MAJOR PROJECT

Study Of Analytical Model And Performance Analysis Of IEEE 802.11 Wireless LAN Using Ns-2

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF ENGINEERING

IN

COMPUTER TECHNOLOGY AND APPLICATION

SUBMITTED BY :

PAYAL SINGLA

(Roll No. 3006)

UNDER THE ESTEEMED GUIDANCE
OF

Dr. ASOK DE

DEPARTMENT OF COMPUTER ENGINEERING
DELHI COLLEGE OF ENGINEERING
UNIVERSITY OF DELHI
2004 -2005

Department of Computer Engineering
Delhi College of Engineering, Delhi-42

CERTIFICATE

This is to certify that the project report entitled

"STUDY OF ANALYTICAL MODEL AND PERFORMANCE ANALYSIS OF IEEE 802.11 WIRELESS LAN USING NS-2 "

being submitted by **PAYAL SINGLA (Roll no.3006)**

is a bonafide record of her own work carried under our guidance and supervision in partial fulfillment for the award of the degree of Master of Engineering in Computer Technology and Applications from Delhi College of Engineering, Delhi.

Prof. Asok De
HOD,
Department of Information Technology,
Delhi College of Engg,Delhi

Dr. D. Roy Choudhury
Professor & HOD,
Department of Computer Engineering,
Delhi College of Engg, Delhi

ACKNOWLEDGEMENT

I appreciate the contribution and support which various individuals have provided for the successful completion of this work. It may not be possible to mention all by name but the following were singled out for their exceptional help.

It is with immense pleasure that I acknowledge my gratitude to my guide **Prof.Asok De,** Head of Department, Department of Information Technology, Delhi College of Engineering, Delhi, for his scholastic guidance and sagacious suggestions throughout this work. His immense generosity and affection bestowed on me goes beyond his formal obligations as guide.

I am deeply indebted to **Dr. D. Roy Choudhury**, HOD, Department of Computer Engineering, Delhi College of Engineering, Delhi, for his constant encouragement, valuable guidance, resourceful suggestions and alignment evaluations throughout the course of this project.

It is with immense affection that I acknowledge my gratitude to **Sh.P.K.Hazra**, Asst. Professor, Department of Computer Sc., Delhi University, for his perpetual encouragement, generous help and inspiring guidance throughout this work.

I would like to express my sincere thanks to **Dr. P. B. Sharma**, Principal, Delhi College of Engineering, Delhi to allow me to perform this work.

The moral support and love given by my friends **Shraddha Singhai**, **Supriya Sharma**, **Shweta Sharma**, and **Sunita Verma** help me stood when were times difficult.

It would be gross mistake on my part, if I forget the love and blessings of **my family** that I could not have raised to this height in my life.

(Payal Singla)

DEDICATED

TO

MY MOTHER

Mrs. Anita Singla

ABSTRACT

The IEEE has standardized the 802.11 protocol for Wireless Local Area Networks. The primary medium access control (MAC) technique of 802.11 is called distributed coordination function (DCF). DCF is a carrier sense multiple access with collision avoidance (CSMA/CA) scheme with binary slotted exponential back off. Here, a simple, but accurate, analytical model to compute the 802.11 DCF throughput has been studied, in the assumption of finite number of terminals and ideal channel conditions. The proposed analysis applies to the packet transmission scheme employed by DCF, namely, the RTS/CTS access mechanisms. By means of the proposed model, in this dissertation we provide an extensive throughput performance evaluation of RTS/CTS access mechanisms of the 802.11 protocol using ns-2 simulator. The study of the QoS enhancements to the IEEE 802.11 standard, named IEEE 802.11e, currently under specification has also been discussed. Both the Enhanced Distributed Coordination Function (EDCF) and Hybrid Coordination Function (HCF) modes of Medium Access Control (MAC) operation are described.

CONTENTS

Page No.

LIST OF FIGURES

Page No.

CHAPTER-1

INTRODUCTION

INTRODUCTION

The use of wireless LAN is quickly becoming an established technology in educational institutions and many commercial places like hotels, cafeterias for internet access, multimedia communication, file transfer etc. There are millions of people around the globe who are using wireless LAN technology. Wireless LAN technology has greatly improved our ability to work and communicate at home or at work in our local and global communities. More and more people are using wireless LAN technology not only for work, but also for the convenience. Wireless local area network or WLAN, provides an excellent way to extend the reach of local area networks, through a wireless connection.

WLAN is a technology that enables connecting computers to a network wirelessly with high bit rates, compared to IR and Blue tooth. The purpose of WLAN was originally to enable in office working without the hassle of network cables, but it has since evolved and is still currently evolving very rapidly towards offering fast connection capabilities within large area. WLAN is in great use due to the following predominant reasons:

- User mobility.
- Availability of higher bandwidth compared to other cellular technologies like GSM, CDMA.
- Availability of infrastructure based as well as infrastructure less WLAN where mobile adhoc networks can be established in far flung areas, ware fields where provision of connectivity is not possible.
- Integration of WLAN with classical Ethernet can be easily done through bridging.
- High bandwidth internet connectivity can be easily incorporated through specially designed routers.

The first WLAN standard, IEEE 802.11 is based on radio technology operating in the 2.4 GHz frequency and has a maximum throughput of 1 to 2 Mbits per second. The general idea of WLAN was basically just to provide a wireless network infrastructure comparable to the wired Ethernet networks in use. Currently, the most spread and deployed standard is the IEEE 802.11b.Standards like the 802.11a, 802.11b and 802.11g have been published and these standards improve on data transfer rates. The 802.11b operates in the 2.4 GHz band, data rates can be as high as 11Mbps.

The 802.11a standard was published as a supplement to the 802.11. It operates in the 5GHz band instead of the traditional 2.4Ghz that the earlier WLAN standards used, thus being subjected to less interference and supports data rates up to 54Mbps. 802.11a is not compatible with 802.11b and therefore its emergence has been quite slow. The disadvantage with the 5Ghz frequency is the reduced working distance.

IEEE 802.11 uses a system known as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as its Distributed Coordination Function (DCF). All stations participating in the network use the same CSMA/CA system to coordinate access to the shared communication medium. DCF is the basic access method in 802.11 and operates both on infrastructure based and infrastructure less. Here in this dissertation, following objectives are performed-

1. Study of analytical model of IEEE 802.11 Wireless LAN.
2. Performance analysis of IEEE 802.11 Wireless LAN using ns-2 simulator.

The IEEE is developing the 802.11e standard as an extension of the 802.11 standard with backward compatibility and Quality of Service (QoS) features. The 802.11e standard mainly concerns the MAC layer protocol. That means the modification is only in the MAC layer. The 802.11e can provide different service quality for traffic streams with different priorities. Two mechanisms have been discussed here- EDCF and HCF.

