditions that determine which contention window size to use. The steps in modified method has been defined in Figure 3.3.

3.4.1 Packet Labeling

3.4.1.1 Learning Phase

To incorporate the idea mentioned above, initially each node has been equipped with window timer, which runs in background. Our protocol relies only on local information that a station can derive from its own experience without using any measure from the carrier sensing mechanism. We need to observe current distribution of packet size and for this packet distribution sensed by individual nodes have been considered here. A node at first learns the sizes of packets it is generating or forwarding as an intermediate node for a certain time interval. Having received a packet p with size s, node increments the arrival frequency count for packet size s. We require packet sizes and arrival frequency to adaptively set the packet labels based on network traffic. It also continues its learning process during the simulation and adjusts the labeling values dynamically from one interval to another.

We are using static network here. So it is assumed that packets received or forwarded through an individual node will remain almost same in subsequent intervals.

3.4.1.2 Packet Allocation in Bins

Based on data collected from learning phase, we have to adaptively set the values of packet *labels* based on network traffic such that around 25% of of packets fall into a specific bin. Total 4 bins have been considered in our proposed algorithm with 3 labels LOW, MEDIUM and HIGH. We have considered 25, 50 and 75 percentile values for labeling packets as LOW, MEDIUM and HIGH.

Total no of different packets is n. X is a random variable denoting packet size. So to calculate f percentile, we need to find the packet size X where f% packets falls below or equal to X. At the end of every window timer, it wakes up and calculates f percentile for the next interval.

Cumulative distribution function(CDF) is used to calculate label values for each bin. Cumulative distribution function $P[X <= f]$, describes the probability that a real valued random variable X with a given probability distribution will be found to have a value less than or equal to x.

Mathematically, it can be described as follows:

Suppose the sizes of packets flowing through a node are $s_1, s_2, s_3, \ldots, s_n$ (in ascending sorted order) with probability $p_1, p_2, p_3, \ldots, p_n$. Then the value of a label is set to a value such that:

$Pr\{X \leq Label\} = f$

Where X is a random variable denoting packet size. For our 3 labels the above equation can be defined as:

$Pr\{X_1 \leq LOW\} = f_1 = 0.25$

$Pr\{LOW < X_2 \leq MEDIUM\} = f_2 = 0.25$

$Pr\{MEDIUM < X_3 \leq HIGH\} = f_3 = 0.25$

$Pr\{X_4 > HIGH\} = f_4 = 0.25$

where f_1, f_2, f_3 and f_4 are the percentile values that fall into a specific bin.

Let us denote P_i be the probability that a packet's size is less than or equal to s_i. Then mathematically,

$P_i = \frac{(\sum_{j=1}^{i} fi)}{(\sum_{k=1}^{n} fk)}$

Where, f_i is frequency count for packet size s_i.

Here P_i is the cumulative distribution function (CDF). Using this CDF and linear interpolation, we calculate values of LOW, MEDIUM and HIGH. After that we update the value of labels to compensate the relative weight of previous label values in determining the current LOW,MID and HIGH labels for this interval. Calculation for LOW label is shown below:

$LOW = \left\{ \frac{(f_1 - P_{LG}) * (S_{GL} - S_{LG})}{(P_{GL} - P_{LG})} \right\} + S_{LG}$

Where,

P_{LG} =Greatest probability less than f_1

P_{GL} =Least probability greater than f_1

S_{LG} =Packet size at P_{LG}

S_{GL} =Packet size at P_{GL}

Values for MEDIUM and HIGH have been calculated in similar way as mentioned above.

3.4.2 Contention Window Adjustment Based on Labels

Based on packet labeling, we partition packets into 4 bins with almost *equal* number of packets in each bin. Now we are going to assign a separate contention window sub-range for packets belonging to a certain bin. But in our proposed algorithm, we have made this random selection dependent on *packet size*. To create separate collision domain for packets in different bin, CW has been *uniformly* partitioned into segments from CWMin to CWMax. LOW labeled packets pick from 0 to $\frac{CW}{4}$, MEDIUM labeled packets from $\frac{CW}{4}$ to $\frac{CW}{2}$, HIGH labeled packets from $\frac{CW}{2}$ to $\frac{3*CW}{4}$ and packets greater than HIGH pick from $\frac{3*CW}{4}$ to CW. After each failure, CW gets doubled and $CWMax$ remains same as per standard protocol. In our scheme, we have considered $CWMin$ as 127 for our initial learning phase.

3.4.2.1 Why Smaller Packets to Smaller CW Sub-Range?

Based on property mentioned below, it can be denoted that if average backoff Time decreases, collision probability increases.

Property : If average Backoff Time decreases, collision probability increases.
Proof: Average probability of each node to send a packet in a idle medium is:

$\partial = \frac{1}{1+BT}$

Where,

∂ =average packet transmission probability of each node if the medium is idle

CW_i= Contention window after i times of collision

$= 2^i * CWMin, i = 0, 1, 2, \ldots, m$

We are considering here only basic mechanism and disabled RTS/CTS scheme in our proposed modified algorithm.

m= Long Retry Limit (LRC)

BT= average backoff timer value of each node= $\frac{(CW_i-1)}{2}$

for ith backoff $= \sum(\frac{(CW_i-1)}{2}) * P_i$ Where, $i = 0, 1, \ldots, LRC$

If no of collisions more, BT will increase.

From another property it can be stated that, collision probability increases with increasing number of nodes, When a node transmits a packet with probability ∂, the probability of collision with other nodes P is,

$P = 1 - (1 - \partial)^{n-1}$

Where $n =$ no of contending nodes

∂ =average packet transmission probability if the medium is idle

From the above equation, if no of contending node that is n increases, collision probability also increases

If $n = 3$,

$P_3 = 1 - (1 - \partial)^2 = 1 - (1 - 2\partial + \partial^2) = 2\partial - \partial^2$

$0 < \partial < 1$

For any two random variable x and y denotes backoff timer where

$x < y$

$(\frac{1}{1+x}) > (\frac{1}{1+y})$

$\partial_x > \partial_y$

From above equation it can be said that, if no of contending nodes are constant, increased BT will cause lower packet sending probability.

We consider number of contending nodes constant. So collision probability for

$n = 3$,

$P_{3x} = 2\partial x - \partial_x^2$

$P_{3y} = 2\partial y - \partial_y^2$

$P_{3x} > P_{3y}$; for $\partial_x > \partial_y$

And we already know that probability of corruption and as well as retransmission is comparatively low for smaller packets. Collision probability is also low for smaller packets. Collision and corruption both probability high for larger packets, so expected number of retransmissions will be more for them. If we consider higher range for larger packets, it will decrease the number of retransmissions. So we consider the analogy smaller delay for smaller packets and longer delay for larger packets in our proposed one.

3.5 Disadvantage of Using Fixed Packet Sizes for Labeling

As the packet size of a network is random and not known before, with a fixed value used for packet labeling (LOW, MEDIUM and HIGH) it may happen that all the packets in the network are having sizes larger than any fixed label. Consequently all of the packets will not fall uniformly under a specific bin. As we are planning to limit the selection of contention window (CW) within a definite sub-range based on packed labeling, uniform distribution of packet sizes in different bins is a prerequisite for it. Otherwise we can't intelligently inter mix the packets among the bins as desired. To accommodate this need, we have to learn the packet sizes flowing through the network and adaptively set the value of labels.

3.6 Proposed MAC Algorithm

In this section we present our modified algorithm. The detailed algorithm is as follows:

Set Packet Label Before Sending Data Packet: In basic 802.11 MAC protocol, `sendData()` function builds MAC header for data packet. This function increases the size of the packet, setting the type and subtype as data. After having a complete MAC header attached to it, packet is then attached to pktTX_ variable.

In our modified algorithm, we have introduced SIZE and STAT variables to define packet labels before sending any data packet. In learning phase of window timer we got following information:

PKT_SIZE_LOW: Derived packet size for label LOW from window timer

PKT_SIZE_MEDIUM: Derived packet size for label MEDIUM from window timer

PKT_SIZE_HIGH: Derived packet size for label HIGH from window timer

PKT_STAT_LOW: Defined packet status for label LOW during `sendData()` function before sending any packet

PKT_STAT_MEDIUM: Defined packet status for label MEDIUM during `sendData()` function before sending any packet

PKT_STAT_HIGH: Defined packet status for label HIGH during `sendData()` function before sending any packet

Algorithm 3.1 Packet label setting during sendData()

if $packet[size]<$PKT_SIZE_LOW **then**

  {$packet[label] \leftarrow$PKT_STAT_LOW}

else if PKT_SIZE_LOW $<$ $packet[size]$ $<$ PKT_SIZE_MEDIUM **then**

  {$packet[label] \leftarrow$PKT_STAT_MEDIUM }

else if PKT_SIZE_MEDIUM $<$ $packet[size]$ $<$ PKT_SIZE_HIGH **then**

  {$packet[label] \leftarrow$PKT_STAT_HIGH}

else if $packet[size]$ $>$ PKT_SIZE_HIGH **then**

  {$packet[label] \leftarrow$PKT_STAT_HIGH_1}

end if {*No packet label set in learning phase* }

Set rTime of Backoff Timer Based on Packet Labeling: After expiration of the backoff timer, an RTS or data packet will be transmitted if either is waiting. This is because a node is supposed to wait a brief period of time before transmitting. In `start()` function of backoff timer random value is picked from 0 to CW to calculate how many slots to be skipped before transmitting a frame.

Algorithm 3.2 Start() function of Backoff Timer in basic MAC

stime← *s.clock*()

rTime ← *(randomly select from 0 to CW) * SlotTime*

schedule backoff timer with this rTime and difs_wait time

Before going through the modified algorithm below mentioned variables have been used to modify selection of waiting period.

PKT_STAT_LOW : Derived packet status for label LOW from `sendData()` function

PKT_STAT_MEDIUM : Derived packet status for label MEDIUM from `sendData()` function

PKT_STAT_HIGH : Derived packet status for label HIGH from `sendData()` function

Algorithm 3.3 Modification in Backoff Timer based on packet labeling

if label ==PKT_STAT_LOW **then**

$\{rTime \leftarrow$ *(randomly select from 0 to* $\frac{CW}{4}$*) * SlotTime* $\}$

else if *label*==PKT_STAT_MEDIUM **then**

$\{rTime$ = *((randomly select from 0 to* $\frac{CW}{4}$*) +* $\frac{CW}{4}$*) * SlotTime*$\}$

else if *label* == PKT_STAT_HIGH **then**

$\{rTime \leftarrow$ *(randomly select from 0 to* $\frac{CW}{4}$*) +*$\frac{CW}{2}$*) * SlotTime* $\}$

else if *label* >PKT_STAT_HIGH_1 **then**

$\{rTime \leftarrow$*(randomly select from 0 to* $\frac{CW}{4}$*) +*$\frac{3*CW}{4}$ *) * SlotTime*$\}$

end if

Packet Data Collection During Function RecvData(): For every CBR packet increase frequency count for a specific size of packet and increase total packet count.

