# Performance Evaluation of Multi-Hop Ad-Hoc Wireless LANs

Farshad Eshghi

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

April 2004

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisisitons et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

*Your file Votre référence*
*ISBN: 0-612-90383-4*
*Our file Notre référence*
*ISBN: 0-612-90383-4*

**CONCORDIA UNIVERSITY**
**SCHOOL OF GRADUATE STUDIES**

This is to certify that the thesis prepared

By: **Farshad Eshghi Esfahani**

Entitled: **Performance Evaluation of Multi-hop Ad-hoc Wireless LANs**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Electrical & Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________ Chair
Dr. J. Svoboda

______________________ External Examiner
Dr. S. Erfani

______________________ External to Program
Dr. J. Opatrny

______________________ Examiner
Dr. E. Plotkin

______________________ Examiner
Dr. M.R. Soleymani

______________________ Thesis Sunervisor
Dr. A.K. Elhakeen

Approved by ______________________
Dr. M. Mehmet-Ali, Graduate Program Director

APR 3 0 2004

____________ 2004

______________________
Dr. N. Esmail, Dean
Faculty of Engineering & Computer Science

# Abstract

Performace Evaluation of Multi-hop Ad-Hoc Wireless LANs

Farshad Eshghi, Ph.D.
Concordia University

Continuing advances in computing devices and mobile communications and the ever-increasing demand for network services have been fueling the interest in wireless packet services. Mobility, installation speed, simplicity, scalability, and less vulnerability to point-of-break problem, make wireless local area networks (WLAN) a good candidate for providing wireless packet services. Among the introduced WLAN standards, IEEE 802.11 due to its compatibility with the IEEE 802 protocol suite has become the dominating standard.

The need for easy and spontaneous communications between proliferated mobile devices has attracted the attention of researchers and industry toward ad-hoc networks over the past few years. Ad-hoc networks can be used in situations where either there is no wired or wireless infrastructure present or, if present, it can not be used because of security, cost, or safety reasons. Due to the employed random back-off algorithm and system complexity, most of the works toward system performance evaluation resort to simulation. In order to optimally select system parameters to fulfill specific needs, a mathematical description of the system turns to be a lot helpful in observing the trend of any parameter changes made.

In this thesis we mathematically model the distributed coordination function (DCF) mode of operation of IEEE 802.11 WLAN. Our model is based on the presentation of the system with a pair of one-dimensional state diagrams which accommodate variations of many input parameters. The model enables us to make optimum choices for the above parameters in different wireless channel qualities represented by the average bit error rate (BER). System performance is evaluated through three following measures: throughput, delay, and probability of fail to deliver of which the last two are crucial for real-time applications. Following that and aiming at multi-hop WLAN communications, we develop a detailed simulation environment spanning both medium access control (MAC) and physical (PHY) layers to investigate a greater number of phenomena such as hidden terminal, mobility, channel fading and path loss, etc. Results show that optimal selection of the system parameters can lead to considerable improvement in performance especially for time-sensitive applications.

# Acknowledgments

So many people have played a part in the process of this research work.

My sincere thanks must go to my supervisor, Professor Ahmed K. Elhakeem, for his patience, guidance, and invaluable insights from which I have benefited over all these years. He has been very supportive and a continuous source of encouragement during my work towards a Ph.D degree. My thanks also go to the members of my supervisory committee whom I have bothered with questions every now and then.

I would like to express my deep gratitude to my wife for her love, encouragement and all the sacrifices she has made. Without her encouraging me, I would definitely not be able to concentrate on my research during my most productive period. Also I would like to thank my little lovely kids for their understanding. Finally , my parents should be acknowledged since they have always been there for me.

Farshad Eshghi
Concordia University
April 2004

# Contents

# List of Tables

# List of Figures

# List of Acronyms

| | |
|---|---|
| ACK | Acknowledgment |
| ACT | Active State |
| AES | Advanced Encryption Standard |
| AODV | Ad hoc On-Demand Distance Vector |
| ARQ | Automatic Repeat Request |
| AP | Access Point |
| BER | Bit Error Rate |
| BPSK | Binary Phase Shift Keying |
| BSA | Basic Service Area |
| BSS | Basic Service Set |
| CAC | Channel Access and Control |
| CBR | Constant Bit Rate |
| CCK | Complementary Code Keying |
| CF | Contention-Free |
| CFP | Contention-Free Period |
| CSMA/CA | Carrier Sense Multiple Access/Collision Avoidance |
| CSMA/CD | Carrier Sense Multiple Access/Collision Detection |
| CTS | Clear to Send |
| CW | Contention Window |
| DBPSK | Differential Binary Phase Shift Keying |

| | |
|---|---|
| DCF | Distributed Coordination Function |
| DIFS | Distributed Inter-Frame Space |
| DS | Direct Sequence |
| DSSS | Direct Sequence Spread Spectrum |
| DTBS | Distributed Time-Bounded Services |
| EIFS | Extended Inter-Frame Space |
| ESS | Extended Service Set |
| ETSI | European Telecommunications Standards Institute |
| e.t.e | end-to-end |
| FCC | Federal Communications Commission |
| FEC | Forward Error Coding |
| FH | Frequency Hopping |
| FHSS | Frequency Hopping Spread Spectrum |
| FR | Fragment |
| FRA | Fixed-Radio Access |
| GFSK | Gaussian Frequency Shift Keying |
| GPS | Global Positioning System |
| GSM | Global System for Mobile Communications |
| HiperLAN | High performance radio Local Area Network |
| IBBS | Independent Basic Service Set |
| IDL | Idle State |
| IFS | Inter-Frame Space |
| IP | Internet Protocol |
| IR | InfraRed |
| ISM | Industrial, Scientific. and Medical |
| LAN | Local Area Network |
| LCD | Liquid Crystal Display |

| | |
|---|---|
| LLC | Logical Link Control |
| MAC | Medium Access Control |
| MANET | Mobile Ad-hoc Network |
| MOK | M-ary Orthogonal Shift Keying |
| MoP | Measure of Performance |
| MPDU | MAC Protocol Data Unit |
| NAV | Network Allocation Vector |
| OFDM | Orthogonal Frequency Division Multiplexing |
| OH | OverHead |
| PAN | Personal Area Network |
| PBCC | Packet Binary Convolutional Coding |
| PC | Point Coordinator |
| PCF | Point Coordination Function |
| PDA | Personal Digital Assistant |
| PFR | Packet Failure Rate |
| PHY | Physical |
| PIFS | Point (coordination function) Inter-frame Space |
| PLCP | Physical Layer Convergence Procedure |
| PMD | Physical Medium Dependent |
| PN | Pseudo-Noise |
| PPM | Pulse Position Modulation |
| PSDU | Physical layer Service Data Unit |
| PSR | Packet Success Rate |
| QAM | Quadrature Amplitude Modulation |
| QL | Queue Length |
| QoS | Quality of Service |
| QPSK | Quadrature Phase Shift Keying |

| | |
|---|---|
| rms | root mean square |
| RF | Radio Frequency |
| RITL | Radio In The Loop |
| RTS | Request to Send |
| RX | Receive or Receiver |
| SIFS | Short Inter-Frame Space |
| SNR | Signal to Noise Ratio |
| TCP | Transport Control Protocol |
| TDM | Time Division Multiplexing |
| TKIP | Temporal Key Integrity Protocol |
| TPDU | Transport Protocol Data Unit |
| TTL | Time-to-Live |
| TX | Transmit or Transmitter |
| UNII | Unlicensed-National Information Infrastructure |
| UWB | Ultra WideBand |
| WBAN | Wireless Body Area Network |
| WECA | Wireless Ethernet Compatibility Alliance |
| WEP | Wired Equivalent Privacy |
| Wi-Fi | Wireless Fidelity |
| WLAN | Wireless Local Area Network |
| WPAN | Wireless Personal Area Network |

# Chapter 1

# Introduction

The ability to communicate with anyone on the planet from anywhere on the planet has been mankind's dream for a long time. Wireless is the only medium that can enable such untethered communication. Among all the wireless services, mobile computing continues to enjoy rapid growth thanks to ongoing technological advances in laptop computers, portable devices such as Personal Digital Assistants (PDA), and in a slower pace though as important, battery production. Availability of higher data rates, and prices continuously dropping are paving the way for the emergence of new bandwidth-hungry applications and leading to escalation of the demand for wireless communication among mobile users. There are two different and distinct approaches for enabling wireless mobile computers to communicate with each other. The first is to utilize the existing cellular network infrastructure originally developed for voice communications, and the second is WLAN deployment. Although up along the hierarchy of cellular system generations, higher data rates are provisioned, but still they are way behind the capabilities of WLANs. WLANs besides, have several other advantages when compared to traditional cellular systems such as lower cost of operation, fault tolerance, and capability of being setup on demand. On the other hand, cellular systems enjoy a global and more reliable coverage.

## 1.1 Wireless Networks

A wireless network is comprised of devices with wireless adapters communicating with each other using radio waves. The signal transmitted can be received only within a certain distance from the sender. Based on the network architecture, wireless networks can be logically divided into two classes: distributed and centralized.

- **Distributed Wireless Networks.** Distributed wireless networks, also called ad-hoc networks, are a collection of communication devices (nodes) that wish to communicate, but have no fixed infrastructure available, and have no predetermined organization of available links. Individual nodes are responsible for dynamically discovering which nodes they can directly communicate with. Not all the nodes can directly communicate with each other, so nodes are required

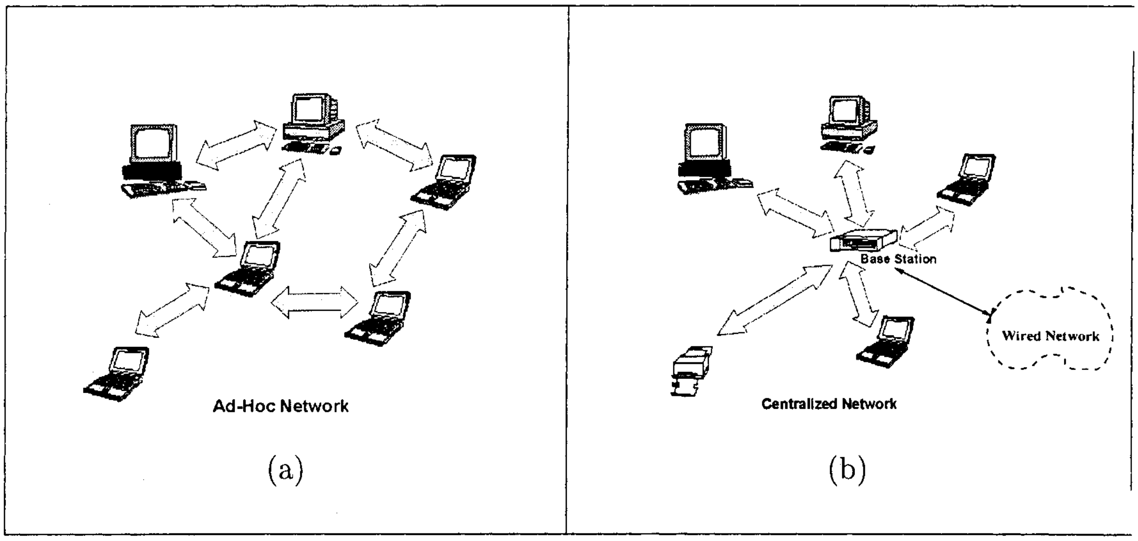


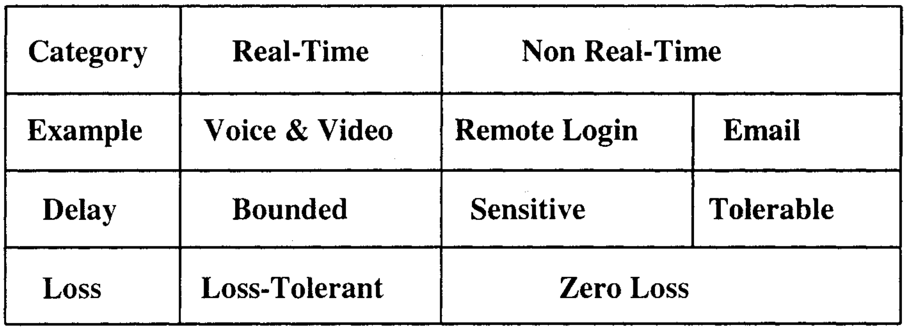
Figure 1.1: Wireless network architectures a) Ad-Hoc network. b) Infrastructured network.

to relay packets on behalf of other nodes in order to deliver data across the network. A typical ad-hoc network is illustrated in Fig. 1.1-a.

- **Centralized Wireless Networks.** Centralized wireless networks, also known as last-hop networks, are extensions to wireline networks with wireless in the last section of the network. These networks have a base station that acts as the interface between wireless and wireline networks. In centralized networks the down-link transmissions (from base station [BS] to wireless nodes) are broadcast and can be heard by all the devices on the network. The up-link (from wireless terminals to the BS) is shared by all the nodes and is therefore a multiple access channel. The existence of a central node like a BS gives a great degree of flexibility in the design of medium access control (MAC) protocols. The BS can control the up-link transmissions which benefit quality of service (QoS) implementation. The system communications of a centralized network is shown in Fig. 1.1-b.

### 1.1.1 Ad-Hoc Networks

The need for easy and spontaneous communications between proliferated mobile devices has attracted the attention of researchers and the industry toward ad-hoc networks over the past few years. Ad-hoc networks can be used in situations where either there is no wired or wireless infrastructure present or, if present, it can not be used because of security, cost, or safety reasons. A few examples include military battlefield communications, sensor networks, an infrastructure-less network of notebook computers in a conference or campus setting, rare animal tracking, space exploration, undersea operations, temporary offices, disaster recovery operations, search and rescue. Portability and mobility are in great consistence with the infrastructure-less nature of the ad-hoc networks. This has made the mobile ad-hoc networks (MANET) a compelling research topic for a while. On the other hand, viewed either as a stand-alone network or as a last-mile wireless connection to the internet, ad-hoc networks have the capability of expansion through multi-hopping. Multi-hop MANETs, in situations where they are supposed to support multimedia, also called multimedia, multi-hop, and mobile ($M^3$) wireless networks, present a challenge in many aspects. This is attributed to the fact that applications with different natures demand different treatments as shown in Fig. 1.2.



| Category | Real-Time | Non Real-Time | |
|---|---|---|---|
| Example | Voice & Video | Remote Login | Email |
| Delay | Bounded | Sensitive | Tolerable |
| Loss | Loss-Tolerant | Zero Loss | |

Figure 1.2: Different applications quality requirements.

Multi-hop MANETs have features that distinguish them from wired and infrastructured wireless networks. These features can be enumerated as: unpredictable link properties, node mobility, limited battery life, route maintenance, security, hidden terminal[1], exposed terminal[2], and capture effect[3]. Figure 1.3 illustrates a stand-alone example of multi-hop ad-hoc WLAN in disaster recovery situation. Figure 1.4 shows how a multi-hop ad-hoc WLAN cluster can be accommodated in a hybrid global telecommunications network. There have been already efforts to integrate WLANs

[1] A hidden terminal can hear the destination node but not the source node which leads to higher probability of collision.

[2] An exposed terminal can hear the source node but not the destination node which leads to bandwidth waste.

[3] Happens when a particular transmission can be recovered out of simultaneous transmission attempts. Capture effect may cause unfairness in bandwidth sharing.

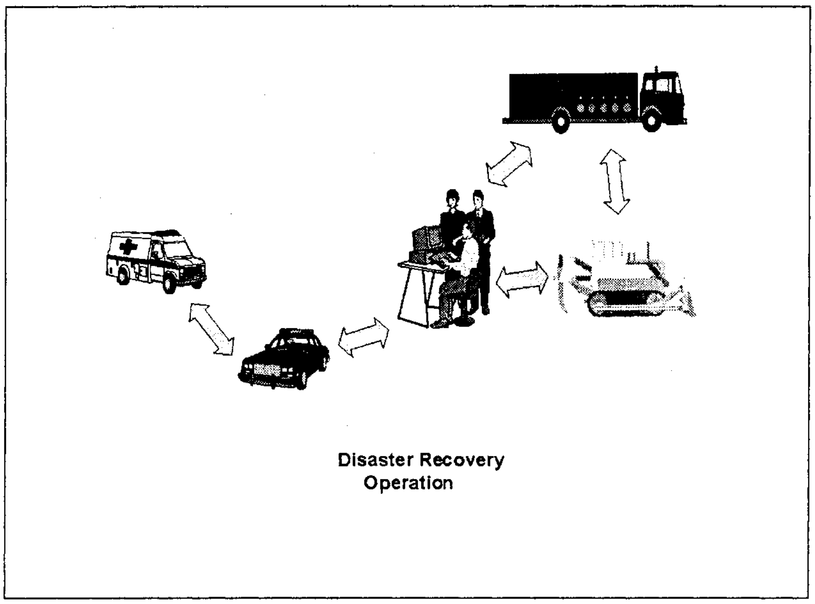


Figure 1.3: A stand-alone example of multi-hop ad-hoc WLAN.

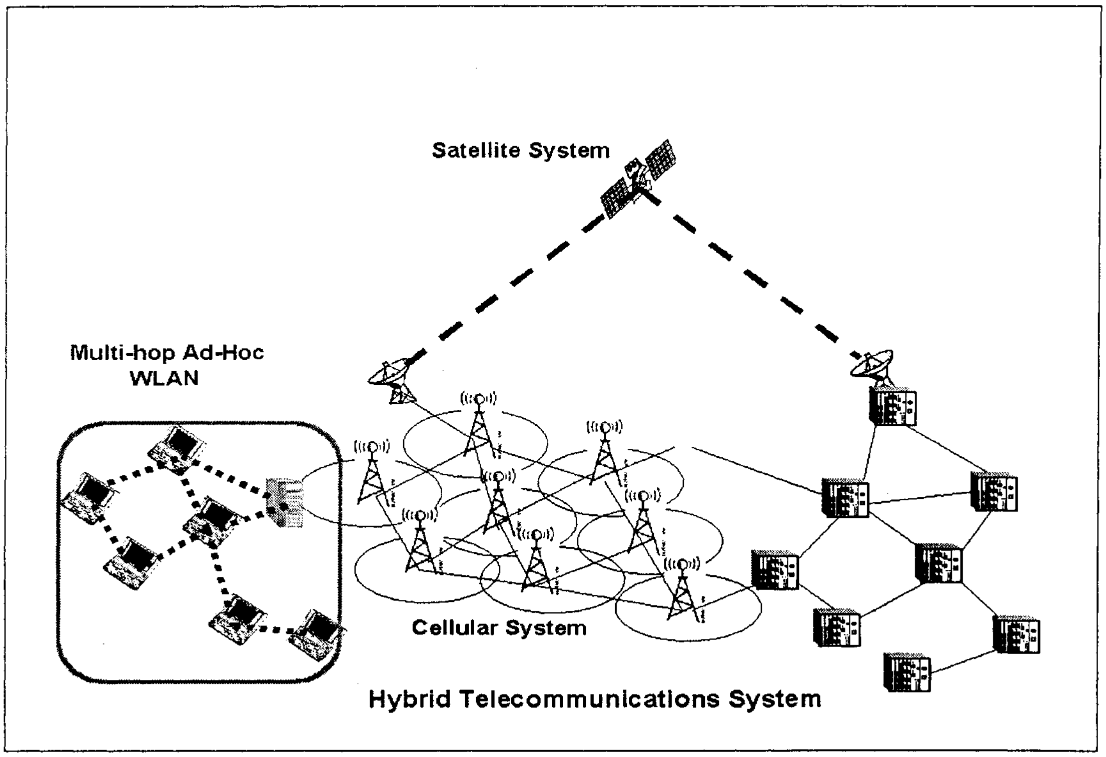


Figure 1.4: A Multi-hop ad-hoc WLAN as a part of global telecommunications Network.

with cellular systems (e.g. see references [2], [3], and [4]).

## 1.2 Motivation

Ad-hoc networking is a multi-layer problem. The physical layer must adapt to rapid changes in link characteristics, and the multi-path faded nature of wireless channel. In this very layer, decisions have to be made on modulation and coding schemes. Similarly, the link layer should react to the changes in link error rate, including the use of automatic repeat request (ARQ). The medium access control (MAC) layer needs to minimize collisions, allow fair access, and reliably transport data over the shared wireless links in the presence of rapid changes and hidden or exposed terminals. The network layer needs to determine and distribute information used to calculate paths in a way that maintains efficiency when links change often and bandwidth is at a premium. The transport layer must be able to handle delay and packet loss statistics that are very different from wired networks. Finally, applications need to be designed to cope with frequent disconnections and reconnections with peer applications as well as widely varying delay and packet loss characteristics.

Thus, there are so many interacting parameters, mechanisms, and phenomena in the area of ad-hoc networking. To achieve a particular performance level, or to be consistent with the network terminology, an intended quality of service (QoS), some suggest a layered approach as by [5] (in which optimization is done independently in each individual layer) while others favor a cross-layering approach (where the physical and MAC layer knowledge of the wireless medium is shared with higher layers) as suggested by [6] and [7]. Ad-hoc networking, in any case, remains open to research efforts for years to come.

In general, performance evaluation and optimization efforts made toward ad-hoc networking have used computers or real-life simulation. While this is the only means in the case of multi-hop scenario, there have been a few analytical approaches in the single-hop case as reviewed in Chapter 2.

The problem with the related past work is that regarding the single-hop case, the proposed analytical models are either too complicated (in terms of model tractability or size) or they adopt too many simplifying and restrictive assumptions which undermine the inherent dynamics of the system. The latter is true with almost all the simulation efforts as well. However, due to the inherent complexity of the matter, assumptions are unavoidable. Our motivations for conducting this research are:

- To propose a simpler and more representative analytical model of the IEEE 802.11 ad-hoc mode of operation.
- To do a more inclusive simulation study of the multi-hop scenario.

## 1.3 Major Contribution

The major contribution of this thesis can be summarized as follows:

- MAC layer fuctionality of the IEEE 802.11 ad-hoc WLAN was mathematically modeled. The model is based on a pair of one dimensional M/M/K/1 state diagrams (rather than the more complex multi-dimensional state diagram techniques used in [1]) which easily accommodates variations of many input parameters and is capable of ingesting many other probable parameters of interest. The model successfully imitates the dynamic behavior of the random access technique as well.
- Regarding multi-hop operation of the IEEE 802.11 ad-hoc WLAN, a detailed simulation environment spanning both MAC and PHY layers was developed

which considers mobility and varying wireless channel. Due to its modular structure, different mobility schemes, channel models, and traffic generators can be plugged in. The thoroughness of our simulation differentiates it from the past simulation efforts.

- Based on the above analytical model and simulation package, system performance was evaluated in terms of throughput, delay, and probability of fail to deliver. The latter two are crucial for real-time applications which we are more interested in (Fig. 1.2). Accordingly, conclusions were drawn towards optimal selection of system parameters.

## 1.4 Thesis Outline

This thesis is concerned with the performance evaluation of IEEE 802.11 WLAN in ad-hoc mode of operation and how it is affected by different choices of system parameters in different situations. It consists of five chapters as follows.

**Chapter 2: Wireless LAN Protocols.** Following a brief introduction to different wireless packet services, and in particular peer local area services, the scope of wireless local area networks (WLAN) is reviewed. The focus is then turned to IEEE 802.11 protocol suite where a literature survey on the protocol, enhancements, capabilities, and challenging issues are presented. Special attention is paid to works related to ad-hoc mode of operation in multi-hop and mobile scenarios.

**Chapter 3: Performance Analysis of Ad-Hoc IEEE 802.11 WLAN.** In this chapter, the Distributed Coordination Function (DCF) of the IEEE 802.11 protocol suite over a single-hop coverage area is reviewed in more detail. Afterwards, our proposed mathematical model of DCF is presented. The mathematical model is used to find measures of performance, particularly those of much interest to real-time applications. Two different fragment transmission strategies, contention per packet and contention per fragment, are then compared in terms of their performance measures using the mathematical model. The chapter concludes with simulating DCF to be compared with analytical approach results based on the model.

**Chapter 4: Performance Evaluation of Multi-hop, Mobile, Ad-Hoc IEEE 802.11 WLAN.** This chapter deals with a simulation effort of multi-hop, mobile, and ad-hoc IEEE 802.11 WLAN. The simulation program has been modularly developed in order to be capable of accommodating a variety of traffic generation, wireless channel, and mobility models. How performance measures are affected by system parameters choice is finally discussed.

**Chapter 5: Conclusion and Future Work.** The contributions of this thesis, along with suggestions for future work are summarized in this chapter.

# Chapter 2

# Wireless LAN Protocols

## 2.1 Introduction

Technological advances, coupled with the flexibility and mobility of wireless systems, are the driving force behind the anyone, anywhere, anytime goal of communications. The compelling feature of integrateability of different kinds of services such as data, voice, and video has pushed digital communications further toward packet communications. Among wireless packet service providers, there are satellite system, wireless local loops (WLL), cellular system, wireless metropolitan area networks (WMAN), wireless local area networks (WLAN), wireless personal area networks (WPAN), and wireless body area networks (WBAN). Figure 2.1 illustrates the area of operation of the aforementioned wireless packet services. Wireless Local Loop, sometimes called radio in the loop (RITL) or fixed-radio access (FRA), is an ideal application to provide network connection besides many other services, to a remote rural area. A wireless MAN provides WLAN-like services, primarily internet access, to an entire city. The air interfaces for WMANs as defined by IEEE 802.16, also known as WirelessMAN,

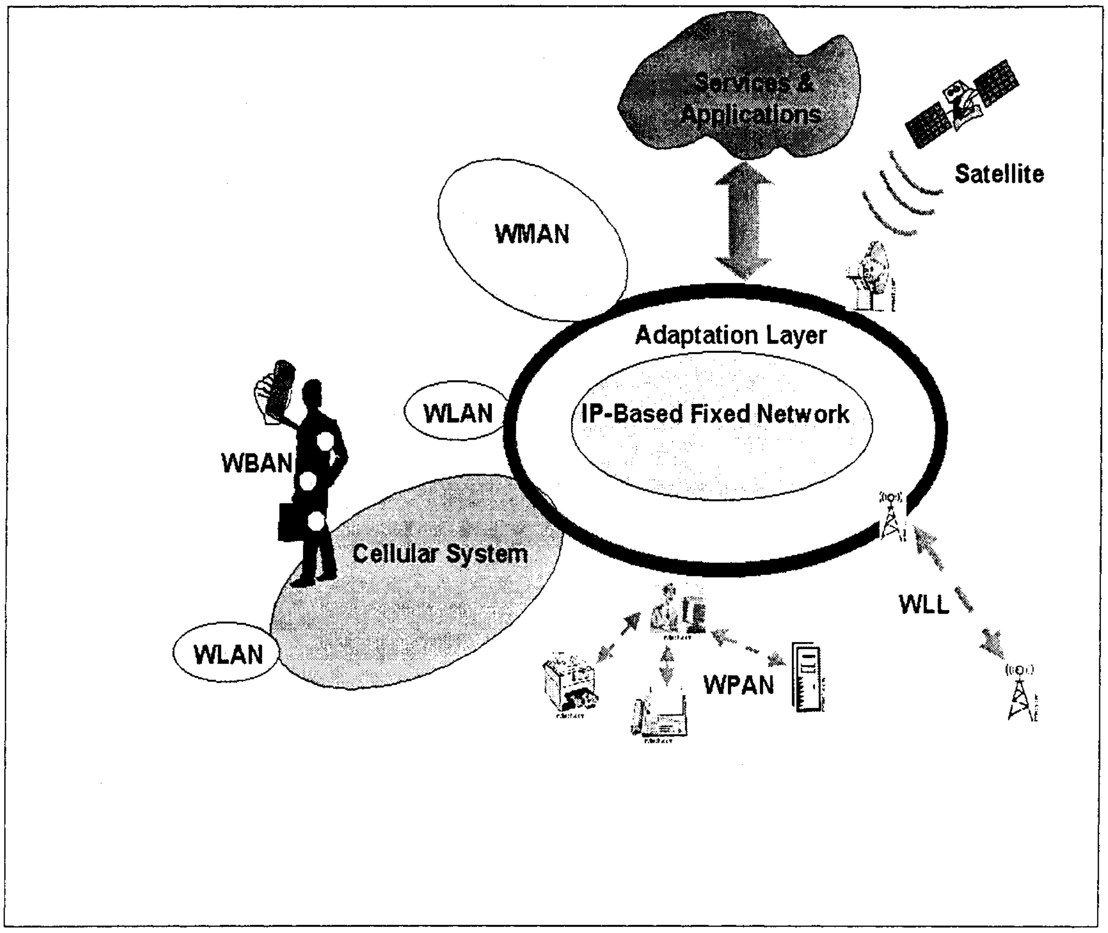


Figure 2.1: Multiple wireless access approaches and networks.

provide inexpensive broadband access to buildings through exterior antennas that communicate with a base station. WPANs are intended to provide wireless connectivity among personal office equipment, and consumer electronic devices. The core representatives of WPANs are IEEE 802.15.1 (Bluetooth [8]) and with lesser importance HomeRF [9]. Bluetooth is a specification to standardize on a low-cost, low-power RF (Radio Frequency) solution for wireless transmission between a wide variety of devices such as PCs, keyboards, cordless telephones, cell phones, headsets, printers, monitors, LCD projectors, and PDAs (Personal Digital Assistants). So the specific features of WPANs are small range of operation and low power consumption.

Wireless area body networks (WBAN) are networks whose nodes are usually placed close to the body or in everyday clothing and so are also called wearable networks. Compared to other wireless networks a WBAN has some distinct features and requirements. Due to the close proximity of the network to the body, electromagnetic pollution should be extremely low. Thus a noninvasive WBAN requires a low transmit power [10]. Accordingly, the ultra wideband (UWB) transmission technique is a potential candidate for WBANs [11]. Table 2.1 summarizes the key features of the wireless networks mentioned.

One of the common features among wireless networks is operation in unlicensed bands. No license is required for using theses bands, however Federal Communications Commission (FCC) regulations should be met. Unlicensed bands are split into industrial, scientific, and medical (ISM) bands and unlicensed-national information infrastructure (UNII) bands. Tables 2.2 and 2.3 define the unlicensed frequencies available in the United States.

Wireless transmission within 2.4 GHz requires spread spectrum utilization. However, the transmissions within the 5 GHz band depart from the spread spectrum technology, instead using a frequency division multiplexing scheme. In making the choice of the band of operation, a designer should consider different attributes of these bands: performance potential, cost, interference, range, and material-penetration capabilities.

In the following section, two major WLAN protocols, HiperLAN and IEEE 802.11 are reviewed. The focus is then narrowed down to IEEE 802.11 protocol described in section 2.3. In this section, past research efforts toward performance evaluation of the protocol, multi-hop operation, interference, interaction with TCP/IP protocol. are mentioned. More emphasis is laid on the DCF mode of operation in single and multi-hop scenarios in keeping with the subject of the thesis.

Table 2.1: Summary of characteristics of some WMAN/WLAN/WPAN/WBAN standards

| Technology | Operational spectrum | Physical layer | Data rate | Channel access | Coverage |
|---|---|---|---|---|---|
| IEEE 802.11b | 2.4 GHz | CCK | 11 Mbps | CSMA/CA | 100 m |
| HomeRF | 2.4 GHz | FHSS/FSK | 10 Mbps | CSMA/CA,TDMA | 50 m |
| Bluetooth | 2.4 GHz | FHSS/GFSK | 1 Mbps | FH/TDD | 10 m |
| HiperLAN2 | 5 GHz | OFDM/QAM | 54 Mbps | TDMA/TDD | 150 m |
| UWB | 3.1-10.6 GHz | Multiband/BPSK,PPM | 110/10,200/4 (Mpbs/m) | XXX | 10 m |
| Ricochet | 900 MHz | FHSS | 176 Kbps | CSMA | City |

Table 2.2: ISM Frequencies

| Specification | ISM-900 | ISM-2.4 | ISM-5.8 |
|---|---|---|---|
| Freq. range | 902-928 MHz | 2400-2483.5 MHz | 5725-5850 MHz |
| Bandwidth | 26 MHz | 83.5 MHz | 125 MHz |
| Max. Power | 1 W | 1 W | 1 W |

Table 2.3: UNII Frequencies

| Specification | UNII indoor | UNII low power | UNII/ISM |
|---|---|---|---|
| Freq. range | 5150-5250 MHz | 5250-5350 MHz | 5725-5825 MHz |
| Bandwidth | 100 MHz | 100 MHz | 100 MHz |
| Max. Power | 50 mW | 250 mW | 1 W |

## 2.2 MAC and Physical layers of WLAN Protocols

The core element in wireless LANs is the MAC layer which outlines the channel access and sharing scheme. An almost complete classification of different MAC protocols is presented in [12]. Reference [13] is also a good source in comparing different access techniques. Different wireless LAN standards or protocol suites may adopt more than one MAC protocol for different modes of operation. In the following subsections two mature WLAN standards are described: IEEE 802.11 and HiperLAN (High performance radio local area network). IEEE 802.11 has been the more successful standard mostly because one of its design requirements is interoperability with any IEEE 802.x based network. HiperLAN is the European standard for wireless LANs, it is not widely used but it is still an interesting standard. IEEE 802.11 standard will be studied in more detail since it forms the basis of our future work. A parallel explanation of IEEE 802.11 and HiperLAN can be found in [14] and [15].

### 2.2.1 HiperLAN

European Telecommunications Standards Institute (ETSI) is one of the world's recognized Standards bodies. It developed global system for mobile communications (GSM) standards for digital cellular telephony, and it also published many other standards dealing with all aspects of telecommunications. HiperLAN was developed by ETSI during the period 1991 to 1996, whose goal was to achieve higher data rate than IEEE 802.11 data rates - 1 to 2 Mbps. HiperLAN includes a family of four standards: HiperLAN/1, HiperLAN/2 [16], HiperAccess (HiperLAN/3), and HiperLINK (HiperLAN/4). In what follows, we focus on the discussion of HiperLAN/1. HiperLAN/1 defines data link layer and physical layer. For local area networks, data

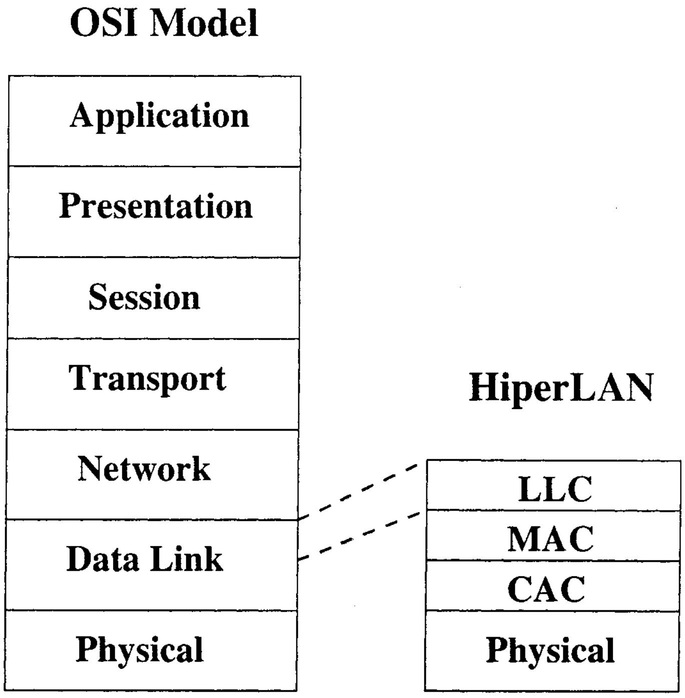


Figure 2.2: HIPERLAN/1 Reference Model.

Link layer is further divided into two sublayers: the logical link control (LLC) and

MAC. HiperLAN only deals with MAC and PHY (Fig. 2.2). An intermediate layer, the channel access and control (CAC) sublayer, is introduced in the HIPERLAN/1 architecture to deal with the channel access signaling and protocol operation required to support packet priority. In fact medium access functionality is done in this very same layer. Regarding the PHY layer, HiperLAN uses the radio frequency band 5150 MHz to 5300 MHz. Table 2.4 shows the nominal frequency of each carrier. It is required that all transmissions shall be centered on one of the nominal carrier frequencies, and all HiperLAN equipments shall operate on all 5 channels. Gaussian minimum shift keying (GMSK) is used as the high bit rate modulation scheme. Regarding the access, the CAC layer defines how a given channel access attempt will be made depending on whether the channel is busy or idle, and at what priority level the attempt will be made, if contention is necessary. A transmission passes through three phases: the prioritization phase, the contention phase and the transmission phase. The transmission phase forms the channel access cycles because during the transmission the medium is considered free. These three phases form a synchronized channel access cycle. CAC works in the following three steps (Fig. 2.3):

1. During the prioritization phase, the data transmissions with highest channel access priority are selected out. This phase has 1-5 slots, packets that are to be transferred each have a priority between 0 and 4 (with 0 being the highest priority). Nodes with priority $p$ transmit in slot $p+1$. When a slot is used, the priority phase ends and all users with lower priority than the nodes emitting first defer their transmission.

2. In the contention phase,all the remaining nodes pick a random number, $i < 12$, where $P(i) = 0.5^i$. Every such node transmits a burst of length i slots (slots are 256 bits in length). When a node ends its transmission, it senses the channel; if the channel is idle then the node knows that it has picked the largest i and,

goes on to the next phase or else it defers its transmission.

3. At the final phase (transmission phase), before transmitting, the remaining nodes backoff for a random period of time and sense the channel. If no activity is sensed then data transmission is initiated.

A data transmission can compete for the channel only if it is ready at the beginning of a channel access cycle. Otherwise, it should wait until the next channel access cycle. The HiperLAN MAC layer defines the various protocols which provide the HiperLAN features of power conservation, security, and multi-hop routing, as well as the data transfer service to the upper layers.

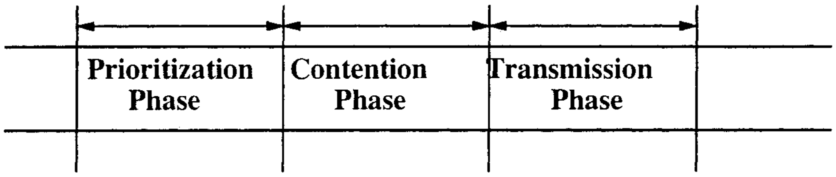


Figure 2.3: Channel access and control

Table 2.4: Nominal carrier center frequencies

| Carrier No. | Center Frequency (MHz) |
|---|---|
| 0 | 5176.4680 |
| 1 | 5199.9974 |
| 2 | 5223.5268 |
| 3 | 5247.0562 |
| 4 | 5270.5856 |

### 2.2.2 IEEE 802.11

The IEEE 802.11 standard places specifications on the parameters of both the physical (PHY) and medium access (MAC) layers of the network. The physical layer is further divided into two sublayers: physical layer convergence procedure (PLCP) and physical media dependent (PMD) sublayers (Fig. 2.4). PLCP adapts the capabilities of the physical medium dependent system to the physical layer service. PLCP defines a method of mapping the 802.11 PHY layer service data units (PSDU) into a framing format suitable for sending and receiving user data and management information between two or more stations using the associated physical medium dependent system. This allows 802.11 MAC to operate with minimum dependence on the PMD sublayer. PMD defines the characteristics and method of transmitting and receiving data through a wireless medium between two or more stations each using the same modulation system. Regarding the PHY layer, the standard provides five different types: frequency hopping spread spectrum (FHSS), direct sequence spread spectrum (DSSS), infrared (IR), orthogonal frequency division multiplexing (OFDM), and ultra wideBand (UWB). The IEEE 802.11 considers the following as possible modulation techniques: binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), Gaussian frequency shift keying (GFSK), M-ary orthogonal shift keying (MOK), com-

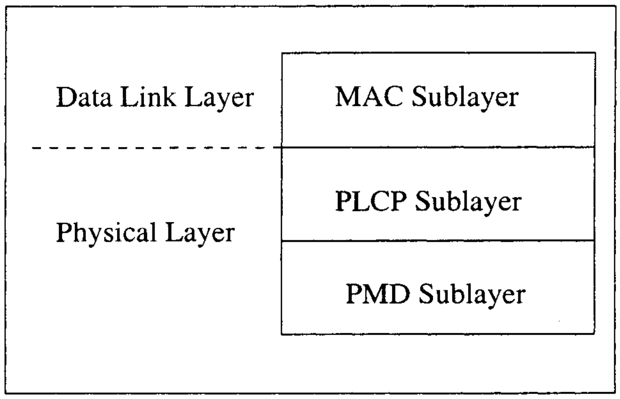


Figure 2.4: IEEE 802.11 Reference Model

plementary code keying (CCK), pulse position modulation (PPM) , and quadrature amplitude modulation (QAM).

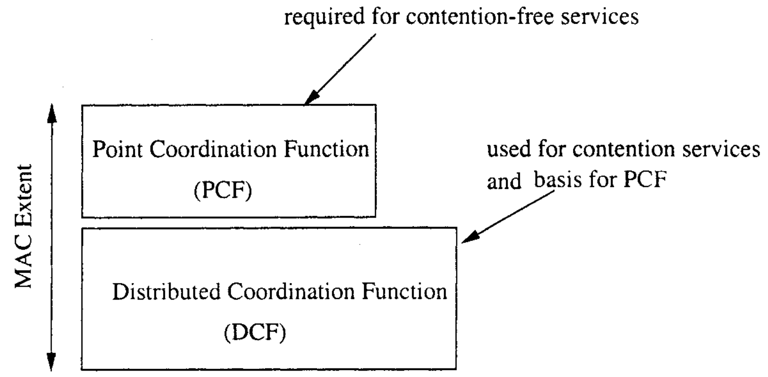

required for contention-free services
Point Coordination Function
(PCF)
used for contention services
and basis for PCF
MAC Extent
Distributed Coordination Function
(DCF)

Figure 2.5: Coexistence of DCF and PCF in IEEE 802.11 MAC layer

Regarding the MAC layer, IEEE 802.11 uses carrier sense multiple access with collision avoidance (CSMA/CA) to access the medium. The basic idea is: if a station wants to transmit, it first senses the medium. If the medium is busy, the state defers its transmission to a later time.

The MAC protocol is formed of two separate coexisting coordination functions that provide support for asynchronous data transfer and optionally distributed time-bounded services (DTBS). Asynchronous data transfer refers to traffic that is insensitive to time delays; it is supported by the distributed coordination function (DCF). Delay constrained traffic is handled by the point coordination function (PCF) of the protocol. As identified in the specification, all stations must support the DCF while PCF is an optional capability providing contention-free services. The MAC architecture is depicted in Fig. 2.5, where the DCF sits directly on top of the physical layer and supports contention services. Interacting users are grouped in basic service sets (BSS) and the geographical area covered by a BSS is called basic service area (BSA). A group of infrastructured WLANs connected together through a distribution network and access points form an extended service set (ESS). Fig. 2.6 illustrates BSS and ESS. A stand-alone ad-hoc network is called independent-BSS (IBSS). The

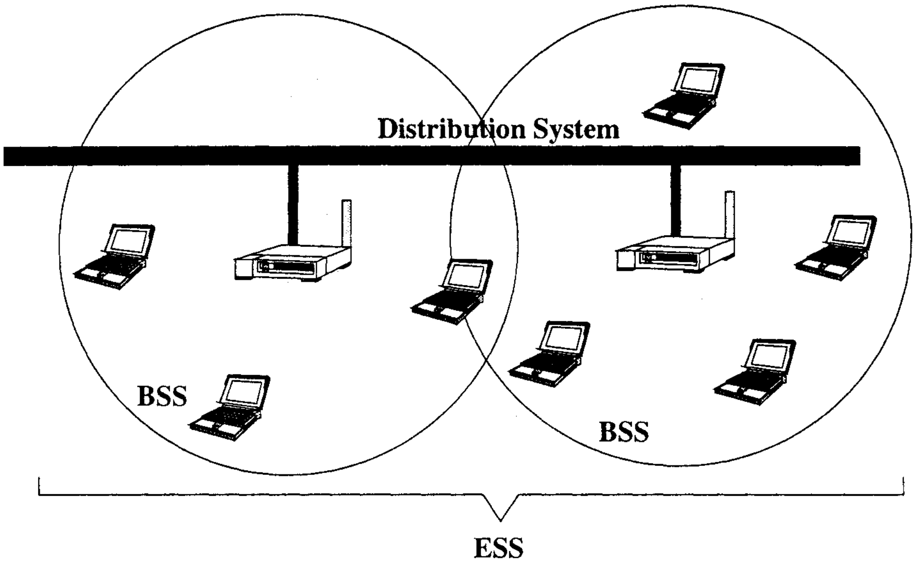


Figure 2.6: A typical 802.11 WLAN

protocol has 3 pre-defined inter-frame space (IFS) time periods with increasing duration: short IFS (SIFS), point coordination function IFS (PIFS) and distributed coordination function DCF-IFS (DIFS). The IFSs are mandatory periods of idle time and are used to control event priorities. Furthermore, every user maintains a network allocation vector (NAV) which is the remaining busy period of the shared channel (in microseconds). Correct receipt of packets is signaled through ACK packets; an ACK packet should be returned SIFS seconds after the successful reception of a packet. Time is divided into contention periods where DCF access is used and contention free periods where PCF access is used. Note that the PCF cannot be used in ad hoc networks because the PCF function needs a Point Coordinator, a role that is usually bestowed upon the Access Points (APs) in infrastructured networks.