We begin in **Chapter 2** by describing the features of the wireless LAN, 802.11 Medium Access Control (MAC) sublayer protocol. This includes a brief description of the Distributed Coordination function (DCF) and the Point Coordination Function (PCF).

Chapter3 gives a detailed description of DCF protocol.

Chapter 4 describes the analytical study of IEEE 802.11 wireless LAN DCF protocol.

Chapter 5 is the heart of the dissertation. It is divided into two parts. The first part describes NS-2 features and the second part describes the experiments that were run and interprets the graphical results.

Chapter 6 gives an idea about the IEEE 802.11e i.e. quality of service of 802.11. EDCF and HCF have been discussed.

Chapter 7 adds some concluding remarks and suggests some future work that could be done to extend this analysis.

CHAPTER-2

WIRELESS LAN

WIRELESS LAN

In this dissertation work is aimed at close range networks, which are often called Wireless Local Area Networks(WLANs). Recently, hardware prices have dropped drastically for infrastructure equipment, and as a result of this, WLANs[1] are deployed almost everywhere. The most common standard for these networks today are the IEEE 802.11 standard[3]. There exists other standards such as HiperLan/2[4], and HomeRF[5] but they are not as widely used.

Wireless LANs are increasingly popular, and more office buildings, airports, and other public places are being outfitted with them. Wireless LANs can operate in one of two configurations, with a base station and without a base station. Consequently, the 802.11 LAN standard takes this into account and makes provision for both arrangements. In the following sections we will look at the protocol stack, physical layer radio transmission techniques, MAC sub layer protocol, frame structure, and services.

2.1 Attributes of Wireless LAN's

Wireless LANs must adhere to the many of the same rules as traditional wired LANs, including full connectivity to stations, the ability to broadcast, high capacity, etc. In addition, wireless LANs have some special requirements unique to their form of communication. A few of these follow:

- **Throughput -** Due to the decreased bandwidth of radio and IR channels, the Medium Access Control (MAC) protocol should make as efficient use of this available bandwidth as possible.

• **Backbone Connectivity** - In most cases, wireless LANs connect to some sort of internal (wired) network. Therefore, facilities must be provided to make this connection. This is usually one station that also serves as the Access Point (AP) to the wired LAN for all stations.

• **Power Considerations -** Often times, wireless stations are small battery powered units. Algorithms that require the station to constantly check the medium or perform other tasks frequently may be inappropriate.

• **Roaming -** Wireless stations should be able to move freely about their service area.

• **Dynamic** - The addition, deletion, or relocation of wireless stations should not affect other users.

• **Licensing** - In order to gain widespread popularity, it is preferred that licenses not be required to operate wireless LAN's.

2.2 Challenges

The link characteristics in wireless environments are very different from that of wired networks. At link layer we have to face following challenges:

Bandwidth: Bandwidth is the one of the scarcest resource in wireless networks. The available bandwidth in wireless networks (2-10Mbps) is far less than the wired links (typically 100Mbps).

Range Issues: The transmission range of stations depends upon the transmitted power and various sensitivity values. Unlike wired networks all stations on a LAN can not listen to one another.

Power: The wireless stations are battery operated and therefore higher transmission power leads to faster degeneration of the batteries. On the other hand, if we keep transmission power too small, the stations may no longer be in range of each other.

Collisions: Since all stations cannot listen to each other, transmission from two stations may lead to collision at another station.

Link Errors: Channel fading and interference cause link errors and these errors may sometimes be very severe.

2.3 The 802.11 Protocol Stack

The protocols used by all the 802 variants, including Ethernet, have a certain commonality of structure. A partial view of the 802.11 protocol stack is given in Fig.2.1. The physical layer corresponds to the OSI physical layer fairly well, but the data link layer in all the 802 protocols is split into two or more sub layers.

In 802.11, the MAC (Medium Access Control) sub layer determines how the channel is allocated, that is, who gets to transmit next. Above it is the LLC (Logical Link Control) sub layer, whose job it is to hide the differences between the different 802 variants and make them indistinguishable as far as the network layer is concerned. The 1997 802.11 standard specifies three transmission techniques allowed in the physical layer. The infrared method uses much the same technology as television remote controls do. The other two use short-range radio, using techniques called FHSS and DSSS. Both of these use a part of the spectrum that does not require licensing (the 2.4-GHz ISM band). Cordless telephones and microwave ovens also use this band. All of these techniques operate at 1 or 2 Mbps and at low enough power that they do not conflict too much. In 1999, two new techniques were introduced to achieve higher bandwidth. These are called OFDM and HRDSSS. They operate at up to 54 Mbps and 11 Mbps, respectively. In

2001, a second OFDM modulation was introduced, but in a different frequency band from the first one.

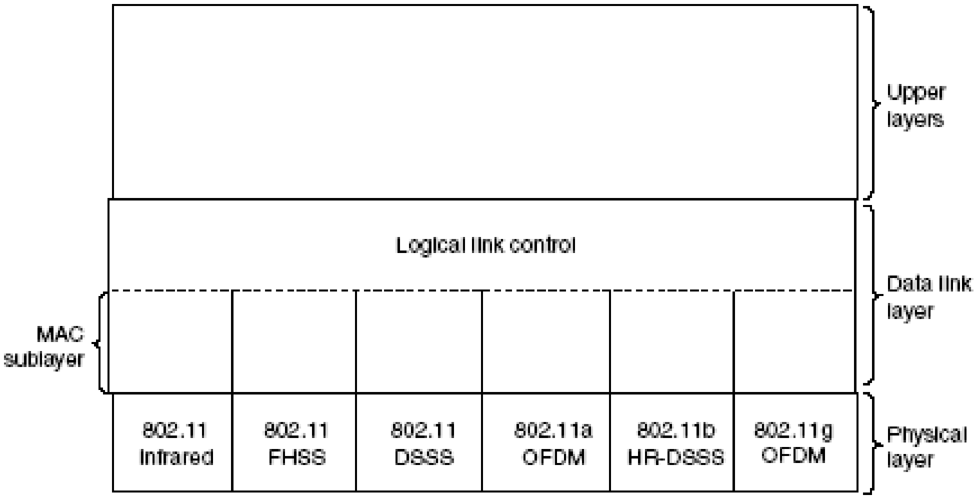

Figure 2.1 802.11 protocol stack.

2.4 The 802.11 Physical Layer

Each of the five permitted transmission techniques makes it possible to send a MAC frame from one station to another. They differ, however, in the technology used and speeds achievable.