Packet [size] ++;

Total_count++;

WindowHandler Function as a Watch Dog: After expiration of window timer, it enters to WindowHandler function. This function initializes and sets the values for packet labels.

Algorithm 3.4 Window Handler Function

1: PKT_SIZE_LOW←*0;*
2: PKT_SIZE_MEDIUM←*0;*
3: PKT_SIZE_HIGH←*0;*
4: *Sort all packet size in ascending order*
5: *Calculate probability for all packets with non zero size*
6: *Calculate value of 25th,50th and 75th Percentile using CDF(Cumulative Distributive Function)*
7: PKT_SIZE_LOW←*value of 25th percentile using linear interpolation*
8: PKT_SIZE_MEDIUM←*value of 50th Percentile using linear interpolation*
9: PKT_SIZE_HIGH←*value of 75th Percentile using linear interpolation*
10: *reset all the values after expiration of WindowHandler timer*
11: *initialize window timer for next interval*

Chapter 4

Simulation Results

In this chapter,we describe the simulation results for modified MAC protocol. Through simulation we study the behavior of our approach and evaluate its performance based on some metrics. We also compare the performance of our approach with original IEEE 802.11 MAC Protocol.

4.1 Simulation Setting

We simulate modified algorithm using NS2, an event driven simulator embedded into the OTCL (object oriented version of TCL) Language developed by LBNL, UC Berkeley, and USC VINT project, with wireless extensions from CMU Monarch project. Undoubtedly, NS-2 has become the most widely used open source network simulator, and one of the most widely used network simulators [23]. Network simulation scripts in NS-2 are used to create the network scenarios and upon the completion of the simulation, trace files that capture events occurring in the network are produced. This captured information could be used in performance study e.g. the amount of packets transferred from source to destination, the delay in packets, packet loss etc.

The main objective of the simulation is to understand the impact of various parameters on the performance issues of the protocol. In this thesis, we have designed and run simulations in Tcl scripts using the simulator objects without changing NS2 core components such as class hierarchy, event schedulers, and other network building blocks. We have only changed the code of IEEE 802.11 MAC layer and it's associated timers. After that we have done post processing to extract required parameters from trace file.

We also simulate original IEEE 802.11 MAC protocol to make a comparison between our approach and existing one. In the following sub sections we describe the various settings of our simulation.

4.1.1 Simulation Environment and Parameters

We run our simulation in two different network topologies.

Grid Topology : We run our simulation for ad hoc wireless networks of size ranging from 4 to 64 nodes, which are placed in a grid format in two dimensional area of 500m $\times$ 500m (from x = 0 to x = 500 and y = 0 to y = 500 in X-Y plane). In most cases, we consider a static network. Sample 5X5 grid topology has been shown in Figure 4.1. For each environment, we run the simulation for 10 times with different seed values and we have taken the average of 10 results with the same setting.

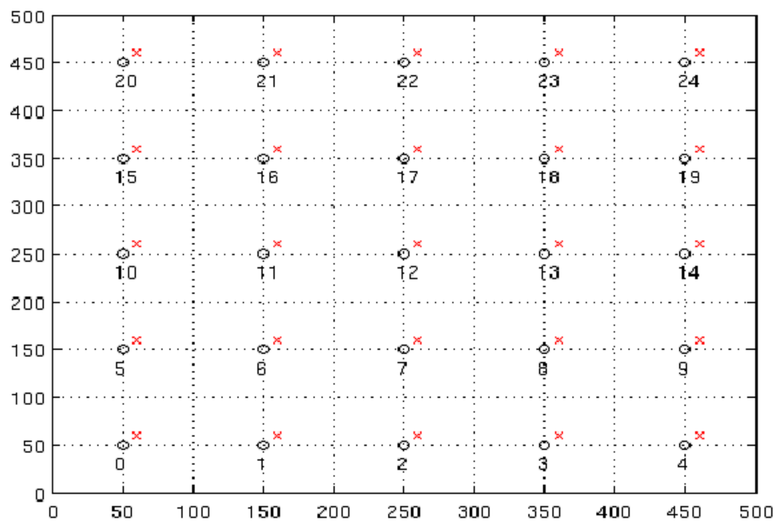


Figure 4.1: 5X5Grid Topology

Random Topology: We run our simulation for ad hoc wireless networks of size ranging from 10 to 60 nodes, which are placed in a random two dimensional area of 500m $\times$ 500m (from x = 0 to x = 500 and y = 0 to y = 500 in X-Y plane). We use multi-hop static environment for simulation purpose. For each environment, we run the simulation for 5 times with different seed values and we have taken the average of 5 results with the same setting. Random placement of nodes have been shown in Figure 4.2

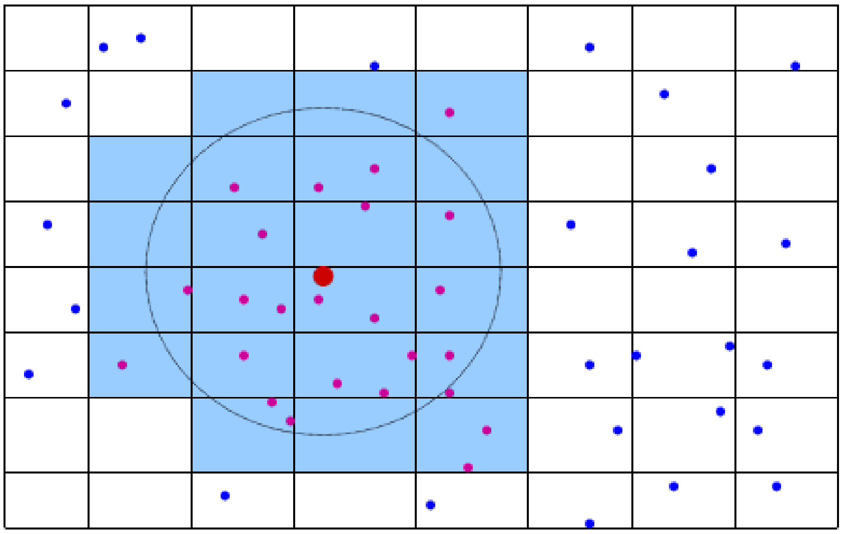


Figure 4.2: Random Topology

We set routing protocol AODV (Ad hoc On-demand Distance Vector Routing). AODV is a reactive routing protocol, which means it starts a route to a destination only on demand. AODV does not need to know the complete path to the destination. In this case, nodes forward the packets to neighbor nodes of the destination.

We have shown basic simulation parameters in Table 4.1 on the next page and 4.2 on page 62.

Table 4.1: Simulation Setting-1

Parameter	**Value**
General Settings for MAC DCF	
Access Medium	MAC/802_11
SIFS	10 µs
DIFS	50 µs
Slot Time	20 µs
CWMin	31
CWMax	1023
Short Retry Limit	7
Long Retry Limit	4
Channel Characteristics	
Channel	Channel/Wireless
Propagation model	Two Ray Ground reflection ($1/r^4$) at far end
Antenna Type	Omni directional
Channel	Symmetric
Propagation Delay	2 µs

Table 4.2: Simulation Setting-2

Parameter	**Value**
TopologySetting	
Simulation scenario	Random and Grid topology
Grid Size (NXN)	2-8
No of Nodes (N)	10,20,30,40,50,60
MaxX (X)	500
MaxY (Y)	500
PacketHeaders	
Preamble Length	144 bits
PLCP Header length	48 bits
PLCP Data rate	1 Mbps
MAC Header	28 bytes
PHY Header	24 bytes
RTS	44 bytes
CTS	38 bytes
ACK	38 bytes

4.1.2 Implementation of Error Model in NS-2

In our simulation, we focus on the Gilbert-Elliott channel model representing a channel with memory. The NS2 simulator supports a multi-state Markov chain error model derived from the Error Model class whose parent class is Connector base class.

With multi-state error model enabled, the link will be in one of the channel states. The multi-state module supports a separate loss module for each channel state with fixed state sojourn times. The unit of error can be specified in term of packet, bit or time. We have used Packet as the unit of error in our simulation. The multi-state error model implements time based error state transitions. Transition to the next error state occurs at the end of the duration of the current state. The next error state is then selected using the transition state matrix.

We define the transition state model matrix and an initial state to control the activation of these loss modules or channel states. To create a multi-state error model, we provide following parameters as defined in ns/tcl/lib/ns-errmodel.tcl.

- States: An array of states (error models) where no packet drop happens in good state and varied packet drop happens in bad state. We use varied error rates like 0.25, 0.5, 0.75, 0.9 and 1.0 in bad state of our simulation.
- Periods: An array of state durations, which defines how much time system stays in one state. After the period the system can change the state using the transition matrix. We use 0.9375s and 0.0625s duration in good and bad state respectively.
- Trans: The transition state model matrix where probability of staying in good and bad state is 0.99 and 0.85 respectively. Suppose transition state model matrix is $mtransmx$.

$$mtransmx = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix}$$

Where, $p_{00} = 0.99$

$p_{01} = 0.01$

$p_{10} = 0.15$

$p_{11} = 0.85$

- transunit: One of [pkt|byte|time]. In our simulation we have used byte.
- sttype: Type of state transitions to use, either time or pkt. We simulate bursty packet loss behavior where error rates actually varies with time.
- nstates: Number of states. Here we use only two states (good and bad).
- start: The initial state.

To add an error model over wireless networks, each node can insert a given statistical error model either over outgoing or incoming wireless channels. For the outgoing link, the error module would be pointed by downtarget_ of the mac module as shown in Figure 4.3, while for the incoming link it would be linked by uptarget_ pointer of the mentioned netif module. And in each case the target_ of the error module points to either the netif or the mac respectively.

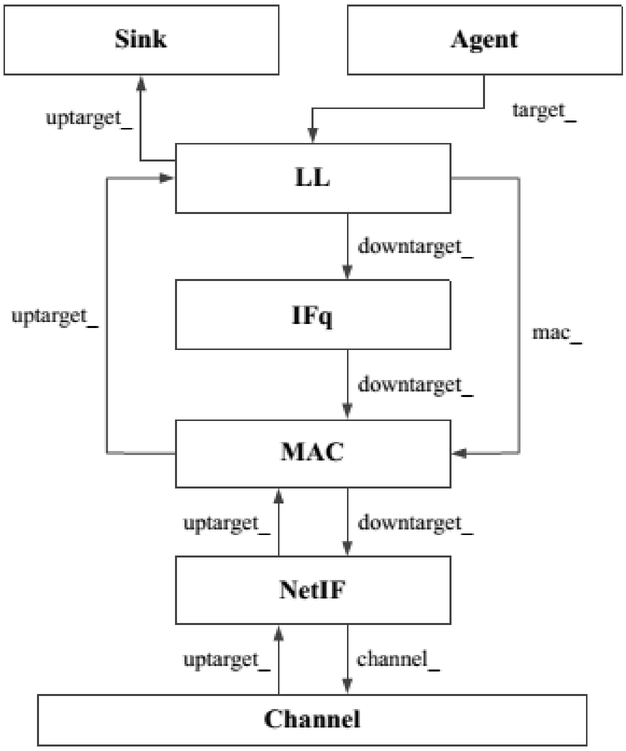


Figure 4.3: Structure of Mobile Node in NS2

The difference of placing over the two different locations is that the outgoing causes all the receivers to receive the packet suffering the same degree of error. Here error is determined before wireless channel module copies the packet. On the other hand, the incoming error module lets each receiver get the packet corrupted with different degree of error since the error is independently computed in each error module.