In the next two paragraphs, the DCF and PCF functions are described and results from the simulation study by Crow et al. are mentioned [17].

**Distribution Coordination Function-** The DCF provides best-effort delivery of data inside a BSS during contention periods, its channel access scheme is very similar to CSMA/CA. Stations can decide to transmit at any time if the channel is sensed to be idle for DIFS seconds and their NAV is zero; having decided to transmit, the user waits for an extra backoff period and then starts transmitting (basic access method). The user can also do virtual carrier sensing by sending a request to send (RTS) packet to the intended receiver and waiting for a clear to send (CTS) packet. The RTS packet broadcasts the duration of the intended transmission; that duration is also transmitted in the CTS response to reduce the possibility of hidden terminal problems. Having detected the CTS or the RTS, other users update their NAV. The two transmission schemes in DCF mode are shown in Figs. 2.7-a and 2.7-b. Note that time is slotted in time periods that correspond to a slot time. Unlike slotted Aloha however, the slot time is much smaller than the packet duration (which is variable). Obviously the use of RTS-CTS entails extra overhead. Intuitively, one might wish to drop the RTS-CTS mechanisms for smaller packets, using them only for larger packets. In fact, a RTS-CTS packet size threshold can be defined whereby data units smaller than the threshold are packetized and transmitted without any RTS and payloads larger than the threshold are transmitted with the use of the RTS-CTS mechanism. Simulations made by [17] suggest that a threshold of 200 bytes is optimal. For a thorough study of the performance of DCF without time-sharing with the PCF, the reader is referred to [18].

**Point Coordination Function-** The PCF is an optional connection-oriented capability. The PCF needs a Point Coordinator (PC) that initiates and controls the contention-free period(CFP) where PCF is used. The PC first senses the channel for PIFS seconds (priority over regular DCF traffic) and then starts a CF period by broadcasting a beacon signal. All regular terminals add CFP-Max-Duration (the maximum possible duration of the contention free period) to their NAV. Later, active users with time-bounded packet streams are continuously polled. The PC can end the contention free period at any time by transmitting a CF-end packet; this occurs frequently when the network is lightly loaded. Figure 2.7-c is a sketch of the CFP repetition interval, illustrating the coexistence of the PCF and DCF. When a terminal's turn in the poll comes, the PC sends a data packet to it (if any such data is buffered) piggybacked by a poll token or simply a poll token. The receiver sends back an ACK after SIFS seconds or any buffered data piggybacked with an ACK. Note that all packets are separated by SIFS seconds, this is why piggybacking is very useful in this transmission scenario. Priority polling mechanisms can be used if different QoS levels are requested by different polled users. Users who are idle repeatedly are removed from the poll cycle after k idle periods and polled again at the beginning of the next CF period. k=1 was found to be optimal by Crow et al. [17] when all time-bounded data are voice data streams; this is explained by the fact that relative to the duration of the CFP, voice streams are sent in slow on-off bursts.

## 2.3 A Closer look into IEEE 802.11

Any WLAN protocol, with no exception of IEEE 802.11, faces almost the same kind of challenges. Among those, coexistence in the same spectrum range, multimedia support, multi hop operation, integration into an all-IP network (e.g. see [19], [20], and [21]), and security (e.g. see [22]) are the major to mention.

Although the use of unlicensed bands facilitates spectrum sharing and allows for an open access to the wireless medium, it also raises serious challenges, such as mutual interference between different radio systems and spectrum utilization inefficiency. A

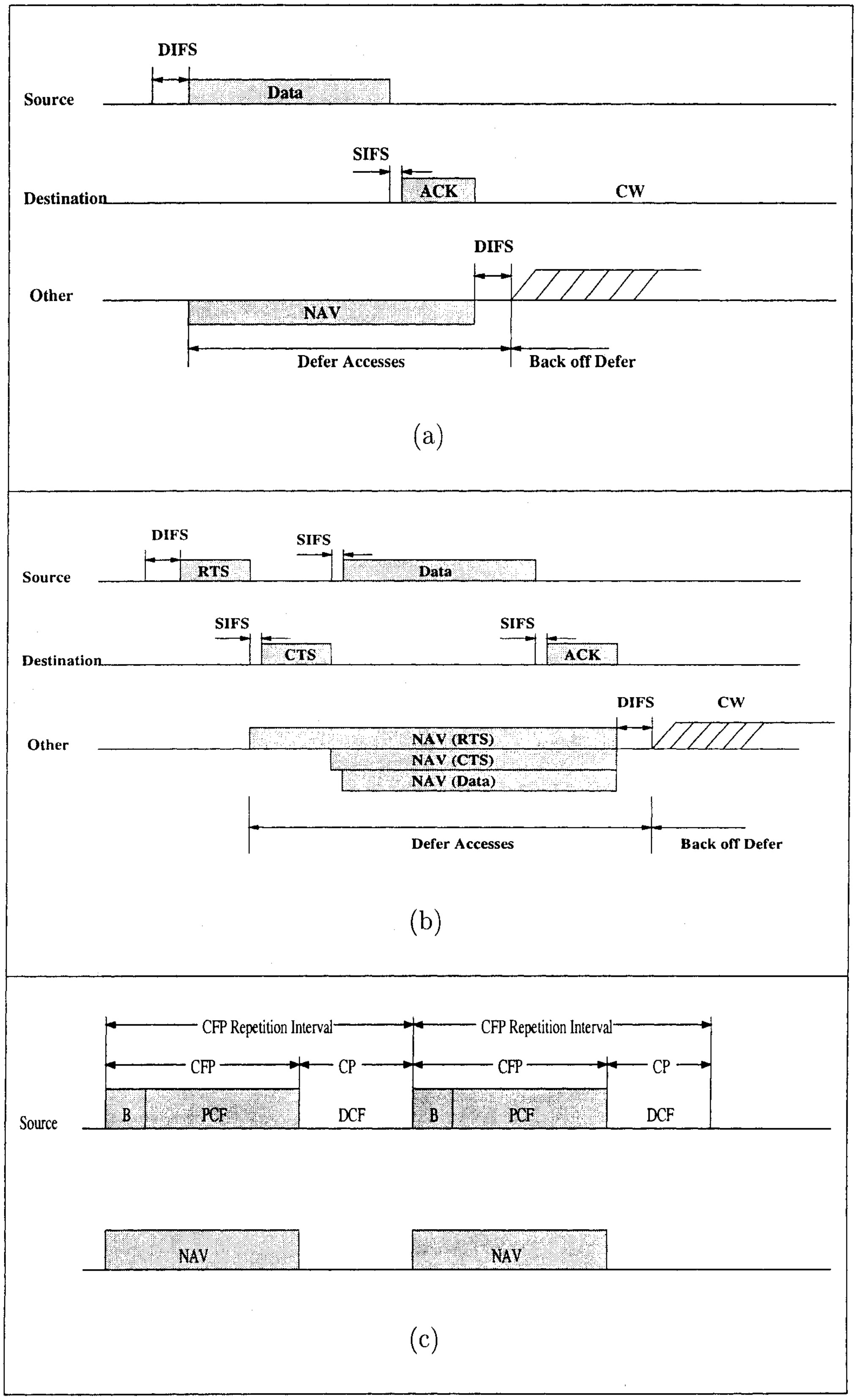


Figure 2.7: a) Transmission of MPDU without RTS/CTS. b) Transmission of MPDU using RTS/CTS. c) Coexistence of the PCF and DCF.

better understanding of the communication channel characteristics in the unlicensed band can be found in [23], [24], [25], [26],and [27] and the coexistence problem has been addressed in [28], [29], and [30]. Under the cap of multimedia, real-time support is the most challenging task to achieve due to its hard-to-meet requirements. Real-time applications are usually more sensitive to delay and less sensitive to loss as compared to their non-real-time counterparts. The issue which boils down to achieving higher throughputs and priority implementation, along with the multi-hop operation, is further dealt with in the sections 2.3.2 and 2.3.3.

### 2.3.1 Different Versions

From its beginnings, the IEEE 802.11 standard has grown into something much more capable and complex. When it was first introduced in 1997, the WLAN standard specified operation at 1 and 2 Mbps in the infrared and in the unlicensed 2.4-GHZ ISM band. Since then, 802.11 has acquired many new attributes. It can now operate at higher speeds and in additional bands. With growth, new issues have arisen such as security, roaming among multiple access points, and even quality of service. These issues are dealt with by extensions to the standard identified by letters of the alphabet derived from the 802.11 task groups that created them:

- **802.11a** (e.g.see [31])**:** Being the fastest flavor of 802.11, this extension defines requirements for a physical layer (which determines, among other parameters, the frequency of the signal and the modulation scheme to be used) operating in the Unlicensed National Information Infrastructure (UNII) band at 5 GHz and data rates ranging from 6 Mbps to 54 Mbps. The layer uses a scheme called orthogonal frequency division multiplexing (OFDM), which transmits data on multiple subcarriers within the communication channel. It is in many ways similar to the physical layer specification for HiperLAN/2.

- **802.11b** (e.g. see [32], [33] and [34]): Commercially trademarked in 1999 by the wireless ethernet compatibility alliance (WECA) as ireless fidelity (Wi-Fi), 802.11b defines operation in the ISM band at 5.5 Mbps and 11Mbps (as well as the fallback rates of 1 Mbps and 2 Mbps). This physical layer uses the modulation schemes complementary code keying (CCK) and packet binary convolutional coding (PBCC). WECA is an industry organization created to certify interoperability among 802.11b products from diverse manufacturers.

- **802.11c:** This task groups's work on wireless LAN bridging has been folded into the 802.1 standard.

- **802.11d:** This task group enhances the 802.11 specification by spelling out its operation in new regulatory domains, such as countries in the developing world. In its initial form, the standard covered operation only in North America, Europe, and Japan.

- **802.11e:** As 802.11 becomes popular, it is being used for real-time applications like voice and video. To ensure that these time-sensitive applications have the network resources they need when they need them, this task group is working on adding mechanisms for ensuring quality of service to MAC layer.

- **802.11f:** 802.11 implementations are evolving from small extensions of wired LANs into larger networks with multiple access points. These access points must communicate with one another to allow users to roam among them. This task group is busy with extensions that enable communication between access points from different vendors.

- **802.11g:** 802.11g is regarded as high speed extension to 802.11b. 802.11g adopts OFDM from 802.11a as well as two additional modulation schemes: PBCC and CCK-OFDM. It enables data rates as high as in 802.11a (54 Mbps) while operating in 2.4 GHZ ISM band.

- **802.11h:** This task group is working on modifications to the 802.11a physical layer to ensure that 802.11a may be used in Europe. The task group is adding dynamic frequency selection and transmit power control, which are required to meet regulations in Europe.
- **802.11i:** the original version of 802.11 incorporated a MAC-level privacy mechanism called wired equivalent privacy (WEP), which has proven inadequate in many situations. This task group is busy with improved security mechanisms. The current draft contains an improvement to WEP called temporal key integrity protocol (TKIP) and a new mode that incorporates the Advanced Encryption Standard (AES), which is widely used in the banking industry.

Table 2.5 gives a brief comparison of different 802.11 standards from physical layer point of view.

### 2.3.2 Point coordination Function (PCF)

The infrastructured mode of operation (PCF) of IEEE 802.11, as discussed earlier, is inherently suitable for real-time traffic (Fig. 2.7). Existence of a centralized controller (AP), makes the priority implementation much easier. In general, efforts towards real-time traffic support realization in PCF mode try to either favorably manipulate the CFP/CP structure, employ a more efficient polling mechanism in CFP, or schedule the real-time traffic itself. Examples of such efforts are [35], [36], and [37] while [38] proposes a new association procedure for multimedia applications. Finally, [17] and [39] give a thorough simulation examination of PCF as well as DCF.

Table 2.5: IEEE 802.11 WLAN Standards

| Standard | 802.11 | 802.11a | 802.11b | 802.11g |
|---|---|---|---|---|
| Spectrum | 2.4 GHz | 5.0 GHz | 2.4 GHz | 2.4 GHz |
| Maximum physical rate | 2 Mbps | 54 Mbps | 11 Mbps | 54 Mbps |
| Transmission | FHSS/DSSS | OFDM | DSSS | OFDM |
| Compatible with | None | None | 802.11 | 802.11/ 802.11b |
| Major disadvantage | Limited bit rate | Smallest range of all 802.11 standards | Bit rate too low for many emerging applications | Limited number of collocated WLANs, higher range than 802.11a |
| Major advantage(s) | Higher range | Higher bit rate in less crowded spectrum | Widely deployed; higher range | Higher bit rate in 2.4-GHz spectrum |

### 2.3.3 Performance Evaluation of Distributed Coordination Function (DCF)

Due to the employed random backoff algorithm and system complexity, most of the works toward system performance evaluation under DCF, resort to simulation ( [17], [40], and [41]). A very limited, (in terms of assumptions and obtaining boundary results), but interesting analytical study of DCF operation, in the presence of hidden terminal and capture effect has been performed in [18] and more thorough in [42]. The only two important and most referred analysis studies of DCF are [1], [43] and its continuation [44] and [45]. In [44], to simplify the protocol analysis, it is assumed that a tagged station for each transmission attempt uses a backoff interval sampled from a geometric distribution. This assumption on the backoff algorithm implies that the future behavior of the station does not depend on the past. Because of this basic assumption, dynamic behavior of the protocol is undermined while it may be useful in finding some boundary results.

The two-dimensional Markov Chain model used in [1] is reproduced in Fig. 2.8. Referring to the model, $W_i$ is the CW size at stage $i$ which there is called backoff stage. The key assumption in the model proposed in [1] is that, at each transmission attempt, and regardless of the number of retransmissions suffered, each packet collides with constant and independent probability $p$. Also channel conditions have been assumed to be ideal. The model though more realistic compared to [44], falls short in taking into account the real-life system parameters (such as limited retransmission and buffer) and accurately following the dynamic behavior of the backoff algorithm but maybe in asymptotic situations. Moreover, the model size (one of the dimusions) is proportional to the number of users in the system. This makes the system mathematically untractable for a large number of users.

Lack of a centralized control as in PCF makes real-time traffic realization over DCF much more challenging. One of the common strategies for applying priority over the DCF is to revise the protocol so that different classes of applications are assigned different initial window size and/or window increasing factor, and/or maximum backoff stage according to their priorities. References [46] and [47] present an analytical approach of the aforementioned strategy; the former employs the model in [1] and the latter uses our model that appears in Chapter 3. A simulation effort of the above strategy can be found in [48]. Another scheme is proposed in [49] which deploys the inter-frame spaces (IFS) in order to enforce priority. A completely different technique is brought up by [50] in which a new distributed multiple access scheme can be overlaid on an IEEE 802.11 implementation to fulfill delay requirements of real-time applications. According to [50], data stations access the channel as IEEE

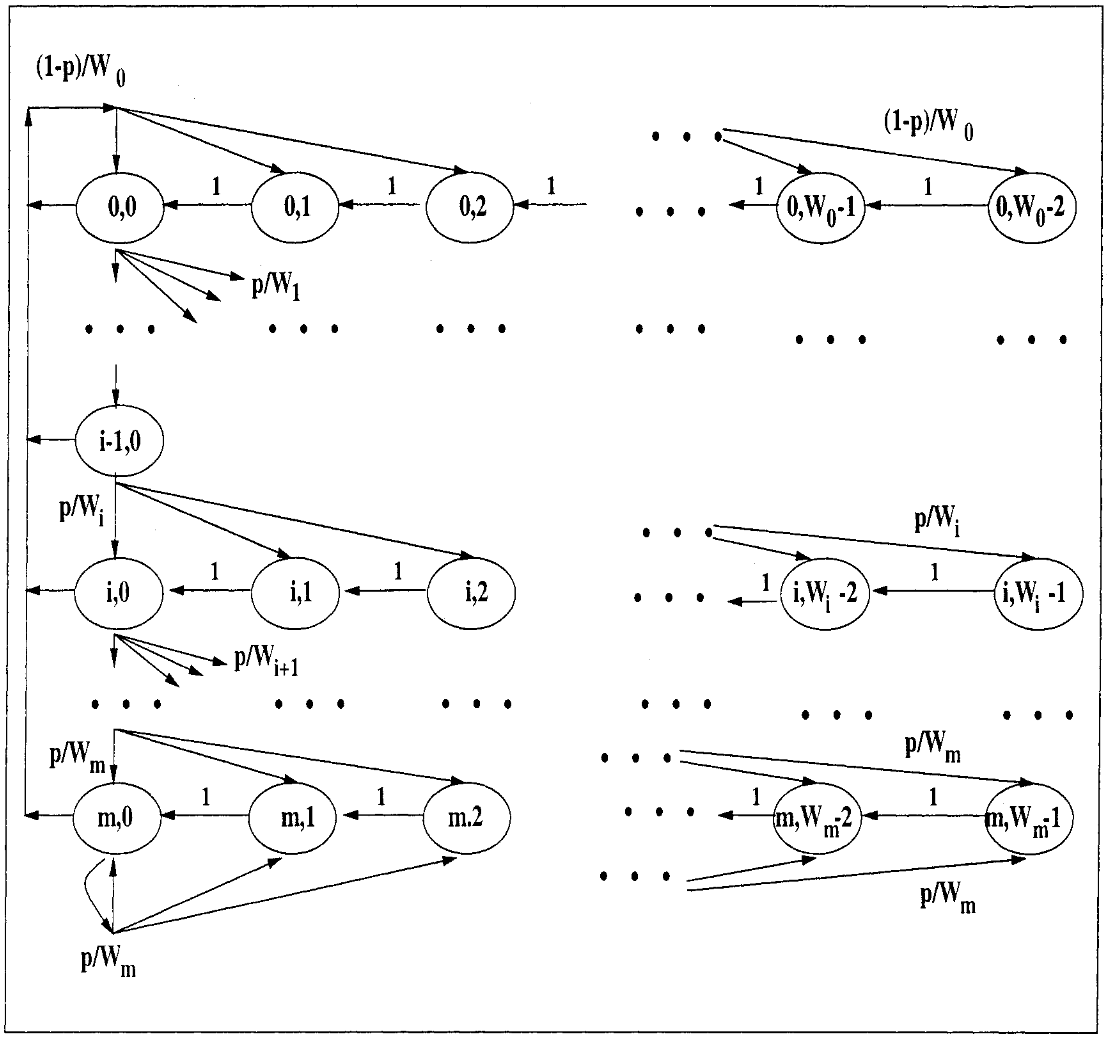


Figure 2.8: Markov Chain model for the backoff window size acquired from [1].

802.11 DCF. Regarding real-time applications (mainly voice is considered), the related station schedules its transmission at specific moments ($t_{sch}$ seconds apart). If after $t_{sch}$, the channel is perceived to be idle, it transmits its packet and if not, waits till the channel becomes idle and then enters a black burst contention period. At this time, the station jams the channel with a number of black slots(pulses of energy of prespecified duration). This number is proportional to the time that the station has been waiting for the channel to become idle (after passing access instant). After transmitting its black burst, the station listens to the channel to see if any other station is transmitting a longer black burst, implying that it would have been waiting longer for access to the channel. If the channel is perceived idle, then the station has an access instant and transmits a packet. Of course all these different actions take place with related interframe spaces which are not mentioned herein for the sake of brevity. By assuming shorter after-channel-idle waiting times, the procedure guarantees higher priority to real-time data over nonreal-time data and by introducing black bursts, delay priority is introduced among real-time packets themselves. This leads to a limited delay feature appealing to real-time applications. Fig. 2.9 shows how the protocol operates.

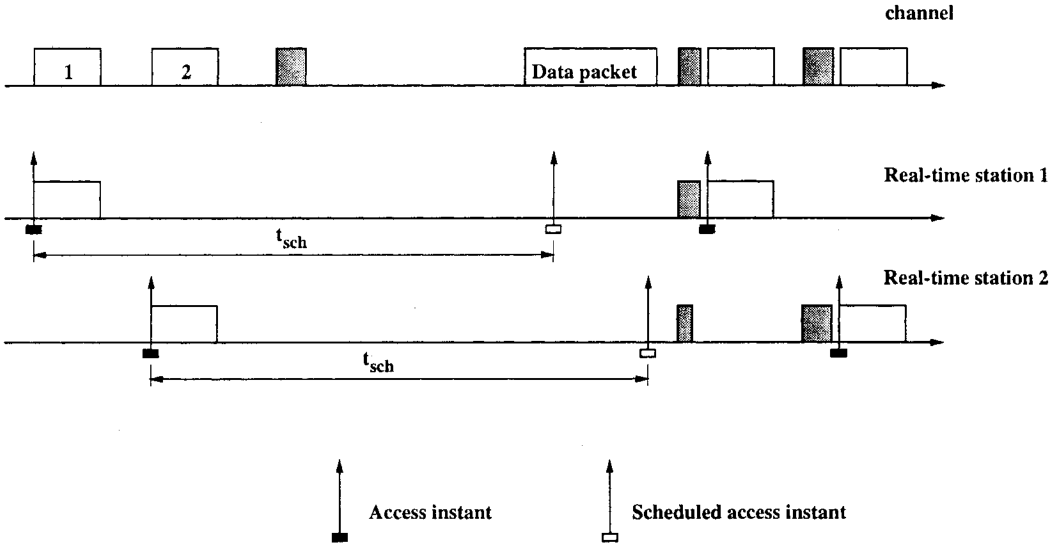


Figure 2.9: A time diagram of the access procedures of real-time stations.

### 2.3.3.1 Multi-hop Challenges

In a multi-hop ad-hoc network, nodes communicate with each other using several wireless links, and there is no fixed infrastructure such as a base station. Each node in the network also acts as a router, forwarding data packets for the other nodes. A central challenge in the design of multi-hop ad-hoc networks is the development of routing protocols that can efficiently find routes between two communication nodes. Several routing protocols have been proposed for MANETs, which can be classified into three broad categories: proactive table-based routing schemes, reactive on-demand source-based routing schemes, and constraint-based routing schemes. References [5], [51] and [52] present an almost complete review of different wireless routing protocols while [52] has a comparative simulation approach as well. There is another approach that utilizes location information (for instance, obtained using the global positioning system (GPS)) to improve performance of routing protocols for ad hoc networks (e.g. refer to [53] and [54]).

Simulation is the only means for performance evaluation of multi-hop ad-hoc networks due to the numerousness and complexity of variables involved. Reference [55] is an attempt to study the performance of the multi-hop ad hoc IEEE 802.11 WLAN in various mobility and traffic volumes. It proposes a piggyback reservation protocol in which the neighbors which hear the data packet are blocked and avoid colliding with returning ACK. Furthermore, they learn about the next packet transmission time. Likewise, the neighbors of the receiver which hear the ACK will avoid transmitting at the time when the receiver is scheduled to receive the next data packet. Another related work [56], compares two different feedback mechanisms, explicit (per hop ACK) versus end-to-end, in multi-hop scenarios. Another somehow compelling work regarding multi-hop ad-hoc operation is [57], where the concept of adaptive clustering is introduced. In the proposed architecture, nodes are organized into non-overlapping clusters. The clusters are independently controlled, and are dynamically reconfigured as nodes move. Adaptive clustering uses code separation among different clusters. However, the work is not particularly about IEEE 802.11 protocol.

Due to interaction between very many phenomena, there is still room for more involved and accurate simulation-type research endeavors regarding multi-hop ad hoc WLANs. In Chapter 4 we will present ours.

## 2.4 Summary

In this chapter we have tried to provide the reader with an insight regarding how wireless networks fit into the hybrid communications system. Different levels of wireless networks, WMAN, WLAN, WPAN, and WBAN, were differentiated and their common feature of unlicensed band operation was addressed. The two most representative of WLAN protocols, HiperLAN and IEEE 802.11 were briefly reviewed with more emphasis on medium access technique employed. This is done in the CAC sublayer in the case of HiperLAN, and in the MAC sublayer in the case of IEEE 802.11. Through a closer look into 802.11, different flavors of the standard were briefly explained. How PCF and DCF have been addressed in the literature were included afterwards. Different approaches toward the performance evaluation of DCF, analysis and simulation, were discussed giving the former more depth in serving as a background prior to chapter 3. Challenging aspects of the multi-hop scenario of the ad-hoc WLANs were mentioned along with the simulation-type past research work.

# Chapter 3

# Performance Analysis of Ad-Hoc IEEE 802.11 WLAN

## 3.1 Introduction

In order to optimize system parameters to fulfill specific needs, a mathematical description of the system turns to be a lot helpful in observing the trend of any parameter changes made. Though not always possible, reasonable assumptions and simplifications may lead to analysis models with the required accuracy. Efforts have been made to introduce a mathematical model which in addition to being simple, and efficient, represents the inherently complex behavior of the random access system. References [1], [43] and its continuation [44] and [45] are important works toward IEEE 802.11 system modeling (Chapter 2) which have adopted a different approach than what is presented here. Compared to the latter, our model enjoys more simplicity and captures the inherent dynamics of the system more efficiently.

In this chapter we will mathematically model the DCF mode of operation of

IEEE 802.11 WLAN. Our model is based on the presentation of the system with a pair of one-dimensional state diagrams which accommodate variations of many input parameters. Among all the parameters, those we pay special attention to are:

- *Packet Fragmentation Factor.* Represents how deep a long packet is divided to smaller parts with independent identities.
- *User Buffer Size.* The number of packets that a user can hold pending transmission.
- *Maximum Retransmission Time.* The number of times a failed transmission is repeated until it is discarded.

In addition, the impact of channel error on the system performance is also considered. The model enables us to make optimum choices for the above parameters in different wireless channel qualities represented by the average bit error rate (BER). System performance is evaluated through the three following measures; throughput, delay, and probability of fail to deliver of which the last two are crucial for real-time applications. A new fragment transmission is also introduced which differs from the standard in the way the channel is treated (released or kept) as a non-terminating-fragment of a particular packet is transmitted. The two transmission strategies are thereafter compared in terms of the measures of performance (MoP) above. Finally, to investigate the validity of the presented mathematical model, the MAC layer functionality is simulated and the results are compared with the analysis results.

In the first section, *System Analysis*, and following a brief review of the transmission mechanism in IEEE 802.11, a state-diagram model representing the system is introduced. This is the core of our mathematical modeling effort. The *Model Parameters* sub-section is dedicated to calculating the state-diagrams parameters using IEEE 802.11 transmission protocol and parameters. System performance is formulated in the *Performance* section which is followed by graphical presentations in a section titled *Numerical Results.* The new fragment transmission strategy and comparison to the standard is presented in section 3.5. Simulation investigation of the analysis results appears in section 3.5.1.

## 3.2 System Analysis

Following a brief review of packet transmission in the IEEE 802.11 protocol suite, the whole process will be mathematically modeled. In the proposed model, channel BER, packet fragmentation and retransmission, and limited buffer size of each user are also taken into account.

### 3.2.1 System Description

Random back-off algorithm is the contention resolution strategy in the IEEE 802.11 WLAN protocol. A typical station with a packet ready to send randomly picks a slot number out of CW with size of $h = 2^3 = CW_{min}$, after observing an idle channel

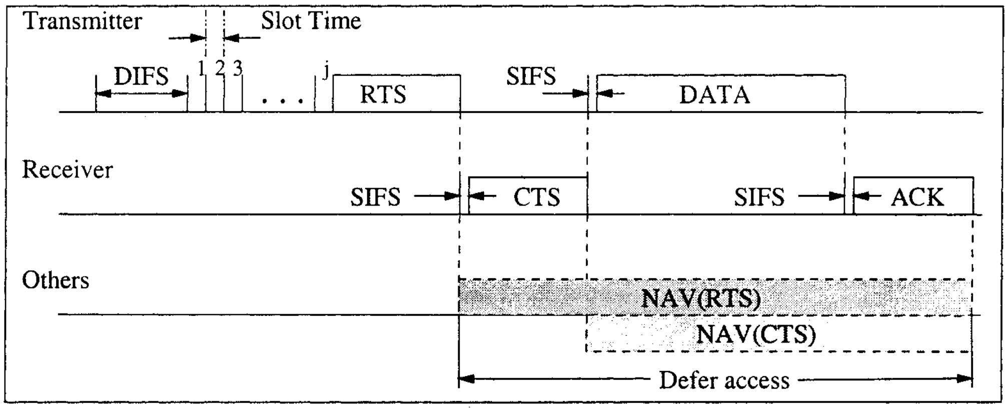


Figure 3.1: Successfully transmitted unicast data frame in RTS/CTS mode of the IEEE802.11 standard.

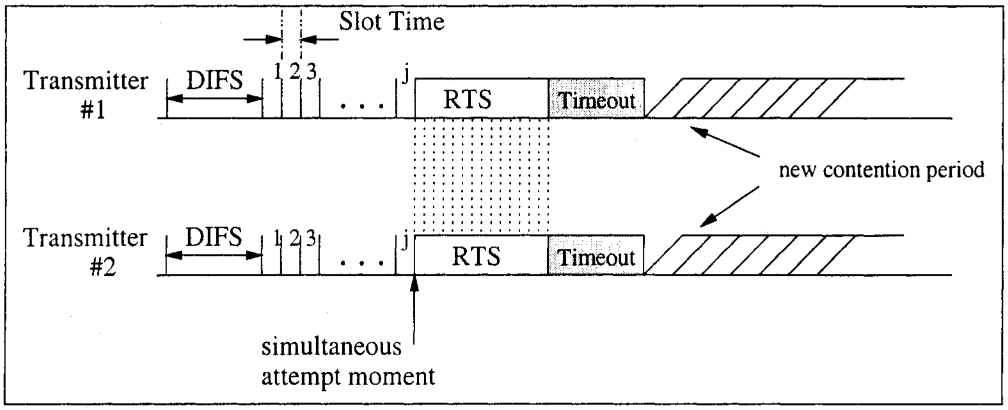


Figure 3.2: Simultaneous transmission attempt in the RTS/CTS mode leading to a collision.

for $T_{DIFS}$ seconds. The station then starts transmission in the picked slot (a slot is typically equal to $2T_{SIFS}$ where $T_{SIFS}$ is the shortest reference period defined in the protocol). Upon occurrence of a collision, stations involved will double their CW size up to a maximum of $h = 2^{3+5} = 2^8$. A station that has successfully transmitted a packet will return its $h$ to minimum value $CW_{min}$.

The key issues in studying the performance of random access systems are success and collision probabilities. Two parameters that quantitatively determine the contention resolution mechanism are contention window (CW) size and number of users involved in contention which will be discussed shortly after. Figs. 3.1 and 3.2 respectively show an individual successful transmission and collision and their time cost in an IEEE 802.11 ad-hoc WLAN using request-to-send/clear-to-send (RTS/CTS) dialogue. As opposed to a successful transmission, collision takes place when there is a simultaneous transmission attempt from at least two of the stations, $T_{DIFS} + jb$ seconds after the channel becomes idle where $j$ denotes the slot number in which the transmission attempt takes place and $b$ is the slot duration. A station that fails to receive a CTS at the end of $T_{Timeout}$ seconds following the transmission of an RTS, assumes that collision has occurred. Assuming that there are $u_a$ active (demanding channel access with packets ready to be sent) users in the system and all of them have the same contention window size (these assumptions are to be discussed later on), $h = 2^{\bar{i}+3}$ $(0 \leq \bar{i} \leq 5)$, for calculating the probabilities of successful and collided attempts initiated in the arbitrary $j$th slot we proceed as follows. Getting to slot $j$ means that there have been no transmission attempts in the previous $j-1$ slots (otherwise a successful or failed transmission would have taken place and the contention period would have been aborted at the previous slots. This also implies that all those not involved in an already initiated transmission would have backed off). This has been illustrated in Fig. 3.3. We define a random experiment as active users in the system selecting a random number $m$, $1 \leq m \leq h$, prior to entering contention period. So the set of all possible outcomes of the suchwise defined random experiment is $S = \left\{(u_{1m_1}, ..., u_{u_a m_{u_a}}) \;\; ; \;\; 1 \leq m_1, m_2, ..., m_{u_a} \leq h\right\}$ wherein $u_{nm}$ denotes the active user $n$ selecting random slot number $m$. This should be differentiated from the sequence of independent binomial trials where the experiment proceeds slot by slot with two possible outcomes of transmission attempt or not. The latter is the case with Ethernet-like random access techniques. Based on the aforementioned experiment, three following events are defined which will guide us toward sought success and collision probabilities..

- $s_j$ ={ Successful transmission in slot $j$ }. $s_j$ occurs when just one user has scheduled its transmission for the $j$th slot and the others have transmission

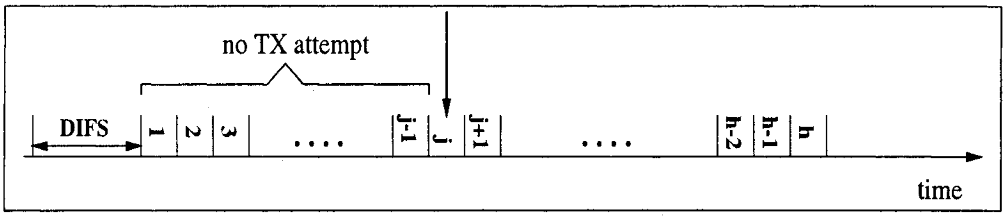


Figure 3.3: A slot can be reached if and only if there are no transmission attempts in prior slots.

intentions over the remaining $h - j$ slots. In advance transmission scheduling (before contention period starts) is once again emphasized. The above statement is mathematically expressed as:

$$s_j = \left\{ (u_{1m_1}, ..., u_{u_a m_{u_a}}) \quad ; \quad \underset{1 \leq n,n' \leq u_a}{\exists n \ \ \forall n'} \mid \quad m_n = j \quad \& \quad j+1 \leq \underset{n' \neq n}{m_{n'}} \leq h \right\}$$

where $n$ and $n^{'}$ are independent active user identification indexes.

- $col_j$ ={ Collision occurrence in slot $j$ }. $col_j$ occurs when there are at least two users which have scheduled their transmission for the $j$th slot and the others have transmission intentions over the remaining $h - j$ slots. The corresponding mathematical expression would then be:

$$col_j = \left\{ (u_{1m_1}, ..., u_{u_a m_{u_a}}) \quad ; \quad \underset{1 \leq n,n' \leq u_a}{\exists \ \ n, n'} \quad \mid \quad \underset{n \neq n'}{m_n = m_{n'}} = j \right\}$$

- $I_j$ ={ No transmission attempt through slot $j$ }. $I_j$'s occurrence means that all the active users have scheduled their transmission over the remaining $h - j$ slots to the end of the contention period. This is mathematically expressed as:

$$I_j = \left\{ (u_{1m_1}, ..., u_{u_a m_{u_a}}) \quad ; \quad j+1 \leq \underset{1 \leq n \leq u_a}{m_n} \leq h \right\}$$

Figure 3.4 is a Venn diagram representation of the universal set ($S$) and the events of interest on it ($s_j$, $col_j$, and $I_j$). From the definitions and Fig. 3.4 it is understood that:

$$\begin{aligned} &\emptyset = I_h \subset I_{h-1} \subset ....I_j \subset I_{j-1} \subset ...I_1 \subset I_0 = S \\ &I_{j-1} = I_j \cup s_j \cup col_j \qquad \& \qquad I_j \cap s_j \cap col_j = \emptyset \end{aligned} \tag{3.1}$$

The second line of (3.1) suggests that $I_{j-1}$ is partitioned by mutually exclusive events $I_j$, $s_j$, and $col_j$. Denoting the corresponding probabilities by $P_{I_{j-1}}$,$P_{I_j}$,$P_{s_j}$, and $P_{col_j}$, we have:

$$P_{I_{j-1}} = P_{I_j} + P_{s_j} + P_{col_j} \tag{3.2}$$

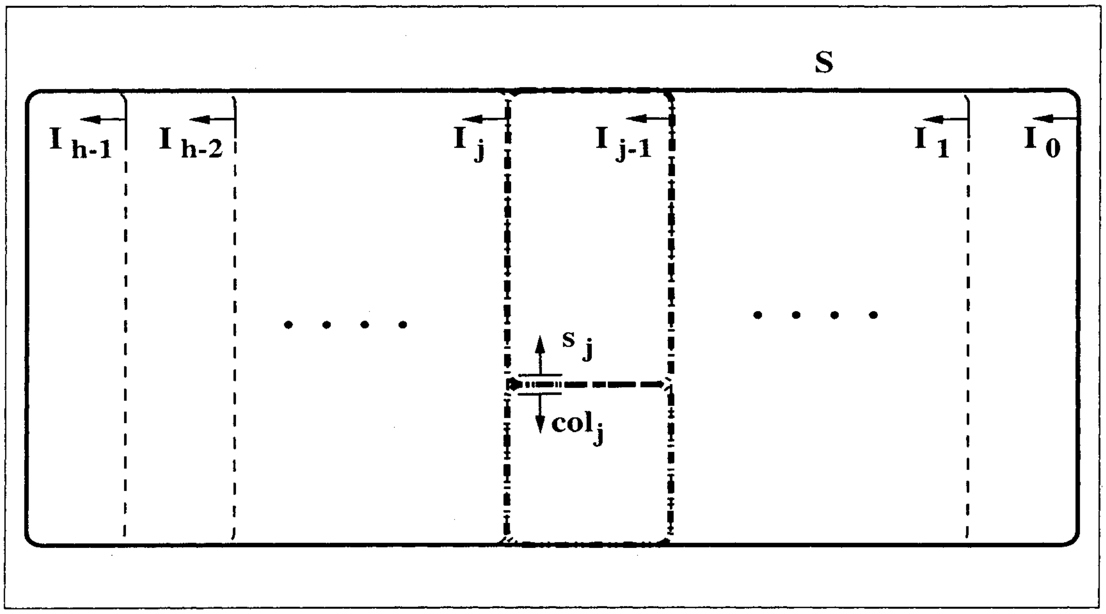


Figure 3.4: Venn diagram representing the relation between successful, collided and idle slots events.

Probabilities in (3.2) can be directly formulated using the definition of the corresponding events presented earlier as follows:

$$\begin{aligned} P_{s_j} &= u_a.(\frac{1}{h}).(\frac{h-j}{h})^{u_a-1} \\ P_{col_j} &= \sum_{k=2}^{u_a} \binom{u_a}{k}.(\frac{1}{h})^k.(\frac{h-j}{h})^{u_a-k} \\ P_{I_j} &= (\frac{h-j}{h})^{u_a} \end{aligned} \tag{3.3}$$

The probabilities in (3.3) can be verified using (3.2) as done below:

$$\begin{aligned}
P_{s_j} + P_{col_j} + P_{I_j} &= u_a.(\frac{1}{h}).(\frac{h-j}{h})^{u_a-1} + \sum_{k=2}^{u_a} \binom{u_a}{k}.(\frac{1}{h})^k.(\frac{h-j}{h})^{u_a-k} \\
&+ (\frac{h-j}{h})^{u_a} \\
&= \sum_{k=1}^{u_a} \binom{u_a}{k}.(\frac{1}{h})^k.(\frac{h-j}{h})^{u_a-k} + (\frac{h-j}{h})^{u_a} \\
&= \left[(\frac{h+1-j}{h})^{u_a} - (\frac{h-j}{h})^{u_a}\right] + (\frac{h-j}{h})^{u_a} \\
&= (\frac{h+1-j}{h})^{u_a} \\
&= P_{I_{j-1}}
\end{aligned} \tag{3.4}$$

However, from this point on we are not interested in $P_{I_j}$s and continue with the other two ($P_{s_j}$ and $P_{col_j}$).

The time costs corresponding to the events $s_j$ and $col_j$ are:

$$\begin{aligned}
T_{s_j} &= T_{DATA} + T_{RTS} + T_{CTS} + T_{ACK} + T_{DIFS} + 3T_{SIFS} + jb \\
T_{col_j} &= T_{RTS} + T_{DIFS} + T_{Timeout} + jb
\end{aligned} \tag{3.5}$$

where $T_{DATA}$, $T_{RTS}$, $T_{CTS}$, $T_{ACK}$, $T_{DIFS}$, $T_{SIFS}$, and $T_{Timeout}$ are respectively the data packet, request to send packet, clear to send packet, acknowledgment packet, channel idle, short inter-frame, and collision recovery timeout periods.

Transmission on unreliable channels necessitates partitioning of longer packets into smaller ones. Figure 3.5 illustrates how the RTS/CTS channel reservation scheme is combined with fragmentation according to the IEEE 802.11 protocol suite. Once a station contends for the channel successfully, it shall continue to send all the fragments ($a$ in Fig. 3.5) belonging to a specific packet before releasing the channel. Also an unacknowledged fragment leads to a channel release and the subject station should contend for the channel to resume transmission. A complete and detailed discussion of IEEE 802.11 can be found in [58].

The flowchart in Fig. 3.6-a provides a useful description of the transmission mechanism in IEEE 802.11 accommodating access, back-off procedure, and fragmentation. The flowchart description helps in developing the proposed state-diagram model that will follow. Stations try to transmit following a $T_{DIFS}$ period during which the medium is determined to be idle, or following a $T_{EIFS}$ period during which the medium is determined to be idle subsequent to a frame that was not received correctly. Following a transmission attempt, unsuccessful RTS/CTS or frame/acknowledgment (FR/ACK) dialogue (due to collision or channel-caused erroneous reception) initiates a new contention process with a doubled CW size ($CW = 2CW_{old}$ where $CW_{old}$ denotes the most recent CW size). In the case of RTS (FR) in error, by not receiving any response in CTSTimeout (ACKTimeout), the transmitter concludes that RTS (FR) did not get through successfully and shall invoke its back-off procedure ( [58]). This means that erroneous RTS (FR) and CTS (ACK) impose different timing cost on the system. However, since CTSTimeout/ACKTimeout is in the order of CTS/ACK time length, equal cost assumption does not change the analysis results much, and the flowchart of Fig. 3.6-a reflects this assumption. Any given unacknowledged fragment

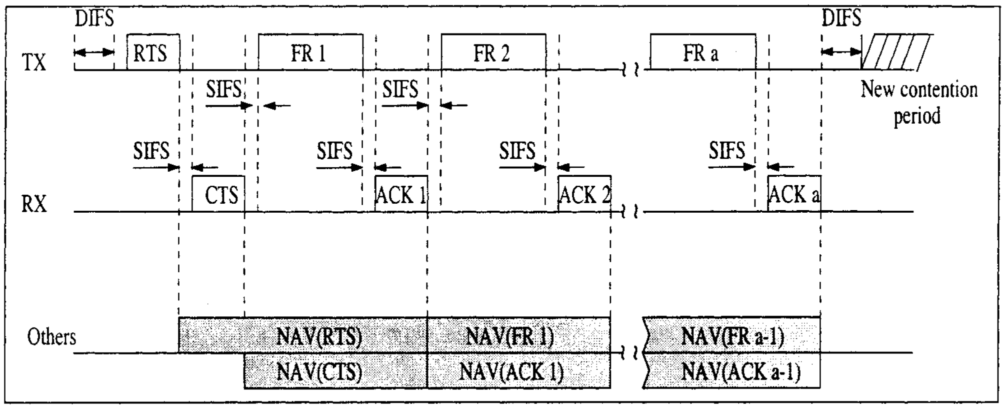


Figure 3.5: RTS/CTS with fragmentation.

is retransmitted within retransmission limit constraint beyond which, it is discarded and CW is reset to $CW_{min}$ (not shown in the flowchart of Fig. 3.6-a). Following a successful fragment burst (the process of sending multiple fragments after contending for the channel is defined as a fragment burst), a new contention process with CW size reset ($CW = CW_{min}$) is initiated. Stations not involved in the transmission process (either failed or successful ones) keep counting down their back-off counters during idle periods of channel greater than $T_{DIFS}$ rather than initiating a new back-off algorithm. The down counting is frozen during busy channel situations. The station will attempt transmission once its counter reaches zero. Real-time traffic, depending on the application, requires a compromise between delay and loss. Late fragments are worth not more than lost ones. To take care of this, and as an amendment to the protocol toward real-time adaptability, unacknowledged fragments are retransmitted for a limited number of times ($R_{max}$) before releasing the channel. Moreover, $R_{max}$ unsuccessful retransmissions are followed by fragment drop to avoid too much latency of the late fragments (Fig. 3.6-b). In both cases: the standard and the amendment presented here, a fragment drop resets the CW size. It should be noted, however, that the state-diagram analysis proposed in the subsequent sections is equally applicable to the original standard (Fig. 3.6-a). In the latter, any variation in the structure of the flowchart would merely change the departure/arrival rates of the state diagrams.