The **infrared** option uses diffused (i.e., not line of sight) transmission at 0.85 or 0.95 microns. Two speeds are permitted: 1 Mbps and 2 Mbps. At 1 Mbps, an encoding scheme is used in which a group of 4 bits is encoded as a 16-bit codeword containing fifteen 0s and a single 1, using what is called **Gray code**. This code has the property that a small error in time synchronization leads to only a single bit error in the output. At 2 Mbps, the encoding takes 2 bits and produces a 4-bit codeword, also with only a single 1, that is one of 0001, 0010, 0100, or 1000. Infrared signals cannot penetrate walls, so cells in different rooms are well isolated from each other. Nevertheless, due to the low bandwidth (and the fact that sunlight swamps infrared signals), this is not a popular option.

FHSS (**Frequency Hopping Spread Spectrum**) uses 79 channels, each 1- MHz wide, starting at the low end of the 2.4-GHz ISM band. A pseudorandom number generator is used to produce the sequence of frequencies hopped to. As long as all stations use the same seed to the pseudorandom number generator and stay synchronized in time, they will hop to the same frequencies simultaneously. The amount of time spent at each frequency, the **dwell time**, is an adjustable parameter, but must be less than 400 msec. FHSS' randomization provides a fair way to allocate spectrum in the unregulated ISM band. It also provides a modicum of security since an intruder who does not know the hopping sequence or dwell time cannot eavesdrop on transmissions. Over longer distances, multipath fading can be an issue, and FHSS offers good resistance to it. It is also relatively insensitive to radio interference, which makes it popular for building-to-building links. Its main disadvantage is its low bandwidth.

The third modulation method, **DSSS** (**Direct Sequence Spread Spectrum**), is also restricted to 1 or 2 Mbps. Each bit is transmitted as 11 chips, using what is called a **Barker sequence**. It uses phase shift modulation at 1 Mbaud, transmitting 1 bit per baud when operating at 1 Mbps and 2 bits per baud when operating at 2 Mbps.

The first of the high-speed wireless LANs, **802.11a**, uses **OFDM** (**Orthogonal Frequency Division Multiplexing**) to deliver up to 54 Mbps in the wider 5- GHz ISM band. Since transmissions are present on multiple frequencies at the same time, this technique is considered a form of spread spectrum, but different from FHSS. Splitting the signal into many narrow bands has some key advantages over using a single wide band, including better immunity to narrowband interference and the possibility of using noncontiguous bands.

Next, we come to **HR-DSSS** (**High Rate Direct Sequence Spread Spectrum**); another spread spectrum technique, which uses 11 million chips/sec to achieve 11 Mbps in the 2.4-GHz band. It is called **802.11b** but is not a follow-up to 802.11a[18,22]. In fact, its standard was approved first and it got to market first. Data rates supported by 802.11b are

1, 2, 5.5, and 11 Mbps. The two slow rates run at 1 Mbaud, with 1 and 2 bits per baud, respectively, using phase shift modulation (for compatibility with DSSS). The two faster rates run at 1.375 Mbaud, with 4 and 8 bits per baud, respectively, using **Walsh/Hadamard** codes. The data rate may be dynamically adapted during operation to achieve the optimum speed possible under current conditions of load and noise. In practice, the operating speed of 802.11b is nearly always 11 Mbps. Although 802.11b is slower than 802.11a, its range is about 7 times greater, which is more important in many situations.

An enhanced version of 802.11b, **802.11g**, was approved by IEEE in November 2001 after much politicking about whose patented technology it would use. It uses the OFDM modulation method of 802.11a but operates in the narrow 2.4- GHz ISM band along with 802.11b. In theory it can operate at up to 54 MBps. It is not yet clear whether this speed will be realized in practice.

2.5 802.11 Family

• 802.11a	54 Mbps	5GHz
• 802.11b	11 Mbps	2.4GHz
• 802.11c	Bridge Operation Procedures	
• 802.11d	Global Harmonization	
• 802.11e	MAC enhancements for QoS	
• 802.11f	Inter Access Point Protocol (roaming)	
• 802.11g	54 Mbps 2.4 GHz	
• 802.11h	Dynamic Frequency Selection	
• 802.11i	Security	

2.6 The 802.11 MAC Sub layer Protocol

Now return from the land of electrical engineering to the land of computer science. The 802.11 MAC sub layer protocol is quite different from that of Ethernet due to the inherent complexity of the wireless environment compared to that of a wired system. With Ethernet, a station just waits until the ether goes silent and starts transmitting. If it does not receive a noise burst back within the first 64 bytes, the frame has almost assuredly been delivered correctly. With wireless, this situation does not hold.

To start with, there is the hidden station problem illustrated in Fig. 2.2(a). Since not all stations are within radio range of each other, transmissions going on in one part of a cell may not be received elsewhere in the same cell. In this example, station *C* is transmitting to station *B*. If *A* senses the channel, it will not hear anything and falsely conclude that it may now start transmitting to *B*. In addition, there is the inverse problem, the exposed station problem, illustrated in Fig. 2.2(b). Here *B* wants to send to *C* so it listens to the channel. When it hears a transmission, it falsely concludes that it may not send to *C*, even though *A* may be transmitting to *D* (not shown). In addition, most radios are half duplex, meaning that they cannot transmit and listen for noise bursts at the same time on a single frequency. As a result of these problems, 802.11 do not use CSMA/CD, as Ethernet does.

To deal with this problem, 802.11 support two modes of operation. The first, called **DCF** (**Distributed Coordination Function**)[12] does not use any kind of central control (in that respect, similar to Ethernet). The other called **PCF** (**Point Coordination Function**)[17,21] uses the base station to control all activity in its cell. All implementations must support DCF but PCF is optional. We will now discuss these two modes in turn.

When DCF is employed, 802.11 uses a protocol called **CSMA/CA** (**Carrier Sense**