The insertion into the wireless protocol stack can be done by calling node-config command with the two options IncomingErrrProc and OutgoingErrProc. We can use these two options at the same time or each one separately.

In our simulation, we have inserted error model in both IncomingErrrProc and OutgoingErrProc.

4.1.3 Random Packet Generation

We have done our simulations considering uniform packet distribution for random packet generation. To add another variant beta packet size distribution has also been used in some experiments. We can generate packets with sizes ranged from a Minimum and Maximum value using Random variable class available in OTcl.

Uniform Distribution Uniform distribution is a family of symmetric probability distributions such that for each member of the family, all intervals of the same length on the distributions support are equally probable. The support is defined by two parameters a and b, which are its minimum and maximum value. We generate random packets from uniform distribution with 128 byte as minimum and 1024 byte as maximum value in our simulation.

Beta Distribution Beta distribution is a family of continuous probability distributions defined on the interval [0, 1] parametrized by two positive shape parameters, denoted by α and β that appear as exponents of the random variable. They control the shape of the distribution.

The random variable X is said to have a beta distribution with parameters n, m if its density is given by

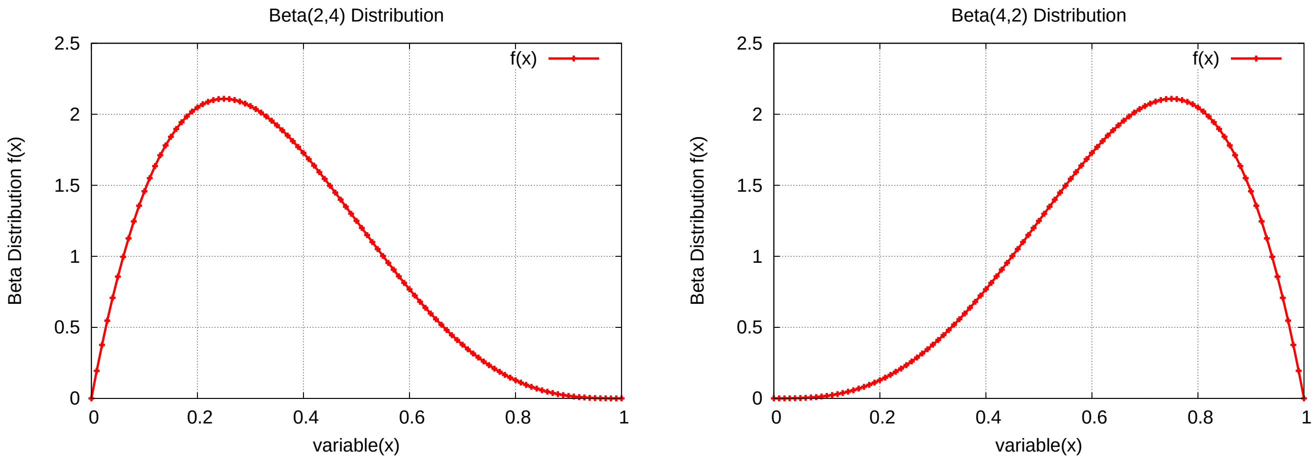
$f\left(x\right)=\frac{(n+m-1)!}{(n-1)!(m-1)!}x^{n-1}\left(1-x\right)^{m-1}$,$0<x<1$

Let U_1, ...,$U_{(n+m-1)}$ be independent uniform (0,1) random variables and consider the nth smallest value of this set U_n. Now U_n will equal x if , of the $n+m-1$ variables,

- $n-1$are smaller than x
- one equals x

- $m - 1$are greater than x

Beta (2, 4) and Beta(4, 2) distribution has been depicted in Figure 4.4. To simulate some variants of our proposed algorithm, we use Beta (2, 4) distribution to generate random packets within 128-1024 bytes range.



(a) Beta (2,4) distribution (b) Beta (4,2) distribution

Figure 4.4: Beta Distribution

4.2 Simulation Technique

At the beginning, we have applied the scenario for legacy DCF. As legacy DCF is already included in NS2, the initial function for wireless LAN will remain same. After that, we re-apply the same scenario for modified one. In order to implement modification in NS2, we have done some code changes in MAC 802.11 and associated timers. Modified codes have been added in appendix.

After completion of simulation, NS2 produces an output file called the trace file. In this case, each row in the trace file presents detailed information about all unique packets generated during simulation time.

We develop some simple Awk scripts to calculate required performance metrics. This script investigates the trace file line by line. For every CBR packet, our script checks whether it is sent or received packet, calculates duration of each packet and then the relevant counter has been updated. A simple equation produces the packet delivery ratio, throughput and average E2E delay (As defined in section 4.2) with this collected data.

After running the simple scenario with the two different MAC protocols, we obtain two unique trace files for basic and modified one. With the help of performance parameters we analyze the results and deduce some statement about performance improvement.

4.2.1 Scenario Generation for Random Topology

In order to carry out meaningful study of different networking issues like effect of network dynamics, different parameters etc, it is necessary to carry simulations on the right kind of scenario. Various scenarios can be used to illustrate interesting network performances. We use NS2 scenario generator to create different random scenarios for simulation. We have taken area of 500m X 500m.

The command in NS-2 is Setdest to generate the scenario.

setdest -v <version_no> -n <no of nodes> -s <speed type> -m <minimum speed> -M <Maximum speed> -t <simulation time> -P <pause type> -p <pause time> -x <Max x dimension of space> -y < Max y dimension of space> >output file name .

Here,

Version No = 2 (For original 1999 CMU version use 1 and for modified 2003 U.Michigan version use 2)

Maxx = 500

Maxy = 500

Minimum speed= 1 (>0)

Maximum speed= 10.0

Speed type=1 (For Uniform speed from min to max use 1 and for normal speed clipped from min to max use 2)

Pause Type= 1 (For Constant pause use 1 and for uniform pause use 2)

Number of nodes=10, 20, 30, 40, 50 or 60 depending on experiment

Simulation time =300.1 s

Pausetime = 310 s (Actually we take a pause-time greater than simulation time to get a static scenario)

Setdest command will generate a 500m X 500m topology with n number of nodes randomly distributed in static environment.

4.2.2 Traffic Generation for Random Topology

The traffic generator used in the simulation is a CBR (Constant Bit Rate) traffic generator, which generates traffic randomly by picking up random node pairs as source and destination.

Random traffic connections of CBR can be setup between mobile nodes using a traffic scenario generator script. In order to create a traffic-connection file, we need to define the type of traffic connection (CBR or TCP), the number of nodes and maximum number of connections to be setup between them, a random seed and in case of CBR connections, a rate whose inverse value is used to compute the interval time between the CBR packets.

Application/Traffic/CBR CBR objects generate packets at a constant bit rate.

cbr start : Causes the source to start generating packets.

cbr stop : Causes the source to stop generating packets.

Configuration parameters are:

PacketSize_: constant size of packets generated.

rate_: sending rate.

interval_: (optional) interval between packets.

random_: whether or not to introduce random noise in the scheduled departure times. default is off.

maxpkts_: maximum number of packets to send.

So the command line looks like the following:

ns cbrgen.tcl [type cbr/tcp][nn nodes][seed seed][mc connections][rate rate][mt max time] >output.tcl

with,

type = cbr

nodes = 10, 20, 30, 40, 50, 60

seed = 1, 3, 5, 7, 10

mc connection = 100

rate = 0.1 or 2

cbrgen.tcl creates traffic having provided rate with n nodes.

We also consider packets with different size from different nodes. We use uniform distribution to generate random packets for different nodes.

4.3 Performance Metrics

The performance metrics we consider in our simulation through trace analyzer are as follows,

- Throughput
- Packet Delivery Fraction (PDF)
- Average E2E Delay

Throughput Throughput can be defined as time average of the number of bits that can be transmitted by each node of the network to its destination.

Packet Delivery Ratio The delivery ratio is the number of received packets divided to the number of generated packets.

Packet Delivery Ratio= $No\ of\ Received\ packets/No\ of\ Generated\ packets$

End to End Delay End-to-end delay refers to the time taken for a packet to be transmitted across a network from source to destination.

$D = |T_d - T_s|$

Where, T_d = Packet Receive time at the destination

T_s =Packet Send time at source Node

Average delay is the summation of the delay of all packets divided by the number of generated packets.

$AverageE2EDelay = \frac{\sum_{n=0}^{recv_{packets}} Delay[n]}{recv_{Packets}}$

We compare the parameters evaluated at error models with different error rates for both original IEEE 802.11 MAC and modified one.

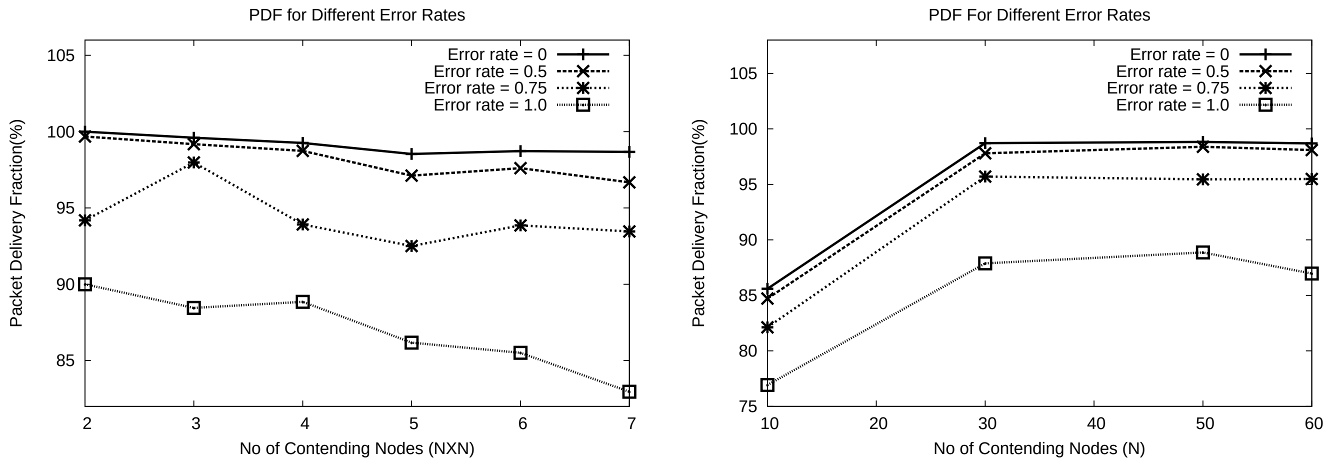
4.4 Performance Evaluation

4.4.1 Impact of Error Prone channel on IEEE 802.11 MAC

We have devised impact of error prone channel on basic IEEE 802.11 MAC protocol for below mentioned performance metrics. Packet error variability causes the variability of IEEE 802.11 protocol performance.