### 3.2.2 System Model

Following an idle channel for $T_{DIFS}$ seconds, only stations with a history of transmission attempt (successful or not) in the previous contention period will invoke back-off algorithm and other active users will continue counting down their back-off counter. Among the first group, given the differing start times and different history of fail/success, it is understood that different users do not share the same CW at any

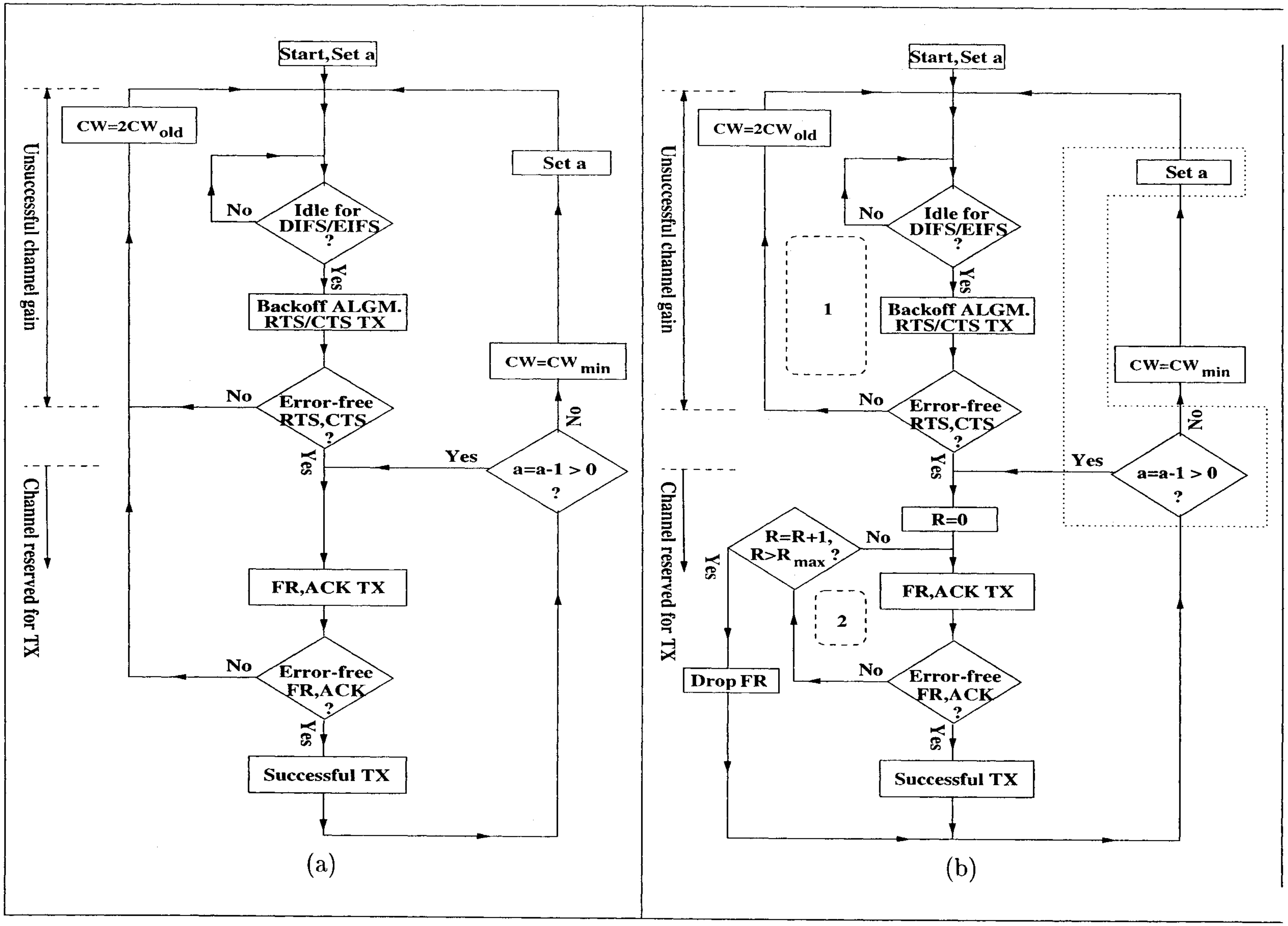


Figure 3.6: Flowchart illustrating the transmission procedure regarding an individual user. a) IEEE 802.11 protocol suite. b) Partially revised to support real-time traffic.

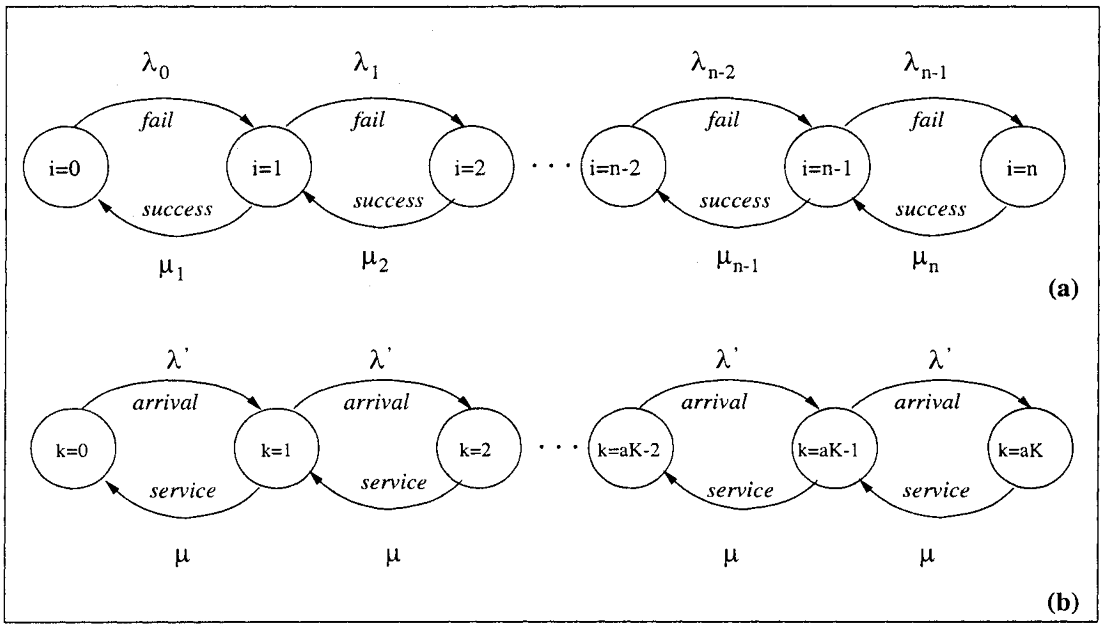


Figure 3.7: State-diagram pair presentation of the system. a) State-diagram with the CW size measure, $i$, as state variable. b) State-diagram with individual user buffer occupancy, $k$, as state-variable.

time. Besides, the number of channel-demanding users fluctuates from one contention period to the other.

The main objective of the proposed model should be to capture the dynamics of the random access technique deployed in 802.11. So any assumptions made, though unavoidable, should not undermine the dynamics of the access technique. In what follows, the key assumptions basic to the model are mentioned.

If one is interested in a microscopic look at the joint-state of all users, simulation studies will be the natural resort due to the complexity of the corresponding classic multi-dimensional Markov chain analysis. However, when homogeneity prevails, the average performance of one user will be reflective of the conditions of all users. Accordingly, for analysis convenience we take a typical user's CW size as representative of all users. In other words, we assume that success and failure in transmission would affect all the stations present in the network equally and in an average sense. In view of this, since upon a successful transmission, just one station is resetting its CW to $CW_{min}$, it is assumed that all stations will shrink their CW by a factor of two rather than one resetting its CW to minimum value which corresponds to a worst case assumption considering performance. On the other hand, since at least two stations are involved in a collision, it is assumed that all stations will enlarge their CW size by a factor of two which shows the overall network as compared to the standard, and results in worse performance (in fact it is like assuming all stations collide). So the transmission procedure regarding a typical user in our analysis is the same as the flowchart in Fig. 3.6-b with $CW_{min}$ replaced by $\frac{CW_{old}}{2}$. Regarding other issues, it is assumed that all users can hear each other which excludes hidden terminal problem. However, hidden terminal effect can be regarded in the model as a multiplicative factor degrading the successful channel contention rate. The wireless channel has been modeled as a time-invariant channel with fixed BER and non-selective frequency features. Propagation, packetization, and higher-layer (higher than MAC) delays are ignored.

Based on the above assumptions, our goal is to approximate the system with an equivalent random access network with a given number of active stations ($u_a$) which contend for the channel, given the same CW size ($h = 2^{\bar{i}+3}$) seen by all users (active and idle). Again implying another worst case scenario, since in reality newly active users will take a smaller window, $CW_{min}$. The ($u_a, \bar{i}$) pair which represents the system in an average sense and at all times is used for later analysis of the system performance.

To do so, we consider a simultaneous pair of one dimensional state-diagrams shown in Fig. 3.7 as descriptive of the system. To reduce the complexity, both state-diagrams are assumed to obey limited-buffer M/M/1 queuing discipline. The transition probabilities of this state-diagram pair are extracted from Fig. 3.6-b. The first state diagram (Fig. 3.7-a) is set up around a common CW size measure, $i$, while considering the number of active users in the system fixed at $u_a$. This state-diagram describes the contention process of all active users in the network (since only active users participate in the contention process). Upon failure to gain channel (loop number 1 in Fig. 3.6-b) with any number of users involved, all stations (both active and inactive, not just those involved in collision or failure) double their CW size (to reflect the increase in the system's common equivalent value of $i$ or typical user's CW size) for the next transmission attempt and similarly upon successful contention, all stations halve their CW size. A successful contention by any user leads to departure (successful transmission or drop) of $a$ fragments. The second state diagram (Fig. 3.7-b) represents the buffer occupancy of each individual user. Buffer length is fixed in units of long packet ($K$) and varies in units of fragment ($aK$) by changing the fragmentation factor $a$. In any case, buffer is filled and emptied in fragment units. New generated fragments arrive at each station at the rate of $\lambda'$ fragments per second , and fragments depart at a service rate delivered based on information from the first state-diagram. It is important to notice that failure and success in the first state-diagram is network-based (failure to gain channel by any number of users and successful transmission by any user) and is packet oriented, while arrival and service in the second state-diagram are per user and fragment-oriented. Simultaneous results from the above two result in a unique $u_a$ (and consequently $\bar{i}$) which is sufficient for deriving the various performance criteria. Related rates are to be calculated in the next section.

### 3.2.3 Model Parameters

#### 3.2.3.1 First State-Diagram (Network State)

Figure 3.7-a shows the state diagram representing the CW size as state variable and how it varies with collision and success rates ($\lambda_i$ and $\mu_i$). Referring to the flowchart in Fig. 3.6-b, the following events lead to an unsuccessful channel access:

collision, RTS in error, and CTS in error. Probability of an unsuccessful channel access initiated in the $jth$ slot, $P_{fail_j}$, and its corresponding cost of channel time averaged over aforementioned three events, $T_{fail_j}$, are:

$$\begin{aligned} P_{fail_j} &= P_{col_j} + P_{s_j}(1-\alpha) \\ T_{fail_j} &= x + jb \end{aligned} \tag{3.6}$$

where $x = T_{RTS}+T_{CTS}+T_{DIFS}+T_{SIFS}$, and $\alpha = (1-P_b)^{L_{RTS}+L_{CTS}}$ with $P_b$ denoting the channels BER, and $L_{RTS}$ and $L_{CTS}$ the number of bits in RTS and CTS packets of Fig. 3.1. $P_{col_j}$ and $P_{s_j}$ are already calculated in (3.3). From (3.6) and on, it is implied that RTS, CTS, and ACK contain PHY overhead. At state $i$, in an average sense, the failed contention rate $\lambda_i$ in Fig. 3.7-a can be written as the ratio of the corresponding probability of failure to its time cost. These two can easily be calculated from (3.6) by averaging over all possible values of slot number $j$ which depends on the CW size state ($i$). So we would have :

$$\lambda_i = \frac{P_{fail}}{\sum_{j=1}^{h} \frac{P_{fail_j}}{P_{fail}} T_{fail_j}} = \frac{P_{fail}^2}{\sum_{j=1}^{h} P_{fail_j} T_{fail_j}} \tag{3.7}$$

$P_{fail} = 1 - P_s\alpha$ is the total probability of a failed channel reservation with $P_s = \sum_{j=1}^{h} P_{s_j}$, and $P_{s_j}$ defined in (3.3). To be noted, all $P_{fail}$, $T_{fail_j}$,....,$P_{s_j}$,$P_s$, etc. are functions of $i$, but this $i$ is dropped for convenience of notation.

CW shrinkage happens when a succession of independent events, i.e. successful channel gain, successful RTS/CTS dialogue, and transmission (successful or unsuccessful) of the $a$ FR/ACKs take place. FR/ACK transmission process may also include retransmissions. Following a successful channel access with probability $P_{success} = P_s\alpha$, both successful transmission and drop (Fig. 3.6-b) lead to a CW shrinkage. However, depending on the number of retransmissions that occurred (which corresponds to the number of times the loop number 2 in Fig. 3.6-b is traced), time cost imposed on the system would be different. The average cost of a successful transmission/drop (a single fragment), not including the channel reservation time cost is found to be:

$$\begin{aligned} C &= \sum_{R=0}^{R_{max}} \left[(1-\beta)^R \beta (1+R) y\right] + (1-\beta)^{R_{max}+1} y (R_{max}+1) \\ &= \frac{y}{\beta}\left[-(R_{max}+1)(1-\beta)^{R_{max}+1}\beta + 1 + (1-\beta)^{R_{max}+2}\right] \end{aligned} \tag{3.8}$$

where the summation accounts for successful transmission following $R$ retransmissions ($R+1$ times looping of loop number 2 in Fig. 3.6-b), and the second term corresponds to drop following $R_{max}$ retransmissions ($R_{max}+1$ times looping of loop number 2 in Fig. 3.6-b). The parameters used above are; $\beta = (1-P_b)^{L_{FR}+L_{ACK}}$, and $y = T_{FR} + T_{ACK} + 2T_{SIFS}$ with $T_{FR} = T_{LDATA}/a + T_{OH(MAC)} + T_{OH(PHY)}$ in which $T_{LDATA}$ is the IEEE 802.11 maximum length data packet duration, and $T_{OH(MAC)}$ and $T_{OH(PHY)}$ are overheads associated with MAC and PHY layers respectively. Including the channel access time, the average CW shrinkage time cost initiated in the $j$th slot is $T_{success_j} = aC + x + jb$. The corresponding occurrence probability is $P_{success_j} = P_{s_j}\alpha$. Following the same reasoning above, the average CW shrinkage rate, $\mu_i$, would be:

$$\mu_i = \frac{P_{success}}{\sum_{j=1}^{h} \frac{P_{success_j}}{P_{success}} T_{success_j}} = \frac{P_s^2 \alpha}{\sum_{j=1}^{h} P_{s_j} T_{success_j}} \tag{3.9}$$

The average CW size measure, $\bar{i}$, is readily available through the solution of the state-dependent classic M/M/1/n queuing model ( [59]), i.e.:

$$\bar{i} = \sum_{i=0}^{n} i P_i \quad ; \quad P_i = \prod_{k=1}^{i} \frac{\lambda_{k-1}}{\mu_k} P_0 \quad ; \quad P_0 = \frac{1}{1 + \sum_{i=1}^{n} \prod_{k=1}^{i} \frac{\lambda_{k-1}}{\mu_k}} \tag{3.10}$$

#### 3.2.3.2 Second State-Diagram (User Buffer Occupancy State)

Figure 3.7-b shows the state diagram representing the user buffer occupancy as state variable and how it varies with new arrival and departure rates ($\lambda'$ and $\mu$). Referring to Fig. 3.6-a, packet departure process breaks down as an unlimited number of collisions followed by a successful channel access, and transmission of $a$ fragments each of which is retransmitted $R \leq R_{max}$ times if in error. This is shown in Fig. 3.8.

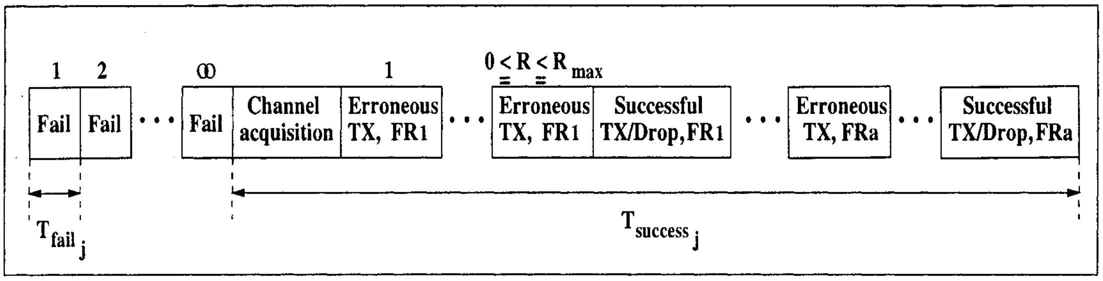


Figure 3.8: Typical packet (fragment) departure process.

Using the results of the first state-diagram, the total packet transmission time is (which partially comprises dropped fragments):

$$
\begin{aligned}
T_d &= \sum_j \left[ \frac{P_{fail_j}}{P_{fail}} T_{fail_j} \sum_{l=0}^{\infty} \left[ P_{fail}^{l} (1 - P_{fail}) l \right] + \frac{P_{success_j}}{P_{success}} T_{success_j} \right] \\
&= \frac{1}{P_{success}} \left[ \sum_j P_{fail_j} T_{fail_j} + P_{success_j} T_{success_j} \right] \qquad (3.11) \\
&= \frac{(1 - P_{success})^2}{P_{success} \lambda_{\bar{i}}} + \frac{P_{success}}{\mu_{\bar{i}}}
\end{aligned}
$$

in which $T_{fail_j}$, unsuccessful channel contention period initiated in the $jth$ slot, has been averaged over slot number $j$ and repeated unlimited times with its corresponding probability, and $T_{success_j}$ just averaged over $j$. In terms of time cost, channel acquisition (successful channel contention that was initiated in the $j$th slot) equals $T_{fail_j}$. In the final step of (3.11), (3.7) and (3.9) are invoked with $i = \bar{i}$ from the first state diagram. $T_d$ actually reflects how long a packet transmission (maybe partially successful) takes to terminate, viewing the whole network. So, it can be easily concluded that an individual user is provided with the fragment service capacity rate ($\mu$ in Fig. 3.7-b) of:

$$\mu = \frac{1}{a u_a T_d} \tag{3.12}$$

in which $u_a$ denotes the average number of the users to share the channel (have fragments in their buffer to send). This service rate does not depend on the user buffer occupancy and thus is state-independent. Denoting the total network packet generation rate in terms of long packets per second with $\lambda$ (assumed to be fixed) and assuming even distribution of new traffic among users, the new fragment arrival rate per user (active or not) in Fig. 3.7-b would be $\lambda' = \frac{a\lambda}{u}$ where $u$ is the total number of users. Consequently, the probability distribution function of the buffer occupancy of each individual user in the second state-diagram is:

$$P_k' = \frac{1 - a\lambda/u\mu}{1 - (a\lambda/u\mu)^{aK+1}} \left(\frac{a\lambda}{u\mu}\right)^k \quad ; \quad 0 \leq k \leq aK \tag{3.13}$$

Accordingly, $P_0'$ is the percentage of the time that a user has an empty buffer and so is not considered an active user. Changing the above time perspective into a quantitative view, one may say, out of the pool of $u$ users sharing the same channel, $u(1 - P_0')$ are active users with queue occupancy greater than or equal to one. So we can write:

$$u_a = u(1 - P_0') \tag{3.14}$$

which should be in conformity with the value of $u_a$ obtained from the first state-diagram. This leads us to a unique $(u_a, \bar{i})$ representing the system state in an average sense. In general, this average representation varies with different choices of

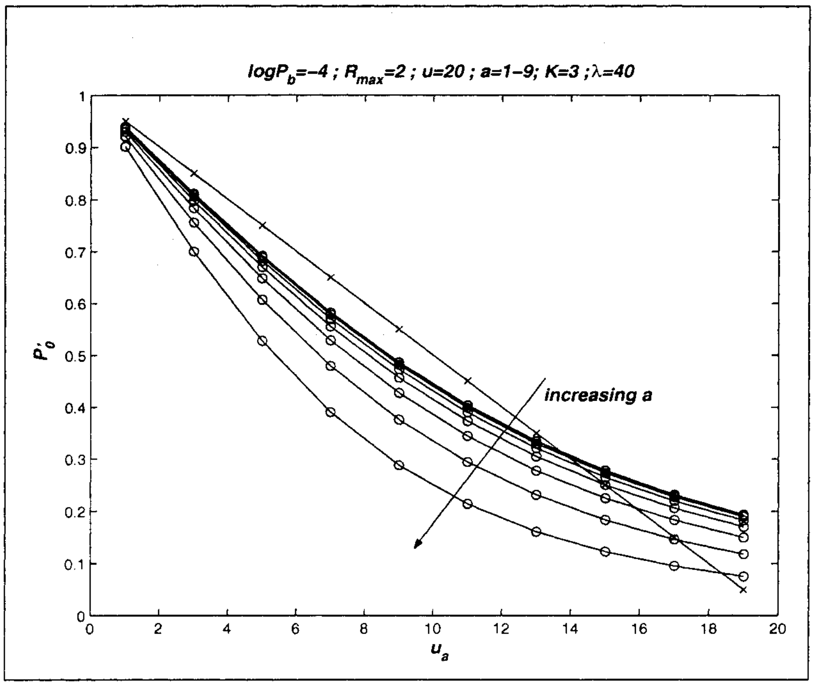


Figure 3.9: Probability of an empty queue for a representative user, $P_0'$, vs. number of active users in the system, $u_a$.

$(a,P_b,R_{max},K)$. To find this unique $(u_a, \bar{\imath})$, one can assume a starting value for $u_a$ and then solve for $\bar{\imath}$ using (3.3)- (3.10). Calculated $\bar{\imath}$ and assumed $u_a$ can give us $P_0'$ through applying (3.11)- (3.13) which in turn leads to a new value for $u_a$ in (3.14). Such iterations will continue until we get a unique $(u_a, \bar{\imath})$ pair. This approach may not converge depending on the selection of initial value for $u_a$. Since only integer rather than precise values of $u_a$ are searched for, another approach is adopted which is based on a graphical solution. As in the first part above, $P_0'$ is calculated and plotted for all values of $u_a$ in the range of $1 \leq u_a \leq u$ (curves with circle marks in Fig. 3.9). Then $P_0'$ is sketched versus $u_a$ based on (3.14) (linear part in Fig. 3.9), superimposed on the previous plot. The intersection of two plots provide us with the intended $u_a$ and subsequently $\bar{\imath}$ from (3.3)- (3.10). Figure 3.9 illustrates the whole procedure for selected values of parameters and different fragmentation factors. Such a figure is repeated for all combination of parameter values discussed in section 3.4. Finally, it should be added that for the specific case when $u_a = 0$, performance measures become meaningless. To avoid this, we have always forced $u_a = 1$ in those occasions.

## 3.3 Performance

Since the second state-diagram shown in Fig. 3.7-b is our basic model for the user performance evaluation, all the analysis presented is in units of fragments. However, by resorting to normalization, results are not length-dependent.

Aiming at real-time applications, we measure the system performance by the following three parameters:

- **Throughput :** By looking at the channel we calculate the period it takes a fragment transmission to be completed successfully. $T_d$ in (3.11) denotes a single total packet transmission time. Thus, the fragment transmission time is $\frac{T_d}{a}$. To regard only successful transmissions, fragments dropped are to be excluded by dividing $\frac{T_d}{a}$ by $(1 - P_{drop})$ where (Fig. 3.6-b):

$$P_{drop} = (1 - \beta)^{R_{max}+1} \tag{3.15}$$

  So the total throughput (considering the effect of both overhead and multiple access) would be the ratio of the single access transmission time of the data part of the fragment, $\frac{T_{LDATA}}{a}$, to the multiple access fragment transmission time as calculated above:

$$\eta = \frac{(1 - P_{drop})T_{L_{DATA}}}{T_d} \tag{3.16}$$

- **Delay:** Regarding delay, the queuing model of Fig. 3.10 illustrating a single user's buffer proves helpful. This model is equivalent to the second state-diagram in Fig. 3.7-b from a delay point of view. In Fig. 3.10, each buffer cell stores fragments. Average delay makes sense only for successfully delivered packets or carried traffic in queuing terminology. Blocked and dropped fragments that are not delivered have infinite delays which are excluded from delay calculation. However, dropped packets affect the average delay pertaining to delivered ones through decreasing $\mu$. On the other hand, in a system which features blocking (as our M/M/1/aK system herein), only part of the arriving traffic would enter the system. Denoting the blocking probability by $P_{block}$, the actual arrival rate into an individual user's buffer is scaled down to $a\lambda(1-P_{block})/u$. $P_{block}$ equals the probability of a full buffer ($k = aK$) in (3.13) and would be:

$$P_{block} = \frac{1-\lambda/u\mu}{1-(\lambda/u\mu)^{aK+1}}\left(\frac{a\lambda}{u\mu}\right)^{aK} \tag{3.17}$$

Little's formula relates the average number of customers to the average delay in the system. This average delay includes both queuing and service times:

$$D = \frac{r}{L_{DATA}}\left[\frac{\overline{k}u}{(1-P_{block})\lambda}\right] \tag{3.18}$$

where $\overline{k} = \sum kP_k^{'}$ is calculated using (3.13). The multiplicative term serves as a normalization with respect to the transmission time of data part of a fragment

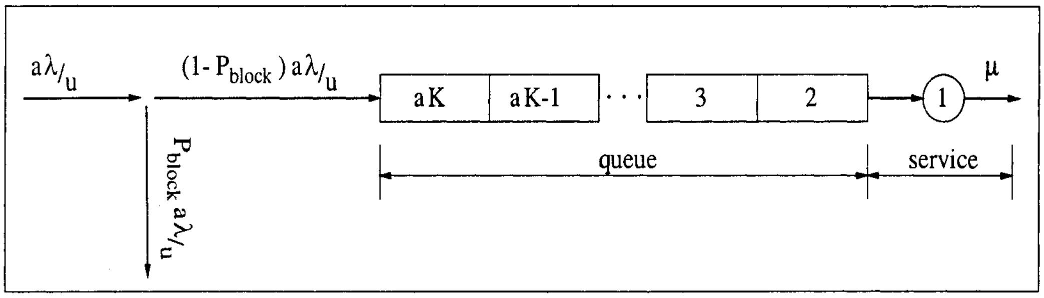


Figure 3.10: A single user's queue.

in a single access environment with $r$ denoting the IEEE 802.11 transmission bit rate. Finally, it is worth mentioning that the delay as defined above corresponds to the total time a fragment spends in the system. In this way there is no distinction between dropped or successfully transmitted fragments. This distinction is made through another performance measure, probability of fail to deliver, as discussed below.

- **Probability of Fail to Deliver:** Failure to deliver the packets (fragments) at the receiver could be due to either blocking because of full buffer or drop because of $R_{max}$ violation. This probability is as follows:

$$P_f = P_{block} + (1 - P_{block})P_{drop} \tag{3.19}$$

$P_{block}$ and $P_{drop}$ in (3.19) are substituted from (3.17) and (3.15) respectively.

## 3.4 Numerical Results

This section is dedicated to graphical presentation of the mathematical expressions of MoPs derived earlier.

Parameters including different packet and overhead lengths, transmission bit rate, etc., used in the earlier sections' results are presented in Table 3.1 (extracted from [58]). The total number of users present in the system, not necessarily demanding channel, is assumed to be $u = 20$ to represent an almost realistic number in a typical WLAN cluster coverage area. With $r = 1$ Mbit/s and referring to this table, the maximum long packet transmission rate on the channel is around 50 packets/s. To better show the effects of some specific factors like maximum allowable number of retransmissions and buffer size, and for the sake of brevity, only results for $\lambda = 40$

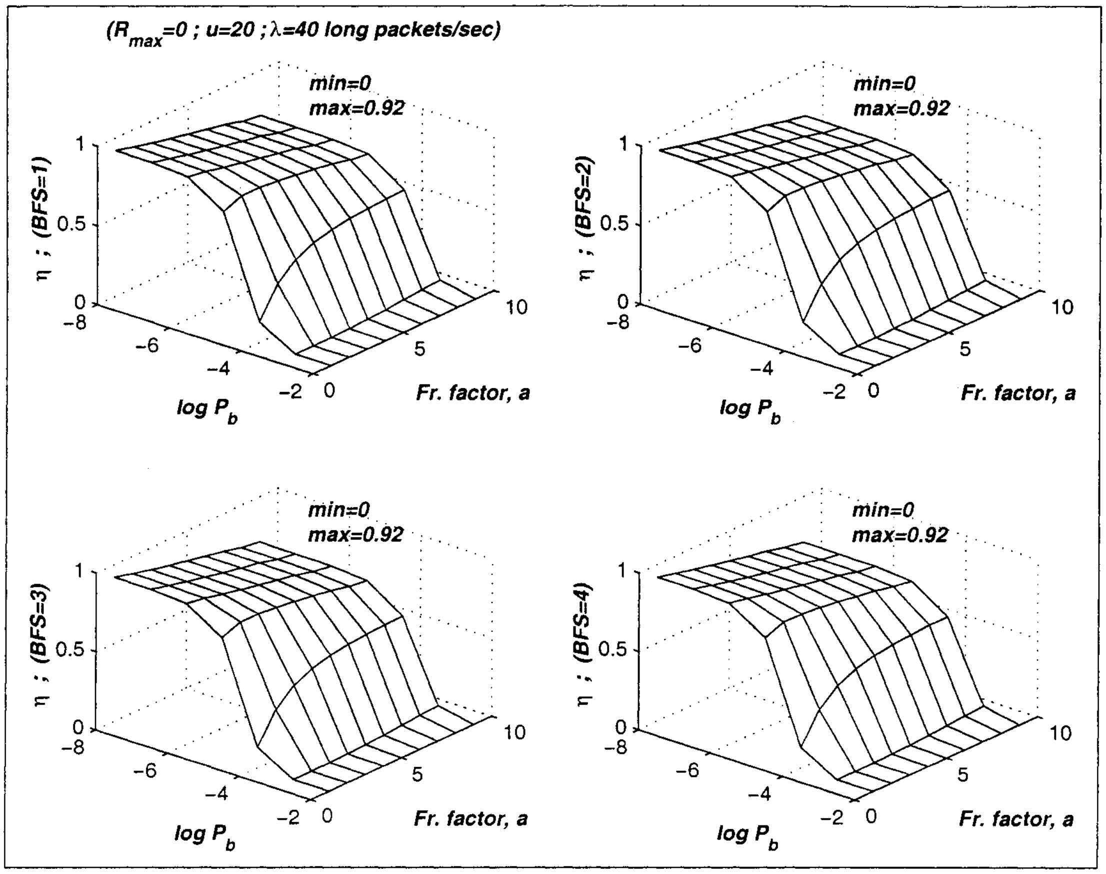


Figure 3.11: Throughput vs. BER and fragmentation factor; $R_{max} = 0$.

packets/s, corresponding to a very high-demand situation are reported. So in interpreting the numerical results, specifically in the context of real-time applications, the latter should be considered.

In the first step, (3.7)- (3.13) provide a relation between $u_a$ and $P_0^{'}$. The second such a relation is acquired from (3.14). The two relations simultaneously equip us with a unique $u_a$ which from now on represents the equivalent access demanding population (active users) in the system. In the second step, relevant performance measures are calculated using (3.15)- (3.19). For each different choice of ($a$,$P_b$,$R_{max}$,$K$), the above calculations are repeated.

Performance measures are illustrated in Figs. 3.11- 3.16. Each measure has been

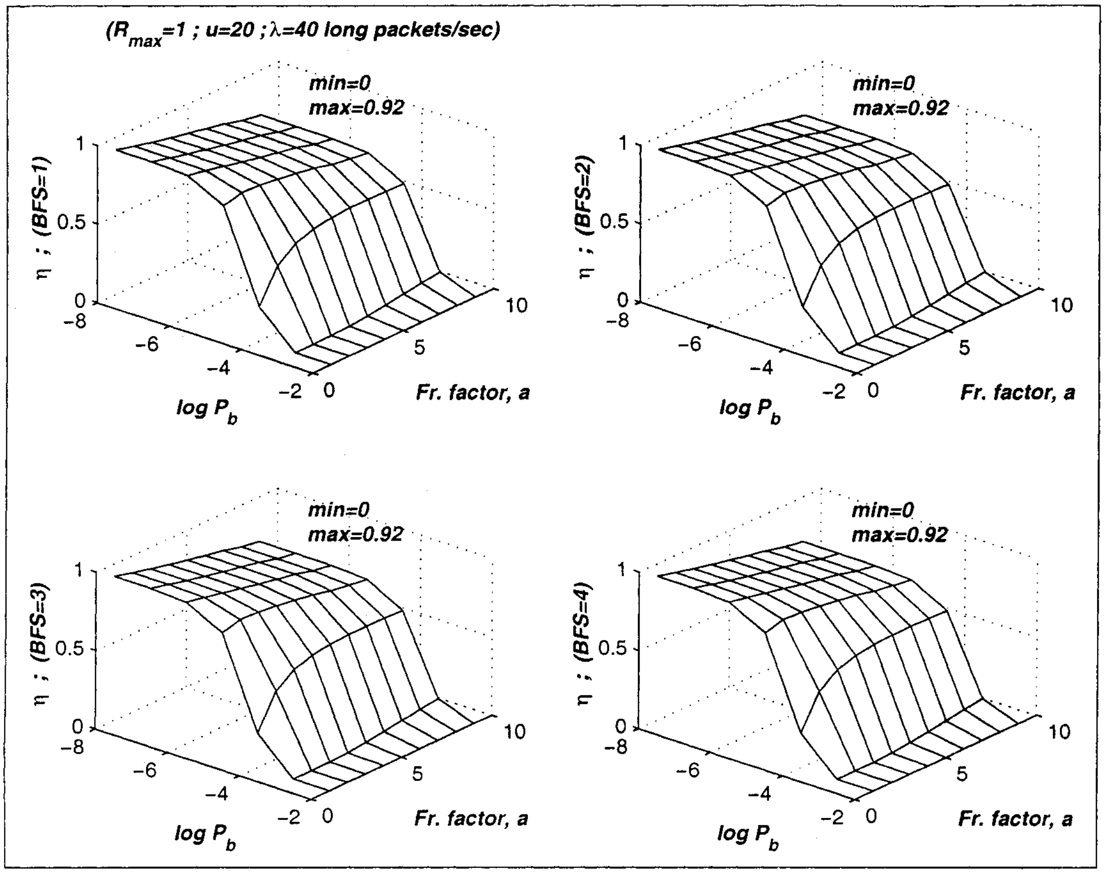


Figure 3.12: Throughput vs. BER and fragmentation factor; $R_{max} = 1$.

sketched for 12 cases of $R_{max} = 0, 1, 2$ and buffer size $(BFS)$=$1, 2, 3, 4$ the latter in terms of long packets. In each case, BER and fragmentation factor ($a$) vary in the range of $10^{-8} - 10^{-2}$ and $1 - 9$ respectively. However, for delay calculation, $10^{-3}$ $\leq$BER$\leq$ $10^{-2}$ range was eliminated due to very large delay results attributed to high $P_{drop}$. In wireless packet transmission, one of the most important choices to be made is the packet length. Results in this section lead us to make such optimum choices. Figures 3.11- 3.13 show that for BERs in the range of $10^{-4} - 10^{-6}$, there is an optimum fragmentation factor which increases with increasing BER. For instance $a$=3,8 are optimum choices for BER=$10^{-5}$, $10^{-4}$ respectively, while fragmentation has no benefit for BER$\leq$ $10^{-6}$. Throughput falls down dramatically for BERs beyond

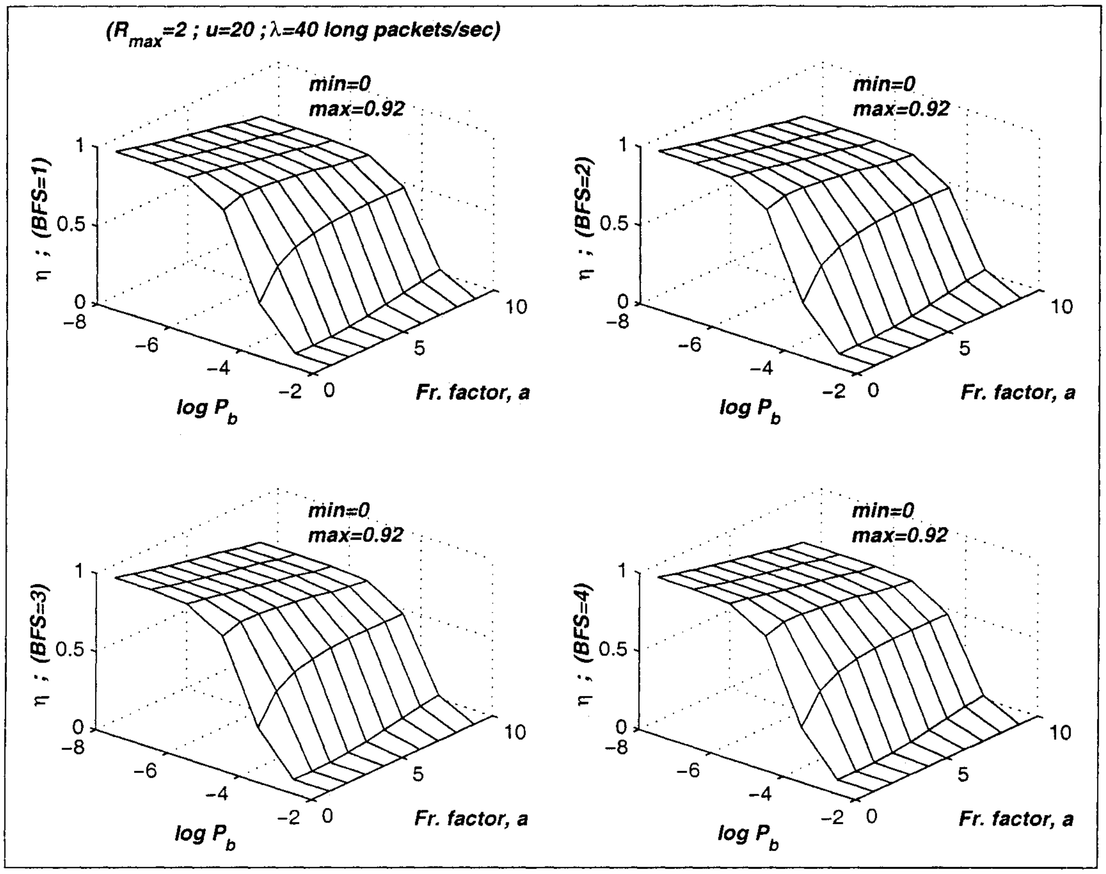


Figure 3.13: Throughput vs. BER and fragmentation factor; $R_{max} = 2$.

$10^{-3}$ inclusive. Finally, unnoticeable differences between Figs. 3.11, 3.12, and 3.13 and their sub-plots show that buffer size and retransmission limit have little effect on throughput defined by (3.16) which in turn emphasizes the significance of tradeoffs among multiple performance measures.

Referring to Figs. 3.14- 3.16, total transmission delay has been normalized with respect to the single access fragment transmission time to facilitate comparison. In general, allowing more retransmission and longer buffer should lead to higher delays. However, for $10^{-8}$ ≤BER≤ $10^{-6}$ with $a \leq 7$ and BER=$10^{-5}$ with $a \leq 6$, regardless of the retransmission limit and the buffer size, delay stays at a constant low. In both cases, further fragmentation leads to delay increase which also obeys the general trend

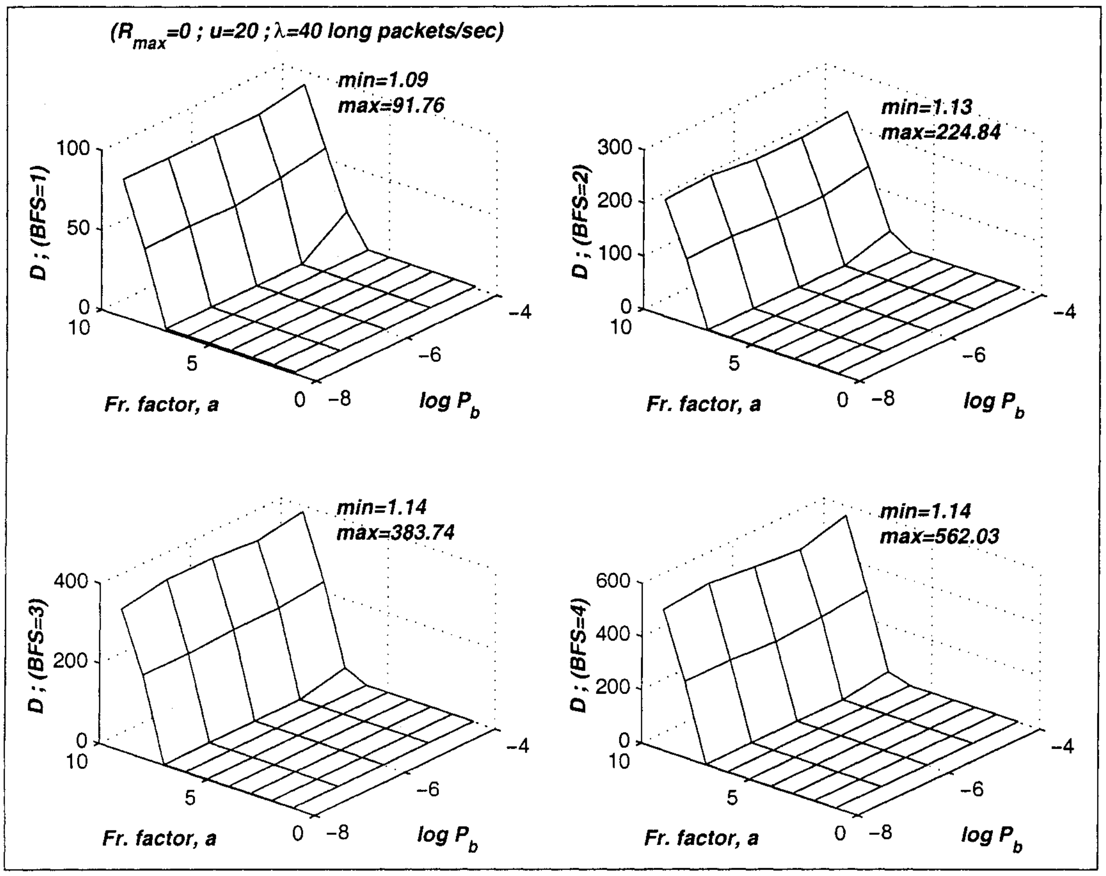


Figure 3.14: Total delay vs. BER and fragmentation factor normalized w.r.t. single access case transmission time); $R_{max} = 0$.

mentioned above. BER=$10^{-4}$ also, demonstrates higher delays with more retransmission and longer buffer which monotonically increases with deeper fragmentation. The only exception is that the delay increase with more fragmentation starts at $a = 6$ where $R_{max} = 0$.