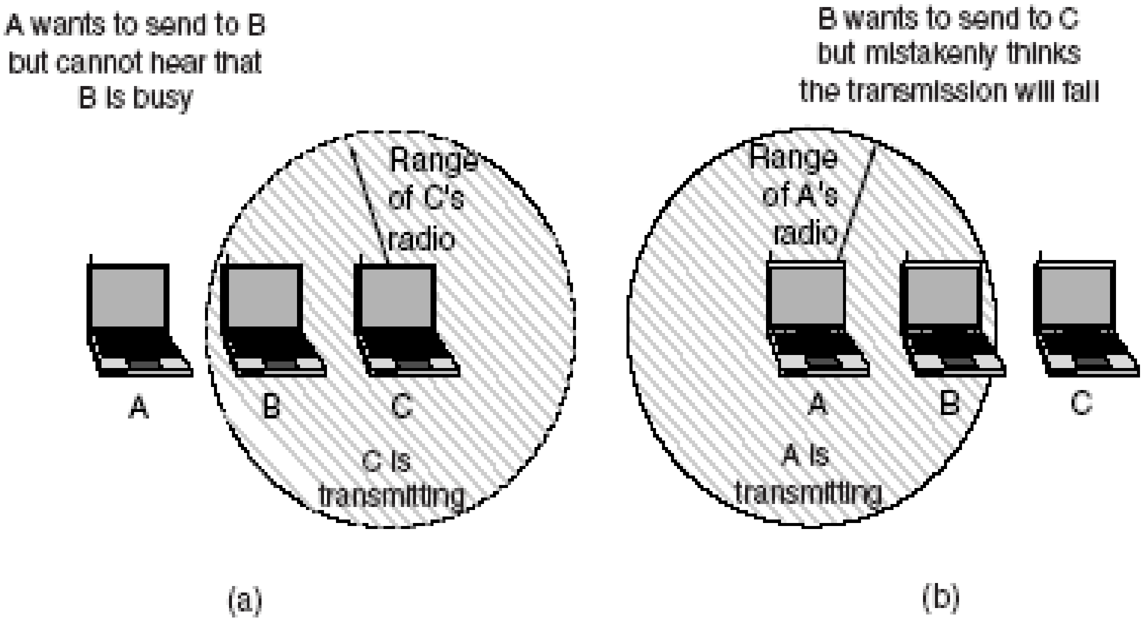

Figure 2.2**.** (a) The hidden station problem. (b) The exposed station problem.

Multiple Access with Collision Avoidance). In this protocol, both physical channel sensing and virtual channel sensing are used. Two methods of operation are supported by CSMA/CA. In the first method, when a station wants to transmit, it senses the channel. If it is idle, it just starts transmitting. It does not sense the channel while transmitting but emits its entire frame, which may well be destroyed at the receiver due to interference there. If the channel is busy, the sender defers until it goes idle and then starts transmitting. If a collision occurs, the colliding stations wait a random time, using the Ethernet binary exponential back off algorithm, and then try again later.

The other mode of CSMA/CA operation uses virtual channel sensing, as illustrated in Fig. 2.3. In this example, *A* wants to send to *B*. C is a station within range of *A* (and possibly within range of *B*, but that does not matter). *D* is a station within range of *B* but not within range of *A*.

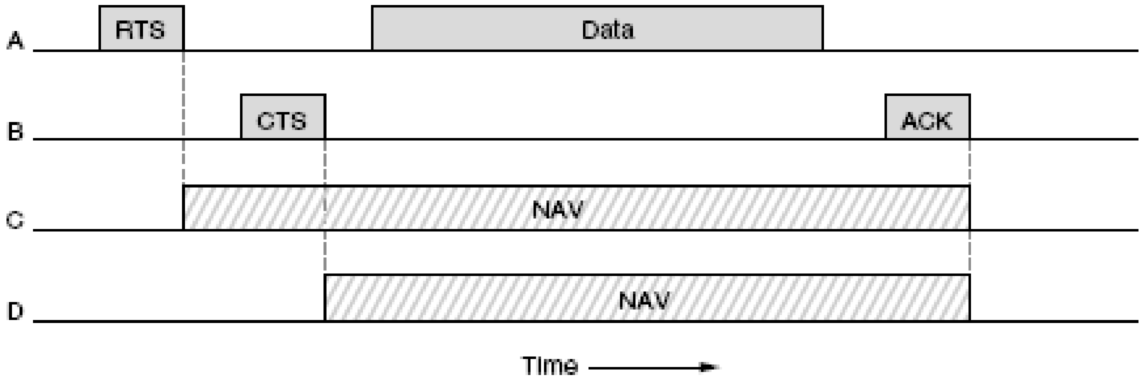

Figure 2.3. The use of virtual channel sensing using CSMA/CA.

The protocol starts when *A* decides it wants to send data to *B*. It begins by sending an RTS frame to *B* to request permission to send it a frame. When *B* receives this request, it may decide to grant permission, in which case it sends a CTS frame back. Upon receipt of the CTS, *A* now sends its frame and starts an ACK timer. Upon correct receipt of the data frame, *B* responds with an ACK frame, terminating the exchange. If *A*'s ACK timer expires before the ACK gets back to it, the whole protocol is run again. Now let us consider this exchange from the viewpoints of *C* and *D*. *C* is within range of *A*, so it may receive the RTS frame. If it does, it realizes that someone is going to send data soon, so for the good of all it desists from transmitting anything until the exchange is completed. From the information provided in the RTS request, it can estimate how long the sequence will take, including the final ACK, so it asserts a kind of virtual channel busy for itself, indicated by NAV (Network Allocation Vector) in Fig. 2.3. *D* does not hear the RTS, but it does hear the CTS, so it also asserts the *NAV* signal for itself. Note that the *NAV* signals are not transmitted; they are just internal reminders to keep quiet for a certain period of time.

In 802.11 DCF mode, there is no central control, and stations compete for air time, just as they do with Ethernet. The other allowed mode is PCF, in which the base station polls the other stations, asking them if they have any frames to send. Since transmission order is completely controlled by the base station in PCF mode, no collisions ever occur.

The basic mechanism is for the base station to broadcast a **beacon frame** periodically (10 to 100 times per second). The beacon frame contains system parameters, such as hopping sequences and dwell times (for FHSS), clock synchronization, etc. It also invites new stations to sign up for polling service. Once a station has signed up for polling service at a certain rate, it is effectively guaranteed a certain fraction of the bandwidth, thus making it possible to give quality-of service guarantees. Battery life is always an issue with mobile wireless devices, so 802.11 pays attention to the issue of power management. In particular, the base station can direct a mobile station to go into sleep state until explicitly awakened by the base station or the user. Having told a station to go to sleep, however, means that the base station has the responsibility for buffering any frames directed at it while the mobile station is asleep. These can be collected later. PCF and DCF can coexist within one cell. At first it might seem impossible to have central control and distributed control operating at the same time, but 802.11 provides a way to achieve this goal. It works by carefully defining the interframe time interval. After a frame has been sent, a certain amount of dead time is required before any station may send a frame. Four different intervals are defined, each for a specific purpose. The four intervals are depicted in Fig. 2.4.