4.4.1.1 Packet Delivery Fraction

With the increase of Error rate in "Bad" state of Two State Markov Error Model, Packet Delivery Fraction(PDF) decreases in original MAC 802.11 MAC Protocol. We can observe the fact for small no of nodes (2X2, 3X3) to high no of nodes (5X5, 6X6, 7X7) with uniform packet distribution in Figure 4.5a and Figure 4.5b. We have done simulation for both grid and random topology and PDF decreases more for error rate 1.0 in comparison with other lower error rates for both scenario.

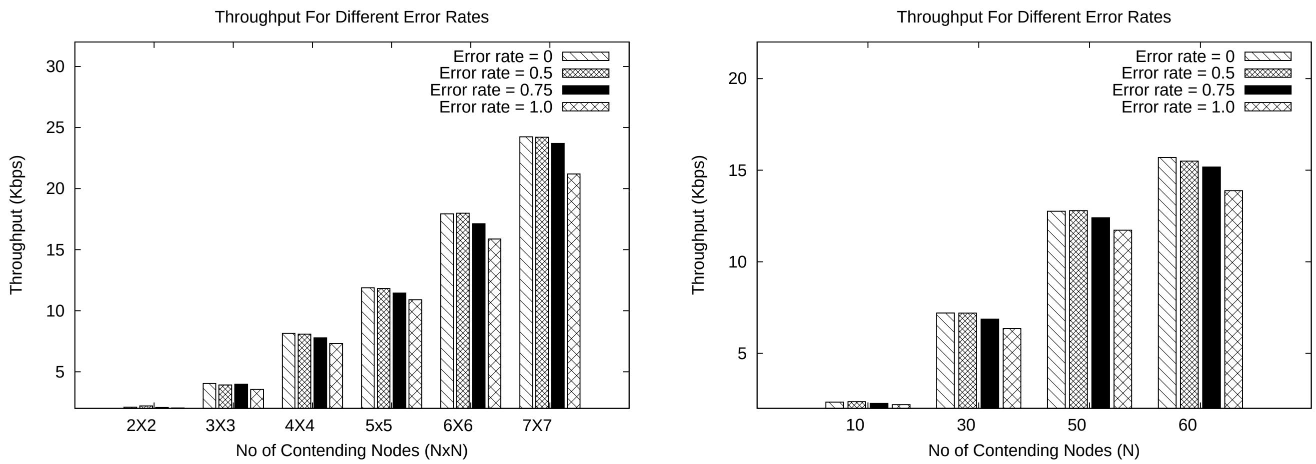


(a) Grid Topology (b) Random Scenario

Figure 4.5: PDF Variation in IEEE 802.11 MAC for Different Error Rates

4.4.1.2 Throughput

Throughput decreases as a result of increasing error rate in bad state of two state markov error model. This is depicted in Figure 4.6a and Figure 4.6b for both random and grid topology. Throughput variation is also more visible for error rate 1.0.

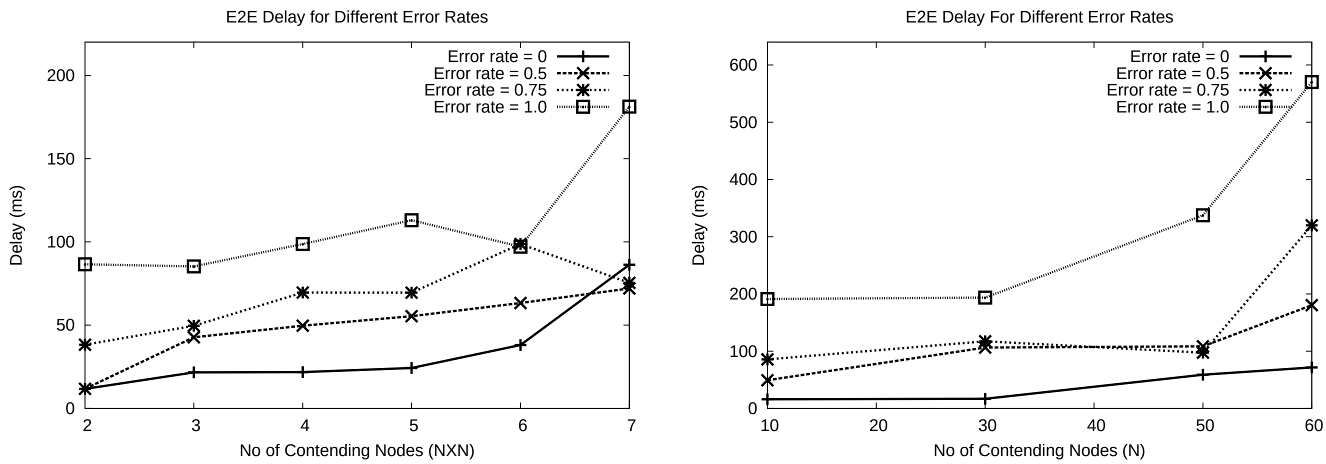


(a) Grid Topology (b) Random Scenario

Figure 4.6: Throughput Variation in IEEE 802.11 MAC for Different Error Rates

4.4.1.3 Average E2E Delay

In most of the cases higher error rate causes more degradation in average E2E delay performance as depicted in Figure 4.7a and Figure 4.7b for random and grid scenario. Higher error causes more retransmissions, which will in turn causes more delay. We can also observe from figures that, E2E delay variation is high for different error rates.



(a) Grid Topology (b) Random Scenario

Figure 4.7: E2E Delay Variation in IEEE 802.11 MAC for Different Error Rates

From the above simulations, we can find that error prone channel has a significant impact in performance metrics of wireless ad hoc network. Impact of error prone channel is more on average E2E delay performance variation rather than PDF and throughput.

4.4.1.4 Impact of Packet Size On IEEE 802.11 MAC Protocol

Under non-ideal channel, a trade off exists between the need to reduce the overhead by adopting larger packet size and the need to reduce packet error by using smaller packet size [3].

A shorter packet is preferred for more error prone channel.

From Figure 4.8, we can observe that with any number of contending nodes for fixed error rate 0.9, uniform packet size distributions with Maximum packet size 600 bytes gives better delay performance in most of the cases. As error rate depends on packet size, smaller packet size is beneficial in most of the cases to minimize the impact of error prone channel. So it can be depicted that packet size has a significant impact on error prone channel.

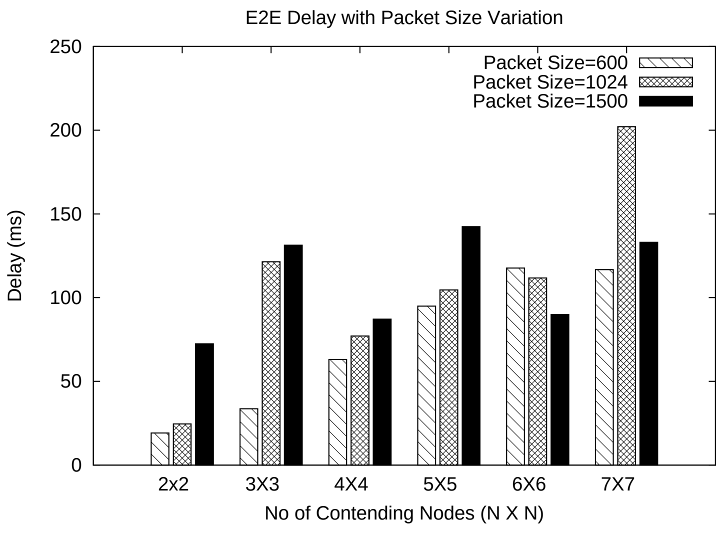


Figure 4.8: E2E delay Variation for Different Max Packet Size

4.4.1.5 Impact of CWMin on 802.11 MAC Protocol

We are not using RTS/CTS scheme here, so there should be an impact of selection of CWMin. It is because shorter control packets RTS and CTS reduce the collision cost, so impact of collision as well as CW is less.

But we can not devise any optimal CWMin for error prone channel. Worse channel condition causes more extra energy consumption in retransmissions and overhearing. CWMin mainly affects the intensity of collision. So in our basic scheme for large number of contending nodes probability of collision increases and increased CWMin causes better performance is some cases. But only setting CWMin to a fixed value is not suffice for error prone channel with different error rates.

From Figure 4.9 we can see that for small no of nodes like 2X2 CWMin=31 gives better E2E delay and for higher no of contending nodes like 7X7 CWMin=127 gives lower E2E for a specific packet distribution for error rate 0.9. But no specific relation can be devised for CWMin with fixed error rate.

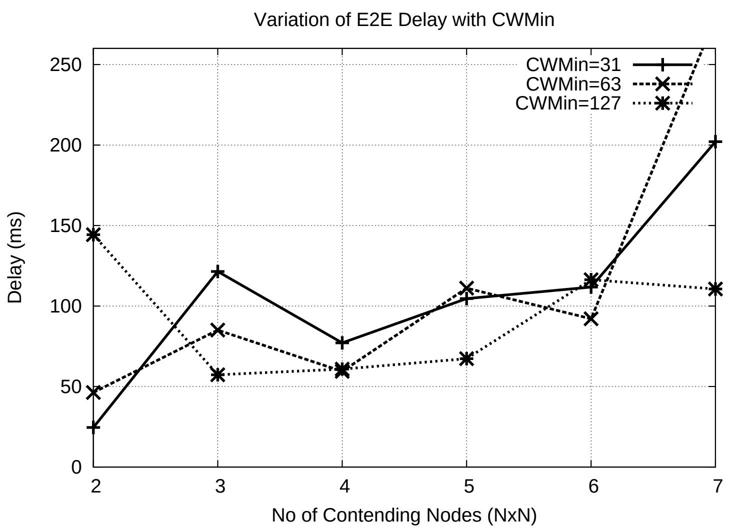


Figure 4.9: E2E delay Variation for CWMin

From the above simulations, we can conclude with the below mentioned facts:

1. Packet error does affect the throughput and delay. The throughput decreases and delay increases with errors.

2. Longer packets are preferred by better channels and shorter packets by error prone channels. Packet size has a significant impact on error prone channel.

3. For small number of contending nodes, default CWMin provides better result whereas for larger no of contending nodes higher CWMin will better. But no specific relation between CWMin and error rate can be devised and we don't find any optimal CWMin for error prone environment.

4. So if we combine CWMin and packet size, that is selection of CW based on dynamic packet size distribution can provide better result for any error rate.

In our modified code we have modified CW based on learned packet size distribution in the network. All the other schemes of basic IEEE 802.11 MAC protocol will remain same. In next section we will compare obtained results from modified code with the original one.