Probability of fail to deliver, $P_f$, is the only measure herein that explicitly includes both blocking probability ($P_{block}$) due to full buffer and drop probability ($P_{drop}$). Intuitively speaking, increasing retransmission limit, $R_{max}$, decreases $P_f$ through lowering $P_{drop}$ while contributing to more occupied buffer and so higher $P_{block}$. On the other hand, larger buffer size lowers the $P_{block}$ and in turn $P_f$. Figures 3.17- 3.19 in ad-

Table 3.1: IEEE802.11 standard parameters for FHSS WLANs

| Parameter | Value |
|---|---|
| Bit Rate ($r$) | 1 Mbit/s |
| $L_{LDATA}$ | 2312 octets |
| $L_{RTS}$ | 20 octets |
| $L_{CTS}$ | 14 octets |
| $L_{ACK}$ | 14 octets |
| $L_{OH(MAC)}$ | 34 octets |
| $L_{OH(PHY)}$ | 16 octets |
| $T_{DIFS}$ | 128 $\mu$sec |
| $T_{SIFS}$ | 28 $\mu$sec |
| $b$ | 50 $\mu$sec |

dition to confirming the above mentioned claim, suggest that major improvement in $P_f$ happens by going from $R_{max} = 0$ to $R_{max} = 1$ and from $BFS = 1$ to $BFS = 2$. So more retransmissions and longer queues are not rewarded proportionally. Moreover, the optimum fragmentation choice corresponding to each BER can be extracted from the figures. For instance, in $(R_{max} = 1, BFS = 2)$ case, $a = 8, 3, 2 - 7$ are the fragmentation choices corresponding to BER=$10^{-4}, 10^{-5}, 10^{-6} - 10^{-8}$ respectively.

## 3.5 Fragment Transmission Strategies

Another issue which constitutes the subject of this section is what fragmentation strategy is adopted regarding transmission order. As was discussed in section 3.2.1 and according to IEEE 802.11 protocol suite, upon successful channel gain, the corresponding user transmits all the fragments belonging to a particular packet before releasing the channel (Fig. 3.5). Thereafter, it releases the channel and a new contention between active users will take place. In another scenario, the user that has the

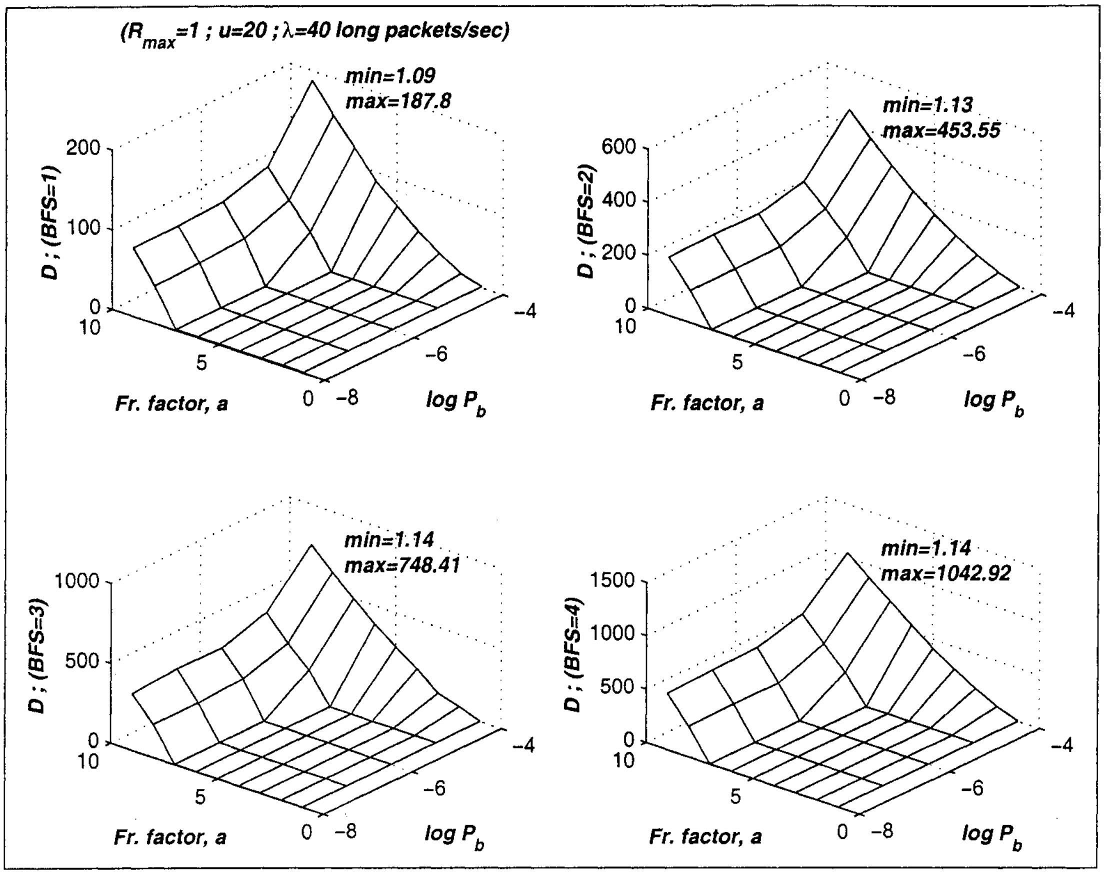


Figure 3.15: Total delay vs. BER and fragmentation factor normalized w.r.t. single access case transmission time); $R_{max} = 1$.

control of the channel, transmits only one fragment associated with a specific packet and then the channel becomes available for another contention period. In other words, packet fragments from different users travel randomly interleaved over the wireless channel. The scenario consistent with IEEE 802.11 protocol suite (Fig. 3.5) provides less contention overhead per packet while the second one gives more fairness regarding resource distribution. Traffic over the wireless channel in terms of fragment orders in both scenarios have been shown in Fig. 3.20. The corresponding flowchart of the second scenario is the same as the flowchart of Fig. 3.6-b with the dotted part removed. The state-diagram pair of Fig. 3.7 is equally applicable here with probably different parameters as follows. To avoid repetition, we try to mention only differing details

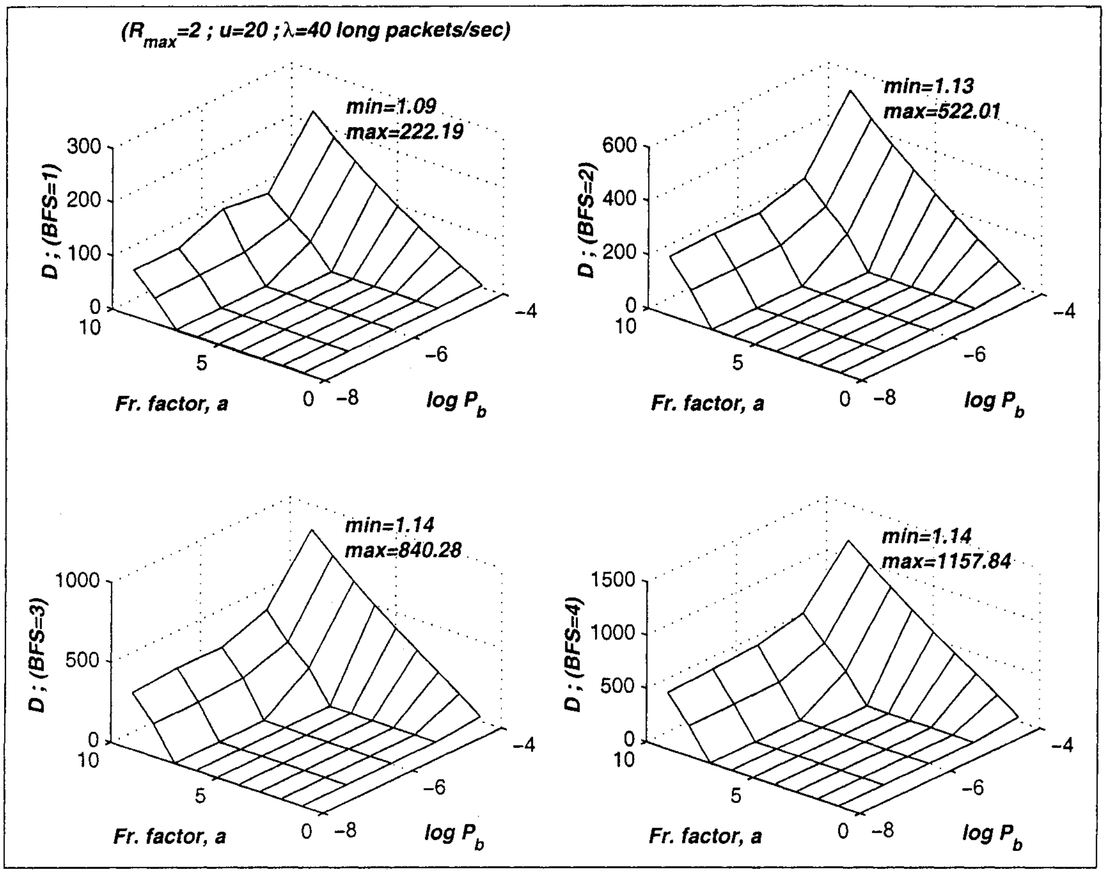


Figure 3.16: Total delay vs. BER and fragmentation factor normalized w.r.t. single access case transmission time); $R_{max} = 2$.

with respect to the earlier analysis. Finally, comparative results will be presented which leads us to draw some conclusions.

As opposed to the 802.11 case, in the second scenario, a successful contention by any user, leads to departure (successful transmission or drop) of a single fragment. Focusing on the first state diagram in Fig. 3.7-a, since the channel cpature mechanism (RTS/CTS dialogue) remains the same, $P_{fail_j}$, $\alpha$, $T_{fail_j}$, $P_s$, and $x$ are unchanged. Subsequently, $\lambda_i$ in (3.7) stays intact. In a very similar way, the average cost of a successful transmission/drop, $C$ in (3.8), prevails for both cases. The average CW shrinkage time cost initiated in the $j$th slot, $T_{success_j} = aC + x + jb$ is

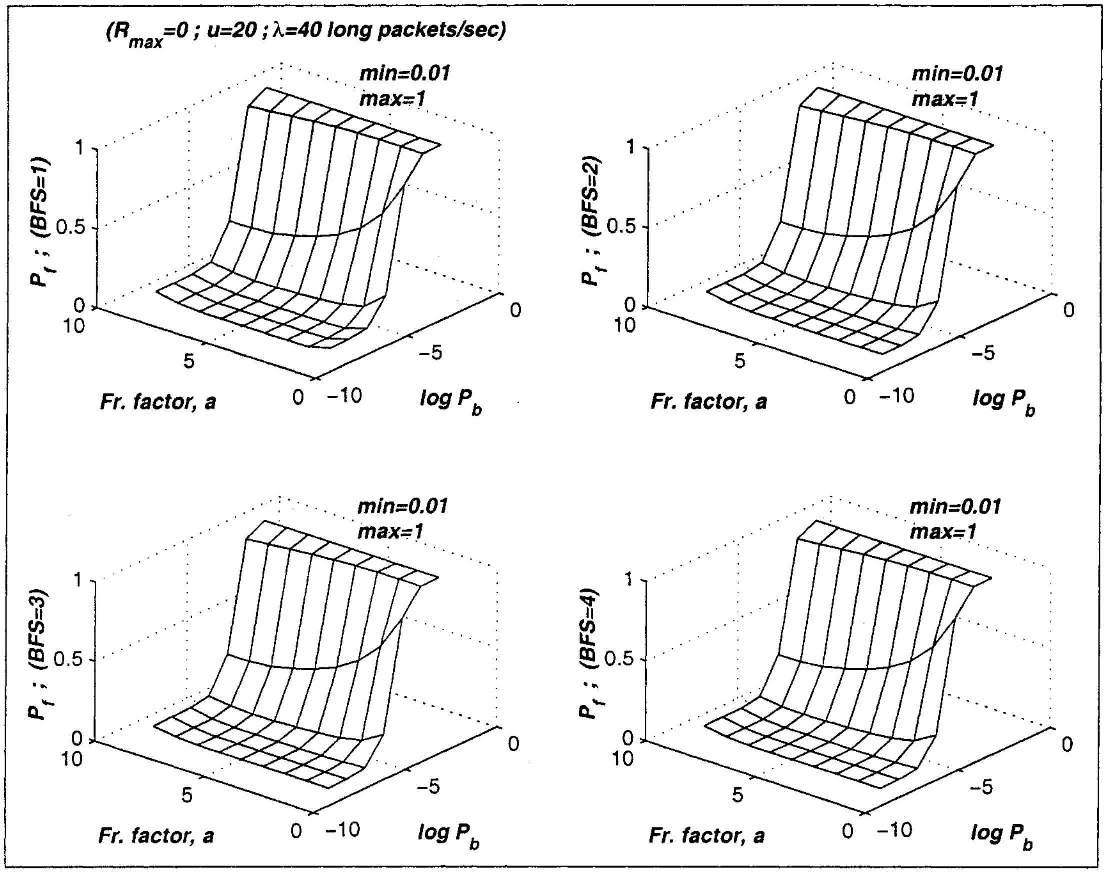


Figure 3.17: Probability of fail to deliver variations vs. BER and fragmentation factor; $R_{max} = 0$.

replaced by $T_{success_j} = C + x + jb$ to account just for a single fragment rather than a full packet. This will inversely affect the average CW shrinkage rate ($\mu_j$ in (3.9)) which now corresponds to fragment departure rate. Equilibrium equations in (3.10) are then applied to yield average CW size measure $\bar{\imath}$. Regarding the second state diagram (Fig. 3.7-b), (3.11) still holds however, $T_d$ is redefined as the total fragment transmission time. As $T_d$ reflects the fragment transmission cost time for the whole system, the fragment service rate available to each user would be

$$\mu = \frac{1}{u_a T_d} \tag{3.20}$$

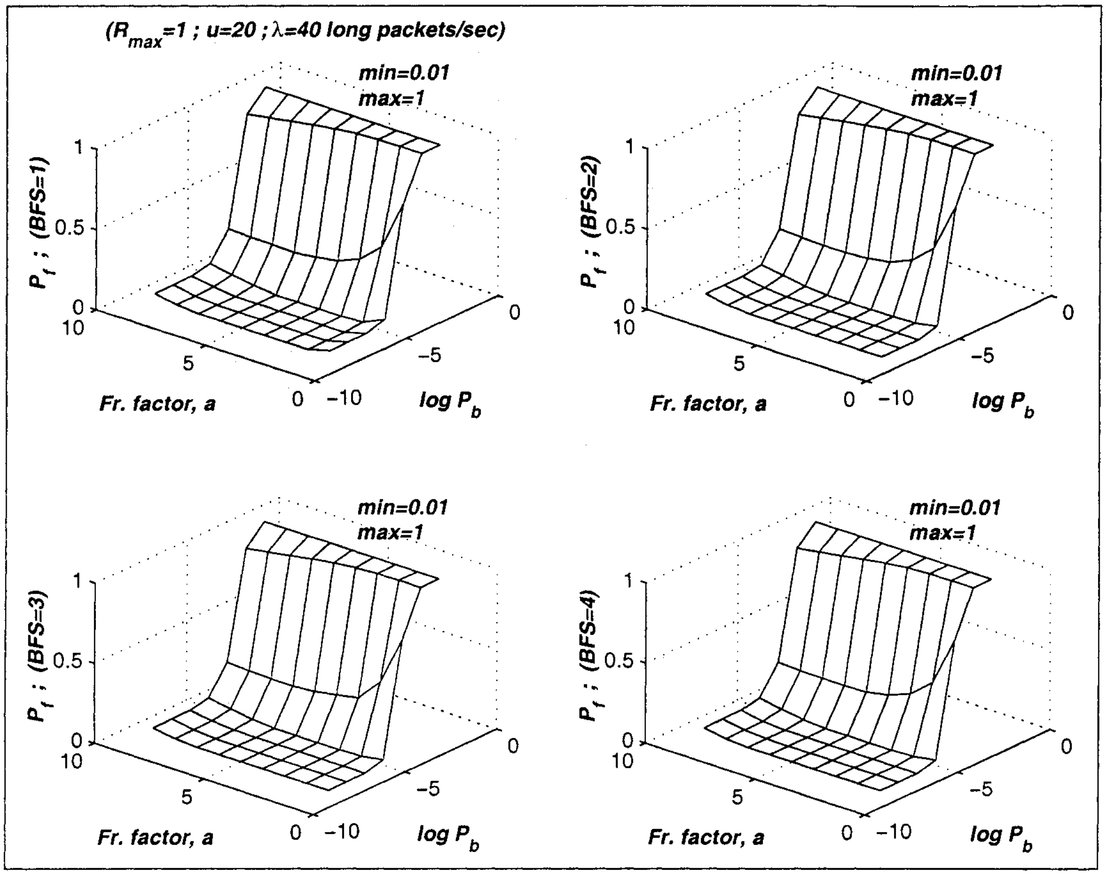


Figure 3.18: Probability of fail to deliver vs. BER and fragmentation factor; $R_{max} = 1$.

The new fragment arrival rate is $\lambda' = \frac{a\lambda}{u}$ as in the previous case. Ultimately, (3.13) and (3.14) are similarly applicable herein. Probability of dropping and blocking a fragment, $P_{drop}$ and $P_{block}$ in (3.15) and (3.17) respectively, remains the same since in both scenarios a fragment is processed as a unit.

Referring to Fig. 3.10, measures of performance of interest; throughput ($\eta_2$), delay ($D_2$), and probability of fail to deliver ($P_{f_2}$) are as below

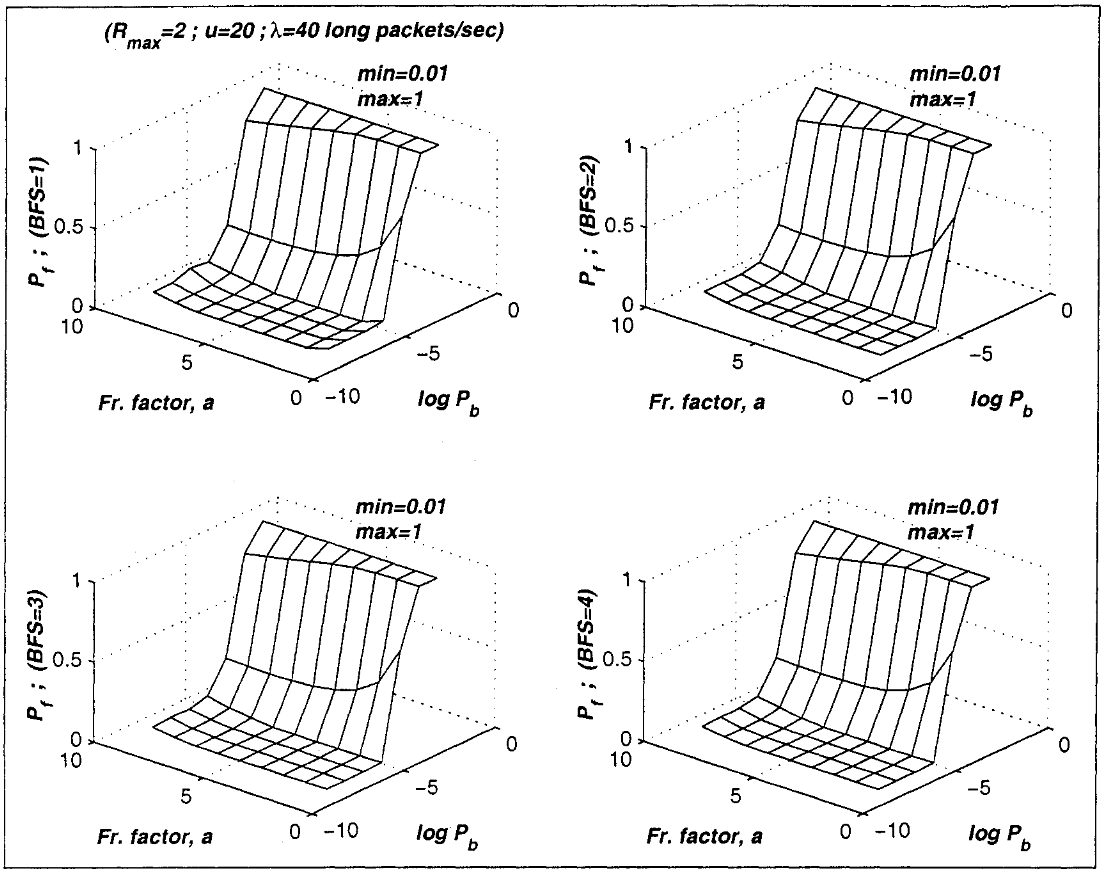


Figure 3.19: Probability of fail to deliver vs. BER and fragmentation factor; $R_{max} = 2$.

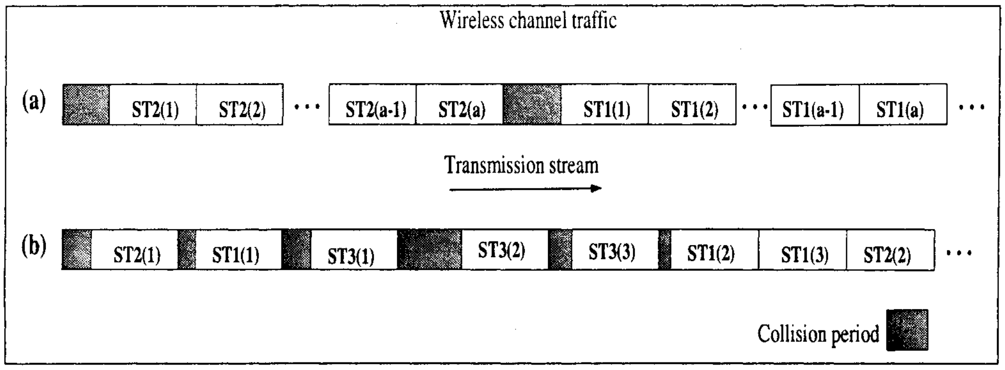


Figure 3.20: Channel traffic according to: a) IEEE 802.11 standard. b) the second scenario.

$$\begin{aligned}
\eta_2 &= \frac{(1-P_{drop})T_{LDATA}}{aT_d} \\
D_2 &= \frac{ar}{L_{DATA}}\left[\frac{\bar{k}u}{(1-P_{block})\lambda}\right] \\
P_{f_2} &= P_{block} + (1-P_{block})P_{drop}
\end{aligned} \tag{3.21}$$

where the subscript 2 is to make the distinction between the second transmission strategy and the first one (which in presenting the comparative results shortly, will be identified by subscript 1). The only differences from the corresponding measures in (3.16) and (3.18) are the extra fragment factor $a$ in $\eta_2$ and $D_2$ which stems from normalization w.r.t fragment rather than packet. Comparative plots of $\eta_1 - \eta_2$, $(D_1/D_2) - 1$, and $P_{f_1} - P_{f_2}$ are illustrated in Figs. 3.21- 3.23. To avoid becoming overwhelmed with figures, only results corresponding to selected values of $R_{max}$ and $BFS$ are shown. However, the selection has been tried to be as representative as possible. To facilitate interpreting the figures, solid circles locate situations where second strategy outperforms the first.

Figure 3.21 shows that for all ranges of BER and fragmentation factor, the first strategy is superior to the second one and this superiority grows toward deeper fragmentation and less BERs. This seems reasonable due to the fact that the second strategy introduces more overhead per fragment which in low BER channels is redundant. At maximum, extra throughput gained by the first strategy is 16%. Buffer size ($BFS$) and retransmission limit ($R_{max}$) parameters have negligible effect on $\eta_1 - \eta_2$.

Illustrated in Fig. 3.22, the second strategy outperforms or equals the first over much of the span of $1 \leq a \leq 4$ except $a = 4$ at BER=$10^{-5}$ for $R_{max} \geq 1$. However, deeper fragmentation favors the first scenario.

Comparison of the two fragment transmission strategies regarding probability of

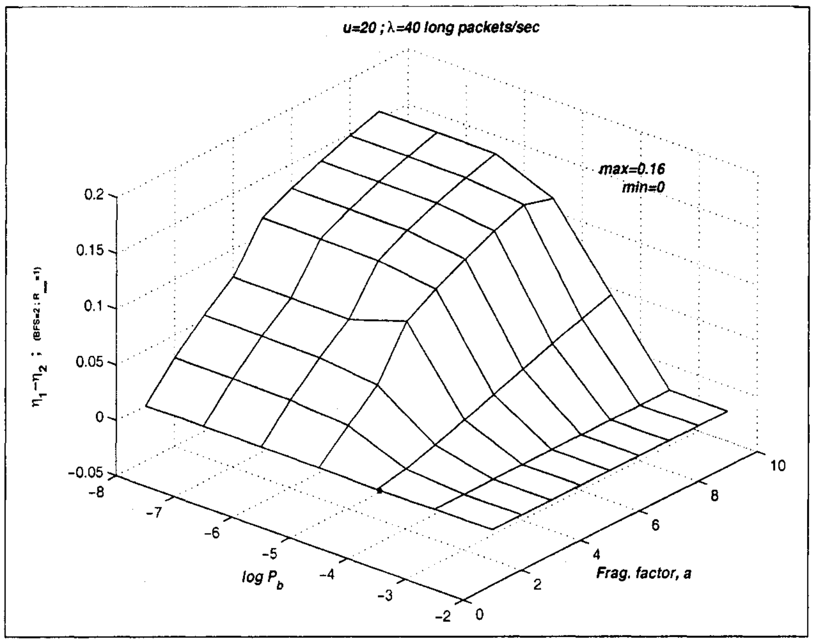


Figure 3.21: Throughput comparison of the two strategies for a selected value of $BFS$ and $R_{max}$.

fail to deliver is presented in Fig. 3.23. Superiority of the second strategy is confined to the $a \leq 4$ and mostly toward less BERs. Also should be noted that at most the second strategy outperforms the first by 4% and under-performs it by up to 18% depending on the choice of parameters.

### 3.5.1 Simulation Results

To investigate the validity of the proposed analysis model in representing system behavior, our simulation package detailed in Chapter 4 is invoked with the following simplifications. The traffic generation module is herein simplified to generate Poisson-distributed traffic rather than on-off process. Regarding the wireless channel, multipath, mobility, path loss, and shadowing phenomena are all eliminated and no FEC is employed. Channel's BER is kept fixed for the whole simulation time for the sake of consistency. All the nodes are placed close to each other's coverage aiming at a

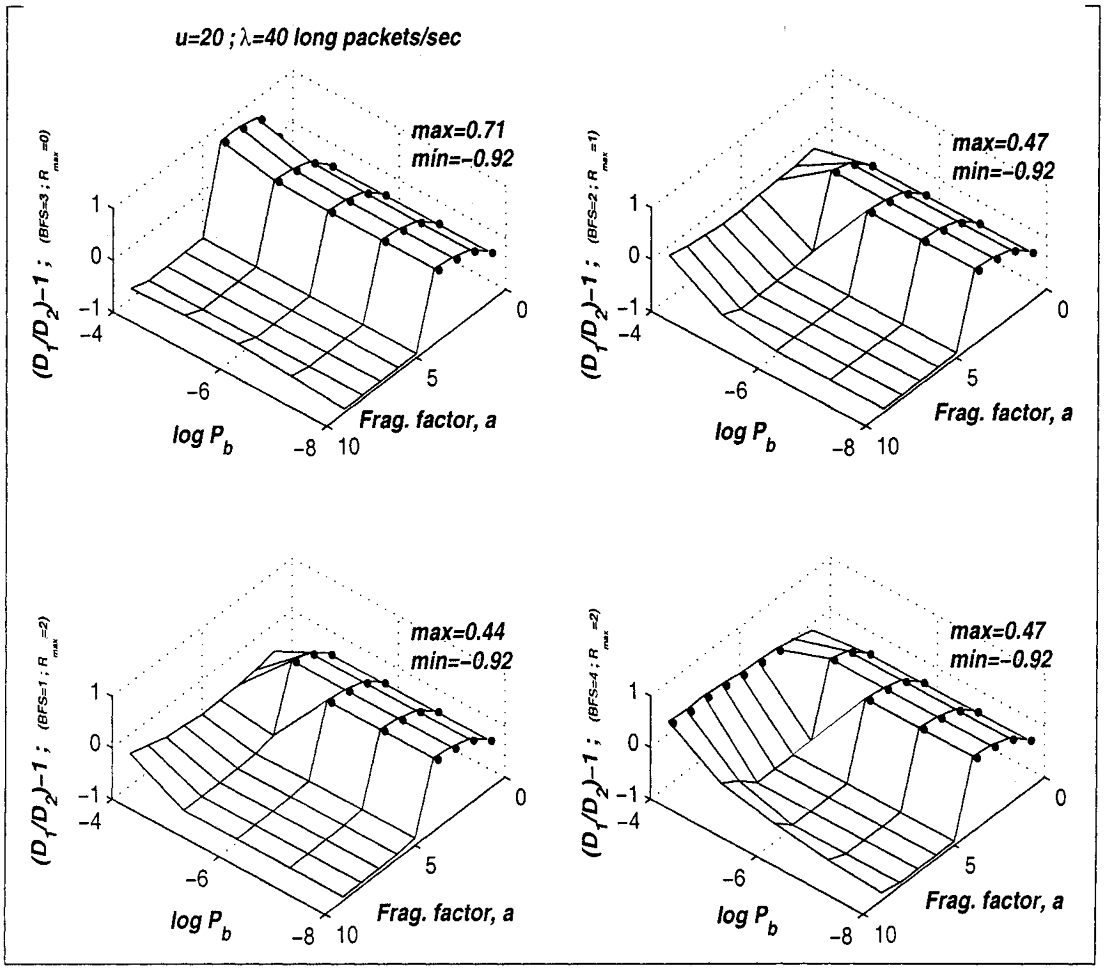


Figure 3.22: Delay comparison of the two strategies for selected values of $BFS$ and $R_{max}$.

single-hop scenario. Just mentioned modifications ensure complete consistency with the analytical approach, and that only 802.11 MAC protocol is simulated. Moreover, in the simulation package, there is a channel contention for each fragment rather than each packet. This corresponds to the second fragment transmission strategy in section 3.5. To better view the convergence of the simulation results, the simulation is executed as follows. First, we let the simulation run for 5.6 seconds ($2 \times 10^5 \times T_{SIFS}$) and repeat 16 ($4 \times 4$) times to yield a single point result by applying the method of batch means [60]. In the second attempt, the simulation time increases to 11.2

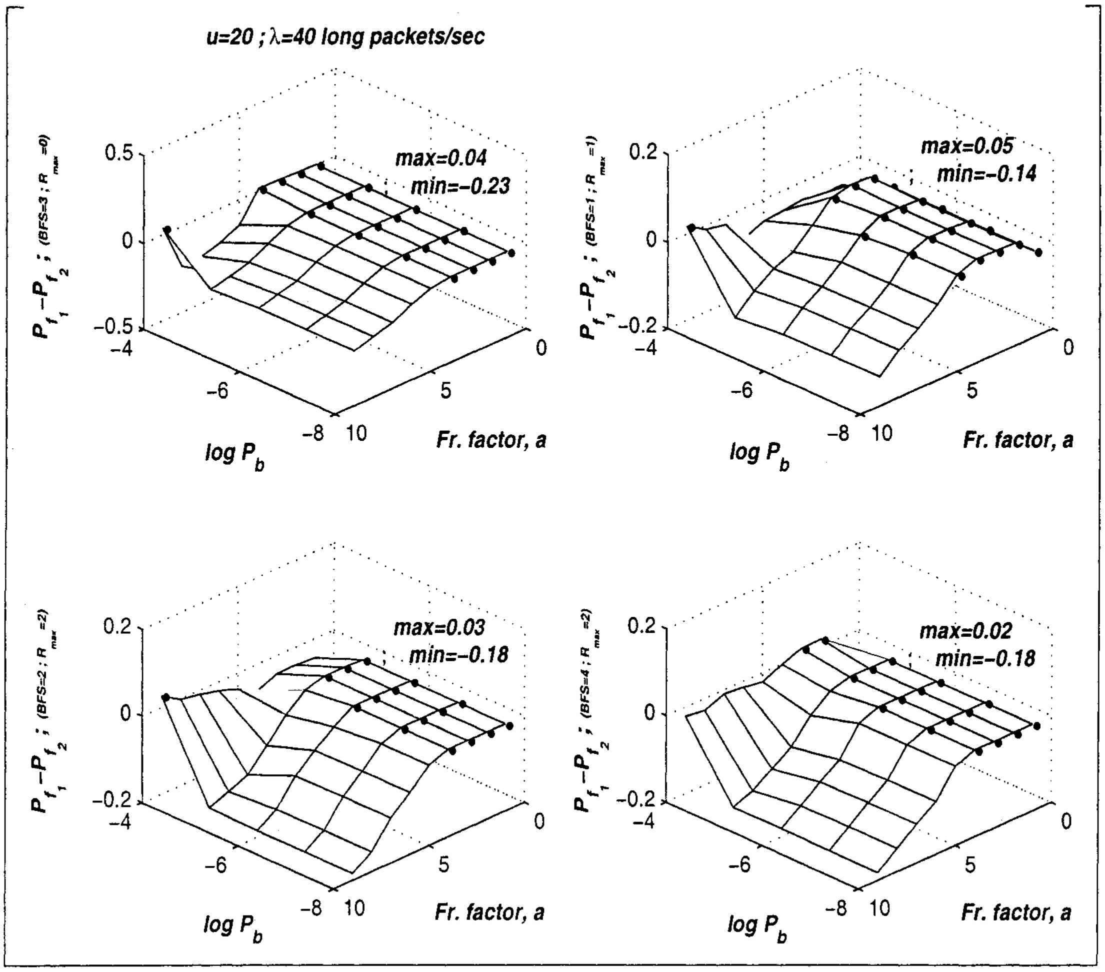


Figure 3.23: Probability of fail to deliver comparison of the two strategies for selected values of $BFS$ and $R_{max}$.

seconds ($4 \times 10^5 \times T_{SIFS}$) and repeats 25 ($5 \times 5$)times. Figures 3.24 and 3.25 show the delay and probability of fail to deliver versus BER and fragmentation factor for special case of $R_{max} = 2$ and $BFS = 3$ and other simulation parameters. In these figures, the upper left and right plots show the normalized delay and probability of fail to deliver respectively. To view the accuracy of the simulation results, normalized standard deviations defined as $\frac{\sigma_D}{D}$ and $\frac{\sigma_{P_f}}{P_f}$ are shown in the lower plots. These figures should be compared to their analytical counterpart as sketched in Fig. 3.26.

Analytical results seem to satisfactorily follow the general trend of variations as

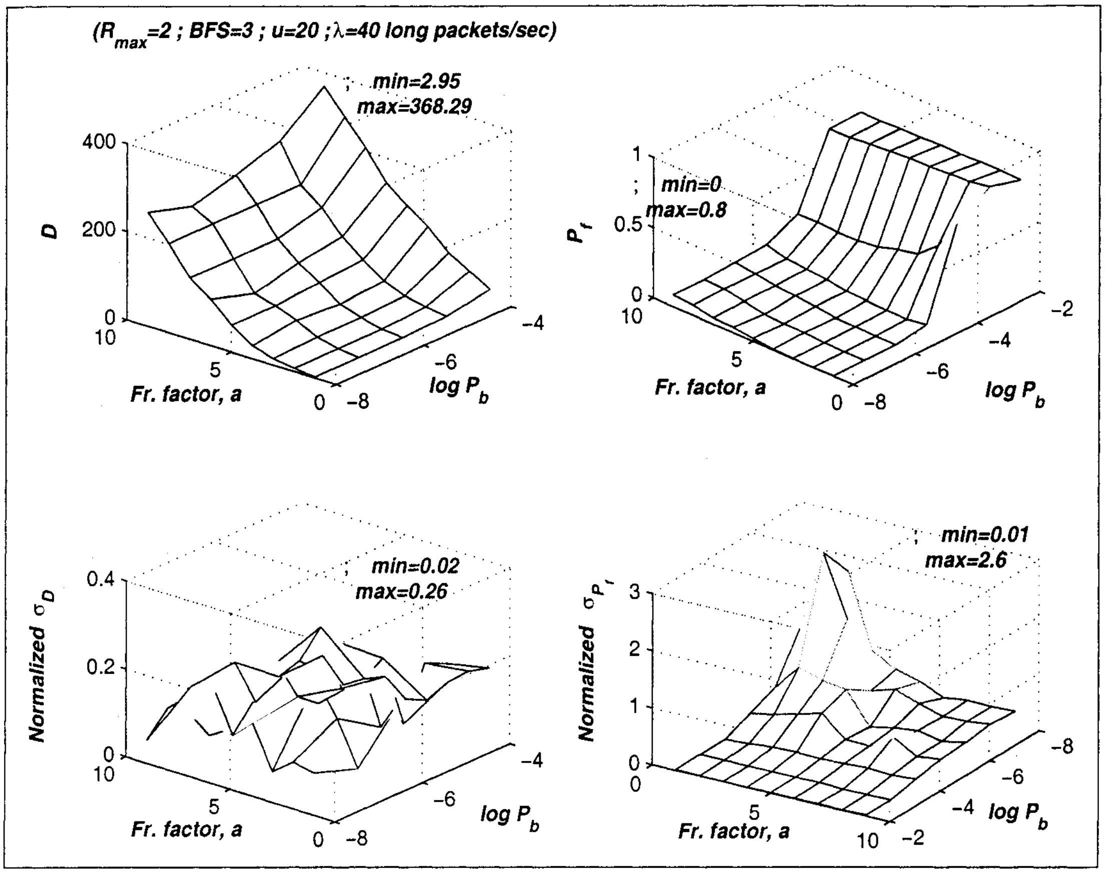


Figure 3.24: Simulation results of the normalized delay and probability of fail to deliver in single-hop 802.11 WLAN corresponding to the second transmission strategy, and with simulation run time of 5.6 seconds and 16 repetitions per point.

shown by simulation results. Regarding the delay, Fig. 3.25 shows more resemblance to Fig. 3.26 than Fig. 3.24. This can be due to the longer run time of the second simulation as the analytical approach aims at system response at $t \rightarrow \infty$. Moreover, Fig 3.25 gives slightly better normalized standard deviations which is attributed to more simulation repetitions. Though longer simulation runs seems to result in higher delays, we expect the analytical results finally to be inferior since as mentioned earlier they are based on a worst case scenario approach. Almost the same comments are applicable to probability of fail to deliver results. However, it should be noted that simulation results present high normalized standard deviations in the regions where

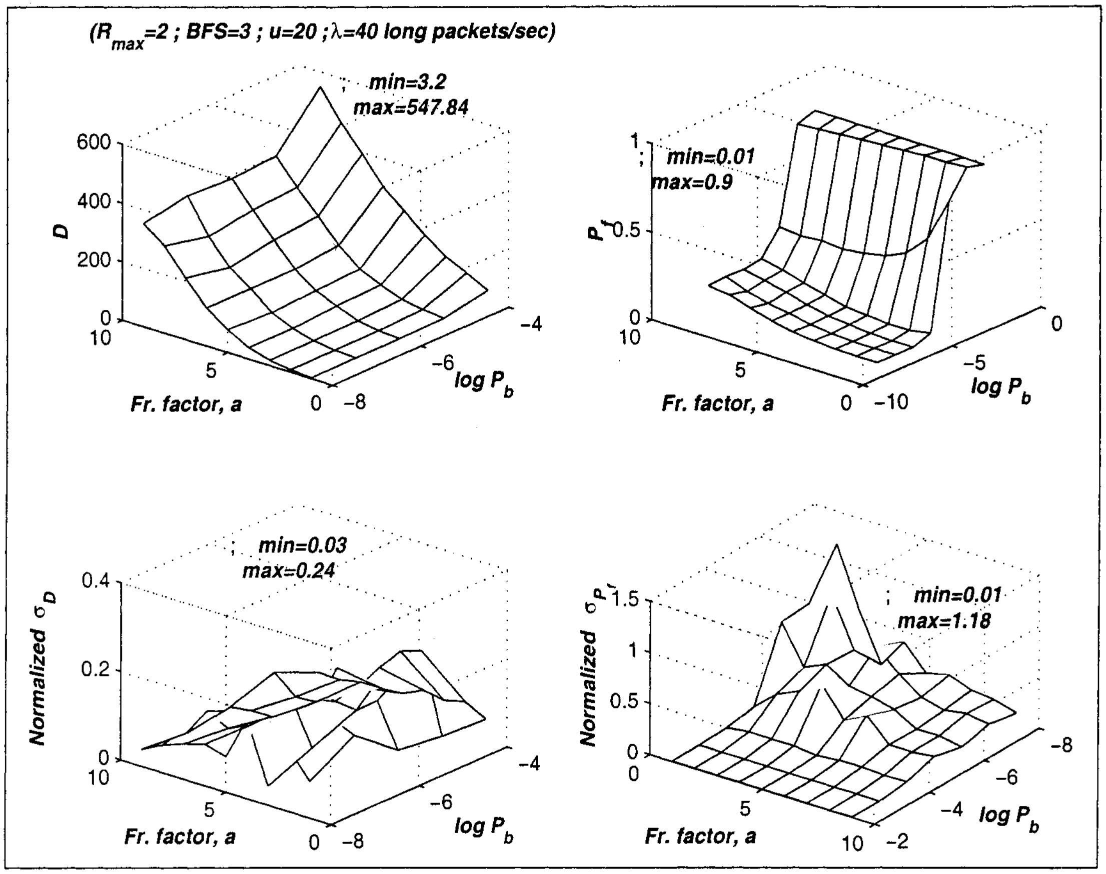


Figure 3.25: Simulation results of the normalized delay and probability of fail to deliver in a single-hop 802.11 WLAN corresponding to the second transmission strategy, and with simulation run-time of 11.2 seconds and 25 repetitions per point.