The shortest interval is **SIFS** (**Short InterFrame Spacing**). It is used to allow the parties in a single dialog the chance to go first. This includes letting the receiver send a CTS to respond to an RTS, letting the receiver send an ACK for a fragment or full data frame. There is always exactly one station that is entitled to respond after a SIFS interval. If it fails to make use of its chance and a time **PIFS** (**PCF InterFrame Spacing**)

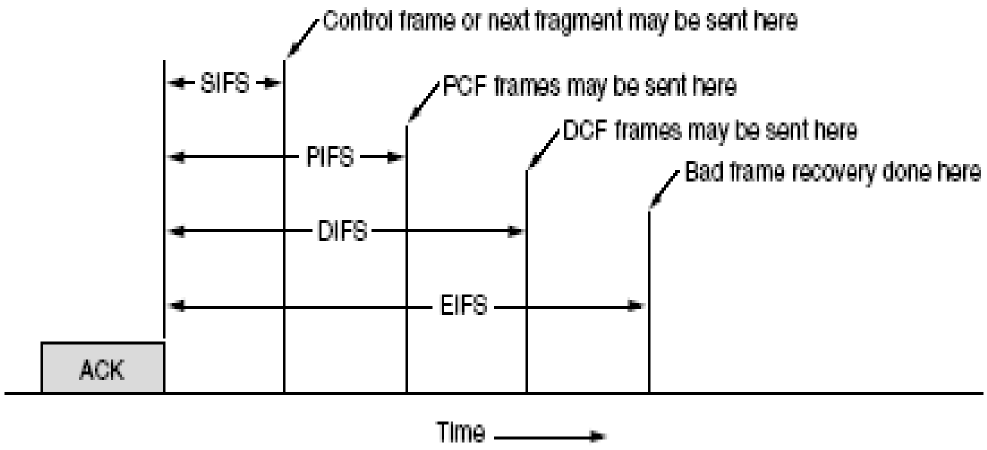
Control frame or next fragment may be sent here
SIFS
PCF frames may be sent here
PIFS
DCF frames may be sent here
DIFS
Bad frame recovery done here
EIFS
ACK
Time

Figure 2.4. Interframe spacing in 802.11

elapses, the base station may send a beacon frame or poll frame. This mechanism allows a station sending a data frame or fragment sequence to finish its frame without anyone else getting in the way, but gives the base station a chance to grab the channel when the previous sender is done without having to compete with eager users. If the base station has nothing to say and a time **DIFS** (**DCF InterFrame Spacing**) elapses, any station may attempt to acquire the channel to send a new frame. The usual contention rules apply, and binary exponential back off may be needed if a collision occurs.

The last time interval, **EIFS** (**Extended InterFrame Spacing**), is used only by a station that has just received a bad or unknown frame to report the bad frame. The idea of giving this event the lowest priority is that since the receiver may have no idea of what is going on, it should wait a substantial time to avoid interfering with an ongoing dialog between two stations.

2.7 Services

The 802.11 standard[24] states that each conformant wireless LAN must provide nine services. These services are divided into two categories: five distribution services and four station services. The distribution services relate to managing cell membership and interacting with stations outside the cell. In contrast, the station services relate to activity within a single cell. The five distribution services are provided by the base stations and deal with station mobility as they enter and leave cells, attaching themselves to and detaching themselves from base stations. They are as follows.

1. **Association**. This service is used by mobile stations to connect themselves to base stations. Typically, it is used just after a station moves within the radio range of the base station. Upon arrival, it announces its identity and capabilities. The capabilities include the data rates supported, need for PCF services (i.e., polling), and power management requirements. The base station may accept or reject the mobile station. If the mobile station is accepted, it must then authenticate itself.

2. **Disassociation**. Either the station or the base station may disassociate, thus breaking the relationship. A station should use this service before shutting down or leaving, but the base station may also use it before going down for maintenance.

3. **Re-association**. A station may change its preferred base station using this service. This facility is useful for mobile stations moving from one cell to another. If it is used correctly, no data will be lost as a consequence of the handover.

4. **Distribution**. This service determines how to route frames sent to the base station. If the destination is local to the base station, the frames can be sent out directly over the air. Otherwise, they will have to be forwarded over the wired network.

5. **Integration**. If a frame needs to be sent through a non-802.11 network with a different addressing scheme or frame format, this service handles the translation from the 802.11 formats to the format required by the destination network.

The remaining four services are intra cell (i.e., relate to actions within a single cell). They are used after association has taken place and are as follows.

1. **Authentication**. Because wireless communication can easily be sent or received by unauthorized stations, a station must authenticate itself before it is permitted to send data. After a mobile station has been associated by the base station (i.e., accepted into its cell), the base station sends a special challenge frame to it to see if the mobile station knows the secret key (password) that has been assigned to it. It proves its knowledge of the secret key by encrypting the challenge frame and sending it back to the base station. If the result is correct, the mobile is fully enrolled in the cell. In the initial standard, the base station does not have to prove its identity to the mobile station, but work to repair this defect in the standard is underway.

2. **Deauthentication**. When a previously authenticated station wants to leave the network, it is deauthenticated. After deauthentication, it may no longer use the network.

3. **Privacy**. For information sent over a wireless LAN to be kept confidential, it must be encrypted. This service manages the encryption and decryption.

4. **Data delivery**. Finally, data transmission is what it is all about, so 802.11 naturally provide a way to transmit and receive data. Since 802.11 is modeled on Ethernet and transmission over Ethernet is not guaranteed to be 100% reliable, transmission over 802.11 is not guaranteed to be reliable either. Higher layers must deal with detecting and correcting errors. An 802.11 cell has some parameters that can be inspected and, in some cases, adjusted. They relate to encryption, timeout intervals, data rates, beacon frequency, and so on.

Wireless LANs based on 802.11 are starting to be deployed in office buildings, airports, hotels, restaurants, and campuses around the world.

CHAPTER -3

IEEE 802.11 DISTRIBUTED COORDINATION FUNCTION

IEEE 802.11 DISTRIBUTED COORDINATION FUNCTION

3.1 Introduction

IEEE 802.11 uses a system known as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as its Distributed Coordination Function (DCF). All stations participating in the network use the same CSMA/CA system to coordinate access to the shared communication medium. DCF is the basic access method in 802.11 and operates both on infrastructure based and infrastructure less. When wireless stations are within transmit range of each other, they form a Basic Service Set (BSS), and can communicate to each other using DCF. If the BSS contains only two stations, it is called Independent Basic Service Set (IBSS). Many BSSs may be connected by a Distribution System (DS) to form an Extended Service Set (ESS). An access point (AP) is the station that provides access to DS services.

DCF describes two techniques to employ for packet transmission. The default scheme is a two-way handshaking technique called basic access mechanism. This mechanism is characterized by the immediate transmission of a positive acknowledgement (ACK) by the destination station, upon successful reception of a packet transmitted by the sender station. Explicit transmission of an ACK is required since, in the wireless medium, a transmitter cannot determine if a packet is successfully received by listening to its own transmission.

In addition to the basic access, an optional four way handshaking technique, known as request-to-send/clear-to-send (RTS/CTS) mechanism has been standardized. Before transmitting a packet, a station operating in RTS/CTS mode “reserves” the channel by sending a special Request-To-Send short frame. The destination station acknowledges the receipt of an RTS frame by sending back a Clear-To-Send frame, after which normal packet transmission and ACK response occurs. Since collision may occur only on the RTS frame, and it is detected by the lack of CTS response, the RTS/CTS mechanism allows to increase the system performance by reducing the duration f a collision when long messages are transmitted. As an important side effect, the RTS/CTS scheme designed in the 802.11 protocol is suited to combat the so-called problem of Hidden Terminals[2], which occurs when pairs of mobile stations result to be unable to hear each other.