To modify basic algorithm, we add below mentioned parameters in simulation settings described in table 4.3.

Table 4.3: Additional Simulation parameters for modified algorithm.

Parameter	**Value**
Simulation Time	300s
Window Timer	30 s
CWMin	127 (In learning Phase)
RTSThreshold	3000 (Disabled)
Error Model	Two state Markov Model
Error Rate	0.25, 0.5, 0.75, 0.9 and 1.0
No of CW sub-range	4
No of Bins	4
PacketLabeling*	
Low	0 to <25%
Medium	25% to <50%
High	50% to <75%
Remaining packets	>75%

*** A value such that all packets ranging from x1% to x2% falls in this label

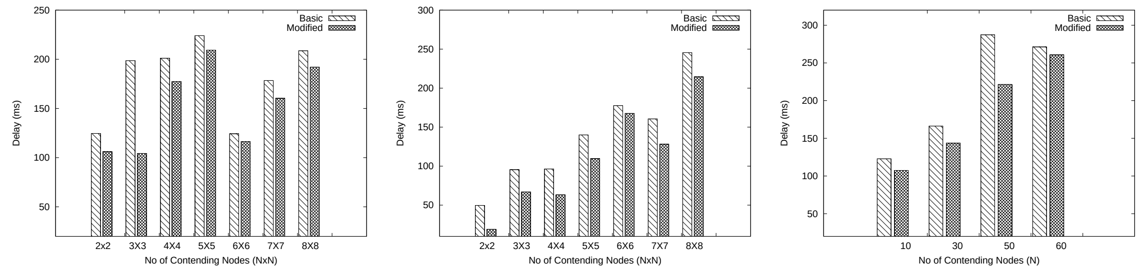
Some observations have been done with simulation time 200s with 30s window timer. But for further simulations to evaluate performance of modified algorithm,it is extended up to 300.1s with 30s window timer so that after learning phase at least 9 iterations can be done. And it gives even better result.

4.4.2 Performance Evaluation of Modified IEEE 802.11 MAC Protocol

We have done extensive simulations to find the performance of modified MAC protocol in comparison with original IEEE 802.11 MAC protocol. In our simulations we vary number of nodes, CBR rate, network topology to observe the behavior of some performance metrics like *E2E delay*, *Throughput* and *Packet Delivery Fraction(PDF)*. To limit the illustrations, we depict only some particular simulation based scenarios, where subtle changes in parameters result in salient differences in the behavior of our performance metrics.

4.4.2.1 E2E Delay Variation

From Figure 4.10a , 4.10b and 4.10c, we observe that in almost all cases average E2E delay decreases in modified code in comparison with original one. We have done this simulation for fixed error rate 0.9 in bad state of two state Markov Model. Average E2E delay performance improves for a range of contending nodes in random and grid topology with different network loads in our modified scheme.



(a) Grid Topology (Heavy Load) (b) Grid Topology (Light Load) (c) Random Scenario

Figure 4.10: E2E Delay Variation with Original IEEE 802.11 MAC for Different Topology

We have shown simulation results for 8 bins with each bin containing almost equal to 12.5% of packets and CW range is also equally partitioned to 8 sub-ranges. Figure 4.11 illustrates the scenario for grid topology.

From the above experiment, we can say that our strategy "*Packets belonging to a certain bin take a CW subrange in proportion to the number of packets reside in that bin*" of uniform binning and CW sub-range scheme decrease average E2E delay in most of the cases. If we divide the packets into more bins and CW into more sub-ranges, computation complexity and protocol overhead will increase, which in turn decrease performance for small no of nodes as shown from Figure 4.11. But for large no of nodes, this will work even better than 4 bin scenario of our proposed scheme.

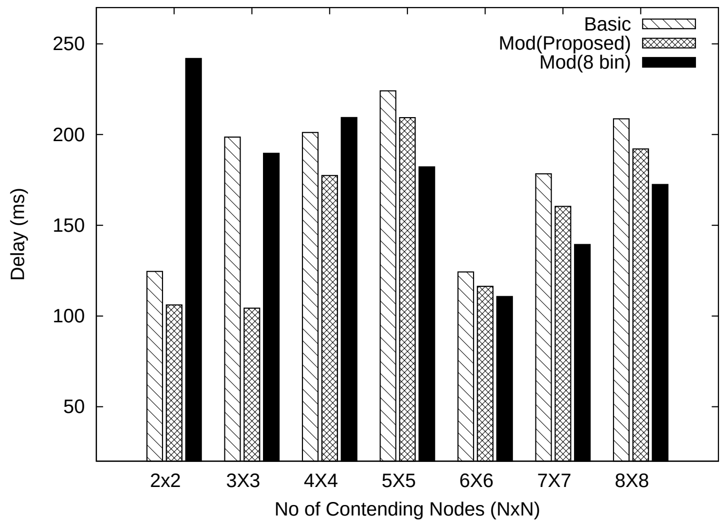
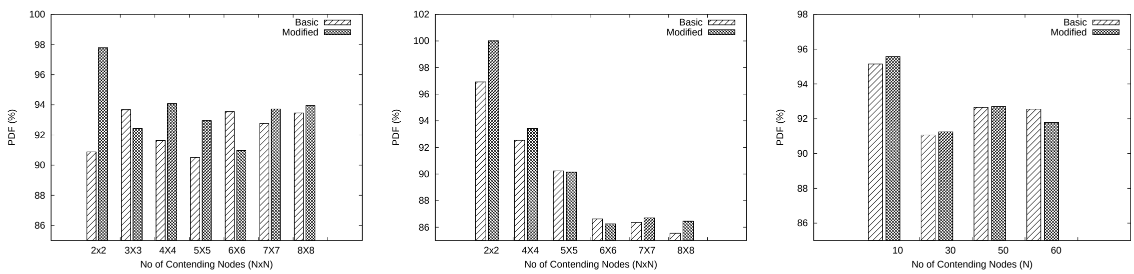


Figure 4.11: E2E delay Variation for Selection of Different No of Bins

4.4.2.2 Packet delivery Fraction (PDF) Variation

PDF remains almost same as original one, which is depicted in Figure 4.12a, 4.12b and 4.12b for both grid and random scenario. We are getting almost 0-4% variation in PDF value between modified and original one.



(a) Grid Topology(Heavy Load) (b) Grid Topology(Light Load) (c) Random Scenario

Figure 4.12: PDF Variation for Different Topology

Selection of 8 bins with 8 equal CW sub-range also gives almost same PDF value as depicted in Figure 4.13.

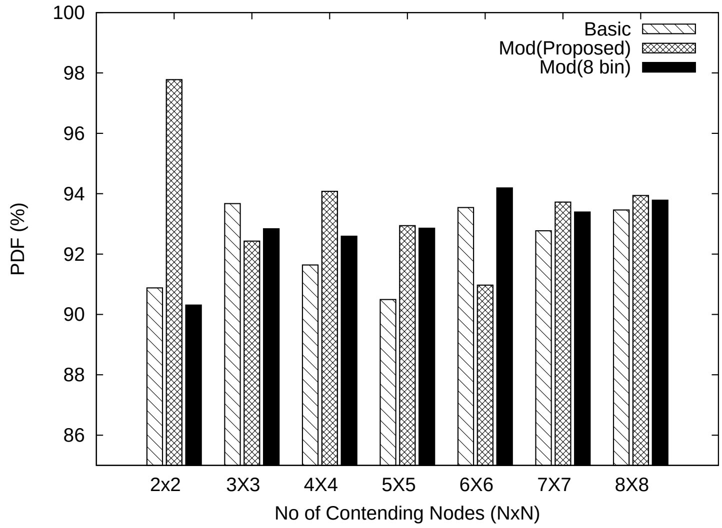
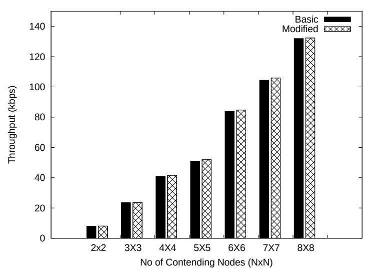


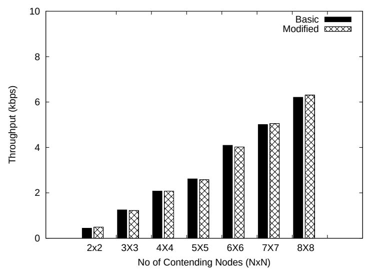
Figure 4.13: PDF Variation for Selection of Different No of Bins

4.4.2.3 Throughput Variation

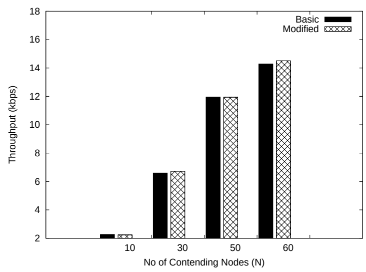
We get almost same throughput in comparison with original one as depicted in Figure 4.14a, 4.14b and 4.14b. Simulation has been done for both grid and random scenario with varied network load.



(a) Grid Topology (Heavy Load)



(b) Grid Topology (Light Load)



(c) Random Scenario

Figure 4.14: Throughput Variation for Different Topology

Throughput also remains same for selection of 8 bins with 8 equal CW sub-range, which is shown in Figure 4.15.

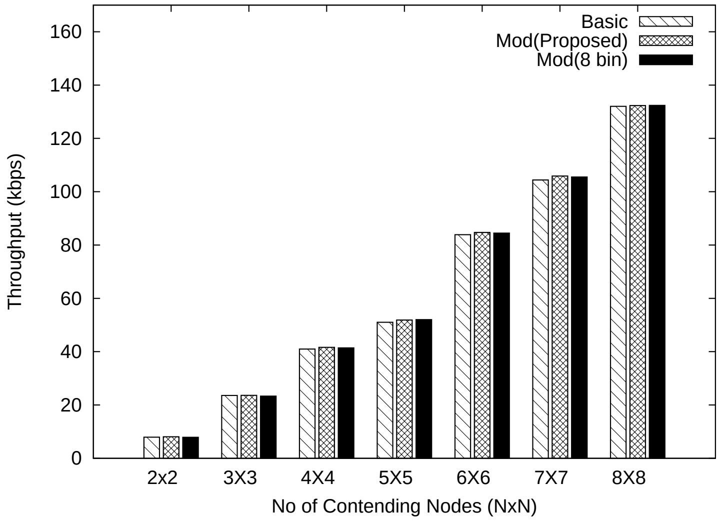


Figure 4.15: Throughput Variation for Selection of Different No of Bins

4.4.2.4 Retransmission Count

From our previous observations we can find that, average E2E delay improves as an outcome of code modification and PDF (Packet delivery Fraction)/Throughput remains almost same as the basic one. This is possible only if no of retransmission count decreases in modified code. From $ACKFailureCount$ of RetransmitData () function this count have been calculated.