$P_f$ assumes very low values. This can be contributed to different sources of calculation errors (e.g. machine-originated) but does not undermine the validity of the analytical approach. While longer simulation run-times and more repetitions should guide us to better results in terms of confidence interval, such an attempt is hampered by excessive simulation times. To give a better sense of real simulation time demand, 25 repetitions of the simulation (Fig. 3.25) take about 75 hours on a Pentium4-2.4 GHz-equipped computer system. At the end it is mentioned that since the simulation package, originally devised for multi-hop WLAN study, does not provide us with the queuing delays involved, throughput results ($\eta = \frac{1}{D-D_Q}$) can not be extracted for

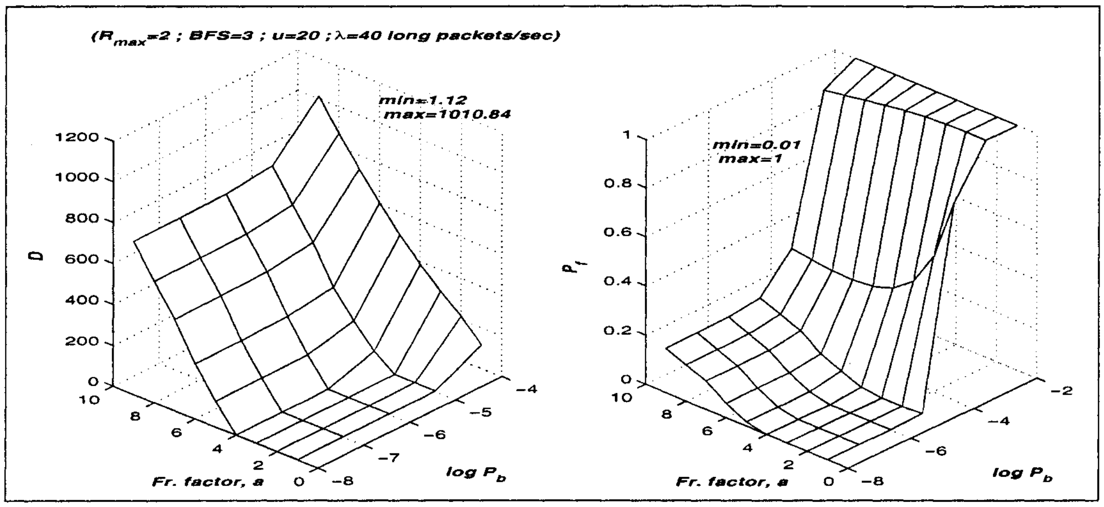


Figure 3.26: Analytical results of the normalized delay and probability of fail to deliver corresponding to the second transmission strategy.

comparison purposes.

## 3.6 Summary

In this chapter, asynchronous mode of operation of IEEE 802.11 WLAN, adapted for real-time traffic, was modeled by a state-diagram pair which turned to be useful in analyzing the effects of several parameters such as packet fragmentation, channel BER, node buffer size, maximum allowable retransmission, and packet arrival rate on the system performance. The model is capable of accommodating many other parameters of interest as well. On the other hand, by changing the CW growth rule, a wide variety of random back-off algorithms can be simulated. Results show that over a range of BERs typical to wireless channels ($10^{-4} - 10^{-6}$), choices of fragmentation factor, buffer size and retransmission limit can be found which optimize a specific performance measure or their joint weighted measure. Among the studied measures, delay and probability of fail to deliver are of much interest to real-time applications.

A new fragment transmission strategy was introduced in section 3.5 which is based on per fragment channel contention rather than per packet in IEEE 802.11. While this strategy proves less efficient in terms of throughput due to its more overhead incurrence, at occasions it shows better performance in terms of delay and probability of fail to deliver.

Finally, by invoking a one-hop version of the computer simulation package developed in Chapter 4, validity of the proposed mathematical model was investigated. Simulation results demonstrate a satisfactory consistency with their analytical counterparts.

# Chapter 4

# Performance Evaluation of Multihop, Mobile, Ad-Hoc IEEE 802.11 WLAN

## 4.1 Introduction

There are situations which give rise to scenarios that involve multi-hop WLAN communications. Among those are: stretching last mile connection beyond coverage radius of access points, and establishing local networks in areas at least larger than what can be reached by an individual node. In an earlier chapter, the mathematical model for ad-hoc mode of operation of IEEE 802.11 was introduced. Going from BSS to ESS, system randomalities grow significantly which make analytical modeling almost impossible. Turning to simulation enables us to investigate a greater number of phenomena such as hidden terminal, mobility, channel fading and path loss, and possibly investigation of more involved models. The simulation is arranged in a modular-based manner where each module copes with one unique process while exchanging data with others. We first start with the topological description of the system. Following that, traffic generation, mobility, channel, and protocol modules are introduced. The computer simulation results are presented afterwards which will lead us to draw conclusions. Key assumptions made throughout the simulation are intermittently mentioned when discussing the modules.

The purpose of the simulation is to evaluate the performance measures of the system such as end-to-end delay and probability of successful delivery in different situations ( e.g. traffic, mobility, etc) and against different choices of system parameters (e.g. fragmentation factor, buffer length, retransmission limit, etc).

## 4.2 System Description

As many as $n_{tot}$ nodes are uniform-randomly located inside a rectangular area, laid diagonally between $(x_{min},y_{min})$ and $(x_{max},y_{max})$. These upper and lower limits, together with transmitter's wireless range, determine whether single-hop or multi-hop scenario is dealt with for the source-destination pair of each call. As mentioned earlier, routing is the first concern that distinguishes single-hop from multi-hop. As the first basic assumption, all nodes are aware of others' geographical locations. Our main emphasis in this research is the interaction of physical and MAC layers, so we simplify the simulation of routing. We assume the information exchange among all nodes to be GPS-based and certainly resulting in some bandwidth cost. Each node with a packet ready to be sent picks another node as its next (intermediate) destination provided that:

- It is inside the source node coverage area
- It has the least Euclidean distance to the packet's final destination

While the above may not be the best routing strategy, in most cases it leads to the least number of hops traversed and is of course easy to implement. However, any other routing algorithm mentioned in section 2.3.3.1 can easily be incorporated in our simulation environment. The most important point which we may not further mention explicitly in what follows is the bandwidth cost of whatever routing strategy is used. This degrades the system performance below the presented results. In the initialization phase, some nodes are uniform-randomly selected as mobile and the rest left as fixed nodes. The mobility status will prevail for the whole simulation time. The parameter $P_{mobile} = \frac{n_{mobile}}{n_{tot}}$ along with other parameters introduced shortly in section 4.4 represent the mobility volume of the system. In a similar way, uniform-randomly selected nodes are responsible for generating traffic while all the nodes might serve as forwarding and final destination nodes at any time during the simulation. A parameter representing traffic generating (active) nodes population is defined as $P_{active} = \frac{n_{active}}{n_{tot}}$. This parameter may serve as the system's traffic volume indicator as well. Each active node is pre-assigned a node as its final destination which remains unchanged throughout the course of simulation.

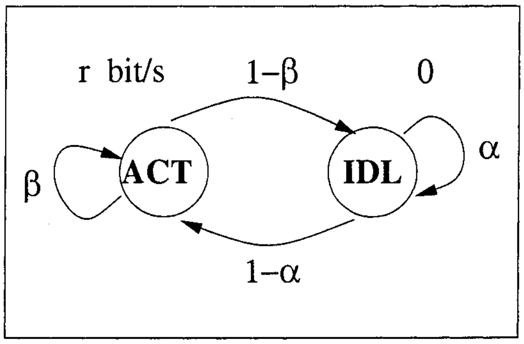


Figure 4.1: On-off traffic generation model.

## 4.3 Traffic Generation Module

Considering voice connection as a representative of real-time applications, the packet generation model employed is an on-off keying model. This model has been used extensively in the literature to model voice communications (e.g. see [37] and [17]), mostly because it can efficiently imitate silence/talk-spurt periods of human conversation and due to ease of computer simulation. Figure 4.1 illustrates the traffic generation model. While in active state (ACT in the figure), the model acts as a constant bit rate (CBR) source generating r bit/s and while in idle state (IDL in the figure), no packet is generated. Parameters $\alpha$ and $\beta$ are self-loop transition probabilities in IDL and ACT states respectively. Slot time of the model is considered to be long enough to accommodate the longest packet size generated above the user's link layer:

$$t_{slot} = \frac{\text{max. packet length}}{\text{source bit rate}} \tag{4.1}$$

where the maximum packet length, source bit rate, $\alpha$, and $\beta$ are the same for all users. Then the average idle and active states dwelling times are:

$$T_{IDL} = t_{slot}(1-\alpha)^{-1} \quad , \quad T_{ACT} = t_{slot}(1-\beta)^{-1} \tag{4.2}$$

The two functions listed below describe the traffic generation status of each node:

*source(ID_NO)*: Determines if a specific node identified by *ID_NO* is a traffic generator.

*status(ID_NO)*: Determines whether a traffic generator node (specified by the function *source(ID_NO)*) is in active or idle mode.

Upon packet generation at source node, functions *pkt*() and *ver*() as presented below are responsible for carrying over all the information pertinent to the generated packet such as: generating node, destination node, intermediate nodes, packet's location in the queue, generation time, receiving time, cumulative time spent in different queues, successful reception indicator, etc.

$$
\begin{aligned}
&pkt(SEQ\_NO, CP\_VER, HOP\_VER) = (SRC, DST, IMDT\_SRC, IMDT\_DST, \\
&\quad Q\_POS, GEN\_TM, RX\_TM, TM\_IN\_Q, SUC\_RX, RE\_TX, DRP, NO\_HOP, \\
&\quad BLK, CNTR, BER1, BER2, ..., BERx, VIA\_1, VIA\_2, ..., VIA\_n) \\
&ver(SEQ\_NO) = LST\_CP\_NO
\end{aligned}
$$

Each packet is uniquely identified by its 3-tuple identifier $(SEQ\_NO, CP\_VER, HOP\_VER)$ as will be explained shortly in section 4.6.

## 4.4 Mobility Module

All nodes designated as mobile, periodically (each $\Delta/v$ seconds where the input parameters $\Delta$ and $v$ denote displacement step length in meters and mobile's speed in m/s respectively) call a uniform-random number generator over interval (0,1) and the outcome, z, changes the nodes' 2-D coordinates as follows:

$$
\begin{aligned}
\Delta x + i\Delta y = \Big(&u(0.5 - z)\big[u(0.25 - z) - u(z - 0.25)\big] + iu(z - 0.5) \\
&\big[u(0.75 - z) - u(z - 0.75)\big]\Big)\Delta
\end{aligned}
\tag{4.3}
$$

where $u()$ is the unit step function, and the updated 2-D coordinates would be:

$$
\begin{aligned}
x_{new} &= min\big(max(x_{old} + \Delta x, x_{min}), x_{max}\big) \\
y_{new} &= min\big(max(y_{old} + \Delta y, y_{min}), y_{max}\big)
\end{aligned}
\tag{4.4}
$$

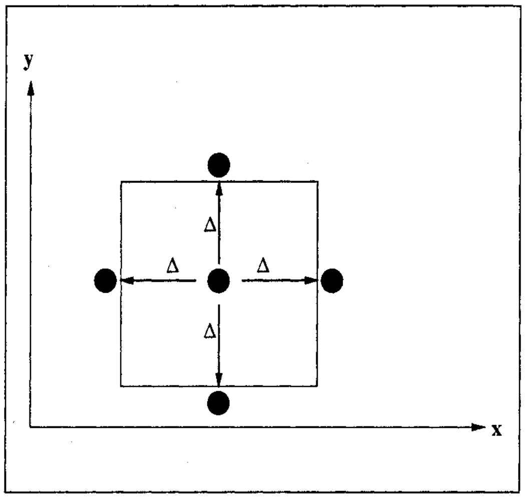


Figure 4.2: Mobile node's movement pattern.

This moves the mobile node to one of its four probable new locations as shown in Fig. 4.2. Equation (4.4) ensures that the nodes do not go beyond the geographical constraints of $(x_{min}, y_{min})$ and $(x_{max}, y_{max})$. The two functions that deal with mobility are *mobile(ID_NO)* and *location(ID_NO)* which respectively specify the initial mobility status and updated location of the nodes throughout the simulation. Large scale fading phenomena are directly related to macro-movements of nodes, and so, they are handled in the mobility module. The first to consider is path loss where we use a two-slope curve model [23]:

$$\begin{aligned} P_L(d) &= P_L(d_0) + 10n_{PL}log_{10}(\frac{d}{d_0}) \quad ; \quad d \geq d_0 \\ P_L(d) &= 20log_{10}(\frac{4\pi d}{\lambda}) \quad ; \quad d < d_0 \end{aligned} \tag{4.5}$$

In (4.5), $n_{PL}$ and $\lambda$ represent terrain-dependent path loss exponent and wavelength respectively. Parameter $d_0$ denotes the break-point distance below where the environment has free space characteristics. The break-point is related to the first Fresnel zone clearance, antenna height, and the carrier frequency. The second large scale phenomenon is shadowing. No shadowing is considered before the break-point, and beyond that, it is modeled as a log-normal distribution with standard deviation $\sigma$. Equation (4.6) can then be augmented to:

$$\begin{aligned} P_L(d) &= P_L(d_0) + 10nlog_{10}(\frac{d}{d_0}) + X(0,\sigma) \quad ; \quad d \geq d_0 \\ P_L(d) &= 20log_{10}(\frac{4\pi d}{\lambda}) \quad ; \quad d < d_0 \end{aligned} \tag{4.6}$$

where $X$ is a normal random variable in dB. For simplicity we do not consider correlation in shadowing from one location to another as opposed to [61]. The displacement, distance, and large scale fading between all pair of nodes are calculated and used for channel BER evaluation as follows in section 4.5.

## 4.5 Channel Module

In this module, the small scale behavior of the channel and in particular the multi-path fading, as the major contributor to such a behavior is simulated. Multi-path fading is caused by propagation on multiple paths and produces short term variations in the received power due to different phases on different paths. Physical characteristics of the channel between transmitter and receiver and its time-variant nature give rise to different fading classes such as fast/slow and flat/frequency-selective fading. There are two important parameters that quantify the channel behavior with regard to the above mentioned fading classes: coherence bandwidth and coherence time. Root mean square(rms) delay spread is the second central moment of the power delay profile and is defined as [62]:

$$\sigma_\tau = \sqrt{\overline{\tau^2} - \overline{\tau}^2} \tag{4.7}$$

where

$$\overline{\tau} = \frac{\sum_k P(\tau_k)\tau_k}{\sum_k P(\tau_k)} \quad \text{and} \quad \overline{\tau^2} = \frac{\sum_k P(\tau_k)\tau_k^2}{\sum_k P(\tau_k)} \tag{4.8}$$

In (4.8) $k$ is the path number that delivers the signal to the receiver, and $P(\tau_k)$ is the signal power received through $k$th path delayed $\tau_k$ seconds. Through its relation with coherence bandwidth, delay spread serves as a measure of frequency selectivity of the channel. A conservative (in terms of frequency selectivity) expression for the coherence bandwidth ($B_c$) is:

$$B_c \approx \frac{1}{50\sigma_\tau} \tag{4.9}$$

over which the correlation between amplitudes of the channel's frequency response at different frequencies is above 0.9. Another less stringent approximation is:

$$B_c \approx \frac{1}{5\sigma_\tau} \tag{4.10}$$

which guarantees the above correlation to be at least 0.5 along the signal bandwidth. Equation (4.10) leads to a larger coherence bandwith compared to (4.9) for the same delay spread. Figure 4.3 shows the power spectral density of the transmitted DS signal in 802.11. As shown, a transmit spectrum mask almost limits the signal bandwidth to around 20-22 MHz which through (4.10) translates to 9-10 nsec delay spread. While this value of delay spread may fall into the ruralarea range, suburban and urban areas show higher delay spreads ( [26] and [27]) and subsequently lower coherence bandwidth. To avoid frequency band variations of the signal strength, at this point we adopt the flat fading channel assumption which may not be valid for all terrain situations. The second parameter, coherence time ($T_c$), quantifies how fast the channel

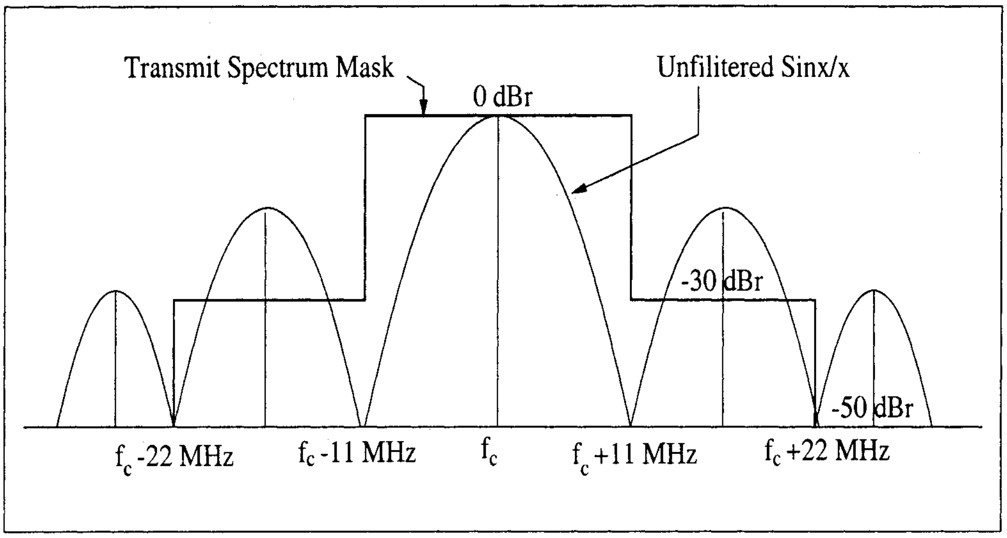


Figure 4.3: Transmit spectrum mask.

changes. A moderate restrictive expression for coherence time follows as ( [62]):

$$T_c \approx \sqrt{\frac{9}{16\pi f_m^2}} = \frac{0.423}{f_m} = \frac{0.423C}{v_m f_c} \tag{4.11}$$

where $f_m$, $v_m$, $f_c$, and $C$ denote maximum Doppler frequency, maximum mobile speed, carrier frequency, and light speed respectively and are related through:

$$f_m = \frac{v_m f_c}{C} \tag{4.12}$$

Regarding the system under study, values of $C = 10^8$ m/s, $f_c = 2.4$ GHz, and $v = 10$ m/s give a coherence bandwidth of $T_c = 5.29$ msec. Compared to the chip period of $T_{chip} \approx 100$ ns ($T_{chip} = \frac{1}{11\times10^6}$), the wireless channel can be considered as slow fading. So at any time we consider the channel as a time-invariant stationary process with different statistics during different coherence-time units. In order to model the time variations of the channel a two-state Markov model (Gilbert model) is adopted. Markovian modeling of fading channels is extensively found in the literature (e.g. see [63], [64], [65], [66], [67], [68], [69], etc.). Most of these references agree on the suitability of the first order Markov chain in modeling slow to medium fading channels while higher order Markov chains are needed for faster fading channels. Moreover, limiting Markov states to two as good and bad channel states has been common due to its efficient tradeoff between accuracy and simplicity, and of course being easily computer-simulated (e.g. refer to [70], [71], [72], [73], [17]). However, good and bad states have been defined in a variety of ways such as having fixed low and high BERs, experiencing no- and deep-fade, or non- and erroneous packet delivery [74]. We assume that a mobile node along its movement alternatively experiences either a line of sight (LOS) or a Rayleigh fading channel along its target node. Figure 4.4 shows the embedded channel model in our simulation with $\alpha'$ and $\beta'$ denoting bad (B) and good(G) states self-loop probabilities. By equating $T_c$ in (4.11) to the model's slot time ($t'_{slot}$), and the average dwelling time of the bad and good states with $T_B$ and $T_G$ respectively, we obtain:

$$T_B = t'_{slot}(1 - \alpha')^{-1} \quad , \quad T_G = t'_{slot}(1 - \beta')^{-1} \tag{4.13}$$

In the good state, the received power ($P_{R_G}$) is calculated from the transmitted power ($P_T$) as $P_{R_G} = P_T - P_L(d)$ where $P_L(d)$ is defined in (4.6). In the bad state, the above obtained $P_{R_G}$ is used to find Rayleigh distributed envelope of the received signal (r)

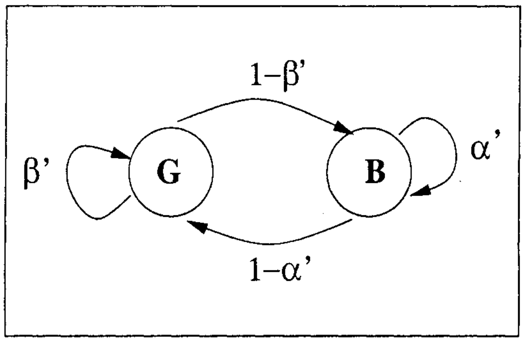


Figure 4.4: Gilbert model regarding time variations of the channel.

through:

$$f_R(r) = \frac{2r}{P_{R_G}} e^{-r^2/P_{R_G}} \tag{4.14}$$

and in turn the bad state received power as $P_{R_B} = r^2$. The received signal power minus noise floor ($NF$) gives the received signal-to-noise ratio $SNR = P_R - NF$, to be used in subsection 4.6.2 to obtain the channel BERs associated with each transmission. Upon expiration of a slot time, the outcome of a uniform random number generator will determine transition between states and if ending up in the bad state, a Rayleigh random number generator is invoked to produce the received power ($P_{R_B}$). Function $ch(ID\_NO, ID\_NO)$ keeps track of the channel status (good or bad) between a pair of nodes.

## 4.6 Protocol Module

IEEE 802.11 protocol implementation in the ad-hoc mode is discussed here. More emphasis is put on the MAC layer while the PHY layer implementation is realized through adopting parameters related to specific choice of modulation, spreading, and FEC coding.

### 4.6.1 MAC Layer

802.11 MAC layer has completely been treated in earlier chapters. However, Figs. 4.5 and 4.6 illustrating successful transmission and collision mechanisms are repeated here for clarity. The flowchart in Fig. 4.7 illustrates the complete MAC implementation based on the transmission mechanism in Figs. 4.5 and 4.6. Due to the ad-hoc nature of the network, this MAC protocol is implemented in a distributed manner (concurrently in all nodes) and thus there are some peculiarities which will be highlighted herein. Throughout the initialization phase, three functions are created that reflect complete status of the node (other than those provided by *source*(), *status*(), and *mobile*()) during the simulation process:

$$\begin{aligned} node(ID\_NO) =& (X, Y, DST, Q\_LTH, BK\_OFF\_CT, CW, ACT\_MOD, TX\_TYP,\\ & SEQ\_NO\_TRNST, CP\_VER, HOP\_VER, NAV, ITER\_TO\_GO,\\ & CH\_IDL\_CT, COLN) \end{aligned}$$

$$\begin{aligned} Q(ID\_NO, CELL\_NO) =& (SEQ\_NO, CP\_VER, HOP\_VER)\\ archive(ID\_NO) =& (SEQ\_NO1, SEQ\_NO2, ....) \end{aligned}$$

Function *node*() gives information such as node's location, destination (in case it is generating any traffic), number of packets in queue, back-off counter's content, contention window size, mode of activity, etc. Function *Q*() on the other hand, provides the specifics about the packets currently in the queue. Sequence number of all

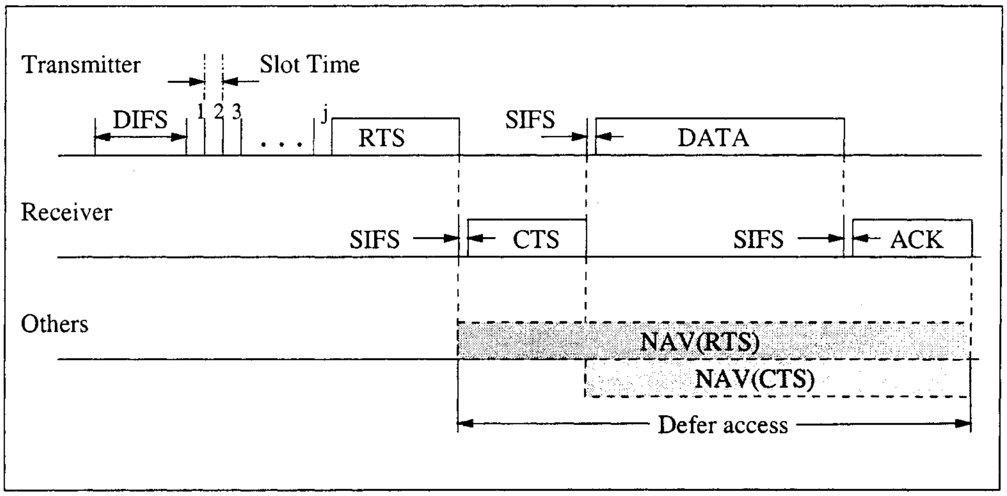


Figure 4.5: Successfully transmitted unicast data frame in the RTS/CTS mode of the IEEE802.11 standard.

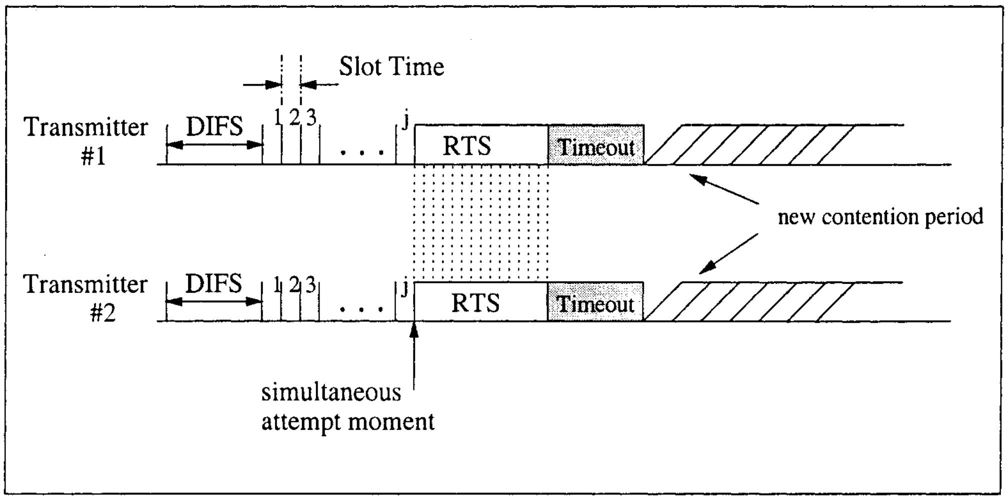


Figure 4.6: Simultaneous transmission attempt in the RTS/CTS mode leading to a collision.

the packets that already have visited a particular node are stored in *archive*(). At each slot, shown for instance in Fig. 4.5, each node checks whether its channel is idle ($ITER\_TO\_GO = 0$). If so, and if $NAV = 0$, proper actions are taken according to its most recently completed activity (e.g. transmission, reception, listening, or being idle) as explained before in Chapter 3. These actions include updating *node*(), *pkt*(), *Q*(), and *archive*() and/or initiating RTS, CTS, DATA, or ACK transmissions. Mechanisms that result in packet elimination are: blocking due to target node's full buffer (or in the generation time), dropping due to reaching some constraints such as time-to-live, maximum hopping, retransmission limit, duplication (explained below), route unavailability, and successful reception. The only verification mechanism regarding a healthy DATA transmission is through ACK exchange (Figs. 4.5 and 4.6) which itself may or may not fail. Thus, situations may arise that several copies of the same packet are hopping to the destination. Figure 4.8 illustrates such situations. In Fig. 4.8-a, node #1 sends packet A to node #2. Although the packet is received correctly at node #2, its corresponding ACK back to node #1 fails which makes node #1 to initiate a retransmission. At the same time, node #2 will forward already queued

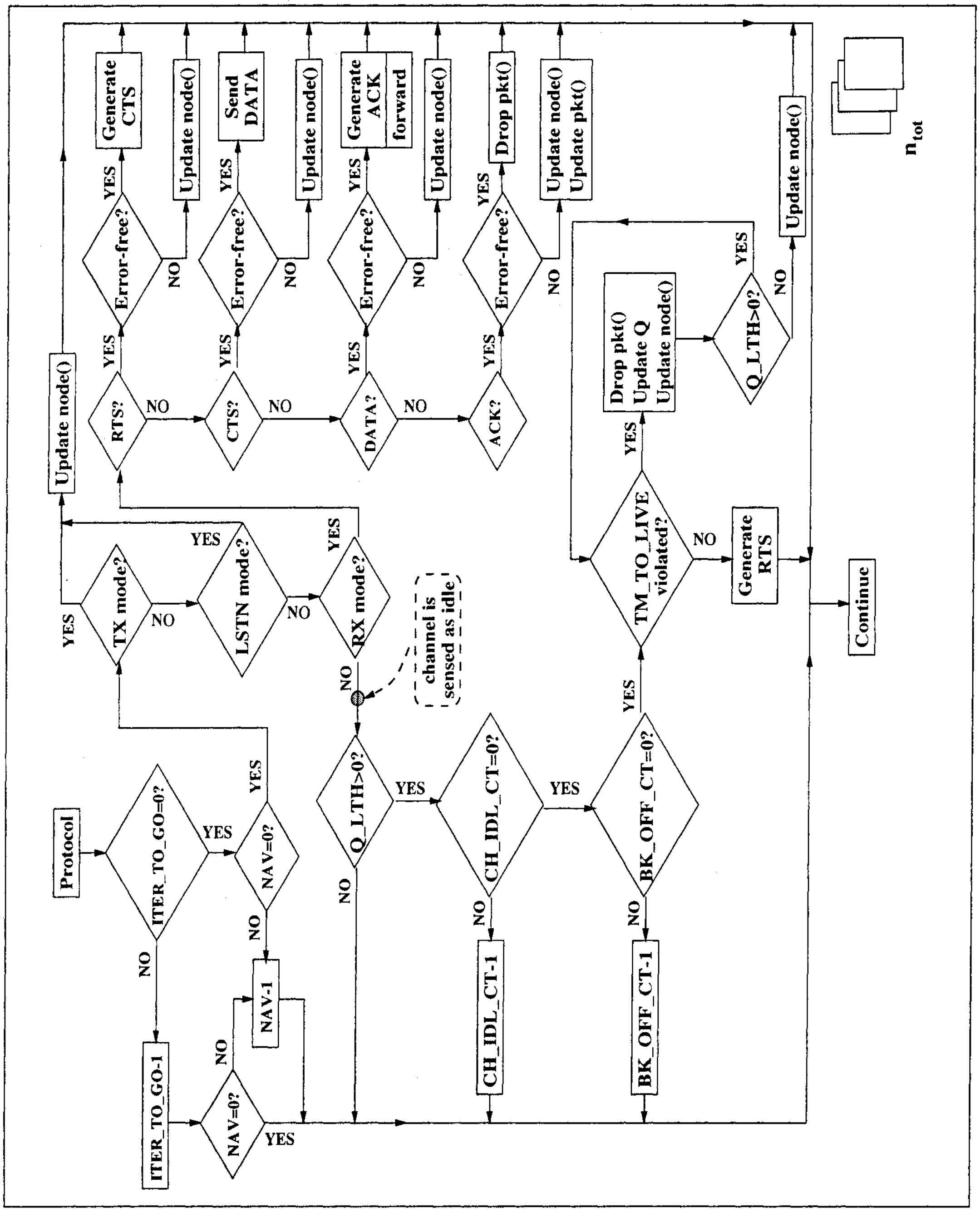


Figure 4.7: MAC layer flowchart.

packet A along its route to the destination. In Fig. 4.8-b, the same event is illustrated except that due to topology changes (as a result of mobility), retransmission of packet A is headed toward a different target, node #4, which in turn will forward it along its route to destination. To better simulate the real-life situation and account for the above possibilities, the following measures are put in place:

- Only originating node can manipulate the function $pkt()$
- Upon correct reception, a new version of the packet with a different $HOP\_VER$ is produced at the destination (The destination is now its new originator) and is queued for further forwarding
- At the time of retransmission, if the current target node is different from the previous transmission attempt, a new version of the packet with a different $CP\_VER$ is produced and the old one is discarded.

With the above provisions, the multi-version packet phenomenon is well simulated. Since all versions of a packet have the same unique sequence number being assigned at the generation time, flooding can be partially prevented. As introduced earlier, the function $archive(ID\_NO)$ encompasses the sequence number of all the packets which have already visited node # $ID\_NO$. As such, a correctly received packet's sequence number is compared against the list in $archive()$. The new version of the packet is either trashed if a match is found, or forwarded otherwise (Fig. 4.8-a). In the flowchart of Fig. 4.7, packet reception checking demands more elaboration due to its ramifications (specifically forwarding). Accordingly, it has been flowcharted in Fig. 4.9. Upon a healthy reception, a copy of the received packet is generated with incremented hop version and an acknowledgment is sent back to the transmitter. From this point on, this newly generated packet (new version) is processed by the receiving node. First, whether the packetis a duplicate is checked, and then it is

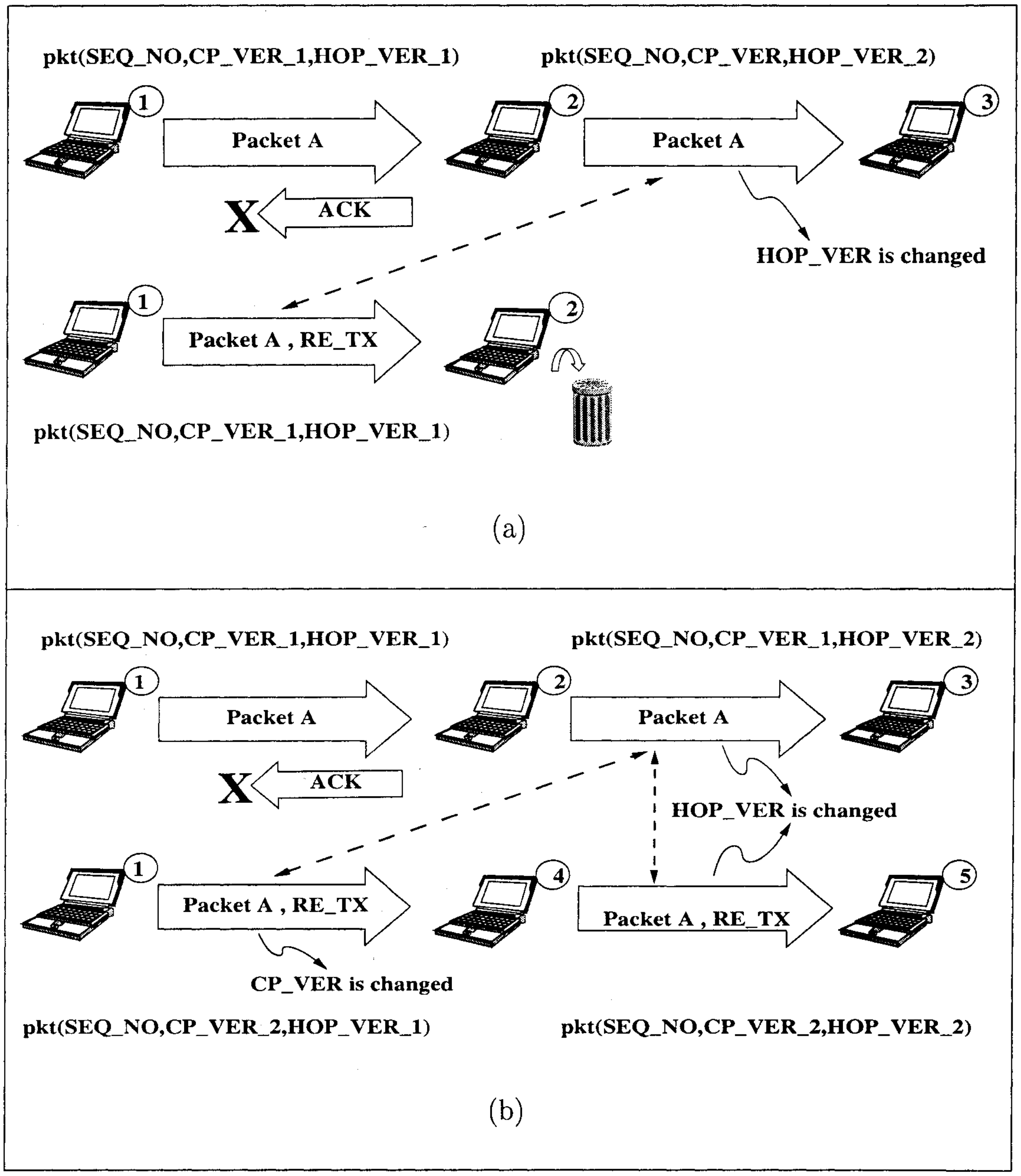

pkt(SEQ_NO,CP_VER_1,HOP_VER_1)
pkt(SEQ_NO,CP_VER,HOP_VER_2)
1
Packet A
2
Packet A
3
X
ACK
HOP_VER is changed
1
Packet A , RE_TX
2
pkt(SEQ_NO,CP_VER_1,HOP_VER_1)
(a)
pkt(SEQ_NO,CP_VER_1,HOP_VER_1)
pkt(SEQ_NO,CP_VER_1,HOP_VER_2)
1
Packet A
2
Packet A
3
X
ACK
HOP_VER is changed
1
Packet A , RE_TX
4
Packet A , RE_TX
5
CP_VER is changed
pkt(SEQ_NO,CP_VER_2,HOP_VER_1)
pkt(SEQ_NO,CP_VER_2,HOP_VER_2)
(b)

Figure 4.8: Upon an ACK failure, retransmission is forwarded to: a) the same previous target node. b) a new target node due to mobility and topology change.

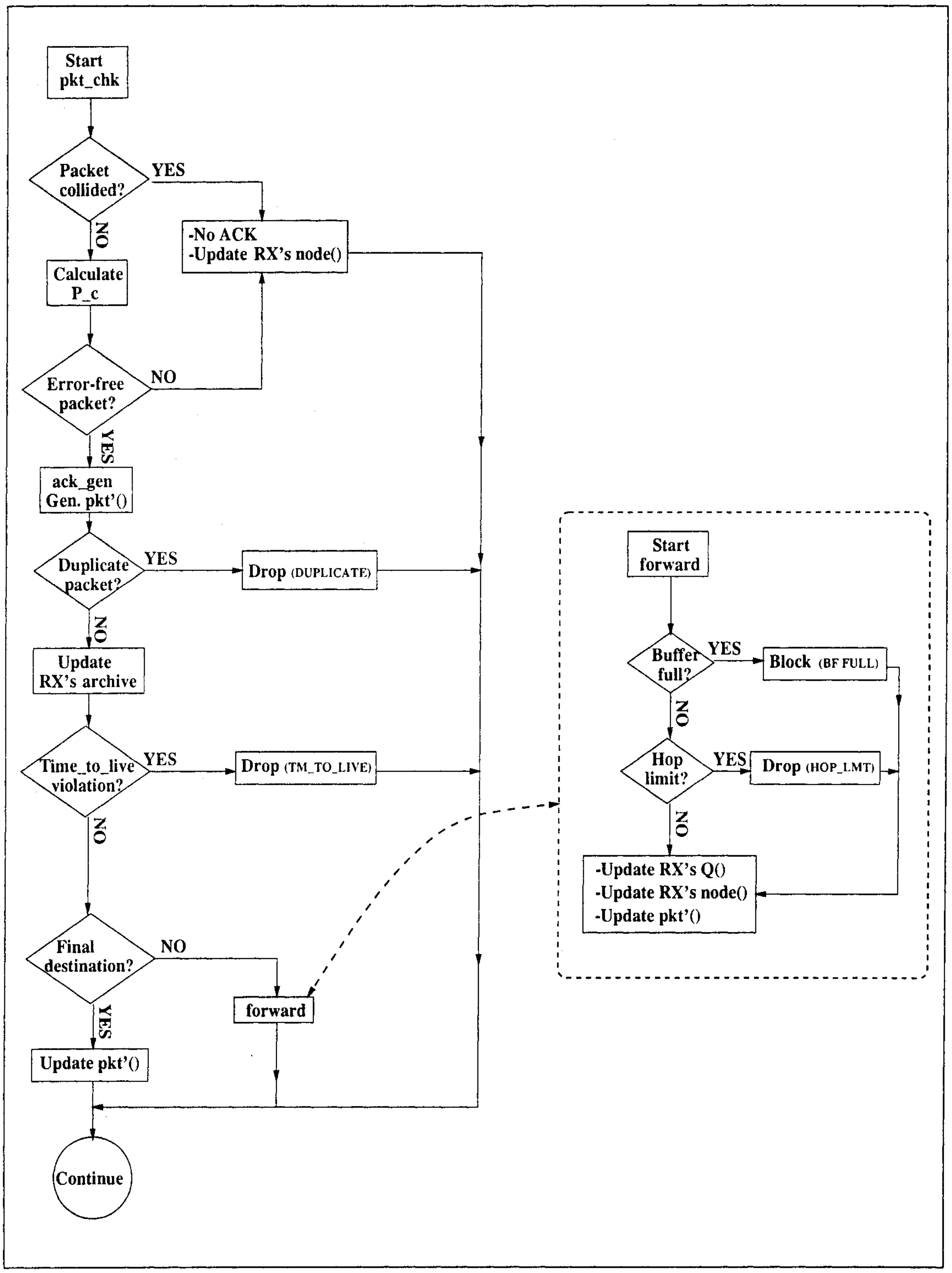


Figure 4.9: Packet reception and forwarding.

checked for time-to-live violation where in either case it is discarded. In the case of survival, if at final destination, a successful packet delivery is recorded and otherwise it is considered for further forwarding. In this stage, it may again get discarded due to a full buffer, maximum hop limit situation, or queued otherwise.

Figure 4.10 illustrates the whole simulation process and how different earlier mentioned modules are embedded.

### 4.6.2 Physical Layer

Spreading, modulation, and coding schemes are considered here. According to the FCC regulations, the DSSS system shall provide a processing gain of at least 10 dB. This is accomplished by chipping the baseband signal at 1 MHz with an 11-chip PN code. The DSSS system uses baseband modulations of BPSK (DBPSK) to provide 1 Mbit/s data rate (Fig. 4.3). Table 4.1 shows physical layer parameters corresponding to DSSS IEEE 802.11. Some MAC layer parameters (RTS/CTS/ACK bit lengths and MAC overhead bits) are also included.