3.2 Distributed Coordination Function

A station that wishes to transmit must first listen to the medium to detect if another station is using it. If so it must defer until the end of that transmission. If the medium is free then that station may proceed.

Two mechanisms are included to provide two separate carrier sense mechanisms. The traditional physical carrier sense mechanism is provided by the physical layer and is based upon the characteristics of the medium. In addition, the Medium Access Control (MAC) layer also provides a virtual mechanism to work in conjunction with the physical one. This virtual mechanism is referred to as the Network Allocation Vector (NAV). The NAV is a way of telling other stations the expected traffic of the transmitting station. A station’s medium is considered busy if either its virtual or physical carrier sense mechanisms indicate busy. Before a station can transmit a frame, it must wait for the medium to have been free for some minimum amount of time. This amount of time is called the Inter-frame Space (IFS). This presents an opportunity to establish a priority mechanism for access to the shared medium. Depending upon the state of the sending station, one of four Inter-Frame spaces is selected. In ascending order, these spaces are the Short IFS (SIFS), PCF IFS (PIFS), DCF IFS (DIFS), and Extended IFS (EIFS). The

MAC protocol defines instances where each IFS is used to support a given transmission priority.

A station wishing to transmit either a data or management frame shall first wait until its carrier sense mechanism indicates a free medium. Then, a DCF Inter-Frame Space will be observed. After this, the station shall then wait an additional random amount of time before transmitting. This time period is known as the back off interval. The purpose of this additional deferral is to minimize collisions between stations that may be waiting to transmit after the same event. This operation is called the Back off Procedure and is shown in figure 3.1[IEE97]

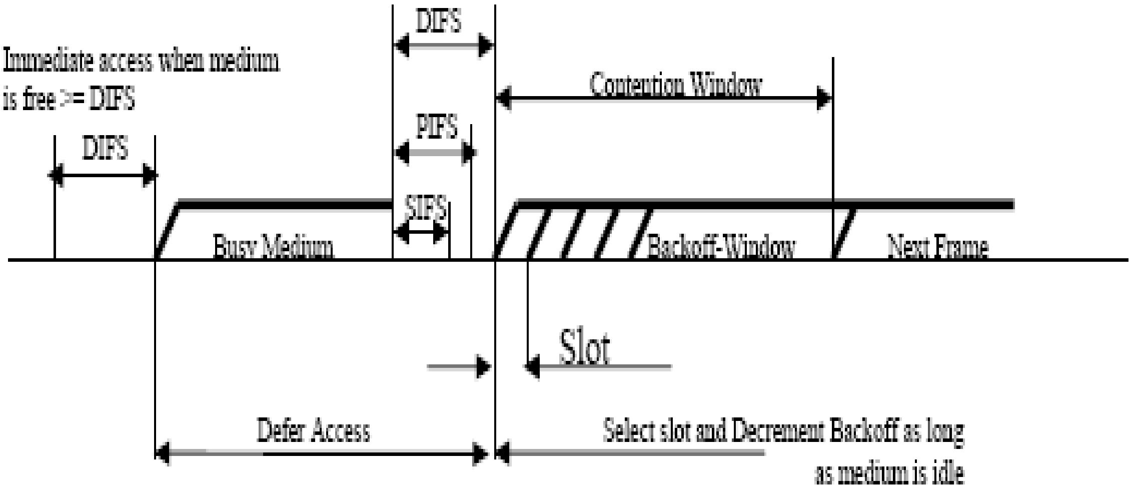
DIFS
Immediate access when medium is free >= DIFS
Contention Window
PIFS
DIFS
SIFS
Busy Medium
Backoff-Window
Next Frame
Slot
Defer Access
Select slot and Decrement Backoff as long as medium is idle

Fig. 3.1 Inter-Frame Space and Back off Window Relationship

Before a station can transmit a frame it must perform this back off procedure. The station first waits for a DIFS time upon noticing that the medium is free. If, after this time gap, the medium is still free the station computes an additional random amount of time to wait called the Back off Timer. The station will wait either until this time has elapsed or until the medium becomes busy, whichever comes first. If the medium is still free after the random time period has elapsed, the station begins transmitting its message. If the medium becomes busy at some point while the station is performing its back off procedure, it will temporarily suspend the back off procedure. In this case, the station must wait until the medium is free again, perform a DIFS again, and continue where it left off in the back off procedure. Note that in this case it is not necessary to re-compute a new Back off Timer. An example of the back off procedure is shown in figure 3.2[IEE97].

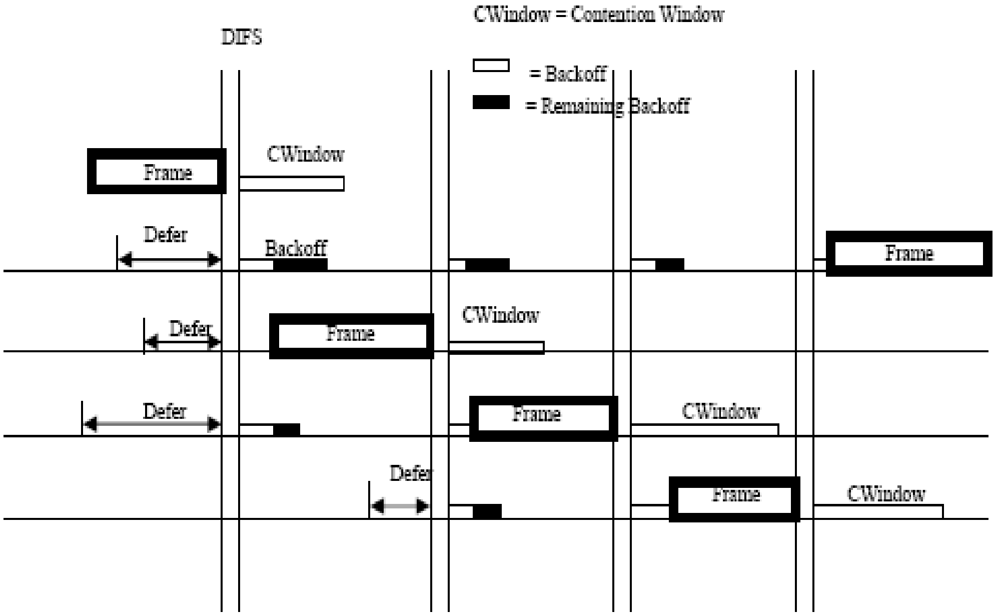

Fig. 3.2 Back off Procedure Example

Upon the reception of directed (not broadcast or multicast) frames with a valid CRC, the receiving station will respond back to the sending station an indication of successful reception, generally an acknowledgement (ACK). This process is known as positive acknowledgement. A lack of reception of this acknowledgement indicates to the sending station that an error has occurred. Of course, it is possible that the frame may have been successfully delivered and the acknowledgement was unsuccessful. This is indistinguishable from the case where the original frame itself is lost. As a result, it is possible for a destination station to receive more than one copy of a frame. It is therefore the responsibility of the destination to filter out all duplicate frames.