For a 4X4 grid simulation, No of Total Retransmission count is 480 in original code where as it is 400 in modified one. 16.67% less retransmission count in modified code also improves energy efficiency and average E2E delay. That's why E2E delay also found better for modified one.

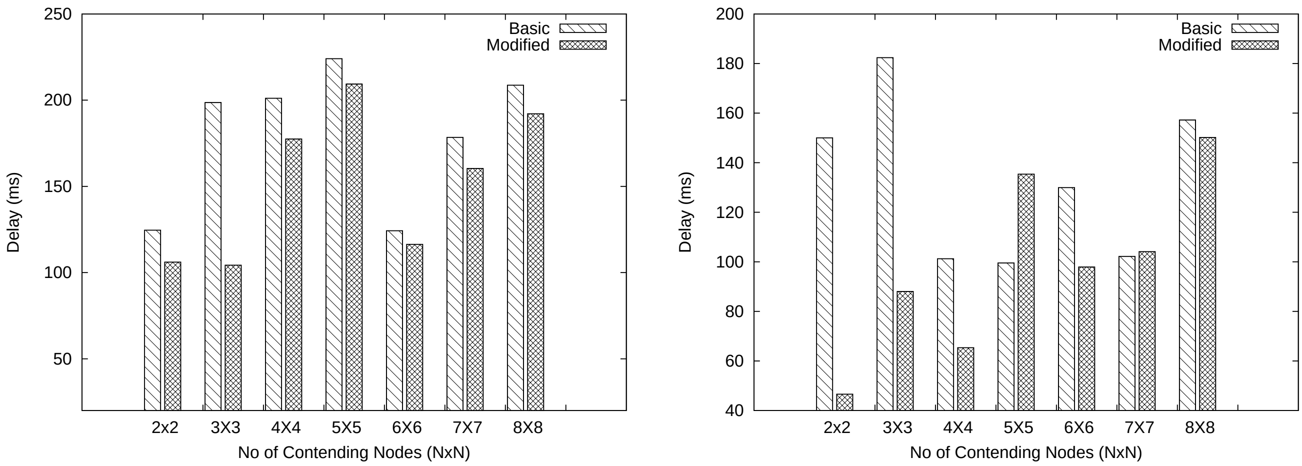
4.4.3 Impact of Varied Schemes in Modified Algorithm

We vary various parameters to validate the selection of introduced parameters in our modified scheme. No of defined classes/bins, packet labeling, class boundary and packet distribution scheme have been changed to observe the behavior of some performance metrics like *E2E delay*, *Throughput* and *Packet Delivery Fraction(PDF)*. From previous simulations, we can find that impact of error prone channel found mostly in average E2E delay parameter. So we simulate further variation of E2E delay with some variants of modified algorithm. For simplicity, we have done these simulations using grid topology with fixed error rate 0.9 for 6X6 contending nodes in a highly loaded network to observe the performance impact on average E2E delay.

4.4.3.1 Selection of Packet Distribution Scheme

We generate random packets in our simulation. In all of our previous simulations we have used uniform distribution to generate these random packets. To insert skewness in packet generation, we use beta distribution to observe the impact of packet distribution scheme in our modified algorithm.

We can observe experimental results from Figure 4.16a and 4.16b for beta and uniform packet distribution, where minimum and maximum packet size ranging from 128-1024 Bytes. From the result we can deduce that our modified algorithm performs better for both type of packet distribution in most of the cases.



(a) Uniform Distribution

(b) Beta Distribution

Figure 4.16: E2E Delay Variation for different Packet Distribution Schemes

4.4.3.2 Random Agnostic Binning

In our proposed scheme, we make selection of CW dependent on packet size distribution. But in this variant, we consider random class assignment instead of packet size based selection to find the effect of packet size. In this simulation, we assign packet labels from random selection over 4 classes/bins. After that, we set CW as per our proposed scheme. Figure 4.17 depicts the scenario with 6X6 grid and error rate 0.9 in bad state of two state markov model.

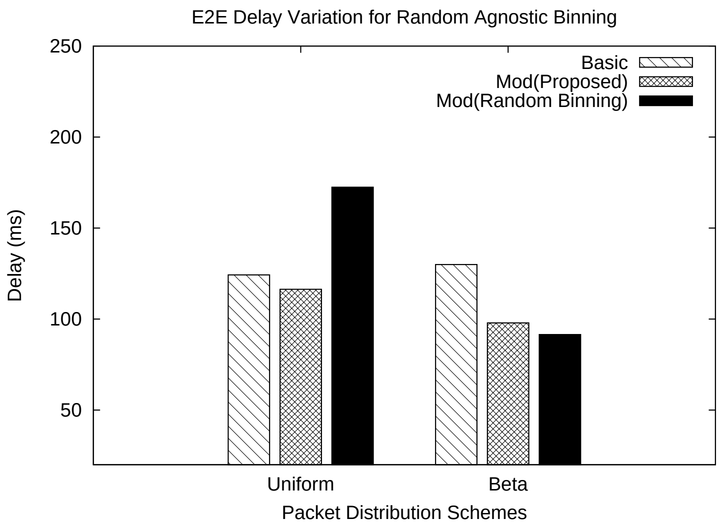


Figure 4.17: E2E delay Variation for Random Agnostic Binning

From the experiment we can say that random class assignment scheme performs worse than our proposed packet distribution based CW selection scheme.

4.4.3.3 Arbitrary Boundary Setting for Bins

In our modified algorithm, we partition packets into a couple of bins based on packet labeling and assign a separate contention window sub-range for packet belonging to a certain bin. We assume equal sized bins for 4 classes. But in this experiment, we assign arbitrary bin boundary instead of equal 25%. Figure 4.18 depicts the scenario for 6X6 contending nodes.

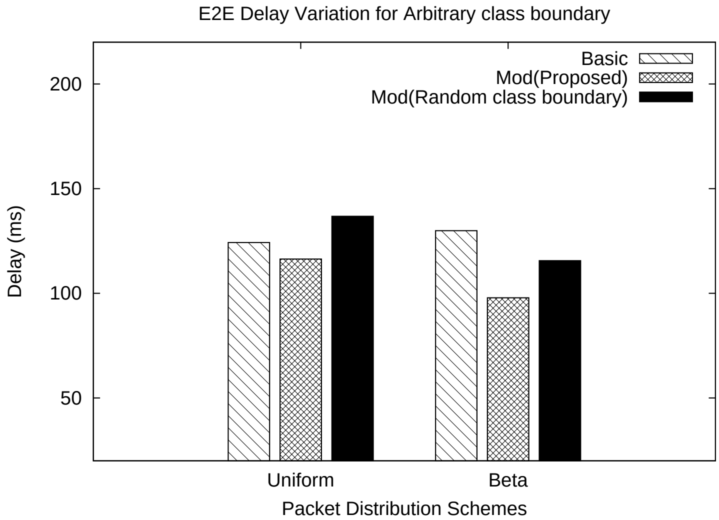


Figure 4.18: E2E delay Variation with Arbitrary boundary setting for bins

From the simulation we can find that our proposed scheme performs better than arbitrary boundary setting scheme for bins.

4.4.3.4 Reverse CW Assignment Scheme

In our modified protocol, we assign smaller CW to bin 1 (smaller packets) and so on. In simulations shown in Figure 4.19 we reverse the assignment that is assign smaller CW to larger packets and so on.

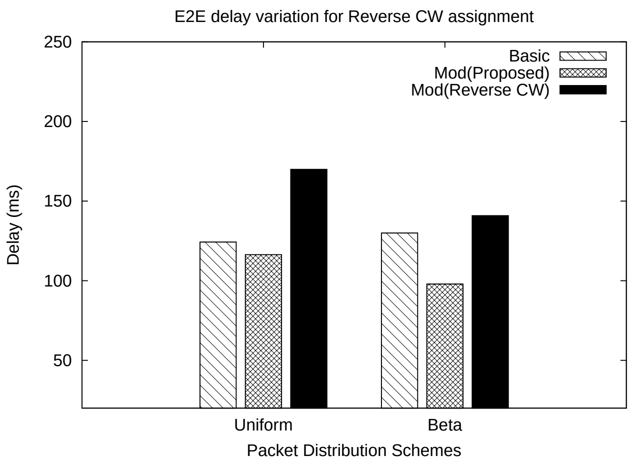


Figure 4.19: E2E Delay Variation for Reverse CW Assignment

From the simulation result we can observe that our modified scheme performs better than this reverse CW assignment scheme.

4.4.3.5 Selection of Different Number of Bins/Classes

In our modified algorithm, we use a simple strategy: *Packets belonging to a certain bin take a CW sub-range in proportion to the number of packets reside in that bin.* Using this concept, we partition packets into 4/8 bins based on packet labeling and assign a separate contention window sub-range for packets belonging to a certain bin. To follow this, first 25% packets belong to bin1, next 25% to bin2 and so on for 4 bins and first 12.5% packets belong to bin1, next 12.5% to bin2 and so on for 8 bins. In this variant, we choose different number of bins with non-equal number of packets in each bin associated with non-equal sub-range. The impact of this choice on average E2E delay has been depicted in Figure.4.20 in comparison with choice of 3, 5 and 6 bins.

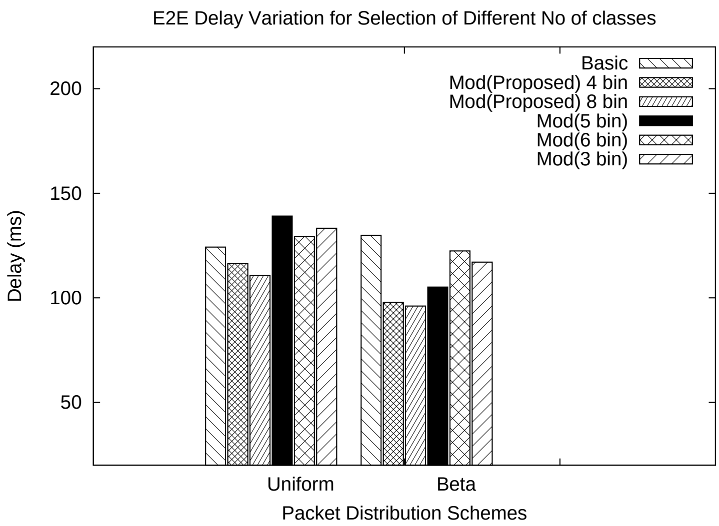


Figure 4.20: E2E delay Variation with Different No of classes

From the experimental result we can observe that in most of the cases our proposed analogy with 4 and 8 equal sized bins will give better result than selecting 3, 5 or 6 non-equal sized bins for 6X6 contending nodes.

4.4.3.6 Learning Time Adjustment of Window Timer

Selection of window timer or interval is important. The length of this interval must not be too short to avoid possible oscillation and not be too long to decrease the reactivity of the algorithm. From below mentioned Figure 4.21, we can find that different window timer value causes different E2E delay with all other settings remain the same.