Table 4.1: IEEE802.11 standard parameters for DSSS WLANs.

| Parameter | Value |
|---|---|
| Channel Rate ($r$) | 1 Mbit/s |
| RTS | 20 octets |
| CTS | 14 octets |
| ACK | 14 octets |
| Overhead(MAC) | 34 octets |
| Overhead(PHY) | 24 octets |
| DIFS | 50 $\mu$sec |
| SIFS | 10 $\mu$sec |
| Slot time | 20 $\mu$sec |

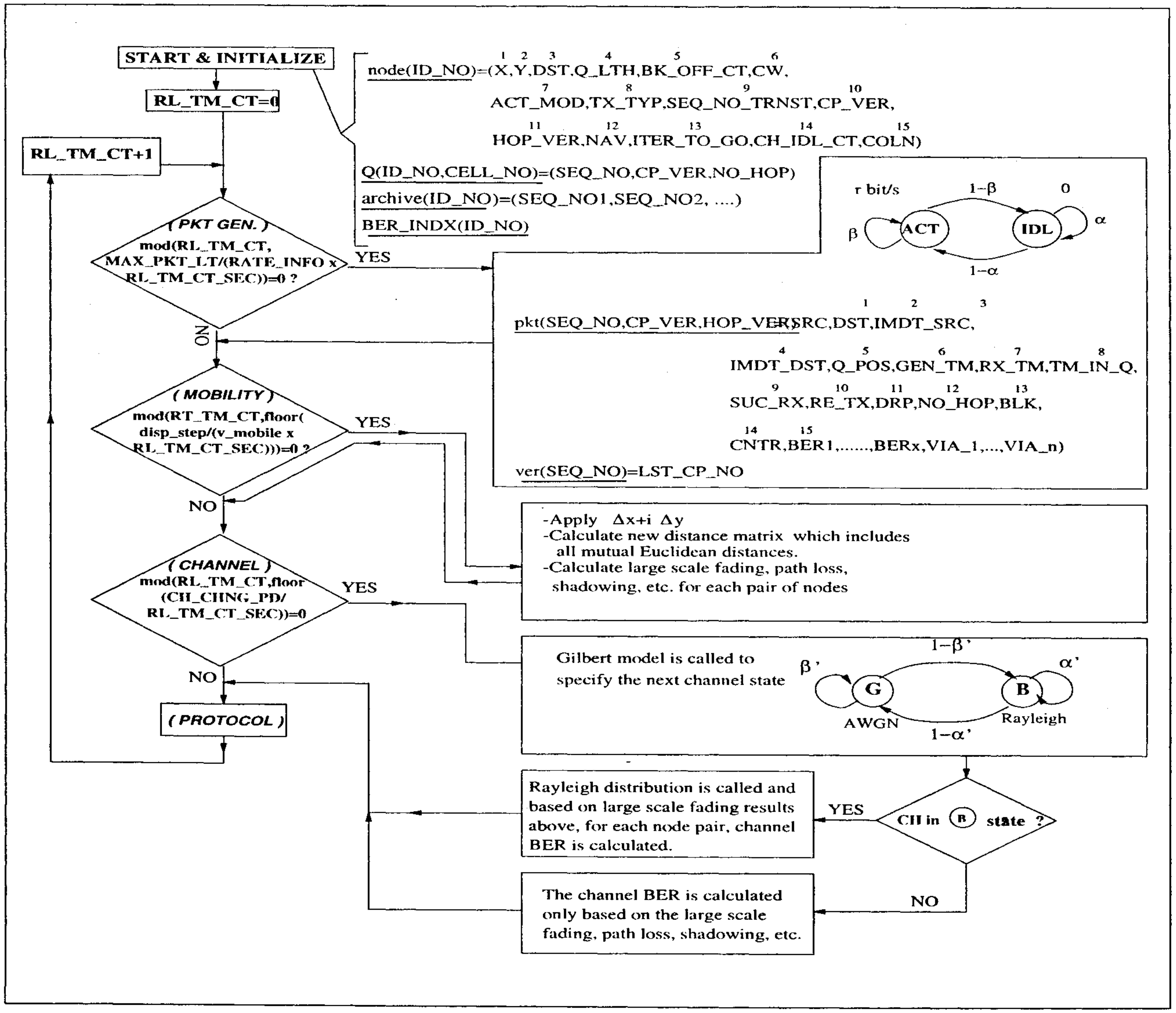

START & INITIALIZE
RL_TM_CT=0
RL_TM_CT+1
node(ID_NO)=(X,Y,DST,Q_LTH,BK_OFF_CT,CW,
ACT_MOD,TX_TYP,SEQ_NO_TRNST,CP_VER,
HOP_VER,NAV,ITER_TO_GO,CH_IDL_CT,COLN)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Q(ID_NO,CELL_NO)=(SEQ_NO,CP_VER,NO_HOP)
archive(ID_NO)=(SEQ_NO1,SEQ_NO2, ....)
BER_INDX(ID_NO)
( PKT GEN. )
mod(RL_TM_CT,
MAX_PKT_LT/(RATE_INFO x
RL_TM_CT_SEC))=0 ?
YES
NO
r bit/s
1−β
0
β
ACT
IDL
α
1−α
pkt(SEQ_NO,CP_VER,HOP_VER,SRC,DST,IMDT_SRC,
IMDT_DST,Q_POS,GEN_TM,RX_TM,TM_IN_Q,
SUC_RX,RE_TX,DRP,NO_HOP,BLK,
CNTR,BER1,......,BERx,VIA_1,...,VIA_n)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
ver(SEQ_NO)=LST_CP_NO
( MOBILITY )
mod(RT_TM_CT,floor(
disp_step/(v_mobile x
RL_TM_CT_SEC)))=0 ?
YES
NO
-Apply Δx+i Δy
-Calculate new distance matrix which includes
all mutual Euclidean distances.
-Calculate large scale fading, path loss,
shadowing, etc. for each pair of nodes
( CHANNEL )
mod(RL_TM_CT,floor
(CH_CHNG_PD/
RL_TM_CT_SEC))=0
YES
NO
( PROTOCOL )
Gilbert model is called to
specify the next channel state
1−β'
β'
α'
G
B
AWGN
Rayleigh
1−α'
CH in B state ?
YES
NO
Rayleigh distribution is called and
based on large scale fading results
above, for each node pair, channel
BER is calculated.
The channel BER is calculated
only based on the large scale
fading, path loss, shadowing, etc.

Figure 4.10: Simulation process flowchart.

The presence of fading in wireless environments motivates the use of channel codes. Channel codes mitigate the adverse effect of fading in wireless channels by adding redundancy and memory to the transmission. Turbo codes are a class of error correction codes that were introduced along with a practical decoding algorithm in 1993. The importance of turbo codes is that they enable reliable communications with power efficiencies close to the theoretical limit predicted by Claude Shannon [75]. The FEC scheme employed in our simulation is a turbo code with the specification as in Table 4.2.The turbo encoder uses two component recursive systematic convolutional (RSC) codes in parallel with constraint length $K$, generator polynomials $G_0$ and $G_1$, $k$ bit input, and $n$ bit output. Encoder/decoder configuration is shown in Fig. 4.11. Regarding interleaver, we consider two types: one which shuffles fixed short size control packets (RTS,CTS, and ACK) to be a block interleaver, and the other one which deals with data packets to be a random separated interleaver. While this coding scheme consumes half the bandwidth in the first place, it is hoped it will retrieve much more through BER improvement. However, there is no claim about the optimality of this coding scheme and no effort is made to evaluate its efficiency since it is not in the main theme of this thesis. Without further discussion over coding details, focus is turned toward how transmission health is assessed in our simulation. For the short packets, depending on the received SNR, corresponding BER is extracted from BER vs. SNR curve in AWGN/BPSK case with above employed coding scheme ( using results in [76]-Figure 5.26). Since short packets experience almost the same channel quality except at boundary situations, packet error rate (PER) is then:

$$PER = 1 - (1 - BER)^{RTS_b/CTS_b/ACK_b} \tag{4.15}$$

where $RTS_b$, $CTS_b$, and $ACK_b$ are the number of bits in the short RTS, CTS, and

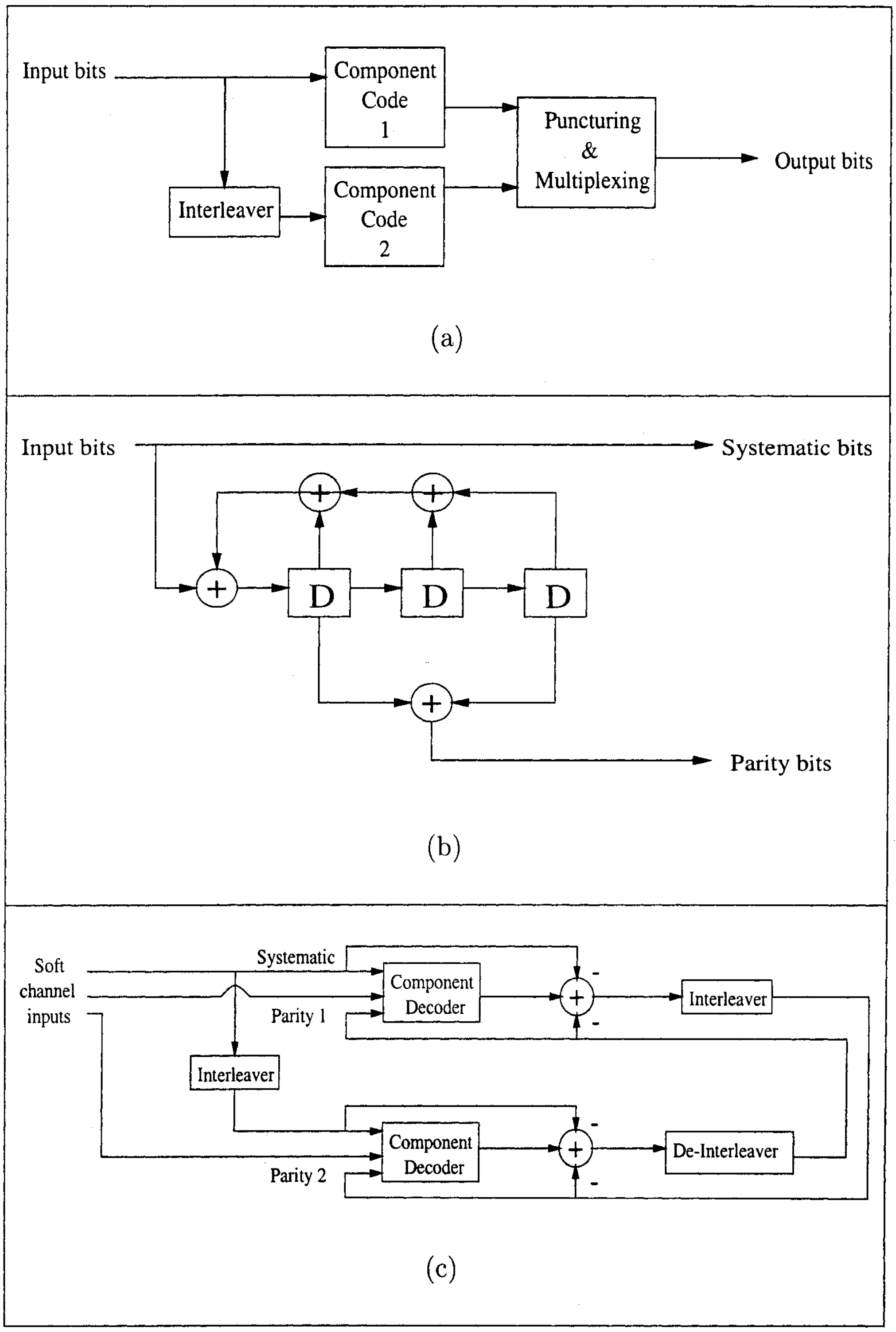


Figure 4.11: Turbo encoder/decoder configuration. a) Encoder Schematic. b) RSC encoder. c) Turbo decoder schematic.

Table 4.2: Standard turbo encoder and decoder parameters used.

| | |
|---|---|
| Component encoders | Two identical Recursive Systematic Convolutional codes |
| RSC parameters | $n = 2$, $k = 1$, $K = 3$, $G_0 = 7$, $G_1 = 5$ |
| Puncturing used | Half parity bits from each component encoder transmitted give half-rate code |
| Component decoders | Log-MAP decoder |
| No. of iterations | 8 |

ACK packets. Data packets depending on their length and time coherence of the channel ($T_c$ in (4.11) experience several channel qualities represented by $BER1, ..., BERx$ in *pkt*() function (Fig. 4.10). The PER is accordingly calculated as:

$$PER = 1 - (1 - BER_x)^{mod(DATA, L_{T_c})} \prod_{m=1}^{x-1} (1 - BER_m)^{L_{T_c}} \qquad (4.16)$$

where $L_{T_c}$ denotes the equivalent bit length to $T_c$, $mod(DATA, L_{T_c})$ accounts for remainder of dividing $DATA$ by $L_{T_c}$, and $x$ is the number of coherence times within one packet. Finally, in the simulation program, the above calculated PER is compared against a uniform-randomly selected number between zero and one to result in a failed or successful transmission.

## 4.7 Simulation Results

In this section, the effects of different choices of MAC layer parameters as well as other external network parameters (e.g. mobility, traffic load, etc.) on the system performance are simulated. In particular, MAC layer packet length (the same fragmentation issue of Chapter 3 is still considered in this simulation by varying the subject packet lengths), maximum life time of a packet (time-to-live, TTL), maximum number of times a packet is retransmitted ($ReTX_{max}$), and a node's maximum buffer size ($Q_{max}$) are examined. This is done in different network situations by varying the percentage of active nodes ($P_{active}$), the percentage of mobile nodes ($P_{mobile}$), and mobile node speed ($v$) while the total number of nodes present in the network ($n_{tot}$), node traffic generating rate, and network geographical dimensions remain fixed at 30, 64 Kbps, and $400m \times 400m$ respectively throughout the simulation. Traffic per user rate of 64Kbps can be representative of a wide variety of real-time applications. A $400m \times 400m$ area together with the node's 150 m coverage radius (which is a typical coverage radius for wireless interfaces in the market) represent our intended multi-hop scenario. Since we are aiming at real-time application support, a larger area means more hops which in turn contributes to unfavorable delay accumulation. The area is limited such that the majority of the the source-to-destination transmissions can take place with number of hoppings less than $HOP\_LMT = 4$. However, the simulation environment developed has no limitation in handling areas of larger sizes. 30 users in the network area is a normal user population and assures a reasonable availability of next hop nodes as well. Parameters used in the simulation are tabulated in Table 4.3. Source and channel models parameters are chosen to be consistent with the corresponding parameters in [17]. The other parameters are partially acquired from [23] and Cisco WLAN specification guides. The simulation time is fixed at 1.0 s to accommodate at least a couple of time-to-lives and letting all the modules activate. In all the simulation results, each point is acquired by applying batch means method [60] over 64 ($8 \times 8$) simulation runs to obtain a good level of confidence. Two major measures considered herein as performance indicatives are end-to-end (e.t.e) delay and packet failure rate (PFR, this is equivalent to the probability of fail to deliver, $P_f$, in Chapter 3) wherein such a delay has been presented as normalized with respect to subject DATA packet time length. The delay includes the time duration from the moment of packet generation until its healthy reception. Not included are packetization delay, interleaving delay in turbo encoding, propagation delay, delays related to turbo decoder iterations, and other packet processing times at intermediate and destination nodes. However, the latter are both fixed and negligible compared to random transmission and queuing delays. In each simulation run, e.t.e delay is calculated by averaging over all healthy received packets. Packet failure rate is the ratio of the number of packets dropped to the total number of generated packets throughout the duration of simulation . Different limit violations in the aggregate, contribute to the total number of dropped packets.

Also measured are the packet success rate (PSR, also called the probability of successful delivery in Chapter 3) and queue occupancy. These two parameters along with the PFR, give a better understanding of packets statistics at the end of the simulation run. Packet success rate is the ratio of the number of packets successfully received to the total number of generated packets, and queue length (QL) shows the average queue length of a typical node at the end of the simulation run. $1 - PSR - PFR$ quantifies the number of pending packets in the user queue.

Figures 4.12-4.17 illustrate how the above mentioned performance measures are affected by varying DATA packet length. To this length, MAC and PHY layer overheads are added and the result is turbo coded and transmitted.

In each subplot of Fig. 4.12, a group of curves corresponding to different values of time to live, 0.1s-0.4s are drawn. In the upper leftmost plot, the normalized delay is decreased by adopting longer packet lengths and less time-to-live values. However, this decrease becomes less significant upon passing 4000-bit DATA packet length. Moreover, from this point (4000-bit) on, the delay corresponding to $TTL$ = 0.3s surpasses the one corresponding to $TTL$ = 0.4s. In the upper rightmost plot, packet failure rate decreases with higher TTLs as may be expected. Again passing 6000-bit DATA packet length, lower $TTL$ of 0.3s gives better PFR than $TTL$ = 0.4s. It

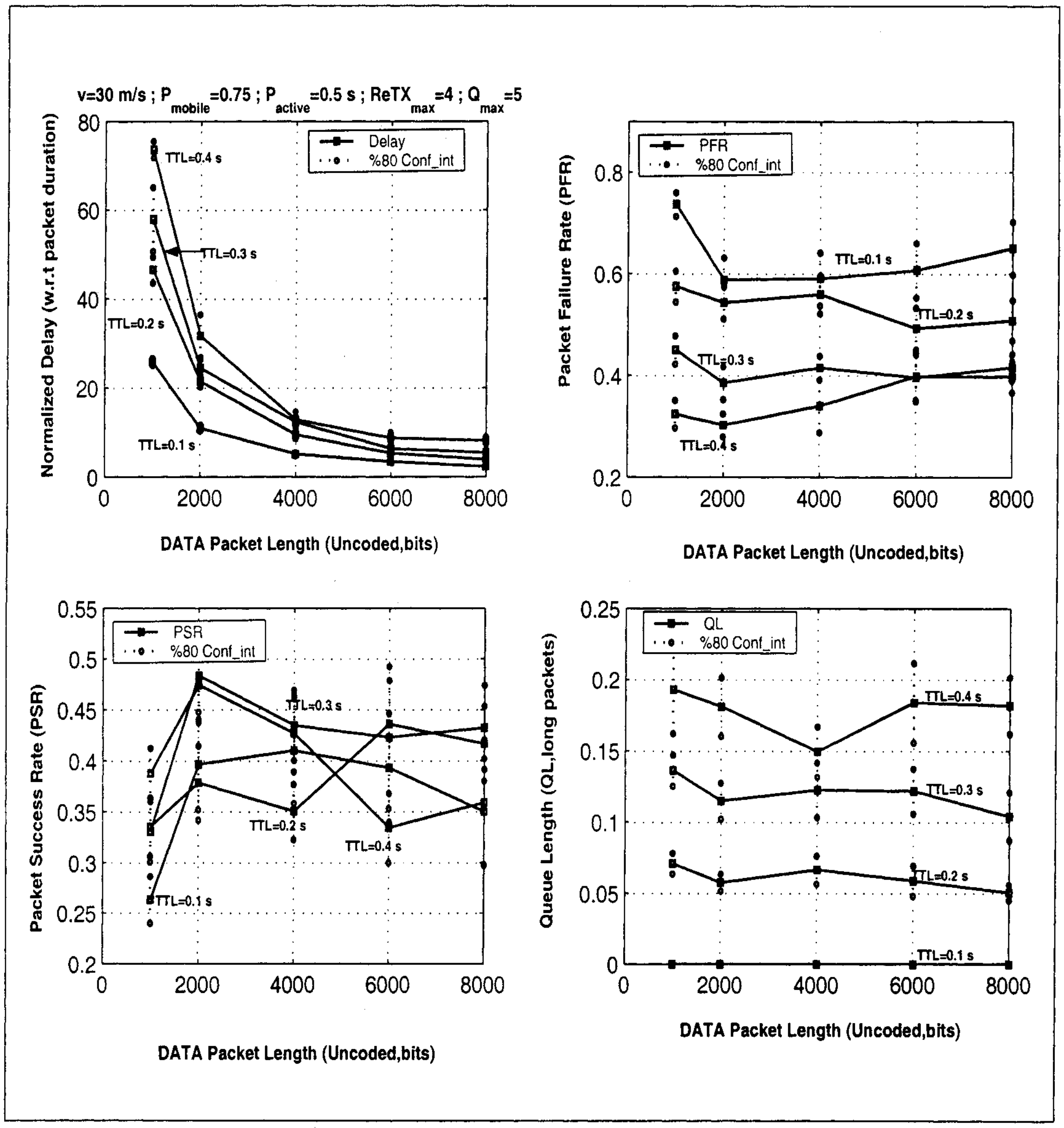


Figure 4.12: Performance measures of the system versus DATA packet length with time-to-live ($TTL$) as parameter.

Table 4.3: Parameters used in the simulation.

| | |
|---|---|
| Network parameters | |
| $n_{tot}$ | 30 |
| Node's coverage radius | 150 m |
| Network area | 400m×400m |
| Displacement step ($\Delta$) | 5 m |
| Simulation time | 1 sec |
| Source model parameters | |
| Node's traffic rate | 64 Kbps |
| Avg. talk spurt period | 1.0 sec |
| Avg. silence period | 1.35 sec |
| Max. generated packet length delivered to the link layer | 8000 bits |
| Channel parameters | |
| Path loss exponent ($n_{PL}$) | 3.2 |
| Break point distance ($d_0$) | 10 m |
| Standard deviation of shadowing ($\sigma$) | 8 dB |
| $\beta'$ | 0.967 |
| $\alpha'$ | 0.9 |
| Miscellaneous parameters | |
| Transmitter power ($P_T$) | 17 dBm |
| Noise floor ($NF$) | -80 dBm |
| Simulation slot time | 10 $\mu$sec |
| Maximum hopping | 4 hops |

means that from this point on, other mechanisms of packet drop rather than time-to-live dominate. In the lower leftmost plot illustrating PSR variations, while $TTL = 0.3, 0.4$s give better success rates for shorter packet lengths up to 4000 bits, $TTL = 0.2$s does so for longer packets. Collectively, they suggest a preferable performance in the range of 4000-6000 bits of DATA packet length and $TTL = 0.2 - 0.3$s, although by trading delay for loss or vice versa, we may end up with other choices. The fourth plot, queue length, shows a very predictable group of curves without any particular suggestion.

A stronger FEC lessens the number of retransmissions needed. Figure 4.13 shows how the performance measures vary against DATA packet length with the number of retransmissions as parameter. Looking at the delay plot, despite multiple curve crossings, all the curves descend by a climbing packet length. The descents become less significant and particularly indistinguishable from the 4000-bit packet length on. On the other hand, in PFR plot and for the same range of DATA packet lengths (4000-bit onward), $ReTX_{max} = 2$ undoubtedly is the best choice. This choice is also confirmed by PSR plot. On the queue length plot, unexpectedly at times, lower queue occupancy is observed for higher values of $ReTX_{max}$ particularly $ReTX_{max} = 3$ and $ReTX_{max} = 4$. This may be attributed to packet drops due to time-to-live limit violation because of higher waiting times corresponding to higher $ReTX_{max}$ cases. Figure 4.14 shows the effect of varying queue size and DATA packet length on the system performance measures. The upper leftmost plot, delay, gives almost the same trend and conclusion as the corresponding plot in Fig. 4.13. Lengthening the queue has conflicting secondary drop effects such as less packet drop due to queue overflow and higher packet drop due to time-to-live violation. Depending on which one has the dominating effect, a better or worse PFR is observed. The PFR and PSR plots suggest that ($Q_{max} = 2, DATA = 6000$ bits), ($Q_{max} = 5, DATA = 4000$ bits), and ($Q_{max} = 5, DATA = 8000$ bits) are the preferable choices. The QL plot shows how the queue occupancy grows with longer queue except for some crossings mostly for $Q_{max} = 3$ and around $DATA = 2000$ bits.

Now we turn our attention toward how other parameters affect the system performance. To evaluate the performance under different traffic loads, the ratio of the active users (generating traffic) is varied over $P_{active} = 0.1 - 0.5$. Another way of increasing the traffic load is to increase each node's traffic generation rate. However,

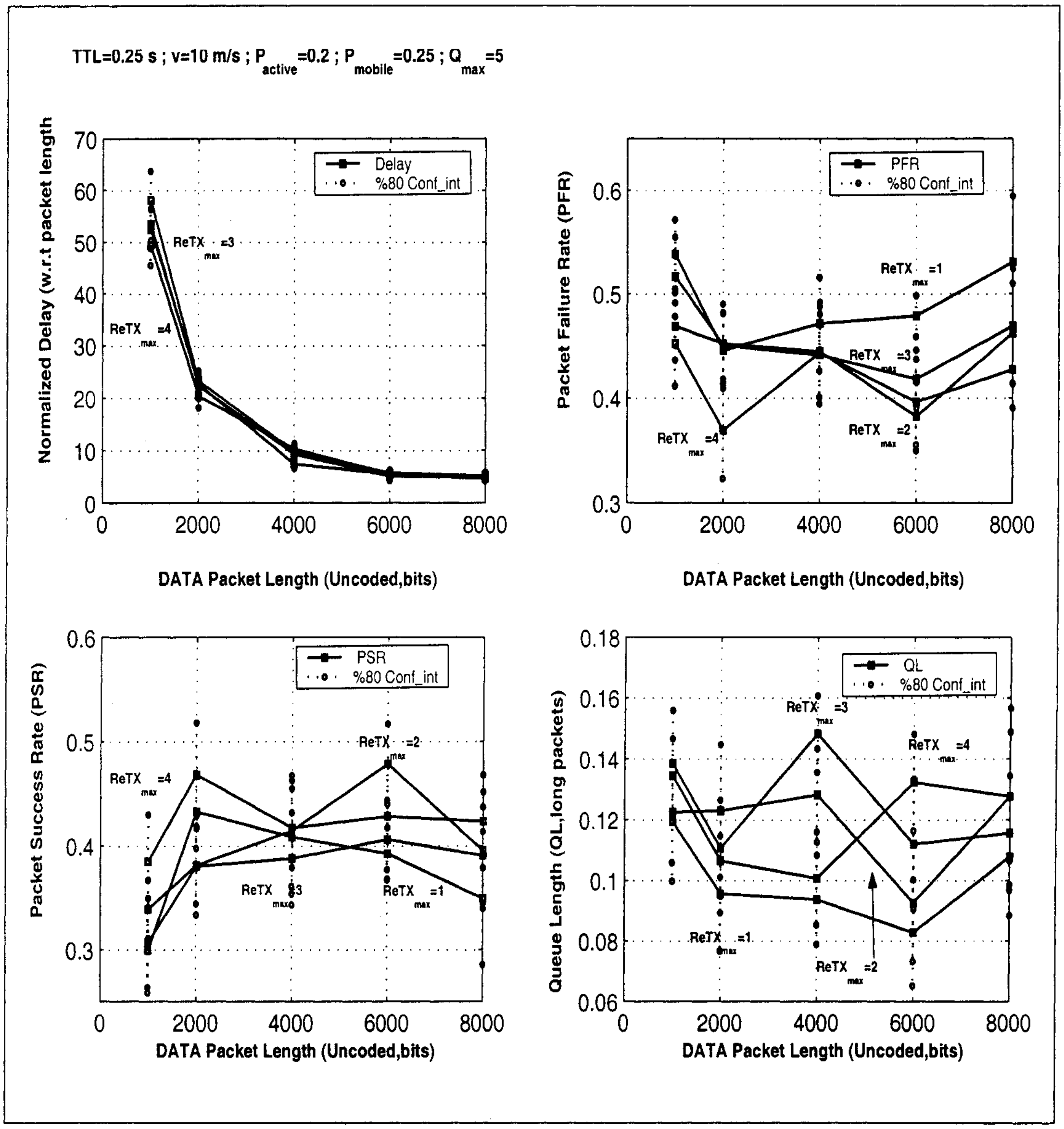


Figure 4.13: Performance measures of the system versus DATA packet length with maximum number of retransmissions ($ReTX_{max}$) as parameter.

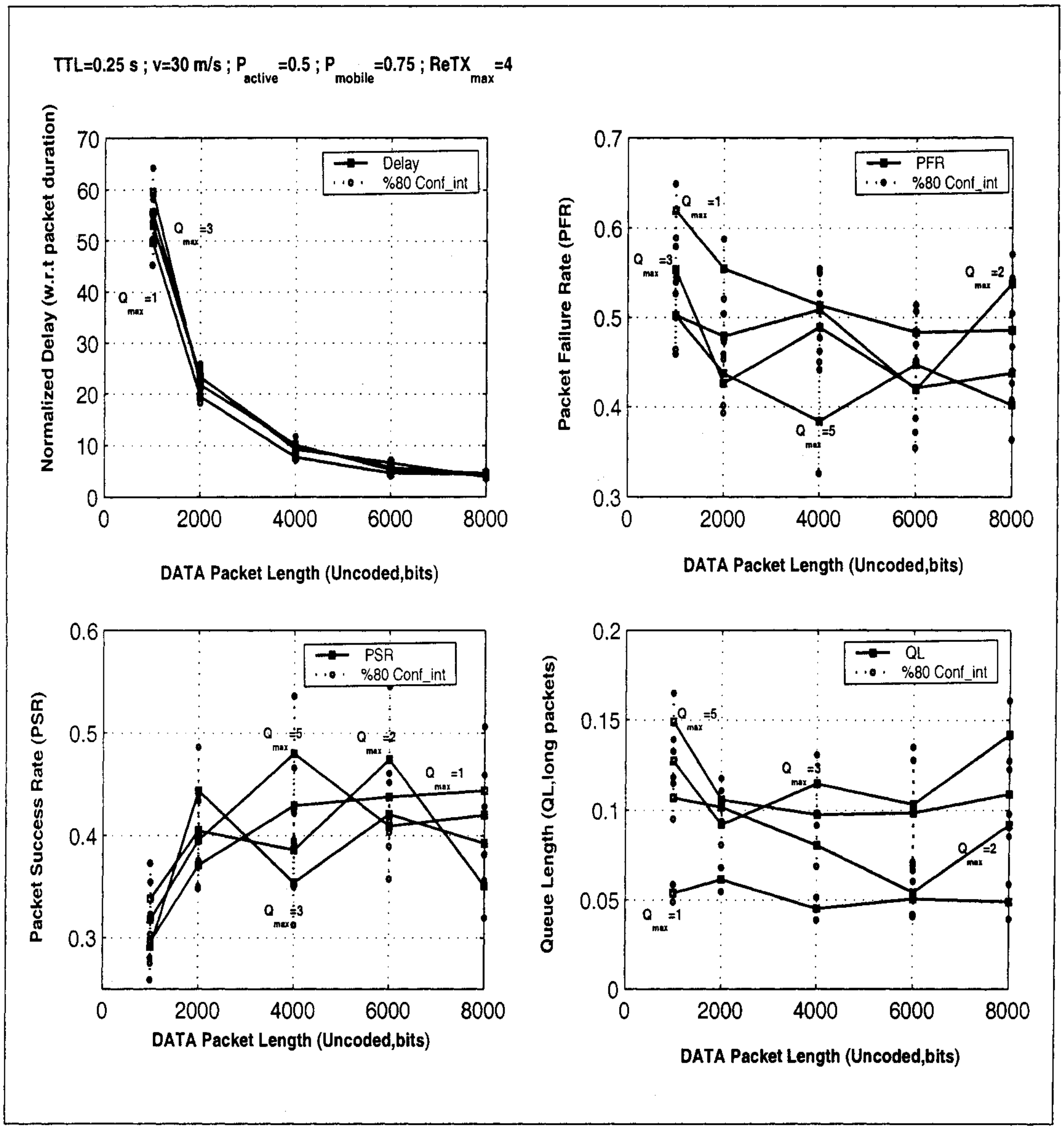


Figure 4.14: Performance measures of the system versus DATA packet length with maximum queue length ($Q_{max}$) as parameter.

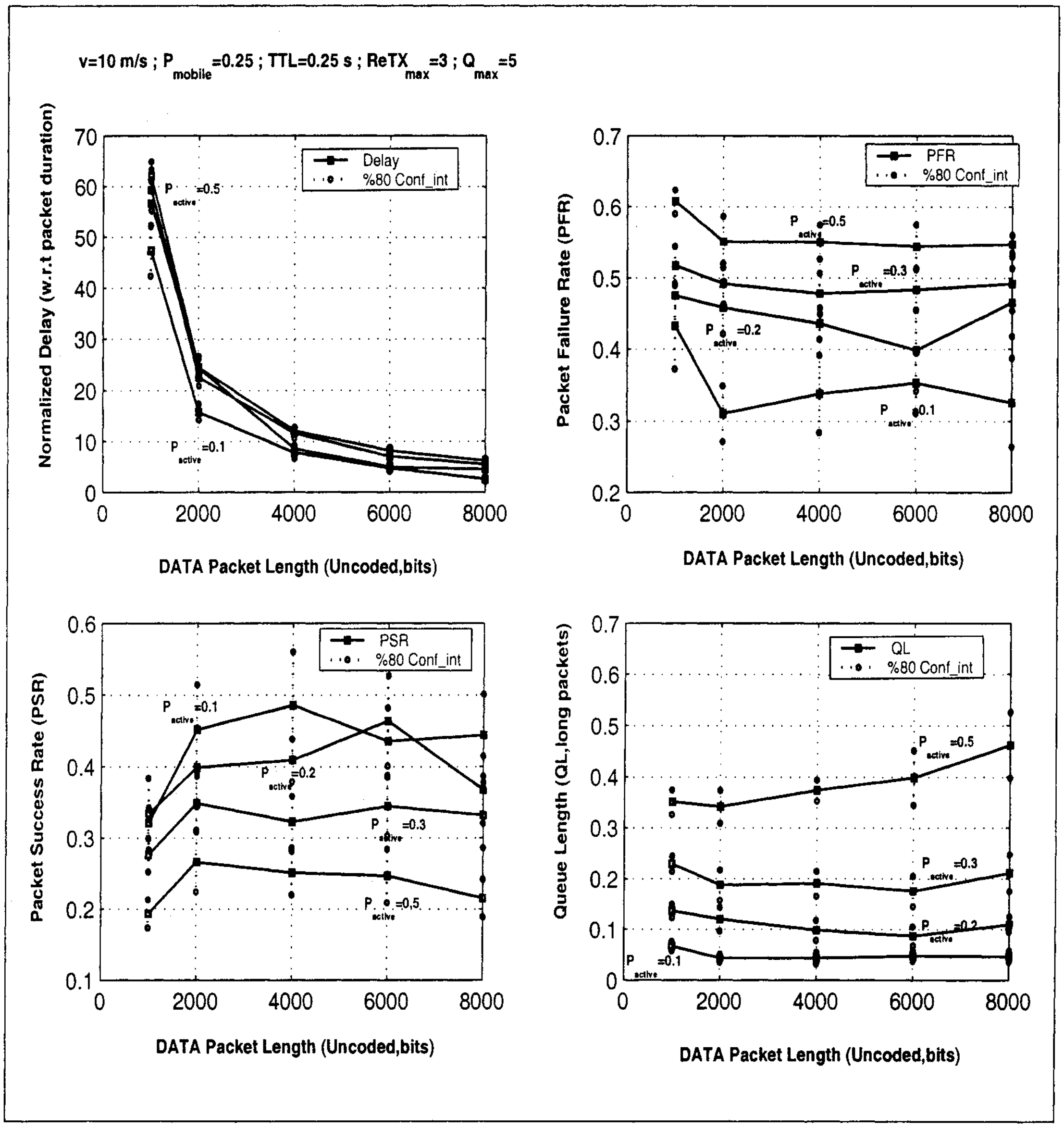


Figure 4.15: Performance measures of the system versus DATA packet length with the percentage of active nodes ($P_{activ}$) as parameter.

since the former is a distributed load increase rather than local in the latter, it is more demanding in terms of bandwidth. In Fig. 4.15, all the plots in general show more degradation with higher loads. Again, delay plot favors longer DATA packets but unlike earlier delay plots, decline continues till an 8000-bit packet length though losing pace gradually. The PFR and PSR plots show that for high volume of traffic case ($P_{active} = 0.3, 0.5$), DATA packet lengths of 2000 bits and above provide no considerable gain in terms of packet delivery. For medium traffic volume case (($P_{active} = 0.2$), 6000-bit packet serves as optimum in terms of both delay and packet delivery. In a light traffic volume corresponding to $P_{active} = 0.1$, even a longer packet size of 8000 bits is beneficial to combined delay-delivery system performance.

The impact of mobility on the system performance is evaluated in two ways. First, by varying the number of mobile users in the system ($P_{mobile}$) with the same speed and then changing the speed of a fixed number of mobile users, particularly $P_{mobile} = 0.25$. Figure 4.16 illustrates the performance reaction to volume of mobility. The delay plot shows the usual descending curves toward longer packets while no major differences are observed due to differing mobilities. Packets longer than 4000 bits give better delay performance. For the same range of packet lengths, the PFR and PSR plots suggest that 6000-bit length is an optimum for all mobility volumes. Interestingly, zero and high mobility volumes ($P_{mobile} = 0, 0.75$) give better delivery performance than low and medium volumes ($P_{mobile} = 0.25, 0.5$). The delay plot of Fig. 4.17 shows slightly higher delays for higher speeds while maintaining the same descending pattern with respect to the packet length. The PFR and PSR plots suggest a near-optimum packet length of 6000 bits in the range of 4000-8000 bits for all speeds. Moreover, in both plots, delivery performance improves with speed. One very important point is large expected errors (relatively large confidence intervals) in PFR and PSR plots of the mobility results in Figs. 4.16 and 4.17. This makes the above related conclusions less certain. Faster mobility causes the packet in transit to

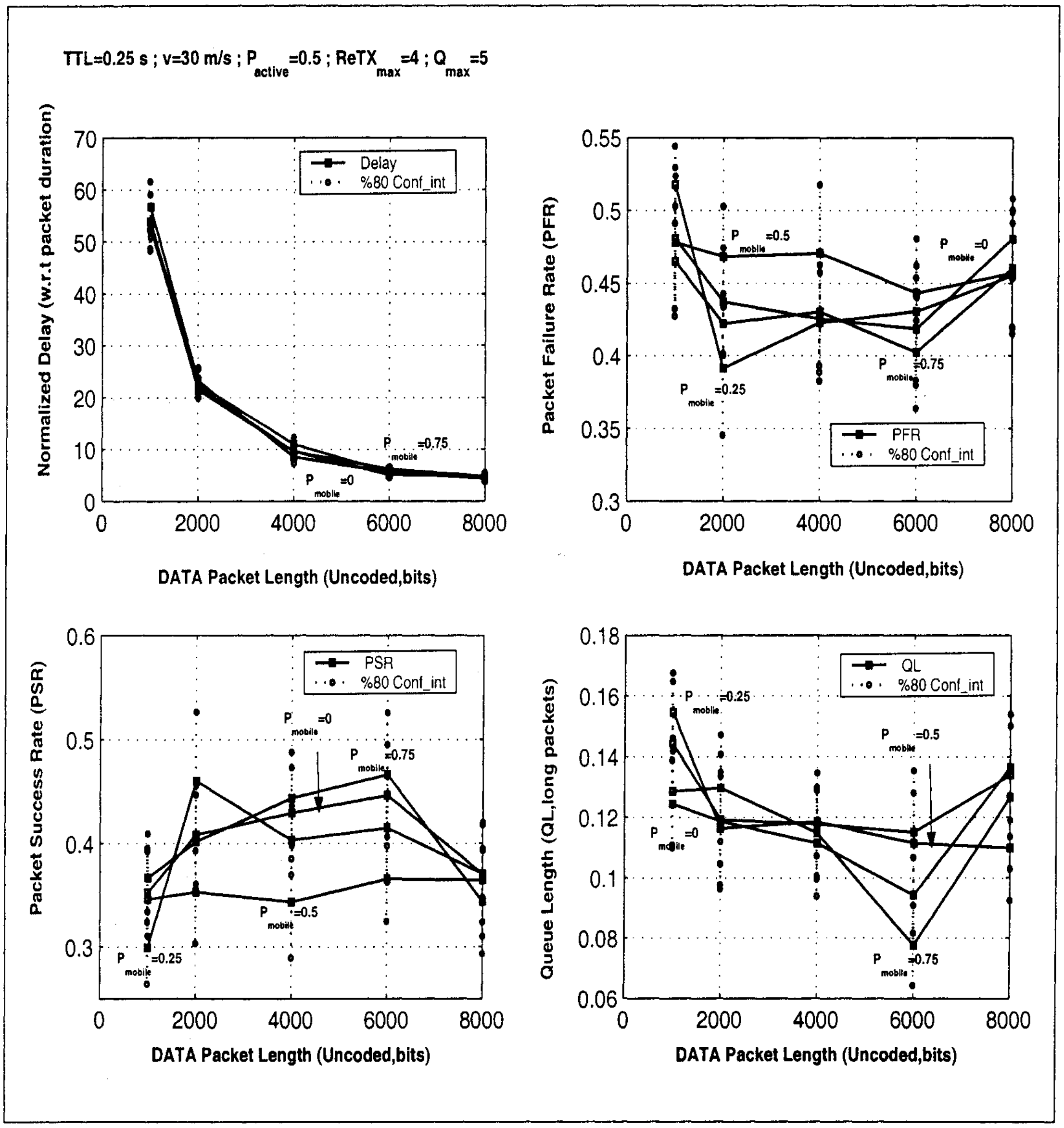


Figure 4.16: Performance measures of the system versus DATA packet length with the percentage of mobile nodes ($P_{mobile}$) as parameter.

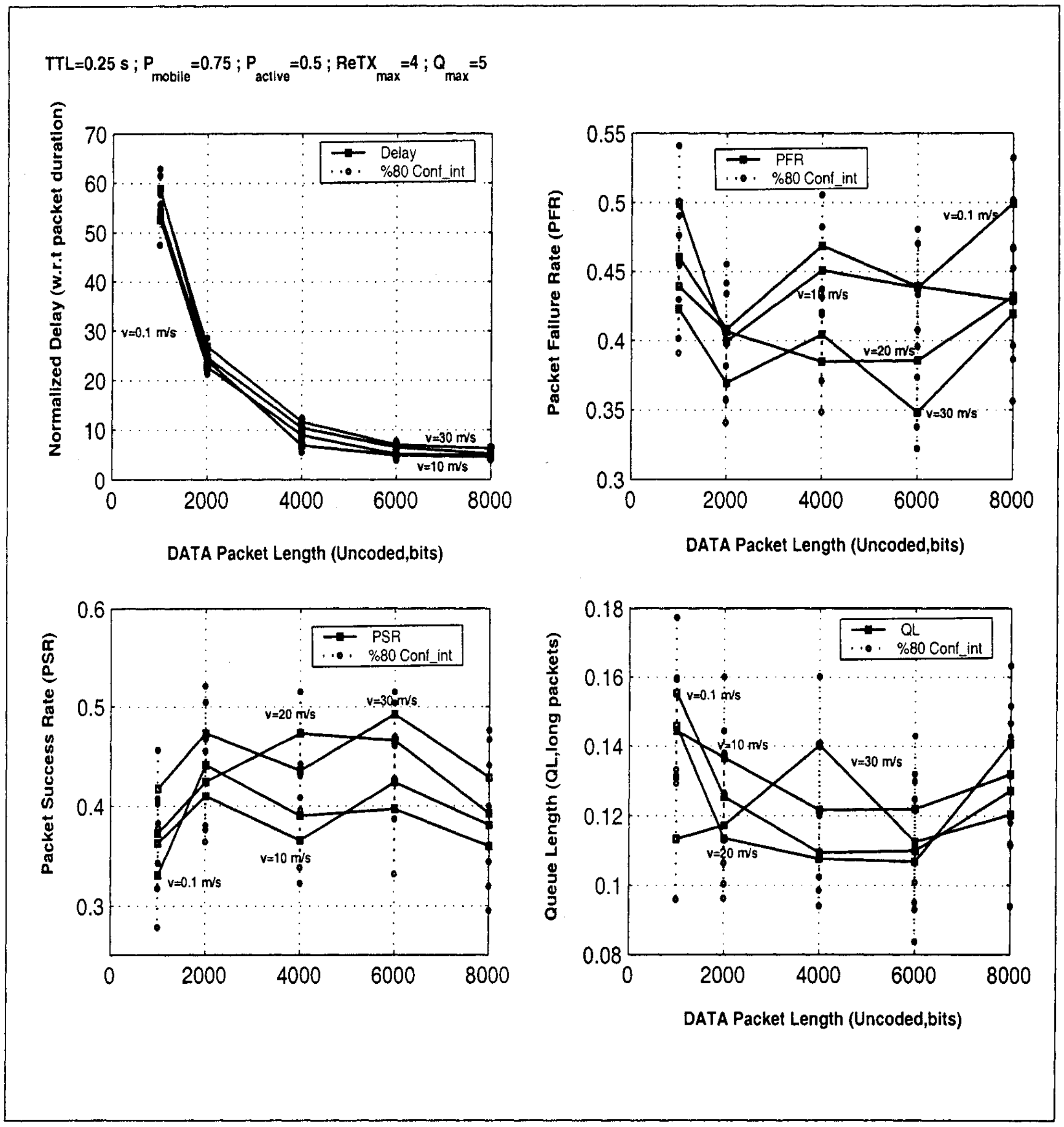


Figure 4.17: Performance measures of the system versus DATA packet length with the mobile nodes' speed ($v$) as parameter.

experience different channel qualities which can be constructive or destructive in a completely random fashion. In a similar way, faster topology changes may improve or degrade availability.

In interpreting the quantities of the performance measures, it should be noted that the results pertain to pure 802.11 protocol and no extra efforts such as revising the protocol, clustering, etc have been made.

## 4.8 Summary

Going from single-hop to multi-hop scenario makes an analytical approach toward performance evaluation almost impossible. On the other hand, a simulation approach enables us to investigate a greater number of phenomena such as hidden terminal, exposed terminal, mobility, channel fading, etc. The simulation comprising different modules; traffic generator, mobility, wireless channel, and IEEE 802.11 protocol is intended to investigate system performance measures: delay, packet failure rate, and packet success rate. These performance measures are studied under different environment scenarios in terms of mobility volume and speed, and traffic load. However, due to the function-like structure of the information flow through the simulation process, many other interesting outputs such as delay per hop, queuing and transmission delays rather than e.t.e, and failed delivery with distinction of cause (drop or block) are easily extractable. The parameters subject to optimization considered herein are packet length, time-to-live, user maximum queue length, and maximum number of retransmissions of a failed attempt. Similarly, the simulation environment developed allows us to consider many more parameters of interest. To avoid the numerousness of the pictorial results however, we refrained from including any more outputs and parameters.