Additionally, 802.11 provide a request-to-send procedure, which is intended to reduce collisions. Stations gain access to the medium in the same way but instead of sending its first data frame, the station first transmits a small Request-to-Send (RTS) frame. The destination replies with a Clear-to- Send (CTS) frame. The NAV setting within both the RTS and CTS frames tell other stations how long the transmission is expected to be. By seeing these frames, other stations effectively turn on their virtual carrier sense mechanism for that period of time. While there may be high contention for the medium while the RTS frame is attempted, the remainder of the transmission should be relatively contention-free. This improves the performance of the protocol because all collisions occur on the very small RTS frames and not on the substantially larger data frames. Figure 3.3[IEE97] shows an example of an RTS exchange.

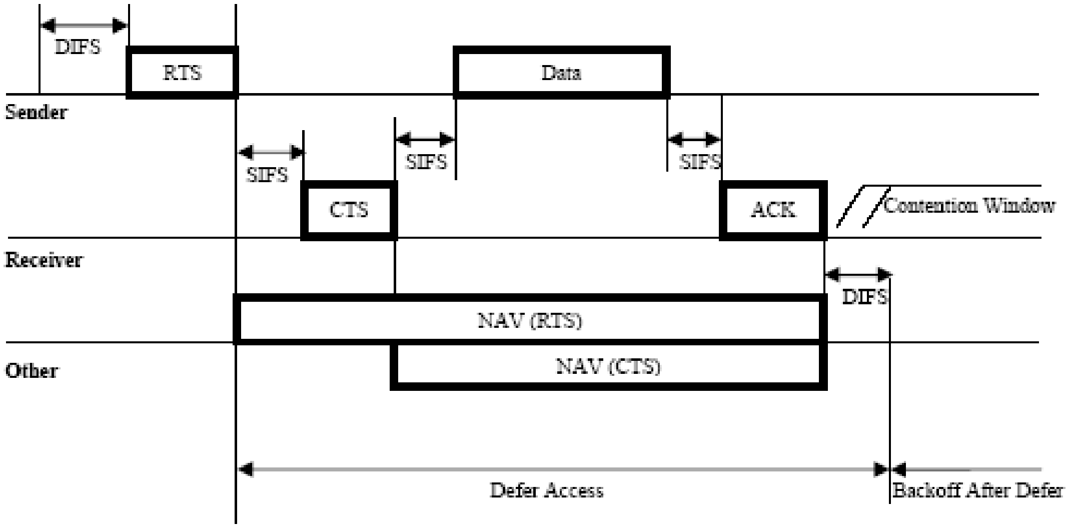

Fig 3.3 RTS exchange example

When beginning a transmission that will include more than one fragment, known as a fragment burst, the rules change slightly. Initially it appears identical to a single fragment transmission. The back off and carrier sense procedures are the same. The difference lies in the IFS used between fragments. Only a SIFS is required between fragments during a fragment burst. The reason for this is to give the sender the highest priority when transmitting a fragment burst. Consider two examples where this may come into play. In the first example a station with no knowledge of the NAV, perhaps having recently joined the network, must try to wait a DIFS before transmitting. After a shorter SIFS the original station takes over the medium with its next fragment and this other station, upon noticing a busy medium, must defer.

When transmitting broadcast or multicast frames, only the basic transfer mechanism is used. No RTS/CTS mechanism is used regardless of the size of the frame. Additionally, no receiving station will ever respond with an ACK to a broadcast or multicast frame.

CHAPTER-4

ANALYTICAL ANALYSIS OF THE IEEE 802.11 WLAN DCF PROTOCOL

ANALYTICAL ANALYSIS OF THE IEEE 802.11 WLAN DCF PROTOCOL

4.1 Introduction

This chapter aims at the analytical study of the saturation throughput based on [25], in the assumption of ideal channel conditions (i.e., no hidden terminals and capture[6]). In the analysis, a fixed number of stations, each always having a packet available for transmission is assumed. In other words, performance is analyzed in *saturation* conditions, i.e., the transmission queue of each station is assumed to be always nonempty. The analysis is divided into two distinct parts. First, the study of the behavior of a single station with a Markov model to obtain the stationary probability τ that the station transmits a packet in a generic (i.e., randomly chosen) slot time. Secondly, the throughput of RTS/CTS access method is expressed by studying the events that can occur within a generic slot time.

Saturation Throughput is a fundamental performance figure defined as the limit reached by the system throughput as the offered load increases, and represents the maximum load that the system can carry *instable conditions*. It is well known that several random access schemes exhibit an unstable behavior. In particular, as the offered load increases, the throughput grows up to a maximum value, referred to as “maximum throughput.” However, further increases of the offered load lead to an eventually significant decrease in the system throughput. This results in the practical impossibility to operate the random access scheme at its maximum throughput for a “long” period of time, and thus in the practical meaning saturation throughput is considered.

4.2 Packet Transmission Probability

Consider a fixed number of contending stations. In saturation conditions, each station has immediately a packet available for transmission, after the completion of each successful transmission[25]. Moreover, being all packets “consecutive,” each packet needs to wait for a random back off time before transmitting.