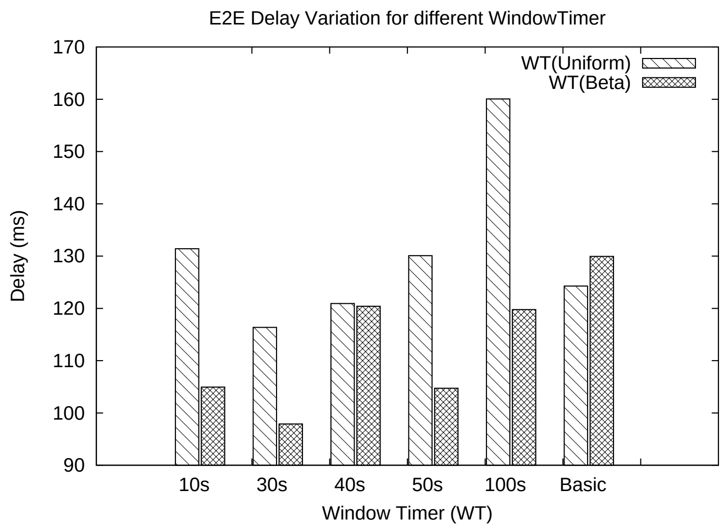


Figure 4.21: E2E Delay Variation for different Window Timer Interval

Here, WT=30s gives optimum result for 300s simulation time and with 6X6 contending nodes. For both beta and uniform packet distribution, WT=30s which is used in our modified algorithm gives lower E2E delay than the others. Actually setting of window timer has a relation with total simulation time. We have to set Window timer with such a value that a considerable number of iterations can be done within simulation time. In our case, after learning phase of window timer 9 iterations has been performed by window timer.

4.5 Summary

In this chapter, we produce the simulation results in support of performance of our modified algorithm. We analyze various performance metrics against performance of original

IEEE 802.11 Mac protocol. Following summary table 4.4 shows the comparison between modified and original one

Table 4.4: Summary of comparison between Modified and Basic MAC.

Parameters	Modified MAC	Basic MAC
Throughput	Almost same	Varies with error rate
Packet Delivery Fraction	Almost 0-4% variation	Varies with error rate
Average E2E Delay	Decreases and Upto 47% improvement in delay	Worse
No of Dropped Packets	Decreases	Worse
Retransmission count	16.67% less retransmission	Worse

Chapter 5

Conclusion

In this last chapter, we draw the conclusion of our thesis by describing major contributions made by research works associated with the thesis followed by some directions for future research over the issue

5.1 Major Contributions

The contributions that have been made in this thesis can be enumerated as follows:

1. In this thesis, we focus on performance issues of wireless error prone network. To the best of our knowledge, this topic has been examined so far, but impact of packet errors on collision probability and packet sending probability has been ignored in previous works. Some previous error models have been considered only for single hop static network.

2. We have considered packet error as an important source of performance degradation along with collision in error prone environment. We observe that, impact of packet error on performance metrics of IEEE 802.11 protocol is different from that of collisions. So the scheme of scaling and randomization of CW does not help when the channel is inherently noisy. As packet error rate depends on BER and packet length, packet size have a greater impact than optimal CW on error prone channel.

3. We design a dynamic CW setting scheme based on packet size distribution in the error prone network. We keep main principle of the MAC Protocol of IEEE 802.11 DCF unchanged. Contrary to BEB algorithm, our protocol dynamically selects contention window sub-range instead of whole selection range based on packet size distribution. Our scheme can accommodate packets with less retransmission, which causes less delay.

4. We formulate a theoretical study of our approach and produce a formal verification of the underlying uniform binning concept in the architecture.

5. We not only develop the modified protocol, but also simulate the protocol using NS2 simulator to make closer observations. We make a rigorous simulation based study of various performance issues of the proposed approach, and analyze the simulation output against the original IEEE 802.11 MAC DCF protocol. The performance of the scheme have been analyzed in terms of average E2E delay, packet delivery fraction (PDF) and throughput and our scheme performs better in most of the cases. Evaluated results portray that our scheme achieves less retransmissions, which causes reduced average E2E delay.

6. And at last, we make some alternative approaches to our modified scheme. We vary packet distribution technique to insert skewness, consider random assignment of packets into bins, and reverse the CW assignment pattern, make boundary setting of each bin arbitrary, change no of bins and select different interval for our window timer. But in most of the cases, our proposed modification outperforms all the variants.

5.2 Future Directions of Further research

Any research on any topic always paves the path of future analysis. Same goes for our one. More research can be possible on this topic. Based on our proposed scheme and results of simulations presented here, we intend to extend our work in future in following directions:

1. We have done most of our analysis and experiments using 4 bins with proportionate CW sub-range. Now we want to extend our analysis for n number of bins and find optimal number of bins that can produce reduced expected delay for small no of contending nodes as well as larger ones.

2. Simulations can be done in more different scenarios other than used random and grid topology.

3. We want to tune used parameters further more to get more better performance in terms of average E2E delay, throughput and packet delivery fraction(PDF) for error prone network.

Bibliography

[1] Razafindralamb, T., Lassous, I. G., "SBA: A Simple Back-off Algorithm for Wireless Ad Hoc Networks", *8th International IFIP-TC 6 Networking Conference*, 2009.

[2] Yin, J., Wang, X., Agrawal, D.P., "Optimal Packet Size in Error-Prone Channel for IEEE 802.11 Distributed Coordination Function", *Wireless Communications and Networking Conference, WCNC, IEEE*, Vol. 3, pp. 1654-1659, 2004.

[3] Wang, X., Yin, J., Agrawal, D.P., "Effects of Contention Window and Packet Size on the Energy Efficiency of Wireless Local Area Network", *Wireless Communications and Networking Conference, IEEE*, Vol. 1, pp. 94 - 99, 13-17, 2005.

[4] Ni, Q., Li, T., Turletti, T., Yang, X., "Saturation Throughput Analysis of Error-Prone 802.11 Wireless Networks", *Wireless Communications and Mobile Computing*, Volume. 5 (8), pp. 945 - 956, 2005.

[5] Balador, A., Movaghar, A., Jabbehdari, S., "History Based Contention Window Control in IEEE 802.11 MAC Protocol in Error Prone Channel", *Journal of Computer Science*, 6 (2), pp. 205-209, 2010.

[6] Chatzimisios, P.,VitsasA, V.,Boucouvalas, C., "On Improving Performance for IEEE 802.11 Wireless Lans under Congested and Error-Prone Environments", *Internet and Multimedia systems International Multi conference*, 24TH IASTED, 13-15, 2006.

[7] Gupta, P.,Kumar, P., "The Capacity of Wireless Networks", *IEEE Transactions on Information Theory*, 46(2):388-402, 2000.

[8] LAI HON, C., “Efficient Coding/Decoding Strategies for Channel with Memory”, *Diploma thesis*, University of British Columbia, Vancouver, Canada, November,1992.

[9] SANNECK, H. ,CARLE, G. ,KOODLI, R., "A framework model for packet loss metrics based on loss runlengths", *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference (MMCN)*, San Jose (California, USA), pp.177â187, 2000.

[10] Bharghavan, V., Demers, A., Shenker, S., Zhang, L., “MACAW: a media access protocol for wireless LAN’s. ”, *ACM SIGCOMM Computer Communication Review*, London, United Kingdom, pp. 212–225, 1994.

[11] Chatzimisios, P., Boucouvalas, A.C., Vitsas, V., Vafiadis, A., Oikonomidis, A.,Huang, P., “ A simple and effective backoff scheme for the IEEE 802.11 MAC protocol”, *Cybernetics and Information Technologies*, Systems and Applications(CITSA), Orlando, Florida, USA, 2005.

[12] Fang, Z., Bensaou, B., Wang, Y., “Performance evaluation of a fair backoff algorithm for IEEE 802.11 DFWMAC”, *ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Lausanne, Switzerland, pp. 48–57 (2002)

[13] Nandagopal, T., Kim, T., Gao, X., Bharghavan, V., "Achieving MAC Layer Fairness in Wireless Packet Networks", *ACM Conference on Mobile Computing and Networking (MOBICOM)*, Boston, Massachusetts, United States, pp. 87–98 (2000)

[14] Luo, H., Lu, S., Bharghavan, V., "A new model for packet scheduling in multihop wireless networks", *ACM Conference on Mobile Computing and Networking (MOBICOM)*, Boston, Massachusetts, United States, pp. 76–86,2000.

[15] Lettieri, P. and M. B. Srivastava, “Adaptive Frame Length Control for Improving Wireless Network Link Throughput, Range and Energy Efficiency”, *Proc. of the IEEE INFOCOM* , 1998..

[16] Chockalingam, A., Zorzi W. Xu, M., and Milstein, L.B., “Energy efficiency analysis of a multichannel wireless access protocol”, *Proc. of The Ninth IEEE International Symposium Personal*, Indoor and Mobile Radio Communications, Vol 3, 1998.

[17] Balador, A., Movaghar, A., Jabbehdari, S., Kanellopoulos, D., "Novel Contention Window Control Scheme for IEEE 802.11 WLANs", *IETE TECHNICAL REVIEW*, Vol. 29(3), pp. 202-212, 2012.

[18] Saraireh, M., AL-Saraireh, J. and Saraireh, S., "A Novel Adaptive Contention Window Scheme for IEEE 802.11 MAC Protocol", *Trends in Applied Sciences Research*, 9: 275-289, 2014.

[19] Ksentini, A., Nafaa, A., Gueroui,A., Naimi, A., "Deterministic Contention Window Algorithm for IEEE 802.11", *IEEE Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 4, pp. 2712-2716, 2005.

[20] Cali, F., Conti, M., Grergori, E., "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit", *IEEE/ACM Transactions on Mobile Computing*, Vol. 1(1), 2002.

[21] Kumar, S. Raghavan, V. Deng, "Medium Access Control protocols for ad hoc wireless networks: A survey", *Ad Hoc Networks* 4, pp. 326-358,2006

[22] Dong, X. J., and P. Varaiya, "Saturation throughput analysis of IEEE802.11 wireless LANs for a lossy channel," IEEE Communications Letters , vol. 9, no. 2, pp. 100–102, Feb. 2005.

[23] ISSARIYAKUL, T.,HOSSAIN, E., "Introduction to Network Simulator NS-2", *Springer Science and Business Media*, LLC, 233 Spring Street, New York, USA, 2009.

[24] Song, W., Michael N. Krishnan, and Zakhor, A., "Adaptive Packetization for Error-Prone Transmission over 802.11 WLANs with Hidden Terminals", *Multimedia Signal Processing,MMSP,IEEE International Workshop*, pp 1-6, 2009.

Chapter 6

Appendix

6.1 What is NS-2

NS is an object-oriented, discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Later NS-2 (version 2) was developed at UC Berkeley in C++ and OTcl (Object-oriented extension of Tcl).