Most of the performance measures above do not demonstrate a monotonic trend against the parameters considered and this shows the importance and necessity of the optimum values sought. While a higher $TTL$ allows a particular packet to stay longer in the system and have more chances of delivery, it contributes to a more crowded system and in turn more chances of collision and failure. In the same way, lengthening the queue has conflicting secondary drop effects such as less packet drop due to queue overflow, and higher packet drop due to time-to-live violation. The same reasoning prevails for the opposite impacts of retransmission limit. The only monotony seen, is the decreasing of the normalized delay versus the packet length. The declining behavior, though dependent on other parameters, is generally more intensive toward short packet lengths and less intensive otherwise. This can mostly be attributed to the ratio of the overheads to the data part of the transmitted packet.

# Chapter 5

# Conclusion and Future Work

## 5.1 Conclusions

To evaluate the performance of the ad-hoc IEEE 802.11 WLAN versus different system parameters and environment conditions, analysis and simulation approaches were employed. The former served to mathematically model a single-hop WLAN cluster while the latter served to simulate a multi-hop WLAN scenario. In order to optimally select system parameters to fulfill specific needs, a mathematical description of the system turns to be a lot helpful in observing the trend of any parameter changes made. Though not always possible, reasonable assumptions and simplifications may lead to analysis models with the required accuracy. However, simulation is the only means for performance evaluation of multi-hop ad-hoc networks due to the numerousness and complexity of variables involved. The performance measures studied are of particular interest to real-time applications.

### 5.1.1 Mathematical Modeling of IEEE 802.11 Ad-Hoc Mode

Asynchronous mode of operation of IEEE 802.11 WLAN, adapted for real-time traffic, was modeled by a state-diagram pair. The adjustable system parameters considered are packet fragmentation, node buffer size, and maximum allowable retransmission while environmental conditions are parameterized by the total number of users, network traffic, and channel BER. The model is capable of accommodating many other parameters of interest as well. Results show that over a range of BERs typical to wireless channels ($10^{-4} - 10^{-6}$), choices of fragmentation factor, buffer size and retransmission limit can be found which optimize a specific performance measure or their joint weighted measure. Also, it is noted that throughput results are not affected by varying a node's buffer size, which just contributes to varying $P_{block}$. A new per fragment channel contention outperforms the per packet contention adopted by IEEE 802.11, for particular choices of parameters and environment conditions.

### 5.1.2 Simulation Study of Multi-hop, Mobile, Ad-Hoc WLAN

In order to investigate the performance of the DCF in multi-hop scenario, a simulation environment was developed. The simulation comprising different modules, traffic generator, mobility, wireless channel, and IEEE 802.11 protocol was intended to evaluate system performance measures such as delay, packet failure rate, and packet success rate. Here we are not considering throughput since the maximum bandwidth available can not clearly be defined due to the multi-channel nature of multi-hop. Most of the performance measures above do not demonstrate a monotonic trend against the parameters considered, and this shows the importance and necessity of the optimum values sought. While a higher $TTL$ allows a particular packet to stay longer in the system and have more chances of delivery, it contributes to a more crowded system and in turn more chances of collision and failure. Similarly, lengthening the queue has conflicting secondary drop effects such as less packet drop due to queue overflow, and higher packet drop due to time-to-live violation. The same reasoning prevails for opposite impacts of retransmission limit. The only monotony seen, is the decreasing of the normalized delay versus the packet length. The declining behavior though dependent on other parameters, is generally more intensive toward short packet lengths and less intensive otherwise. This can mostly be attributed to the ratio of the overheads to the data part of the transmitted packet.

An important observation is that apart from the maximum delay which was constrained to $TTL$ (which was chosen to fulfill many of the time-sensitive applications requirements), other measure results do not seem promising in view of real-time applications. However, one should consider that the above is the pure 802.11 DCF mode of operation in the multi-hop situation without any possible priority, route reservation, and clustering considerations. Hence, it can be regarded as almost the worst case performance. Finally, it is reminded that, being beyond the scope of this work, no effort was made to employ the optimum FEC scheme.

## 5.2 Recommended Future Research

Multi-hop operation of the ad-hoc 802.11 WLANs can be improved in many ways:

- **Smarter Multiple Access Technique:** Random access technique is the basic access means and also a decisive factor in terms of performance limit, in wireless networks without a centralized coordination. All the priority schemes discussed in section 2.3.3.1 are implemented in an isolated manner inside each individual node. However, a distributed priority scheme proves closer to an ideal TDM-like channel sharing.

- **GPS-Based Routing:** Routing is one of the most important challenges of multi-hop communications. Only a few works benefiting from the existing GPS technology can be tracked down in the literature. GPS routing combined with directional antenna technique seems to lead to more favorable performance results.

- **Clustering:** Successful dynamic clustering, imitates the behavior of the infrastructured mode of operation which is without a doubt, suitable to time-bounded services. However, any such attempt should seriously consider compatibility with the dominating standard, IEEE 802.11.

- **WLAN/3G Combined Service:** Accessibility of 3G, affordability and higher data rates of WLAN, can together materialize the seamless high speed wireless data connectivity. A mobile user roaming around is connected to the wired network through WLAN, and whenever the WLAN coverage is lost, it automatically resumes connectivity through 3G. New air interfaces that can detect both technologies, smooth handoffs, and applications that are resistant to temporary disconnections are among issues to be addressed.

# References

[1] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[2] A. Doufexi, E. Tameh, A. Nix, S. Armour, and A. Molina, "Hotspot Wireless LANs to Enhance the Performance of 3G and Beyond Cellular Networks," *IEEE Communications Magazine*, pp. 58–65, July 2003.

[3] H. Honkasalo et. al., "WCDMA and WLAN for 3G and Beyond," *IEEE Wireless Communications*, pp. 14–18, Apr. 2002.

[4] A. K. Salkintzis, C. Fors, and R. Pazhyannur, "WLAN-GPRS Integration for Next-Generation Mobile Data Networks," *IEEE Wireless Communications*, pp. 112–124, Oct. 2002.

[5] P. Mohapatra, J. Li, and C. Gui, "QoS in Mobile Ad Hoc Networks," *IEEE Wireless Communications*, pp. 44–52, June 2003.

[6] S. Shakkottai, T. S. Rappaport, and P. C. Kalsson, "Cross-Layer Design for Wireless Networks," *IEEE Communications*, pp. 74–80, Oct. 2003.

[7] K. Chen, S. H. Shah, and K. Nahrstedt, "Cross-Layer Design for Data Accessibility in Mobile Ad Hoc Networks," *Wireless Personal Communications*, , no. 21, pp. 49–76, 2002.

[8] J. C. Haartsen, "The Blue-tooth Radio System," *IEEE Personal Communications*, pp. 6–14, Feb. 2000.

[9] K. J. Negus, A. P. Stephens, and J. Lansford, "HomeRF: Wireless Networking for the Connected Home," *IEEE Personal Communications*, pp. 20–27, Feb. 2000.

[10] T. Zasowski et al., "UWB for Noninvasive Wireless Body Area Networks: Channel Measurements and Results," in *IEEE Conference on Ultrawideband Systems and Technologies, UWBST'2003, Reston, Virginia, USA*, Nov. 2003.

[11] D. Porcino and W. Hirt, "Ultra-Wideband Radio Technology: Potential and Challenges Ahead," *IEEE Communications Magazine*, pp. 66–74, July 2003.

[12] A. Chandra, V. Gummalla, and J. O. Limb, "Wireless Medium Access Control Protocols," *IEEE Communications Surveys,http://www.comsoc.org/pubs/surveys*, pp. 2–15, 2000, Second Quarter.

[13] I. F. Akyildiz, J. McNair, L. C. Martorell, R. Puigjaner, and Y. Yesha, "Medium Access Control Protocols for Multimedia Traffic in Wireless Networks," *IEEE Network*, pp. 39–42, Aug. 1999.

[14] R. V. Nee, G. Awater, M. Morikura, H. Takanashi, M. Webster, and K. W. Halford, "New High-Rate Wireless LAN Standards," *IEEE Communications Magazine*, pp. 82–88, Dec. 1999.

[15] R. O. LaMaire, A. Krishna, P. Bhagwat, and J. Panian, "Wireless LANs and Mobile Networking: Standards and Future Directions," *IEEE Communications Magazine*, pp. 86–94, Aug. 1996.

[16] J. Khun-Jush et. at., "Overview and Performance of HiperLAN Type 2 - A Standard for Broadband Wireless Communications," in *IEEE Vehicular Technology Conference, VTC2000, Boston, MA, USA*, 2000.

[17] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, pp. 116–126, Sept. 1997.

[18] H. S. Chhaya and S. Gupta, "Performance of Asynchronous Data Transfer Methods of IEEE 802.11 MAC Protocol," *IEEE Personal Communications*, pp. 8–15, Oct. 1996.

[19] G. Xylomenos and G. C. Polyzos, "Internet Protocol Performance over Networks with Wireless Links," *IEEE Network*, pp. 55–63, Aug. 1999.

[20] Y. C. Tseng, C. C. Shen, and W. T. Chen, "Integrating Mobile IP with Ad Hoc Networks," *Computer*, vol. 36, no. 5, pp. 48–55, May 2003.

[21] M. Gerla, K. Tang, and R. Bagrodia, "TCP Performance in Wireless Multi-hop Networks," in *IEEE WMCSA*, 1999.

[22] W. A. Arbaugh, N. Shankar, and Y. C. J. Wan, "Your 802.11 Wireless Network Has No Clothes," *IEEE Wireless Communications*, pp. 44–51, Dec. 2002.

[23] M. J. Ho, M. S. Rawles, M. Vrijkorte, and L. Fei, "RF Challenges for 2.4 and 5 GHz WLAN Deployment and Design," in *WCNC*, Mar. 2002, vol. 2, pp. 783–788.

[24] S. C. Kim, H. L. Bertoni, and M. Stern, "Pulse Propagation Characteristics at 2.4 GHz Inside Buildings," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 3, pp. 579–592, Aug. 1996.

[25] R. Anderson et. at., "In-Building Wideband Multipath Characteristics at 2.5 & 60 GHz," in *IEEE Vehicular Technology Conference, VTC2002, Vancouver, BC, Canada*, Sept. 2002.

[26] G. Woodward, I. Oppermann, and J. Talvitie, "Outdoor-Indoor Temporal & Spatial Wideband Channel Model for ISM Bands," in *VTC, Amsterdam, The Netherlands*, Sept. 1999.

[27] T. A. Wysocki and H. J. Zepernick, "Characterization of the Indoor Radio Propagation Channel at 2.4 GHz ," *Journal of Telecommunications and Information Technology*, , no. 3-4, pp. 84–90, 2000.

[28] C. F. Chiasserini and R. R. Rao, "Coexistence Mechanisms for Interference Mitigation between IEEE 802.11 WLANs and Bluetooth," in *IEEE INFOCOM, Newyork, NY, USA*, June 2002.

[29] I. Howitt, "WLAN and WPAN Coexistence in UL Band," *IEEE Transactions on Vehicular Technology*, pp. 1114–1124, July 2001.

[30] F. Eshghi and A. K. Elhakeem, "Error Performance of FH Wireless LANs in DS and Tone Interference," *Wireless Personal Communications*, vol. 23, no. 2, pp. 243–251, Nov. 2002.

[31] T. H. Meng, B. McFarland, D. Su, and J. Thomson, "Design and Implementation of an All-CMOS 802.11a Wireless LAN Chip," *IEEE Communications Magazine*, pp. 160–168, Aug. 2003.

[32] C. Andren, K. Halford, and M. Webster, "CCK, the New IEEE 802.11 Standard for 2.4 GHz Wireless LANs," in *IIC-Taipei, Conference Proceedings*, 1999.

[33] H. H. Chen, J. F. Yeh, and N. Suehiro, "A Multicarrier CDMA Architecture Based on Orthogonal Complementary Codes for New Generations of Wideband Wireless Communications," *IEEE Communications Magazine*, pp. 126–135, Oct. 2001.

[34] B. Pearson, "Complementary Code Keying Made Simple," Tech. Rep., Intersil, Application note AN9850.1, May 2000.

[35] C. Coutras, S. Gupta, and N. B. Shroff, "Scheduling of Real-Time Traffic in IEEE 802.11 Wireless LANs," *Wireless Networks*, , no. 6, pp. 457–466, 2000.

[36] H. Liu and J. C. Wu, "Packet Telephony Support for the IEEE 802.11 Wireless LAN," *IEEE Communications Letters*, pp. 286–288, Sept. 2000.

[37] M. A. Visser and M. El Zarki, "Voice and Data Transmission over an 802.11 Network ," in *Proceedings of PIMRC, Toronto, Canada*, Sept. 1995, pp. 648–652.

[38] A. Ganz and K. Wongthavarawat, "IEEE 802.11 Wireless LAN Association Procedure for Multimedia Applications," in *MILCOM*, 1999, pp. 1287–1291.

[39] B. P. Crow, I. Widjaja, J. G. Kim, and P. Sakai, "Investigation of the IEEE 802.11 Medium Access Control (MAC) Sublayer Functions," in *IEEE INFOCOM, Kobe, Japan*, Apr. 1997, pp. 126–133.

[40] Benny Bing, "Measured Performance of the IEEE 802.11 Wireless LAN," in *LCN'99*, Oct. 1999, pp. 34–42.

[41] G. Anastasi and L. Lenzini, "QoS Provided by the IEEE 802.11 Wireless LAN to Advanced Data Applications: A Simulation Analysis," *Wireless Networks*, , no. 6, pp. 99–108, 2000.

[42] H. S. Chhaya and S. Gupta, "Performance Modeling of Asynchronous Data Transfer Methods of IEEE 802.11 MAC Protocol," *Wireless Networks*, , no. 3, pp. 217–234, Oct. 1997.

[43] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement," in *INFOCOM*, Mar. 1998, pp. 142–149.

[44] F. Cali, M. Conti, and E. Gregori, "Dynamic IEEE 802.11: Design, Modeling, and Performance Evaluation," in *Lectures Notes in Computer Science*, 2000, pp. 786–798.

[45] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1774–1786, Sept. 2000.

[46] Y. Xiao, "A Simple and Effective Priority Scheme for IEEE 802.11," *IEEE Communications Letters*, vol. 7, no. 2, pp. 70–72, Feb. 2003.

[47] S. P. Chung and T. C. Peng, "Prioritized Medium Access Control in Ad Hoc Wireless LANs with Real-Time Traffic," *Submitted to IEEE Communications Letters*, 2004.

[48] A. Banchs and X. Perez, "Providing Throughput Guarantees in IEEE 802.11 Wireless LAN," in *Wireless Communications and Networking Conference, WCNC'2002, Orlando, FL, USA*, Mar. 2002.

[49] D. J. Deng and R. S. Chang, "A Priority Scheme for IEEE 802.11 DCF Access Method," *IEICE Transactions on Communications*, vol. E82-B, no. 1, pp. 96–102, Jan. 1999.

[50] J. L. Sobrinho and A. S. Krishnakumar, "Real-Time Traffic Over The IEEE 802.11 Medium Access Control Layer," *Bell Labs Technical Journal*, pp. 172–187, Nov. 1996.

[51] N. Milanovic, M. Malek, A. Davidson, and V. Milutinovic, "Routing and Security in Mobile Ad Hoc Networks," *Computer*, pp. 61–65, Feb. 2004.

[52] J. Broch et al., "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," in *MobiCom, Dallas, Texas, USA*, Oct. 1998, pp. 85–97.

[53] Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad hoc Networks," *Wireless Networks*, , no. 6, pp. 307–321, 2000.

[54] A. K. Elhakeem et al., "Forwarding Data Bases and Ad-Hoc Routing Techniques for Nested Clusters of Wireless LANs," in *The 13th International Conference on Wireless Communications, Wireless 2001, Calgary, Alberta, Canada*, July 2001.

[55] C. R. Lin and M. Gerla, "Real-time Support in Multihop Wireless Networks," *Wireless Networks*, , no. 5, pp. 125–135, 1999.

[56] M. Kazantzidis and M. Gerla, "End-to-end versus Explicit Feedback Measurement in 802.11 Networks," in *Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)*, 2002, pp. 429–434.

[57] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications*, pp. 1265–1275, Sept. 1997.

[58] LAN/MAN Standards Committee of the IEEE Computer Society, *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ANSI/IEEE Std. 802.11*, 1999.

[59] T. N. Saadawi, M. H. Ammar, and A. Elhakeem, *Fundamentals of Telecommunication Networks*, John Wiley & Sons, Inc., 1994.

[60] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, Addison-Wesley Publishing Company, Inc., 1994.

[61] E. H. K. Wu, J. T. C. Tsai, and M. Gerla, "The Effect of Radio Propagation on Multimedia, Mobile, and Multihop Networks: Models and Countermeasures," in

*The IEEE Singapore International Conference on Networks (SICON), Singapore*, Apr. 1997.

[62] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall PTR, second edition, 2002.

[63] R. Rao R. V. and T. S. Lamba, "Markovian Modeling of Fading Channels-Application in Forward Error Correction," in *NCC, Bombay, India*, Jan. 2002.

[64] H. S. Wang and P. C. Chang, "On Verifying the First-Order Markovian Assumption for a Rayleigh Fading Channel Model," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, pp. 353–357, May 1996.

[65] H. S. Wang and N. Moayeri, "Finite-State Markov Channel-A Useful Model for Radio Communication Channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, Feb. 1995.

[66] W. Turin and R. V. Nobelen, "Hidden Markov Modeling of Flat Fading Channels," *IEEE Journal on Selected Areas In Communications*, vol. 16, no. 9, pp. 1809–1817, Dec. 1998.

[67] F. Swarts and H. C. Ferreira, "Markov Characterization of Digital Fading Mobile VHF Channels," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 4, pp. 977–986, Nov. 1994.

[68] C. C. Tan and N. C. Beaulieu, "On First-Order Markov Modeling for the Rayleigh Fading Channel," *IEEE Journal on Selected Areas in Communications*, vol. 48, no. 12, pp. 2032–2040, Dec. 2000.

[69] F. Babich and G. Lombardi, "A Markov Model for the Mobile Propagation Channel," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 1, pp. 63–73, Jan. 2000.

[70] A. Konrad, A. D. Joseph, R. Ludwig, and B. Y. Zhao, "A Markov-Based Channel Model Algorithm for Wireless Networks," Tech. Rep., Computer Science Division (EECS), University of California at Berkeley , Report No. UCB/CSD-01-1142, May 2001.

[71] Y. Y. Kim and S. Li, "Modeling Multipath Fading Channel Dynamics for Packet Data Performance Analysis," *Wireless Networks*, , no. 6, pp. 481–492, 2000.

[72] J. R. Yee and E. J. Weldon, "Evaluation of the Performance of Error-Correcting Codes on a Gilbert Channel," *IEEE Transactions on Communications*, vol. 43, no. 8, pp. 2316–2323, Aug. 1995.

[73] M. Zorzi, "Performance of FEC and ARQ Error Control in Bursty Channels under Delay Constraints," in *VTC*, May 1998.

[74] S. Choi and K. G. Shin, "A Unified Wireless LAN Architecture for Real-Time and Non-Real-Time Communications Services," *IEEE Journal on Networking*, pp. 44–59, Feb. 2000.

[75] M. C. Valenti, *Turbo Codes and Iterative Processing*, www.ee.vt.edu/valenti/turbo.html, 1999.

[76] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding*, John Wiley & Sons LTD., 2002.

# Appendix

# Simulation Source Code

For evaluating the performance of the multi-hop ad-hoc 802.11 WLAN, a detailed simulation program has been written and tested in MATLAB environment (version 6.5.1 release 13). The simulation program features modularity which facilitates the replacement of different processes (such as, channel, mobility, and traffic generation). The modularity also makes it easy to understand and follow the flow of the events.

Routines are listed in alphabetical order. System parameters are applied through the *par_net_ds.m*, *init_ds.m*, and *main_ds.m* routines. The latter also serves as the main routine to the program and produces all the results. *main_tot_ds.m* puts together all the results related to a specific figure by calling *main_ds.m* repeatedly. Finally, the results are pictorially presented by *respic_ds.m.*