The core of the wireless module in NS-2 is mobile node, which was originally ported as CMU's (Carnegie Mellon University) Monarch group's mobility extension to NS [52]. It is a basic node object equipped with wireless functionalities, and the mobile node can move within the given topology, receive and send radio signals through wireless channel. The characteristics of mobility like node movement, periodic location update, topology maintenance, etc., are implemented by C++, while the internal network components (like classifier, Link Layer (LL), Media Access Control (MAC), Channel, etc.) are assembled using OTcl, which is an object oriented extension of the scripting language Tcl (Tool Command Language).

It consists of Agent/Sink, Link Layer, Interface Queue, MAC Layer, Network Interfaces and Channel. Each of the main components is briefly described here.

Agent

Agent represents endpoint where network layer packets are constructed, and is used in the implementation of protocols, such as TCP and UDP (User Datagram Protocol) protocols. Any node in NS-2 as a source of data flow needs to attach a specific Agent to generate packets.

Sink

Sink, as the opposite of Agent, is the consumer of the network layer packets. The node acts as areceiver needs to attach a specific Sink to receive packets.

LL (Link Layer)

Link Layer connects to a ARP (Address Resolution Protocol) module, which is used to resolve the MAC address using IP address. All the packets sent out by Agent will be transferred to Link Layer, and then Link Layer puts them into the IFq. For the packets received, MAC transfers them to Link Layer, and then Link Layer would pass them to its up target Sink.

IFq (Interface Queue)

There are two kinds of Interface Queue. When the routing protocol is DSR (Dynamic SourceRouting), a special queue called CMUPriQueue would be used, because different kinds of packets would be assigned different priorities using DSR. Otherwise, another queue called PriQueue, which pushes the packets at the head of the queue, will be used. The encoding and decoding operations.

MAC (Media Access Control)

MAC layer is implemented as DCF (Distributed Coordination Function) MAC protocol in IEEE 802.11. There is a Tap Agent (Tap Agent is application level process on NS-2 node that converts network packets between the simulated and the real network.) could be registered itself with theMAC object using method installTap(). As the requirements of the network coding, the Tap Agent will promiscuously snoop all the packets received by MAC layer, before address filtering is done, and pass them up to the specific layer IFq.

NetIF (Network Interfaces)

NetIF layer in NS-2 acts as the hardware interface used by mobile node to access the channel and simulates signal integrity, collision, transmission error, etc. The Radio Propagation Model (Radio Propagation Model exists in physical layer of each mobile node and is used to predict the received signal power of each packet.) has been integrated into NetIF layer. NetIF marks each transmitted packet with transmission power, wavelength, etc. and decides whether coming packet can be received by the mobile node with given distance, transmit power and wavelength. In addition to that, Omni Directional Antenna module, which has unity gain for all direction, has been implemented into this layer too.

Channel

Channel simulates the practical transmission of the packet at the physical layer. Contention mechanisms have been realized on Channel, which allows the MAC layer to carry out contention, carrier sense and collision detection. If more than one transmissions happens at the same time, Channel mark the collision flag. By checking this flag, MAC layer could be aware of this collision and handle it

6.2 Installation

- Download NS2 version 2.34
- Unzip the tar archive
- Open Terminal and Go to the ns2 root directory. Here ~/ns-allinone-2.34
- Run ./install script

6.3 Set Path

- After installation to set the path type the following contents

1. Export LD_LIBRARY_PATH=~ns-allinone-2.34/otcl-1.13
2. Export LD_LIBRARY_PATH=~ns-allinone-2.34/ns-2.31/lib
3. Export TCL_LIBRARY=~ns-allinone-2.34/tcl8.4.14/library

6.4 Scenario Generation

- This is done using the setdest command. There are two versions for setdest. Version 2 is most recently implemented and used.
- Version 2 signature of setdest command is $setdest - v < 2 > -n < nodes > -s < speedtype > -m < minspeed > -M < maxspeed > -t < simulationtime > -P < pausetype > -p < pausetime > -x < maxX > -y < maxY >$

1. -v version number; Here 2
2. -n number of nodes. Generated node number will be 0 to (n-1)
3. -s speed type (uniform, normal); s=1 uniform speed from min to max; s=2 normal speed clipped from min to max
4. -m minimum speed > 0
5. -M maximum speed

6. -P pause type (constant, uniform); P=1 constant pause; P=2 uniform pause [0, 2*p] o -p pause time (a median if uniform is chosen)

7. -x x dimension of space

8. -y y dimension of space

- After running the command a scenario will be generated. Pipe the scenario in file
- Example : setdest -v 2 -n 10 -m 10 -M 100 -t 20 -P 1 -p 10 -x 200 -y 400 > scen-exp1
- For Further Explanation: Go through the source code ~ns-allinone-2.34/ns-2.34/indep-utils/cmu-scen-gen/setdest/setdest.cc

6.5 CBR Traffic Generation

- This is done with the help of cbrgen.tcl file executing with ns command
- Open terminal in ~ns-allinone-2.34/ns-2.34/indep-utils/cmu-scen-gen
- ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate rate]

1. -type traffic type (tcp, udp/cbr)
2. -nn the highest node number(node number will be 0 to nn)
3. -seed seed for random variable generation which is used to create random number of source-destination pair
4. -mc maximum number of connections; i.e.; source-destination pair
5. -rate it is the inverse of the interval between packet transmission & should be <0

- After running the command a cbr traffic will be generated. Pipe the traffic in file
- Example : ns cbrgen.tcl -type cbr -nn 9 -seed 1 -mc 10 -rate .25 > cbr-exp1
- For Further Explanation: Go through the source code ~ns-allinone-2.34/ns-2.34/indep-utils/cmu-scen-gen/cbrgen.tcl

6.6 Writing Simulation Generation File

• A wireless simulation is made up of some number of MobileNodes. Each such mobile node needs some options to be configured like routing protocol, MAC layer protocol, antenna type, channel type etc. A list of all parameters is given below.

1. Channel type(Channel/WirelessChannel) ;# Set channel for wireless media
2. Propagation model(Propagation/TwoRayGround);
3. Interface type (Phy/WirelessPhy) ;
4. MAC layer protocol (Mac/802_11) ;# The name of MAC layer protocol
5. Routing protocol (AODV) ;# The name of routing protocol
6. Interface Queue type (CMUPriQueue - for DSR) ;
7. Interface Queue Length(50) ;# Queue length of a node for storing packets
8. Antenna type (Antenna/OmniAntenna) ;#
9. LL type (LL) ;# Link Layer type

- Set the required parameters in the tcl file for the specific scenario and traffic. These are the configuration parameters for the topology structure, like the dimensions of the grid, number of nodes present etc. A list of all of them is given below for reference. Example(For the previous scenario and traffic)

1. set val(x) 200 ;# X dimension of the topography
2. set val(y) 400 ;# Y dimension of the topography
3. set val(ifqlen) 50 ;# max packet in ifq
4. set val(seed) 0.0 ;
5. set val(adhocRouting) AODV ;# The routing protocol
6. set val(nn) 10 ;# how many nodes are simulated
7. set val(cp) "cbr-exp1" ;# the absolute or relative path of the traffic file
8. set val(sc) "scen-exp1" ;# the absolute or relative path of the scenario file

9. set val(stop) 200.0 ;# the simulation time

- Turn on the trace for the required layers
- Example

1. -agentTrace ON \
2. -routerTrace ON \
3. -macTrace ON \
4. -movementTrace OFF

- For generating the trace file in new format, remove the comment or vice versa

$ns_ use-newtrace

- Set the output trace file with absolute or relative path

Example: set tracefd [open exp1.tr w]

6.7 Trace Generation

- Paste the tcl file to the ~ns-allinone-2.34/ns-2.34/tcl/ex directory
- Open terminal in ~ns-allinone-2.34/ns-2.34/tcl/ex
- Execute commandns <tclscript> Example: ns sample.tcl
- If the simulation is successful, the desired trace file will be generated.

6.8 Trace Analysis

- The various traces begin with a single character or abbreviation that indicates the type of trace, followed by a fixed or variable trace format. Example of new wireless trace format is

- s -t 163.001503520 -Hs 0 -Hd -2 -Ni 0 -Nx 300.00 -Ny 500.00 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw — -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 2.0 -It cbr -Il 200 -If 1 -Ii 77 -Iv 32 -Pn cbr -Pi 32 -Pf 0 -Po 0 ;
- Illustration of various fields :
- Field 0: Event type:

Event type (as in the older trace format) In the traces above, describes the type of event taking place at the node and can be one of the four types

s	send
d	drop
r	receive
f	forward

- Field 1: General Tag :

The second field starting with "-t" may stand for time o -t : time

- Field 3: Next hop info

1. -Hs: id for this node
2. -Hd: id for next hop towards the destination

- Field 4: Node property type tag:

This field denotes the node properties like node-id, the level at which tracing is being done like agent, router or MAC. The tags start with a leading "-N" and are listed as below:

o -Ni: node id

o -Nx Ny -Nz: node s x/y/z coordinate

o -Ne: node energy level

o -Nl: trace level, such as AGT, RTR, MAC

o -Nw: reason for the event .

The different reasons for dropping a packet are given below:

"END" DROP_END_OF_SIMULATION

"COL" DROP_MAC_COLLISION

"DUP" DROP_MAC_DUPLICATE

"ERR" DROP_MAC_PACKET_ERROR

"RET" DROP_MAC_RETRY_COUNT_EXCEEDED

"STA" DROP_MAC_INVALID_STATE

"BSY" DROP_MAC_BUSY131

"NRTE" DROP_RTR_NO_ROUTE i.e no route is available.

"LOOP" DROP_RTR_ROUTE_LOOP i.e there is a routing loop

"TTL" DROP_RTR_TTL i.e TTL has reached zero.

"TOUT" DROP_RTR_QTIMEOUT i.e packet has expired.

"CBK" DROP_RTR_MAC_CALLBACK

"IFQ" DROP_IFQ_QFULL i.e no buffer space in IFQ.

"ARP" DROP_IFQ_ARP_FULL i.e dropped by ARP

"OUT" DROP_OUTSIDE_SUBNET i.e i.e dropped by base stations on receiving routing updates from nodes outside its domain.

- Field 5: Packet information at IP level:

The tags for this field start with a leading "-I" and are listed along with their explanations as following:

o -Is: source address. Source port number

o -Id: destination address.destination port number

o -It: packet type

o -Il: packet size

o -If: flow id

o -Ii: unique id

o -Iv: ttl value

Field 6: Packet information at MAC level:

This field gives MAC layer information and starts with a leading "-M" as shown below

o -Ma: duration

o -Md: destinations ethernet address

o -Ms: sources ethernet address

o -Mt: ethernet type

- Field 7: Packet info at Application level:

The packet information at the application level consists of the type of application like arp, tcp, the type of adhoc routing protocol like DSDV, DSR, AODV etc. The field consists of a leading P and the list of tags for different applications. For example with TCP protocol.

o -P tcp: Information about TCP flow is given by the following subtags.

o -Ps: seq number

o -Pf: how many times this pkt was forwarded

o -Po: optimal number of forwards