The program can easily be configured to simulate the single-hop scenario as in section 3.5.1.

```
% ack_chk_ds.m
% Checks whether the received ACK is error-free

P_c=(1-BER_mat1(xa,ya))^ACK_UNCOD;          % Probability of correct CTS
if (rand(1,1)<P_c) & (node(xa,15)==0)       % ACK is not collided either
    % **Update TX's queue
    % **TX's node()
    node(xa,6)=0;                 % CW reset to min
    node(xa,5)=min(floor(rand(1,1)*(2^(3+node(xa,6)))), ... % BK_OFF_CT
            2^(3+node(xa,6))-1)*2;
    % **Update pkt()
    pkt(node(xa,9),node(xa,10),node(xa,11),5)=0;      % Q_POS=0
    pkt(node(xa,9),node(xa,10),node(xa,11),11)=6;     % Drop due to successfully IMDT TX.
    pkt(node(xa,9),node(xa,10),node(xa,11),14)=0;     % CNTR=0
    % Update traversed hop no.
    pkt(node(xa,9),node(xa,10),node(xa,11),12)=pkt(node(xa,9),node(xa,10),node(xa,11),12)+1;
    % **Update  TX's Q
     if  node(xa,4)>1
       for jj=1:node(xa,4)-1                % jj=1:Q_LTH-1
         Q(xa,jj,:)=Q(xa,jj+1,:);           % Shift the queue.
         pkt(Q(xa,jj,1),Q(xa,jj,2),Q(xa,jj,3),5)= ...
           pkt(Q(xa,jj,1),Q(xa,jj,2),Q(xa,jj,3),5)-1;
       end
       pkt(Q(xa,1,1),Q(xa,1,2),Q(xa,1,3),8)=pkt(Q(xa,1,1),Q(xa,1,2),Q(xa,1,3),8) ...
       +RL_TM_CT-pkt(Q(xa,1,1),Q(xa,1,2),Q(xa,1,3),14); % TM_IN_Q updated
       pkt(Q(xa,1,1),Q(xa,1,2),Q(xa,1,3),14)=0;
     else
     end
     Q(xa,node(xa,4),:)=0;          % Last occupied location is set to zero.
     node(xa,4)=node(xa,4)-1;        % Q_LTH-1
else
   %cw_add_ds                 % Double CW due to collision effect
   if (pkt(node(xa,9),node(xa,10),node(xa,11),10)==RE_TX_LMT) | ...
     (RL_TM_CT-pkt(node(xa,9),node(xa,10),node(xa,11),6)>TM_TO_LIVE)
     % **Drop pkt
     pkt(node(xa,9),node(xa,10),node(xa,11),5)=0;       % Q_POS=0
     pkt(node(xa,9),node(xa,10),node(xa,11),14)=0;       % CNTR=0
     if pkt(node(xa,9),node(xa,10),node(xa,11),10)==RE_TX_LMT
       pkt(node(xa,9),node(xa,10),node(xa,11),11)=2;  % Drop due to reTX limit
     else
       pkt(node(xa,9),node(xa,10),node(xa,11),11)=3;  % Drop due to TM_TO_LIVE
     end
     % **Update Q
     if  node(xa,4)>1
       for jj=1:node(xa,4)-1  % jj=1:Q_LTH-1
         Q(xa,jj,:)=Q(xa,jj+1,:); % Shift the queue.
         pkt(Q(xa,jj,1),Q(xa,jj,2),Q(xa,jj,3),5)= ...
           pkt(Q(xa,jj,1),Q(xa,jj,2),Q(xa,jj,3),5)-1;
       end
       pkt(Q(xa,1,1),Q(xa,1,2),Q(xa,1,3),8)=pkt(Q(xa,1,1),Q(xa,1,2),Q(xa,1,3),8) ...
```

```
         +RL_TM_CT-pkt(Q(xa,1,1),Q(xa,1,2),Q(xa,1,3),14); % TM_IN_Q updated
         pkt(Q(xa,1,1),Q(xa,1,2),Q(xa,1,3),14)=0;
      else
      end
      Q(xa,node(xa,4),:)=0;        % Last occupied location is set to zero.
      node(xa,4)=node(xa,4)-1;        % Q_LTH-1
   else
   end
end
% **At TX
node(xa,7:11)=0;       % ACT_MOD/TX_TYP/DES_NO_TRNST/CP_VER/HOP_VER=0
node(xa,14)=5;         % CH_IDL_CT=5
node(xa,15)=0;         % COLN=0

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ack_gen_ds.m
% Initiates the process of generating ACK.

% **Update TX's node()
node(xa,14)=5;        % CH_IDL_CT=5
node(xa,13)=ACK+SIFS;  % ITER_TO_GO=ACK+SIFS
node(xa,7)=2;         % ACT_MOD=RX
node(xa,8)=4;         % TX_TYP=ACK
node(xa,15)=0;         % COLN=0
% **Update RX's node()
node(ya,7)=1;         % ACT_MOD=TX
node(ya,8)=4;         % TX_TYP=ACK
node(ya,13)=ACK+SIFS;  % ITER_TO_GO=ACK+SIFS
node(ya,14)=5;         % CH_IDL_CT=5
% **At other nodes inside coverage area of receiver (now is transmitting CTS)
adj=nodecover_ds(ya,d,n_tot,cvr_rad);
if adj(1)~=0
   for jj=1:length(adj)
      if adj(jj)~=xa
         node(adj(jj),14)=5;                         % CH_IDL_CT=5
         node(adj(jj),13)=max(node(adj(jj),13),ACK+SIFS);   % ITER_TO_GO
         switch node(adj(jj),7)       % ACT_MOD
            case 2                % RX
               node(adj(jj),15)=1;   % COLN is recorded.
            case 3                % LSTN
               node(adj(jj),15)=1;    % COLN is recorded.
            case 0            % IDL
               node(adj(jj),7)=3; % node is turned to LSTN mode.
               node(adj(jj),8)=4; % TX_TYP=ACK
               node(adj(jj),9)=node(xa,9);   % Set SEQ_NO_TRNST
               node(adj(jj),10)=node(xa,10);  % Set CP_VER
               node(adj(jj),11)=node(xa,11);  % Set HOP_VER
            otherwise
         end
```

```
    else
    end
  end
else
end
clear adj

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ch_ber1_ds.m
% Assigns a BER to channel SNR for RTS,CTS,ACK  using reference [76], Fig.5-26

if snr(i)<-0.5
  BER_1(i)=0.5;
elseif snr(i)<0.6
    BER_1(i)=0.1;
elseif  snr(i)<1.5
    BER_1(i)=10^(-2);
elseif snr(i)<2.15
    BER_1(i)=10^(-3);
elseif snr(i)<2.55
    BER_1(i)=10^(-4);
elseif snr(i)<3.20
    BER_1(i)=10^(-5);
elseif snr(i)<3.90
    BER_1(i)=10^(-6);
elseif snr(i)<4.40
    BER_1(i)=10^(-7);
else
    BER_1(i)=10^(-8);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ch_ber2_ds.m
% Assigns a BER to channel SNR for DATA packet in case  of PKT_INFO+464=1000 using reference [76], Fig.5-
26

if snr(i)<-0.5
  BER_2(i)=0.5;
elseif snr(i)<0.55
    BER_2(i)=0.1;
elseif  snr(i)<1.15
    BER_2(i)=10^(-2);
elseif snr(i)<1.55
    BER_2(i)=10^(-3);
elseif snr(i)<1.95
    BER_2(i)=10^(-4);
elseif snr(i)<2.50
```

```
        BER_2(i)=10^(-5);
elseif snr(i)<3.0
        BER_2(i)=10^(-6);
elseif snr(i)<3.55
        BER_2(i)=10^(-7);
else
        BER_2(i)=10^(-8);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ch_ber3_ds.m
% Assigns a BER to channel SNR for DATA packet in case of PKT_INFO+464=2000 using reference [76], Fig.5-
26

if snr(i)<-0.5
    BER_2(i)=0.5;
elseif snr(i)<0.55
        BER_2(i)=0.1;
elseif  snr(i)<1.1
        BER_2(i)=10^(-2);
elseif snr(i)<1.3
        BER_2(i)=10^(-3);
elseif snr(i)<1.7
        BER_2(i)=10^(-4);
elseif snr(i)<2.15
        BER_2(i)=10^(-5);
elseif snr(i)<2.55
        BER_2(i)=10^(-6);
elseif snr(i)<2.93
        BER_2(i)=10^(-7);
else
        BER_2(i)=10^(-8);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ch_ber4_ds.m
% Assigns a BER to channel SNR for DATA packet in case of PKT_INFO+464=4000  using reference [76], Fig.5-
26

if snr(i)<-0.5
    BER_2(i)=0.5;
elseif snr(i)<0.55
        BER_2(i)=0.1;
elseif  snr(i)<.97
        BER_2(i)=10^(-2);
elseif snr(i)<1.2
        BER_2(i)=10^(-3);
```

```
elseif snr(i)<1.4
      BER_2(i)=10^(-4);
elseif snr(i)<1.75
      BER_2(i)=10^(-5);
elseif snr(i)<2.09
      BER_2(i)=10^(-6);
elseif snr(i)<2.4
      BER_2(i)=10^(-7);
else
      BER_2(i)=10^(-8);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ch_ber5_ds.m
% Assigns a BER to channel SNR for DATA packet in case  of PKT_INFO+464=8000 using reference [76], Fig.5-
26

if snr(i)<-0.5
   BER_2(i)=0.5;
elseif snr(i)<0.55
      BER_2(i)=0.1;
elseif  snr(i)<.90
      BER_2(i)=10^(-2);
elseif snr(i)<1.1
      BER_2(i)=10^(-3);
elseif snr(i)<1.2
      BER_2(i)=10^(-4);
elseif snr(i)<1.35
      BER_2(i)=10^(-5);
elseif snr(i)<1.5
      BER_2(i)=10^(-6);
elseif snr(i)<1.65
      BER_2(i)=10^(-7);
else
      BER_2(i)=10^(-8);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% channel_ds.m
% implements Gilbert model for the  channel

%  SNR and  BER matrices
% implementing uniform random generation over the paths
xx=rand(1,n_tot*(n_tot-1)/2);
for i=1:n_tot*(n_tot-1)/2
   if ((ch(i)==0) & (xx(i)<=beta1)) | ((ch(i)==1) & ...
            (xx(i)>alpha1))
```

```
    P_rx(i)=P_rx_avg(i);
    ch(i)=0;        % channel is in Good state
  else
    %P_rx(i)=raylrnd(sqrt((P_rx_avg(i)/2)));
    P_rx(i)=(raylrnd(sqrt((P_rx_avg(i)/2))))^2;
    ch(i)=1;        % channel is in Bad State
  end
  P_rx_dbm(i)=10*log10(P_rx(i));      % received power in dBm.
  snr(i)=P_rx_dbm(i)-noise_floor_dbm;
  snr(i)=snr(i)+3;   % uncoded SNR is 3 dB higher than channel SNR in half-rate turbo coding scheme.
  ch_ber1_ds
  switch (PKT_INFO+464)
    case 1000
      ch_ber2_ds
    case 2000
      ch_ber3_ds
    case 4000
      ch_ber4_ds
    case 6000
      ch_ber5_ds
    case 8000
      ch_ber5_ds
    otherwise
  end
end
BER_mat1=squareform(BER_1);
BER_mat2=squareform(BER_2);
for ID_NO=1:n_tot
  if (node(ID_NO,13)>0) & (node(ID_NO,7)==1) & (node(ID_NO,8)==3) % Node is transmitting DATA.
    if BER_INDX(ID_NO)<BER_TOT_NO-1
      BER_INDX(ID_NO)=BER_INDX(ID_NO)+1;
      ya=pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11),4); % Identify IMDT_DST
      pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11),15+BER_INDX(ID_NO))= ...
      BER_mat2(ID_NO,ya);      % BERi is set
    else
    end
  else
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% cts_chk_ds.m
% Checks whether the received CTS is error-free

if node(xa,15)==0
  P_c=(1-BER_mat1(xa,ya))^CTS_UNCOD;           % Probability of correct CTS
  if rand(1,1)<P_c       % CTS is not collided either
    % **Initiate DATA packet
    pkt_gen_ds
    else
      % **At transmitter
      %cw_add_ds           % Double CW due to collision effect.
      node(xa,7:11)=0;     % ACT_MOD/TX_TYP/SEQ_NO_TRNST/CP_VER/HOP_VER
      node(xa,14)=5;       % CH_IDL_CT=5
      node(xa,15)=0;       % COLN=0
    end
else
    % **At transmitter
    %cw_add_ds           % Double CW due to collision effect.
    node(xa,7:11)=0;     % ACT_MOD/TX_TYP/SEQ_NO_TRNST/CP_VER/HOP_VER
    node(xa,14)=5;       % CH_IDL_CT=5
    node(xa,15)=0;       % COLN=0
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% cts_gen_ds.m
% Initiates the process of generating CTS.

% **Update TX's node()
node(xa,14)=5;         % CH_IDL_CT=5
node(xa,13)=CTS+SIFS;  % ITER_TO_GO=CTS+SIFS
node(xa,7)=2;          % ACT_MOD=RX
node(xa,8)=2;          % TX_TYP=CTS
node(xa,15)=0;         % COLN=0
% **Update RX's node()
node(ya,7)=1;          % ACT_MOD=TX
node(ya,8)=2;          % TX_TYP=CTS
node(ya,13)=CTS+SIFS;  % ITER_TO_GO=CTS+SIFS
node(ya,14)=5;         % CH_IDL_CT=5
% **At other nodes inside coverage area of receiver (now transmitting CTS)
adj=nodecover_ds(ya,d,n_tot,cvr_rad);
if adj(1)~=0
   for jj=1:length(adj)
      if adj(jj)~=xa
         node(adj(jj),14)=5;          % CH_IDL_CT=5
         node(adj(jj),13)=max(node(adj(jj),13),CTS+SIFS);   % ITER_TO_GO
         switch node(adj(jj),7)       % ACT_MOD
            case 2              % RX
               node(adj(jj),15)=1;   % COLN is recorded.
            case 3           % LSTN
               node(adj(jj),15)=1;    % COLN is recorded.
            case 0           % IDL
               node(adj(jj),7)=3; % node is turned to LSTN mode.
               node(adj(jj),8)=2; % TX_TYP=CTS
               node(adj(jj),9)=node(ya,9);    % Set SEQ_NO_TRNST
               node(adj(jj),10)=node(ya,10);  % Set CP_VER
               node(adj(jj),11)=node(ya,11);  % Set HOP_VER
            otherwise
      end
    else
    end
  end
else
end
clear adj

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% cw_add_ds.m
% This module doubles the contention window size.

if  node(xa,6)<5
  node(xa,6)=node(xa,6)+1;
  node(xa,5)=min(floor(rand(1,1)*(2^(3+node(xa,6)))), ...
           2^(3+node(xa,6))-1)*2;
else
  node(xa,5)=min(floor(rand(1,1)*256),255)*2;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% duplicate.m
% Verifies if the received packet has already been met.

function x=duplicate_ds(a,b)  % "a" is a scalar and "b" is a 1xn matrix.
x=0;
for m=2:b(1)+1
  if b(m)==a
    x=1;
  else
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% forward_ds.m
% Implements traffic forwarding.

% **Update receiver's queue and new version of packet  %%%%
if node(ya,4)<BFR_LTH   % Buffer not full
  if pkt(node(ya,9),node(ya,10),node(ya,11)+1,12)<HOP_LMT
    node(ya,4)=node(ya,4)+1;   % Q_LTH is incremented
    % **Update RX's Q()
    Q(ya,node(ya,4),1:3)=[node(ya,9) node(ya,10) node(ya,11)+1]; % Q is updated
    % **Update pkt'()
    pkt(node(ya,9),node(ya,10),node(ya,11)+1,5)=node(ya,4);   % Q_POS
```

```
      if node(ya,4)~=1
        pkt(node(ya,9),node(ya,10),node(ya,11)+1,14)=RL_TM_CT;  % Start accumulating dwelling times in
                                                                 %queue
      else
        pkt(node(ya,9),node(ya,10),node(ya,11)+1,14)=0;
      end
      m=pkt(node(ya,9),node(ya,10),node(ya,11)+1,12);
      pkt(node(ya,9),node(ya,10),node(ya,11)+1,14+BER_TOT_NO+m)=ya;   % VIA_m is set.

    else
      % **Drop the pkt'()
      pkt(node(ya,9),node(ya,10),node(ya,11)+1,5)=0;         % Q_POS=0
      pkt(node(ya,9),node(ya,10),node(ya,11)+1,11)=1;         % Drop due to max hop violation.
      pkt(node(xa,9),node(xa,10),node(xa,11)+1,14)=0;          % CNTR=0
      drop=drop+1;
    end
else
  % **Drop the pkt'()
  pkt(node(ya,9),node(ya,10),node(ya,11)+1,5)=0;          % Q_POS=0
  pkt(node(ya,9),node(ya,10),node(ya,11)+1,13)=1;          % Blocked due to full buffer.
  pkt(node(xa,9),node(xa,10),node(xa,11)+1,14)=0;           % CNTR=0
  block=block+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% init_ds.m
% initializes node(n_tot,15) and link channel model [ch] as below:
% node(ID_NO)=(X,Y,Q_LTH,BK_OFF_CT,CW,ACT_MOD,SEQ_NO_TRNS,CP_VER,HOP_VER, ...
%              TX_TYP,NAV,ITER_TO_GO,CH_IDL_CT,COLN)
% Also the mobile nodes will be selected randomly.
% Each node communicates with its next (in terms of identity no.) node.

for ID_NO=1:2:n_tot-1
  dst(ID_NO)=ID_NO+1;
  dst(ID_NO+1)=ID_NO;
end
% **Initializing node()
location=(ulimit-llimit)*rand(n_tot,2)+llimit; % 2-D initial node placement
Q_LTH=zeros(n_tot,1);
BK_OFF_CT=min(floor(8*rand(n_tot,1)),7)*2;
ACT_MOD=zeros(n_tot,1);
SEQ_NO_TRNST=zeros(n_tot,1);
CP_VER=zeros(n_tot,1);
HOP_VER=zeros(n_tot,1);
NAV=zeros(n_tot,1);
CW=zeros(n_tot,1);
TX_TYP=zeros(n_tot,1);
CH_IDL_CT=5*ones(n_tot,1);
ITER_TO_GO=zeros(n_tot,1);
```

```
COLN=zeros(n_tot,1);
node=[location dst' Q_LTH BK_OFF_CT CW ACT_MOD SEQ_NO_TRNST CP_VER HOP_VER  ...
  TX_TYP NAV ITER_TO_GO CH_IDL_CT COLN];  % n_tot x 15
Q=zeros(n_tot,BFR_LTH,3);        % Initiates each node's queue.
archive=zeros(n_tot,1);          % First element of archive shows no. of packets in archive.

% **Initializing channel small scale status for each
% pair of nodes.
ch=sign(sign(rand(1,n_tot*(n_tot-1)/2)-0.5)+1); % 0 means Good channel and 1 means Bad channel.
                                 % this a symetrical n_tot x n_tot matrix
                                 % with diagonal irrelevant.
% **Initializing pkt()
%pkt=zeros(100,3,5,15+BER_TOT_NO);        % pkt(SEQ_NO,CP_VER,HOP_VER)=(..........)
pkt(1,1,1,1:14+BER_TOT_NO+HOP_LMT-1)=0;
BER_INDX=zeros(n_tot,1);
% **Selecting mobile nodes
mobile=sign(sign(rand(n_tot,1)-(1-n_mob_per))+1);      % Fixed/mobile (0/1).

% **Initializing traffic sources
source=sign(sign(rand(n_tot,1)-n_active)+1); % Randomly puts each source in traffic gen.
                             % mode (0) or inactive mode (1).
                             % source is a n_tot x 1 matrix.
yy=1;
for g=1:n_tot                    % Puts all pkt gen. nodes in PKT_GEN.
  if source(g)==0
    PKT_GEN(yy)=g;
    yy=yy+1;
  else
  end
end
if yy>1
  status=sign(sign(rand(length(PKT_GEN),1)-0.5)+1); % Initializes the starting status
                                 % of each pkt gen. node as ACT(0)
                                 % or IDLE(1)
else
end
SEQ_NO=0;
rcvd=0;
pending=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% main_ds.m
RL_TM_CT=0;
switch e1
  case 1
    %n_active=0.1;     % P_active
    v_mobile=0.1;
    %TM_TO_LIVE=0.1;   % TTL
    %n_mob_per=0;      % P_mobile
```

```
   case 2
      %n_active=0.2;
      v_mobile=10;
      %TM_TO_LIVE=0.2;
      %n_mob_per=0.25;
   case 3
      %n_active=0.3;
      v_mobile=20;
      %TM_TO_LIVE=0.3;
      %n_mob_per=0.5;
   case 4
      %n_active=0.5;
      v_mobile=30;
      %TM_TO_LIVE=0.4;
      %n_mob_per=0.75;
   otherwise
end
switch e2
   case 1
      PKT_INFO=536;
   case 2
      PKT_INFO=1536;
   case 3
      PKT_INFO=3536;
   case 4
      PKT_INFO=5536;
   case 5
      PKT_INFO=7536;
   otherwise
end
par_net_ds
init_ds
for iii=1:CNT_LMT
      if mod(RL_TM_CT,MAX_PKT_LT/(RATE_INFO*RL_TM_CT_SEC))==0      % Arrival process.
         src_mod_ds
      else
      end
   if  mod(RL_TM_CT,floor(disp_step/(v_mobile*RL_TM_CT_SEC)))==0  % Mobility process.
      mobility_ds
   else
   end
   if  mod(RL_TM_CT,floor(CH_CHNG_PD/RL_TM_CT_SEC))==0  % Channel process.
      channel_ds
   else
   end
   protocol_ds
   RL_TM_CT=RL_TM_CT+1;
   %st=[e1 e2 e3 e4]
end
pkt_demo_ds
res_show_ds
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% main_tot_ds.m

%%  *Plots the results
clear
Xdelay(1,1,1)=0;
Xpfd(1,1,1)=0;
Xpds(1,1,1)=0;
XQL(1,1,1)=0;

delay(1,1)=0;
std_delay(1,1)=0;
pfd(1,1)=0;
std_pfd(1,1)=0;
pds(1,1)=0;
std_pds(1,1)=0;
QL(1,1)=0;
std_QL(1,1)=0;

for e1=1:4              % Regarding 4 parameters per subplot.
  for e2=1:5            % Regarding 5 points per curve.
    XSEQ_NO(e1,e2)=0;
    Xpending(e1,e2)=0;
    Xrcvd(e1,e2)=0;
    for e3=1:8          % Regarding 8x8 sample size.
      for e4=1:8
        main_ds
        Xdelay_a(e4)=delay_a;
        Xpfd_a(e4)=pfd_a;
        Xpds_a(e4)=pds_a;
        XQL_a(e4)=QL_a;
        XSEQ_NO(e1,e2)=XSEQ_NO(e1,e2)+SEQ_NO;
        Xpending(e1,e2)=Xpending(e1,e2)+pending;
        Xrcvd(e1,e2)=Xrcvd(e1,e2)+rcvd;
        save temp_res_P_mobile.mat e1 e2 e3 e4  XSEQ_NO Xpending Xrcvd Xdelay_a Xpfd_a Xpds_a XQL_a
              Xdelay Xpfd Xpds XQL ...
              delay std_delay pfd std_pfd pds std_pds QL std_QL
        clear
        load temp_res_P_mobile.mat
      end
      Xdelay(e1,e2,e3)=mean(Xdelay_a);
      Xpfd(e1,e2,e3)=mean(Xpfd_a);
      Xpds(e1,e2,e3)=mean(Xpds_a);
      XQL(e1,e2,e3)=mean(XQL_a);
    end
    delay(e1,e2)=mean(Xdelay(e1,e2,:));
    std_delay(e1,e2)=std(Xdelay(e1,e2,:));
    pfd(e1,e2)=mean(Xpfd(e1,e2,:));
```

```
        std_pfd(e1,e2)=std(Xpfd(e1,e2,:));
        pds(e1,e2)=mean(Xpds(e1,e2,:));
        std_pds(e1,e2)=std(Xpds(e1,e2,:));
        QL(e1,e2)=mean(XQL(e1,e2,:));
        std_QL(e1,e2)=std(XQL(e1,e2,:));
    end
end
save result_P_mobile.mat delay std_delay pfd std_pfd pds std_pds QL std_QL XSEQ_NO Xrcvd Xpending

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% mobility_ds.m
% Calculates current location of all nodes
% and their mutual Euclidean distance.

z=rand(n_tot,1);
delta_x=unit_step_ds(0.5-z).*(unit_step_ds(0.25-z)-unit_step_ds(z-0.25));
delta_y=unit_step_ds(z-0.5).*(unit_step_ds(0.75-z)-unit_step_ds(z-0.75));
delta=disp_step.*[delta_x  delta_y].*[mobile mobile];
location=location+delta;
a=llimit*ones(n_tot,2); % assures that updated location does not
b=ulimit*ones(n_tot,2); % violate limits
location=max(location,a);
location=min(location,b);
d=pdist(location);
%d=squareform(temp);  % Euclidean distances are calculated
% Average received signal power
for i=1:n_tot*(n_tot-1)/2
    if d(i)==0
       d(i)=0.5;
    else
    end
    if d(i)<=d0
         P_rx_avg_dbm(i)=P_tx_dbm-20*log10(4*pi*d(i)*2.4*10^9/3*10^8); % no shadowing
    else
         P_rx_avg_dbm(i)=P_tx_dbm-PL_d0_db-10*n_pl*log10(d(i)/d0) ...
                  -sigma_shadow*randn(1,1);
    end
    P_rx_avg(i)=10.^(P_rx_avg_dbm(i)/10);  % Received power in mW.
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% nexthop_ds.m
% Determines the next hop node.
% For each packet, and its related RTS, CTS, and ACK, next hop
% remains unchanged. So it is invoked prior to RTS/CTS dialogue.

function x=nexthop_ds(source,destination,d,n_tot,cvr_rad)
```

```
d_mat=squareform(d);
if d_mat(source,destination)<cvr_rad
   x=destination;
else
   c=nodecover_ds(source,d,n_tot,cvr_rad);   % nodecover finds the nodes inside coverage
                                  % of the source.
   if c~=0
      x=c(1);
      for y=2:length(c)         % Selects the closest node to final destination
                       % among nodes inside current source coverage
                       % area.
         if d_mat(destination,c(y))<d_mat(destination,x)
            x=c(y);
         else
         end
      end
   else
      x=0;        % No next hop available
   end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% nodecover_ds.m
% nodecover lists the nodes inside coverage
% area of ID_NO.

function x=nodecover_ds(ID_NO,d,n_tot,cvr_rad)
d_mat=squareform(d);
ii=1;
for y=1:n_tot
   if  (y~=ID_NO) & (d_mat(ID_NO,y)<cvr_rad)
      x(ii)=y;
      ii=ii+1;
   else
   end
end
if ii==1
   x=0;
else
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% par_net_ds.m
% initializes network, source model, and fading channel parameters.

% **Network parameters
n_tot=30;                % Total number of nodes in the network.
```

```
n_mob_per=0.25;          % Percentage of the mobile users.
n_active=0.2;            % Percentage of the traffic generating nodes.
ulimit=400;              % (m) Upper limit of location coordinates.
llimit=0;                % lower limit of location coordinates.
%v_mobile=10;            % (m/s) Node's average speed=v_mobile*3600 km/h.
RE_TX_LMT=3;             % Retransmission limit.
TM_TO_LIVE=0.25;         % (sec) Maximum allowable delay.
cvr_rad=150;             % Coverage radius.
disp_step=5;             % (m) Displacement step which is 5 m.
CNT_LMT=100000;          % No. of loops the program does.

% **Source model parameters
%PKT_INFO=536;           % Data receiving at MAC level in bits.
RATE_INFO=64000;         % (bit/sec) bit rate generation of information packets.
CH_RATE=10^(6);          % (bit/sec) Channel bit rate.
ACT_PRD=1.0;             % (s) average talk spurt duration.
SLC_PRD=1.35;            % (s) average silence period.
MAX_PKT_LT=8000;
burst=floor(MAX_PKT_LT/PKT_INFO);  % No. of packets produced during each slot time
                         % while in active (ON) state.
source=sign(sign(rand(n_tot,1)-0.5)+1) ;    % Source in ACT/IDL (0/1) mode.
src_slot_time=MAX_PKT_LT/RATE_INFO;  % source traffic model slot time.
beta=1-src_slot_time/ACT_PRD;     % prob. of staying in active mode.
alpha=1-src_slot_time/SLC_PRD;    % prob. of staying in silence mode.
% beta=1;
% alpha=1;
% **Large scale path loss
n_pl=3.2;                % path loss exponent.
d0=10;                   % (m) break point.
PL_d0_db=20*log10(4*pi*d0*2.4*10^9/(3*10^8)); % (dB) break point path loss.
sigma_shadow=8;               % (dB) standard variation of shadowing.
P_tx_dbm=17;                  % (dBm) transmitted power.
%P_sens_dbm=-80;              % (dBm) received power sensitivity.
noise_floor_dbm=-80;          % (dBm) noise floor power.

% **Fading channel parameters
% (ref{Cro:Wid:Kim:Sak}) Pr.{G--->B}=1/30 (sec^-1) & Pr.{B--->G}=0.1(sec^-1)
beta1=0.967;          % prob. of staying in Good channel.
alpha1=0.9;           % prob. of staying in Bad channel.
CH_CHNG_PD=0.423*3*(10^8)/(v_mobile*2.4*(10^9));  % (sec) The period over which channel is unchanged
due to fading.
%CH_CHNG_PD=5*10^(-3);  % (sec) The period over which channel is unchanged due to fading.
BER_TOT_NO=ceil((PKT_INFO+464)*2/(CH_CHNG_PD*CH_RATE));  % No. of different BERs a packet
experiences.

% **Protocol parameters
% RL_TM_CT is 10 micro seconds long and all the following parameters
% are expressed in terms if this unit.
RL_TM_CT_SEC=10^(-5);
RTS=70;                  % (20x8+144+48)x2=704 ---> 70 units
CTS=60;                  % (14x8+144+48)x2=604 ---> 60 units
```

```
ACK=60;                 % (14x8+144+48)x2=604 ---> 60 units
DIFS=5;                 % 5 units
SIFS=1;                 % 1 units
PRTCL_SLT_TM=2;         % 2 units
DATA=floor((PKT_INFO+464)*2/10); % Data packet length in 10 micro secs units.
RTS_UNCOD=352;          % (bits) RTS before coding.
CTS_UNCOD=304;          % (bits) CTS before coding.
ACK_UNCOD=304;          % (bits) ACK before coding.
DATA_UNCOD=PKT_INFO+464;  % (bits) DATA before coding.
L_T_c=CH_CHNG_PD*CH_RATE;  % Bit length equivalent of coherence time.


% **Miscellaneous parameters
MAX_Q=5;                % Q length in units of long packets.
BFR_LTH=MAX_Q*burst;            % Each node's buffer length.
HOP_LMT=4;              % Maximum no of hops allowed.

% **Turbo coding and decoding process times.
%COD_TM=floor((PKT_INFO+464)*g1/10);    % Units of RL_TM_CT
%DECOD_TM=floor((PKT_INFO+464)*2*g2/10); % Units of RL_TM_CT

success=0;
block=0;
drop=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% pkt_chk_ds.m
% Checks if the received packet is error-free.
if node(ID_NO,15)==0
  %P_c=(1-max(pkt(node(xa,9),node(xa,10),node(xa,11),15:14+BER_TOT_NO)))^DATA_UNCOD;
  P_c=1;
  if BER_TOT_NO-1>=15
    for u=15:14+BER_TOT_NO-1
      P_c=P_c*((1-pkt(node(xa,9),node(xa,10),node(xa,11),u))^L_T_c);
    end
  else
  end
  P_c=P_c*(1-pkt(node(xa,9),node(xa,10),node(xa,11),14+BER_TOT_NO))^mod(DATA_UNCOD,L_T_c);
  if (rand(1,1)<P_c)          % DATA packet received error-free
    ack_gen_ds              % ACK is generated.
    % **Generate pkt'()
    pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,:)=pkt(node(ID_NO,9),node(ID_NO,10),node
              (ID_NO,11),:); % New version of pkt s generated.
    pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,12)=pkt(node(ID_NO,9),node(ID_NO,10),node
              (ID_NO,11)+1,12)+1; % NO_HOP updated.
    pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,3)=pkt(node(ID_NO,9),node(ID_NO,10),node
              (ID_NO,11)+1,4); % IMD_SRC=old(IMD_DST)
    pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,4)=0; % IMD_DST=0
    pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,10)=0;        % RE_TX reset.
    pkt(node(ya,9),node(ya,10),node(ya,11)+1,14:13+BER_TOT_NO+HOP_LMT)=0;      % CNTR/BERi reset.
    % **Verify duplication
    result=duplicate_ds(node(ID_NO,9),archive(ID_NO,:));
    if result==0            % No Duplication
      archive(ID_NO,archive(ID_NO,1)+2)=node(xa,9);  % New entity added to archive.
      archive(ID_NO,1)=archive(ID_NO,1)+1;        % archive counter is updated.
      if ((RL_TM_CT-pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,6))*10^(-5)<TM_TO_LIVE)  % No
                                                                                 % violation of time limit.
        if  pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,3)== pkt(node(ID_NO,9),node(ID_NO,10),
              node(ID_NO,11)+1,2)   % IMDT_DST=DST ?
          pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,5)=0;      % Q_POS=0
          pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,7)=RL_TM_CT;   % RX_TM
          pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,9)=1;      % SUC_RX=1,Packet
                                                        % successfully delivered from Rx's view
          success=success+1;                            % Total number of successful packets.
        else
          forward_ds
        end
      else
        % **Drop the pkt'()
        pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,5)=0;       % Q_POS=0
        pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,11)=3;      % Drop due to expiration.
        pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,14)=0;      % CNTR=0
        drop=drop+1;
      end
    else
      % **Drop the pkt'()
      pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,5)=0;       % Q_POS=0
      pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,11)=4;      % Drop due to duplication.
      pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11)+1,14)=0;      % CNTR=0
      drop=drop+1;
    end
  else
    % **At receiver
    node(ID_NO,7:8)=0;   % ACT_MOD=IDLE/TX_TYP=0
    node(ID_NO,9:11)=0;
    node(ID_NO,14)=5;  % CH_IDL_CT=5
    node(ID_NO,15)=0;  % COLN=0
  end       % Continue
else
  % **At receiver
   node(ID_NO,7:8)=0;   % ACT_MOD=IDLE/TX_TYP=0
   node(ID_NO,9:11)=0;
   node(ID_NO,14)=5;  % CH_IDL_CT=5
   node(ID_NO,15)=0;  % COLN=0
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% pkt_gen_ds.m
% Initiates the process of generating DATA packet.

pkt(node(xa,9),node(xa,10),node(xa,11),10)=pkt(node(xa,9),node(xa,10),node(xa,11),10)+1; % increment
                                                                                           % RE_TX
% **At transmitter
node(xa,7)=1;         % ACT_MOD=TX
node(xa,8)=3;         % TX_TYP=DATA
node(xa,13)=DATA+SIFS; % ITER_TO_GO=DATA+SIFS
node(xa,14)=5;         % CH_IDL_CT=5
BER_INDX(xa)=0;        % BER_INDX reset
pkt(node(xa,9),node(xa,10),node(xa,11),15)=BER_mat2(xa,ya);  % BER1 is set.
% **At receiver
node(ya,14)=5;         % CH_IDL_CT=5
node(ya,13)=DATA+SIFS;  % ITER_TO_GO=DATA+SIFS
node(ya,7)=2;         % ACT_MOD=RX
node(ya,8)=3;         % TX_TYP=DATA
node(ya,15)=0;         % COLN=0
% **At other nodes inside coverage area of transmitter.
adj=nodecover_ds(xa,d,n_tot,cvr_rad);
if adj(1)~=0
   for jj=1:length(adj)
      if adj(jj)~=ya
         node(adj(jj),14)=5;           % CH_IDL_CT=5
         node(adj(jj),13)=max(node(adj(jj),13),DATA+SIFS);   % ITER_TO_GO
         switch node(adj(jj),7)        % ACT_MOD
            case 2              % RX
               node(adj(jj),15)=1;   % COLN is recorded.
            case 3            % LSTN
               node(adj(jj),15)=1;    % COLN is recorded.
            case 0            % IDL
               node(adj(jj),7)=3; % node is turned to LSTN mode.
               node(adj(jj),8)=3; % TX_TYP=DATA
               node(adj(jj),9)=node(xa,9);   % Set SEQ_NO_TRNST
               node(adj(jj),10)=node(xa,10);  % Set CP_VER
               node(adj(jj),11)=node(xa,11);  % Set HOP_VER
            otherwise
         end
      else
      end
   end
else
end
clear adj

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% protocol_ds.m
% Implements 802.11 MAC  protocol
```

```
for ID_NO=1:n_tot
   if node(ID_NO,13)==0            % ITER_TO_GO==0
      switch node(ID_NO,7)         % ACT_MOD=?
         case 1              % ACT_MOD=TX
            node(ID_NO,7:8)=0; % ACT_MOD/TX_TYP=0
            node(ID_NO,14)=5;  % CH_IDL_CT=5
            node(ID_NO,15)=0;  % COLN=0
         case 3              % ACT_MOD=LSTN
            xa=pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11),3);     % IMDT_SRC
            ya=pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11),4);     % IMDT_DST
            switch node(ID_NO,8)   % TX_TYP
                 case 1            % TX_TYP=RTS
                    % **Set NAV
                    if node(ID_NO,15)==0
                       P_c=(1-BER_mat1(xa,ID_NO))^RTS_UNCOD;            % Probability of correct RTS
                       if rand(1,1)<P_c   % Correct RTS and no collision
                          node(ID_NO,12)=max(CTS+DATA+ACK+3*SIFS,node(ID_NO,12));
                                                                          % NAV=CTS+DATA+ACK+3xSIFS
                       else
                       end
                    else
                    end
                 case 2            % TX_TYP=CTS
                    % **Set NAV
                    if node(ID_NO,15)==0
                       P_c=(1-BER_mat1(ya,ID_NO))^CTS_UNCOD;            % Probability of correct RTS
                       if rand(1,1)<P_c   % Correct RTS and no collision
                          node(ID_NO,12)=max(DATA+ACK+2*SIFS,node(ID_NO,12));
                                                                                     % NAV=DATA+ACK+2xSIFS
                       else
                       end
                    else
                    end
                 otherwise
            end
            % **Reset listening node except its NAV which is set above.
            node(ID_NO,7:11)=0; % ACT_MOD/TX_TYP/SEQ_NO_TRNST/CP_VER/HOP_VER=0
            node(ID_NO,14)=5;  % CH_IDL_CT=5
            node(ID_NO,15)=0;  % COLN=0
         case 5
            node(ID_NO,7:11)=0; % ACT_MOD/TX_TYP/SEQ_NO_TRNST/CP_VER/HOP_VER=0
            node(ID_NO,14)=5;  % CH_IDL_CT=5
            node(ID_NO,15)=0;  % COLN=0
         otherwise
      end
   else
   end
end
for ID_NO=1:n_tot
   if node(ID_NO,13)==0            % ITER_TO_GO==0
      if node(ID_NO,12)==0         % NAV==0
      switch node(ID_NO,7)      % ACT_MOD=?
        case 2                % ACT_MOD=RX
          xa=pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11),3);     % IMDT_SRC
          ya=pkt(node(ID_NO,9),node(ID_NO,10),node(ID_NO,11),4);     % IMDT_DST
          switch node(ID_NO,8)
            case 1            % TX_TYP=RTS
              rts_chk_ds
            case 2            % TX_TYP=CTS
              cts_chk_ds
            case 3            % TX_TYP=DATA
              pkt_chk_ds
            case 4            % TX_TYP=ACK
              ack_chk_ds
            otherwise
          end
        otherwise             % ACT_MOD=IDLE
            if node(ID_NO,4)>0            % Q_LTH>0
              if node(ID_NO,14)==0          % CH_IDL_CT==0
                if node(ID_NO,5)==0        % BK_OFF_CT==0
                  % **Initiate RTS
                  rts_gen_ds
                else
                  node(ID_NO,5)=node(ID_NO,5)-1;  % BK_OFF_CT-1
                end
              else
                node(ID_NO,14)=node(ID_NO,14)-1;   % CH_IDL_CT-1
              end
            else
            end
        end
    else
      node(ID_NO,12)=node(ID_NO,12)-1;          % NAV-1
    end
  else
    node(ID_NO,13)=node(ID_NO,13)-1;
    if node(ID_NO,12)~=0               % NAV~=0
      node(ID_NO,12)=node(ID_NO,12)-1;   % NAV-1
    else
    end
  end
end                              % Continue

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% res_show_ds.m
% Calculates delay and queueing delay of
% successfully received packets.

w=size(pkt);
v1=1;
for q1=1:SEQ_NO
```

```
   for q2=1:w(2)
      for q3=1:w(3)
         if pkt(q1,q2,q3,9)==1
            delay_tot(v1)=(pkt(q1,q2,q3,7)-pkt(q1,q2,q3,6))/DATA;
            delay_Q(v1)=pkt(q1,q2,q3,8)/DATA;
            v1=v1+1;
         else
         end
      end
   end
end
if v1==1
   delay_tot=0;
   delay_Q=0;
else
end
v=1;
IN_Q(1)=0;
for g1=1:n_tot
   if node(g1,4)>0
      for g2=1:node(g1,4)
         IN_Q(v)=Q(g1,g2,1);
         v=v+1;
      end
   else
   end
end
if length(IN_Q)>1
   for c1=1:length(IN_Q)-1
      for c2=c1+1:length(IN_Q)
         if (IN_Q(c1)==IN_Q(c2)) & (IN_Q(c2)~=0)
            IN_Q(c2)=0;
         else
         end
      end
   end
   pending=0;
   for c3=1:length(IN_Q)
      if IN_Q(c3)~=0
         pending=pending+1;
      else
      end
   end
else
   if IN_Q(1)==0;
      pending=0;
   else
      pending=length(IN_Q);
   end
end
rcvd=v1-1;
```

```
delay_a=mean(delay_tot);
if SEQ_NO~=0
   pfd_a=(SEQ_NO-pending-rcvd)/SEQ_NO;
   pds_a=rcvd/SEQ_NO;
else
   pfd_a=0;
   pds_a=0;
end
QL_a=mean(node(:,4))/burst;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% respic_ds.m
% Sketches the graphical results

x=[.1 .15 .2 .3 .5];
figure(1)
subplot(2,2,1)
plot(x,delay,'-rs','LineWidth',1,'MarkerSize',4)
hold on
for n=1:5
   xx=[x(n) x(n)];
   %y=[Xdelay(n)+Xstd_delay(n) Xdelay(n)-Xstd_delay(n)];
   y=[delay(n)+conf_delay(n) delay(n)-conf_delay(n)];
   plot(xx,y,'--bo','LineWidth',1,'MarkerSize',3)
   hold on
end
title(['{ \bf \fontsize{7}  n_{tot}=',num2str(n_tot),' ; P_{active}=', ...
      num2str(n_active),' ; Traffic per node=',num2str(RATE_INFO/1000),' Kb/s ; Packet length=',num2str
(DATA*10),' bits}'])
ylabel('{\bf \fontsize{8} Normalized Delay}')
xlabel('{\bf \fontsize{8} Time To Live (sec)}')
%h=legend('{\fontsize{7} Delay}','{\fontsize{7}\pm \sigma_{Delay}}',2)
h=legend('{\fontsize{7} Mean_{Delay}}','{\fontsize{7} %90 Conf. Int.}',2)
subplot(2,2,2)
plot(x,pfd,'-rs','LineWidth',1,'MarkerSize',4)
hold on
for n=1:5
   xx=[x(n) x(n)];
   %y=[Xpfd(n)+Xstd_pfd(n) Xpfd(n)-Xstd_pfd(n)];
   y=[pfd(n)+conf_pfd(n) pfd(n)-conf_pfd(n)];
   plot(xx,y,'--bo','LineWidth',1,'MarkerSize',3)
   hold on
end
ylabel('{\bf \fontsize{8} Prob. of Fail to Deliver (PFD)}')
xlabel('{\bf \fontsize{8} Time To Live (sec)}')
%h=legend('{\fontsize{7} PFD}','{\fontsize{7}\pm \sigma_{PFD}}',2)
h=legend('{\fontsize{7} Mean_{PFD}}','{\fontsize{7} %90 Conf. Int.}',2)
subplot(2,2,3)
plot(x,pds,'-rs','LineWidth',1,'MarkerSize',4)
```

```
hold on
for n=1:5
   xx=[x(n) x(n)];
   %y=[Xpds(n)+Xstd_pds(n) Xpds(n)-Xstd_pds(n)];
   y=[pds(n)+conf_pds(n) pds(n)-conf_pds(n)];
   plot(xx,y,'--bo','LineWidth',1,'MarkerSize',3)
   hold on
end
ylabel('{\bf \fontsize{8} Prob. Deliver Successfully (PDS)}')
xlabel('{\bf \fontsize{8} Time To Live (sec)}')
%h=legend('{\fontsize{7} PDS}','{\fontsize{7}\pm \sigma_{PDS}}',2)
h=legend('{\fontsize{7} Mean_{PDS}}','{\fontsize{7} %90 Conf. Int.}',2)
subplot(2,2,4)
plot(x,QL,'-rs','LineWidth',1,'MarkerSize',4)
hold on
for n=1:5
   xx=[x(n) x(n)];
   %y=[XQL(n)+Xstd_QL(n) XQL(n)-Xstd_QL(n)];
   y=[QL(n)+conf_QL(n) QL(n)-conf_QL(n)];
   plot(xx,y,'--bo','LineWidth',1,'MarkerSize',3)
   hold on
end
ylabel('{\bf \fontsize{8} Average Queue Length (AQL)}')
xlabel('{\bf \fontsize{8} Time To Live (sec)}')
%h=legend('{\fontsize{7} QL}','{\fontsize{7}\pm \sigma_{QL}}',2)
h=legend('{\fontsize{7} Mean_{AQL}}','{\fontsize{7} %90 Conf. Int.}',2)  % respic_ds.m,

x=[1000 2000 4000 6000 8000];
figure(1)
std_QL=std_QL*0.5;  % n=8 ; t(0.8,7=1.415
std_delay=std_delay*0.5;
std_pds=std_pds*0.5;
std_pfd=std_pfd*0.5;
subplot(2,2,1)
for m=1:4
   switch m
      case 1
         TTL=0.1;
         n_active=0.1;
         n_mob_per=0;
         v_mobile=0.1;
         RE_TX_LMT=1;
      case 2
         TTL=0.2;
         n_active=0.2;
         n_mob_per=0.25;
         v_mobile=10;
         RE_TX_LMT=2;
      case 3
         TTL=0.3
         n_active=0.3;
         n_mob_per=0.5;
         v_mobile=20;
         RE_TX_LMT=3;
      case 4
         TTL=0.4
         n_active=0.5;
         n_mob_per=0.75;
         v_mobile=30;
         RE_TX_LMT=4;
      otherwise
   end
   plot(x,delay(m,:),'-rs','LineWidth',1,'MarkerSize',3)
   hold on
   for n=1:5
      xx=[x(n) x(n)];
      y=[delay(m,n)+std_delay(m,n) max(delay(m,n)-std_delay(m,n),0)];
      plot(xx,y,':bo','LineWidth',1,'MarkerSize',2)
      hold on
   end
   %gtext(['{ \bf \fontsize{6} TTL=',num2str(TTL),' s}'])
   %gtext(['{ \bf \fontsize{6} P_{active}=',num2str(n_active),'}'])
   %gtext(['{ \bf \fontsize{6} P_{mobile}=',num2str(n_mob_per),' }'])
   gtext(['{ \bf \fontsize{6} v=',num2str(v_mobile),' m/s}'])
   %gtext(['{ \bf \fontsize{6} ReTX_{max}=',num2str(RE_TX_LMT),' }'])
end

%gtext(['{ \bf \fontsize{8}  n_{tot}=',num2str(n_tot),' ; P_{active}=', ...
      %num2str(n_active),' ; TTL=',num2str(TM_TO_LIVE),' s ; P_{mobile}=',num2str(n_mob_per),' ; v_{mobile}
=',num2str(v_mobile), ...
   %' m/s ; RE\_TX\_LIMIT=',num2str(RE_TX_LMT),' ; Q_{max}=',num2str(MAX_Q),' }'])

%%%% P_active
 %gtext(['{ \bf \fontsize{8}  TTL=',num2str(TM_TO_LIVE),' s ; P_{mobile}=',num2str(n_mob_per),' ; v_{mobile}
=',num2str(v_mobile), ...
 %   ' m/s ; ReTX_{max}=',num2str(RE_TX_LMT),' ; Q_{max}=',num2str(MAX_Q),' }'])

%%%%% TTL
 %gtext(['{ \bf \fontsize{8}  v=',num2str(v_mobile),' m/s ; P_{mobile}=',num2str(n_mob_per),' ; P_{active}
=',num2str(n_active), ...
  %  ' s ; ReTX_{max}=',num2str(RE_TX_LMT),' ; Q_{max}=',num2str(MAX_Q),' }'])

%%%%%% P_mobile
%gtext(['{ \bf \fontsize{8}  TTL=',num2str(TM_TO_LIVE),' s ; v=',num2str(v_mobile),' m/s ; P_{active}=',num2str
(n_active), ...
 %   ' ; ReTX_{max}=',num2str(RE_TX_LMT),' ; Q_{max}=',num2str(MAX_Q),' }'])

%%%%%% v_mobile
gtext(['{ \bf \fontsize{8}  TTL=',num2str(TM_TO_LIVE),' s ; P_{mobile}=',num2str(n_mob_per),' ; P_{active}
=',num2str(n_active), ...
   ' ; ReTX_{max}=',num2str(RE_TX_LMT),' ; Q_{max}=',num2str(MAX_Q),' }'])
 %%%%%%% ReTX_max
 %gtext(['{ \bf \fontsize{8}  TTL=',num2str(TM_TO_LIVE),' s ; v=',num2str(v_mobile),' m/s ; P_{active}
=',num2str(n_active), ...
  %  ' ; P_{mobile}=',num2str(n_mob_per),' ; Q_{max}=',num2str(MAX_Q),' }'])

ylabel('{\bf \fontsize{8} Normalized Delay (w.r.t packet duration)}')
xlabel('{\bf \fontsize{8} DATA Packet Length (Uncoded,bits)}')
h=legend('{\fontsize{7} Delay}','{\fontsize{7} %80 Conf\_int}',2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,2)
for m=1:4
  switch m
    case 1
      TTL=0.1;
      n_active=0.1;
      n_mob_per=0;
      v_mobile=0.1;
      RE_TX_LMT=1;
    case 2
      TTL=0.2;
      n_active=0.2;
      n_mob_per=0.25;
      v_mobile=10;
      RE_TX_LMT=2;
    case 3
      TTL=0.3
      n_active=0.3;
      n_mob_per=0.5;
      v_mobile=20;
      RE_TX_LMT=3;
    case 4
      TTL=0.4
      n_active=0.5;
      n_mob_per=0.75;
      v_mobile=30;
      RE_TX_LMT=4;
    otherwise
  end
  plot(x,pfd(m,:),'-rs','LineWidth',1,'MarkerSize',3)
  hold on
  for n=1:5
    xx=[x(n) x(n)];
    y=[pfd(m,n)+std_pfd(m,n) max(pfd(m,n)-std_pfd(m,n),0)];
    plot(xx,y,':bo','LineWidth',1,'MarkerSize',2)
    hold on
  end
  %gtext(['{ \bf \fontsize{6} TTL=',num2str(TTL),' s}'])
  %gtext(['{ \bf \fontsize{6} P_{active}=',num2str(n_active),'}'])
  %gtext(['{ \bf \fontsize{6} P_{mobile}=',num2str(n_mob_per),' }'])
  gtext(['{ \bf \fontsize{6} v=',num2str(v_mobile),' m/s}'])
  %gtext(['{ \bf \fontsize{6} ReTX_{max}=',num2str(RE_TX_LMT),' }'])
```

```
end

ylabel('{\bf \fontsize{8} Packet Failure Rate (PFR)}')
xlabel('{\bf \fontsize{8} DATA Packet Length (Uncoded,bits)}')
h=legend('{\fontsize{7} PFR}','{\fontsize{7}%80 Conf\_int}',2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,3)
for m=1:4
   switch m
      case 1
         TTL=0.1;
         n_active=0.1;
         n_mob_per=0;
         v_mobile=0.1;
         RE_TX_LMT=1;
      case 2
         TTL=0.2;
         n_active=0.2;
         n_mob_per=0.25;
         v_mobile=10;
         RE_TX_LMT=2;
      case 3
         TTL=0.3
         n_active=0.3;
         n_mob_per=0.5;
         v_mobile=20;
         RE_TX_LMT=3;
      case 4
         TTL=0.4
         n_active=0.5;
         n_mob_per=0.75;
         v_mobile=30;
         RE_TX_LMT=4;
      otherwise
   end
   plot(x,pds(m,:),'-rs','LineWidth',1,'MarkerSize',3)
   hold on
   for n=1:5
      xx=[x(n) x(n)];
      y=[pds(m,n)+std_pds(m,n) max(pds(m,n)-std_pds(m,n),0)];
      plot(xx,y,':bo','LineWidth',1,'MarkerSize',2)
      hold on
   end
   %gtext(['{ \bf \fontsize{6} TTL=',num2str(TTL),' s}'])
   %gtext(['{ \bf \fontsize{6} P_{active}=',num2str(n_active),'}'])
   %gtext(['{ \bf \fontsize{6} P_{mobile}=',num2str(n_mob_per),' }'])
   gtext(['{ \bf \fontsize{6} v=',num2str(v_mobile),' m/s}'])
   %gtext(['{ \bf \fontsize{6} ReTX_{max}=',num2str(RE_TX_LMT),' }'])
end

ylabel('{\bf \fontsize{8} Packet Success Rate (PSR)}')
```

```
xlabel('{\bf \fontsize{8} DATA Packet Length (Uncoded,bits)}')
h=legend('{\fontsize{7} PSR}','{\fontsize{7}%80 Conf\_int}',2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,4)
for m=1:4
   switch m
      case 1
         TTL=0.1;
         n_active=0.1;
         n_mob_per=0;
         v_mobile=0.1;
         RE_TX_LMT=1;
      case 2
         TTL=0.2;
         n_active=0.2;
         n_mob_per=0.25;
         v_mobile=10;
         RE_TX_LMT=2;
      case 3
         TTL=0.3
         n_active=0.3;
         n_mob_per=0.5;
         v_mobile=20;
         RE_TX_LMT=3;
      case 4
         TTL=0.4
         n_active=0.5;
         n_mob_per=0.75;
         v_mobile=30;
         RE_TX_LMT=4;
      otherwise
   end
   plot(x,QL(m,:),'-rs','LineWidth',1,'MarkerSize',3)
   hold on
   for n=1:5
      xx=[x(n) x(n)];
      y=[QL(m,n)+std_QL(m,n) max(QL(m,n)-std_QL(m,n),0)];
      plot(xx,y,':bo','LineWidth',1,'MarkerSize',2)
      hold on
   end
   %gtext(['{ \bf \fontsize{6} TTL=',num2str(TTL),' s}'])
   %gtext(['{ \bf \fontsize{6} P_{active}=',num2str(n_active),'}'])
   %gtext(['{ \bf \fontsize{6} P_{mobile}=',num2str(n_mob_per),' }'])
   gtext(['{ \bf \fontsize{6} v=',num2str(v_mobile),' m/s}'])
   %gtext(['{ \bf \fontsize{6} ReTX_{max}=',num2str(RE_TX_LMT),' }'])
end

ylabel('{\bf \fontsize{8} Queue Length (QL,long packets)}')
xlabel('{\bf \fontsize{8} DATA Packet Length (Uncoded,bits)}')
h=legend('{\fontsize{7} QL}','{\fontsize{7}%80 Conf\_int}',2)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% rts_chk_ds.m
% Checks to see if the received RTS is error-free
if node(ID_NO,15)==0
  P_c=(1-BER_mat1(xa,ya))^RTS_UNCOD;          % Probability of correct RTS
  if rand(1,1)<P_c  % RTS is not collided either
    % **Initiate CTS
    cts_gen_ds
  else
    % **At transmitter
    %cw_add_ds           % Double CW due to collision effect
    % **At receiver
    node(ya,7:11)=0;      % ACT_MOD/TX_TYP/SEQ_NO_TRNST/CP_VER/HOP_VER=0
    node(ya,14)=5;       % CH_IDL_CT=5
    node(ya,15)=0;       % COLN=0
  end
else
  % **At transmitter
  %cw_add_ds           % Double CW due to collision effect
  % **At receiver
  node(ya,7:11)=0;      % ACT_MOD/TX_TYP/SEQ_NO_TRNST/CP_VER/HOP_VER=0
  node(ya,14)=5;       % CH_IDL_CT=5
  node(ya,15)=0;       % COLN=0
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% rts_gen_ds.m
% Initiates the process of generating RTS.

xa=ID_NO;
cw_add_ds              % Double CW size for the next attempt
% Examine if the head of the Q has not violated TM_TO_LIVE
p1=Q(ID_NO,1,1);       % SEQ_NO of the first packet in the Q
p2=Q(ID_NO,1,2);       % CP_VER "   "     "   "
p3=Q(ID_NO,1,3);       % HOP_VER "   "     "   "
while  (((RL_TM_CT-pkt(p1,p2,p3,6))*10^(-5)>TM_TO_LIVE)|(pkt(p1,p2,p3,10)==RE_TX_LMT)) ...
    &(node(ID_NO,4)>0)
  % **Drop pkt
  pkt(p1,p2,p3,5)=0;     % Q_POS=0
  if pkt(p1,p2,p3,10)==RE_TX_LMT
    pkt(p1,p2,p3,11)=2;     % Drop due to RE_TX_LMT.
  else
    pkt(p1,p2,p3,11)=3;     % Drop due to expiration.
  end
  pkt(p1,p2,p3,14)=0;     % CNTR=0
  drop=drop+1;
  % **Update node and Q
  if node(ID_NO,4)>1
    for jj=1:node(ID_NO,4)-1  % jj=1:Q_LTH-1
      Q(ID_NO,jj,:)=Q(ID_NO,jj+1,:); % Shift the queue.
      pkt(Q(ID_NO,jj,1),Q(ID_NO,jj,2),Q(ID_NO,jj,3),5)=pkt(Q(ID_NO,jj,1),Q(ID_NO,jj,2),Q(ID_NO,jj,3),5)-1;
                                                        % Q_POS updated.
    end
    pkt(Q(ID_NO,1,1),Q(ID_NO,1,2),Q(ID_NO,1,3),8)=pkt(Q(ID_NO,1,1),Q(ID_NO,1,2),Q(ID_NO,1,3),8)+ ...
    RL_TM_CT-pkt(Q(ID_NO,1,1),Q(ID_NO,1,2),Q(ID_NO,1,3),14); % TM_IN_Q update
    pkt(Q(ID_NO,1,1),Q(ID_NO,1,2),Q(ID_NO,1,3),14)=0;
  else
  end
  Q(ID_NO,node(ID_NO,4),:)=0; % Last occupied location is set to zero.
  node(ID_NO,4)=node(ID_NO,4)-1;      % Decrement Q_LTH
  if node(ID_NO,4)>0
    p1=Q(ID_NO,1,1);       % SEQ_NO of the first packet in the Q
    p2=Q(ID_NO,1,2);       % CP_VER "   "       "    "
    p3=Q(ID_NO,1,3);       % HOP_VER "   "       "    "
  else
  end
end
if node(ID_NO,4)>0
  % **Update TX's node()
  p1=Q(ID_NO,1,1);       % SEQ_NO of the first packet in the Q
  p2=Q(ID_NO,1,2);       % CP_VER "   "       "    "
  p3=Q(ID_NO,1,3);      % HOP_VER "   "       "    "
  p4=nexthop_ds(ID_NO,pkt(p1,p2,p3,2),d,n_tot,cvr_rad);
  if p4==0      % No next hop available so drop the packet from Q.
    node(ID_NO,7:11)=0;       % ACT_MOD/TX_TYP/SEQ_NO_TRNST/CP_VER/HOP_VER=0
    node(ID_NO,14)=5;        % CH_IDL_CT=5
    node(ID_NO,15)=0;        % COLN=0
    node(ID_NO,6)=0;        % CW is reset to min.
    node(ID_NO,5)=min(floor(rand(1,1)*(2^(3+node(ID_NO,6)))), ... % BK_OFF_CT
              2^(3+node(ID_NO,6))-1)*2;
    % Drop procedure
    pkt(p1,p2,p3,5)=0;  % IMDT_DST/Q_POS=0
    pkt(p1,p2,p3,11)=5; % Drop due route unavailability.
    pkt(p1,p2,p3,14)=0; % CNTR=0;
    drop=drop+1;
    % Update TX's Q
    if node(ID_NO,4)>1
      for jj=1:node(ID_NO,4)-1  % jj=1:Q_LTH-1
        Q(ID_NO,jj,:)=Q(ID_NO,jj+1,:); % Shift the queue.
        pkt(Q(ID_NO,jj,1),Q(ID_NO,jj,2),Q(ID_NO,jj,3),5)=pkt(Q(ID_NO,jj,1),Q(ID_NO,jj,2),Q(ID_NO,jj,3),5)-1;
                                                                      % Q_POS updated.
      end
      pkt(Q(ID_NO,1,1),Q(ID_NO,1,2),Q(ID_NO,1,3),8)=pkt(Q(ID_NO,1,1),Q(ID_NO,1,2),Q(ID_NO,1,3),8)+ ...
      RL_TM_CT-pkt(Q(ID_NO,1,1),Q(ID_NO,1,2),Q(ID_NO,1,3),14); % TM_IN_Q update
      pkt(Q(ID_NO,1,1),Q(ID_NO,1,2),Q(ID_NO,1,3),14)=0;
    else
    end
    Q(ID_NO,node(ID_NO,4),:)=0; % Last occupied location is set to zero.
  node(ID_NO,4)=node(ID_NO,4)-1;     % Decrement Q_LTH
else
  if p4~=pkt(p1,p2,p3,4)       % If the new IMDT_DST is not the same as
                               % the old IMDT_DST in the previous TX
                               % attempt, then a new copy version is
                               % created.
    if  pkt(p1,p2,p3,4)==0     % This is the first access channel trial
                             % for a spacific packet.
    else
      ver(p1)=ver(p1)+1;
      pkt(p1,ver(p1),p3,:)=pkt(p1,p2,p3,:);
      pkt(p1,p2,p3,5)=0;     % Q_POS=0           % Old copy is eliminated.
      pkt(p1,p2,p3,14)=0;     % CNTR=0;
      p2=ver(p1);
      % **Replace the old copy with the new one in the Q.
      Q(ID_NO,1,2)=ver(p1);
    end
    pkt(p1,p2,p3,4)=p4;      % IMDT_DST
  else
  end
  node(ID_NO,9)=p1;        % Set SEQ_NO_TRNST
  node(ID_NO,10)=p2;        % Set CP_VER
  node(ID_NO,11)=p3;        % Set HOP_VER
  node(ID_NO,7)=1;         % ACT_MOD=TX
  node(ID_NO,8)=1;         % TX_TYP=RTS
  node(ID_NO,13)=RTS+SIFS;  % ITER_TO_GO=RTS+SIFS
  node(ID_NO,14)=5;        % CH_IDL_CT=5
  node(ID_NO,15)=0;        % COLN=0
  % Update RX's node()
  node(p4,13)=max(RTS+SIFS,node(p4,13));  % ITER_TO_GO=max(RTS+SIFS,old value)
  node(p4,14)=5;       % CH_IDL_CT=5
  if (node(p4,7)==0) & (node(p4,12)==0)        % ACT_MOD=IDL & NAV=0?
    node(p4,7)=2;        % ACT_MOD=RX
    node(p4,8)=1;        % TX_TYP=RTS
    node(p4,9)=node(ID_NO,9);  % Set SEQ_NO_TRNST
    node(p4,10)=node(ID_NO,10);   % Set CP_VER
    node(p4,11)=node(ID_NO,11);   % Set HOP_VER
    node(p4,15)=0;        % COLN=0
  else
    if  node(p4,7)~=0
      node(p4,15)=1;        % COLN=1
    elseif (node(p4,12)~=0)
      node(p4,7)=5;
      node(p4,8)=1;
      node(p4,9)=node(ID_NO,9);  % Set SEQ_NO_TRNST
      node(p4,10)=node(ID_NO,10);   % Set CP_VER
      node(p4,11)=node(ID_NO,11);   % Set HOP_VER
      node(p4,15)=0;        % COLN=0
    end
  end
  % At other nodes inside the coverage area of transmitter
```

```
      adj=nodecover_ds(ID_NO,d,n_tot,cvr_rad);
      if adj(1)~=0
         for jj=1:length(adj)
            if adj(jj)~=p4
               node(adj(jj),14)=5;         % CH_IDL_CT=5
               node(adj(jj),13)=max(node(adj(jj),13),RTS+SIFS);   % ITER_TO_GO
               switch node(adj(jj),7)       % ACT_MOD
                  case 2               % RX
                     node(adj(jj),15)=1;   % COLN is recorded.
                  case 3            % LSTN
                     node(adj(jj),15)=1;    % COLN is recorded.
                  case 0            % IDL
                     node(adj(jj),7)=3; % node is turned to LSTN mode.
                     node(adj(jj),8)=1; % TX_TYP=RTS
                     node(adj(jj),9)=node(ID_NO,9);   % Set SEQ_NO_TRNST
                     node(adj(jj),10)=node(ID_NO,10);  % Set CP_VER
                     node(adj(jj),11)=node(ID_NO,11);  % Set HOP_VER
                  otherwise
               end
            else
            end
         end
      else
      end
      clear adj
   end
else
   node(ID_NO,6)=0;
   node(ID_NO,5)=min(floor(rand(1,1)*(2^(3+node(ID_NO,6)))), ...
                 2^(3+node(ID_NO,6))-1)*2;
   node(ID_NO,7:11)=0;
   node(ID_NO,14)=5;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% src_mod_ds.m
% implements the traffic generation.

%for y=1:length(PKT_GEN)
for y=1:yy-1
   ID_NO=PKT_GEN(y);
   x=rand(1,1);
   % **If source ends up in active state it produces a packet
   if ((status(y)==0) & (x<=beta)) | ((status(y)==1) & (x>alpha))
      for aa=1:burst
         SEQ_NO=SEQ_NO+1;          % A packet is generated
         ver(SEQ_NO)=1;
         pkt(SEQ_NO,1,1,1)=ID_NO;
         pkt(SEQ_NO,1,1,2)=dst(ID_NO);
         pkt(SEQ_NO,1,1,3)=ID_NO;
         pkt(SEQ_NO,1,1,6)=RL_TM_CT;
         status(y)=0;            % The source ends up in active state.
         if node(ID_NO,4)<BFR_LTH     % Generated packet sits in the queue.
            node(ID_NO,4)=node(ID_NO,4)+1;
            pkt(SEQ_NO,1,1,5)=node(ID_NO,4);
            archive(ID_NO,archive(ID_NO,1)+2)=SEQ_NO;  % Gen. packet goes in archive.
            archive(ID_NO,1)=archive(ID_NO,1)+1;       % archive length is incremented.
            if pkt(SEQ_NO,1,1,5)~=1
               pkt(SEQ_NO,1,1,14)=RL_TM_CT; % Record time of enterance to Q.
            else
            end
            Q(ID_NO,node(ID_NO,4),1)=SEQ_NO; % Q() is updated.
            Q(ID_NO,node(ID_NO,4),2)=1;
            Q(ID_NO,node(ID_NO,4),3)=1;
        else
            pkt(SEQ_NO,1,1,13)=1;         % Packet blocked due to full buffer.
            block=block+1;
        end
     end
   else
      status(y)=1;            % Source  becomes idle
   end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% unit_step_ds.m
% implemets unit step function u(x)

function y=unit_step(x)
y=sign(sign(x)+1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```