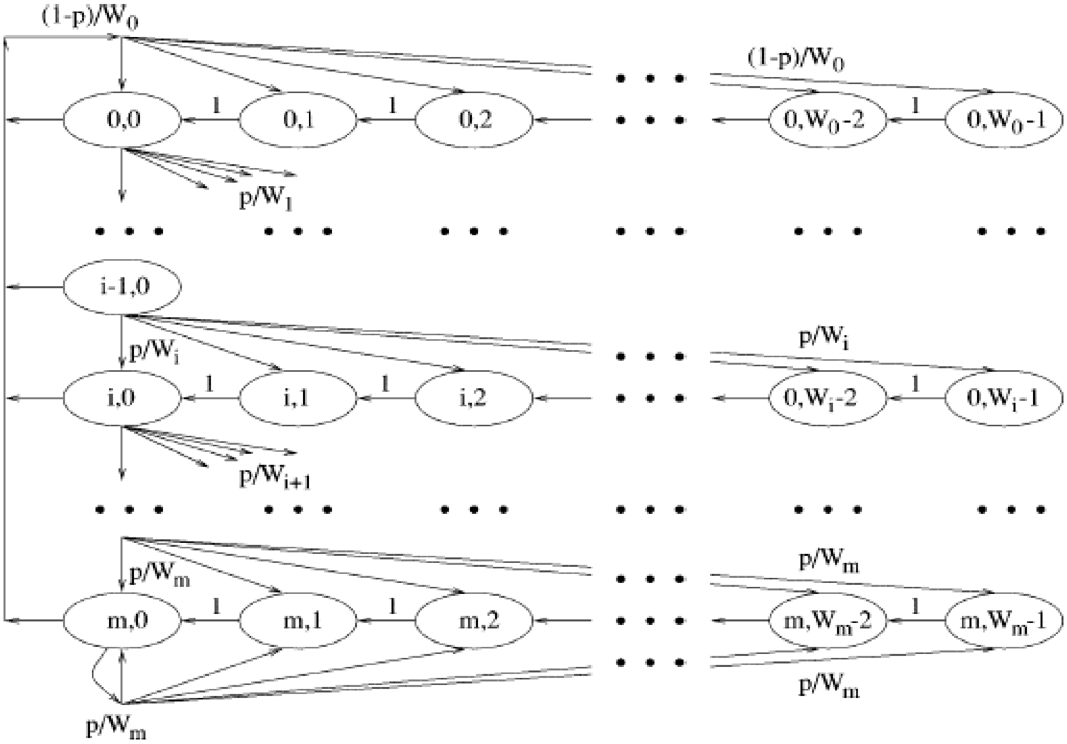

Figure 4.1 Markov Chain model for the back off window size.

Let b(t) be the stochastic process representing the back off time counter for a given station. A discrete and integer time scale is adopted: t and t+1 correspond to the beginning of two consecutive slot times, and the back off time counter of each station decrements at the beginning of each slot time. The back off time decrement is stopped when the channel is sensed busy, and thus the time interval between two consecutive slot time beginnings may be much longer than the slot time size σ , as it may include a packet transmission. In what follows, unless ambiguity occurs, with the term slot time we will refer to either the (constant) value σ, or the (variable) time interval between two consecutive back off time counter decrements.

Since the value of the back off counter of each station depends also on its transmission history (e.g., how many retransmission the head-of-line packet has suffered), the stochastic process is non-Markovian. However, define for convenience W = CW_{min}. Let m , “maximum back off stage,” be the value such that $CW_{max} = 2^m W$, and let us adopt the notation $W_i = 2^i W$, where $i \in (0, m)$ is called “back off stage.” Let s(t) be the stochastic process representing the back off stage (0,…..,m)of the station at time t .

The key approximation in our model is that, at each transmission attempt, and regardless of the number of retransmissions suffered, each packet collides with constant and independent probability p . It is intuitive that this assumption results more accurate as long as W and n get larger. p will be referred to as *conditional collision probability*, meaning that this is the probability of a collision seen by a packet being transmitted on the channel.

Once independence is assumed, and is supposed to be a constant value, it is possible to model the bidimensional process{s(t) , b(t)} with the discrete-time Markov chain depicted in Fig. 4.1. In this Markov chain, the only non null one-step transition probabilities are

$$\begin{array}{lll} P\{i, k \mid i , k + 1\} = 1 & k \in (0, W_i - 2) & i \in (0, m) \\ P\{0, k \mid i , 0\} = (1 - p) / W_0 & k \in (0, W_0 - 1) & i \in (0, m) \\ P\{i, k \mid i - 1 , 0\} = p / W_i & k \in (0, W_i - 1) & i \in (1, m) \\ P\{m, k \mid m ,0\} = p / W_m & k \in (0, W_m - 1) & \end{array} \qquad (1)$$

The first equation in (1) accounts for the fact that, at the beginning of each slot time, the back off time is decremented. The second equation accounts for the fact that a new packet following a successful packet transmission starts with back off stage 0, and thus the back off is initially uniformly chosen in the range(0, W_0 - 1) . The other cases model the system after an unsuccessful transmission. In particular, as considered in the third equation of (1), when an unsuccessful transmission occurs at back off stage i - 1 , the back off stage increases, and the new initial back off value is uniformly chosen in the range (0, W_i). Finally, the fourth case models the fact that once the back off stage reaches the value m , it is not increased in subsequent packet transmissions.

Let $b_{i,k} = \lim_{t \to \infty} P\{s(t) = i, b(t) = k\}$, $i \in (0,m)$, $k \in (0, W_i - 1)$ be the stationary distribution of the chain. It can be shown that it is easy to obtain a closed-form solution for this Markov chain. First, note that

$$b_{i-1,0} . p = b_{i,0} \rightarrow b_{i,0} = p^i b_{0,0} \qquad 0 < i < m$$

$$b_{m-1,0} . p = (1 - p)b_{m,0} \rightarrow b_{m,0} = (p^m b_{0,0}) / (1 - p)$$

(2)

Owing to the chain regularities, for each $k \in (1, W_i - 1)$, it is

$$b_{i,k} = \frac{Wi - k}{Wi} . \begin{cases} (1-p)\sum_{j=0}^{m} b_{j,0} & i = 0 \\ p.b_{i-1,0} & 0 < i < m \\ p.(b_{m-1,0} + b_{m,0}) & i = m \end{cases}$$

(3)

By means of relations (2), and making use of the fact that $\sum_{i=0}^{m} bi,o = {}^{bo,o}\!\big/_{(1-p)}$, (3) can be rewritten as

$$b_{i,k} = \frac{Wi-k}{Wi} \quad b_{i,0} \qquad i \in (0,m), \qquad k \in (0, W_i - 1) \tag{4}$$

Thus, by relations (2) and (4), all the values $b_{i,k}$ are expressed as functions of the value $b_{0,0}$ and of the conditional collision probability p. $b_{0,0}$ is finally determined by imposing the normalization condition, that simplifies as follows:

$$1 = \sum_{i=o}^{m}\sum_{k=0}^{Wi-1} b_{i,k} = \sum_{i=0}^{m} b_{i,0} \sum_{k=0}^{Wi-1}\frac{Wi-k}{Wi} = \sum_{i=0}^{m} b_{i,0} \frac{Wi+1}{2}$$

$$= \frac{bo,o}{2}\left[W\left(\sum_{i=0}^{m-1}(2p)^{i} + \frac{(2p)^{m}}{1-p}\right) + \frac{1}{1-p}\right] \tag{5}$$

from which

$$b_{i,k} = \frac{2(1-2p)(1-p)}{(1-2p)(W+1) + pW\left(1-(2p)^{m}\right)} \tag{6}$$

The probability τ that a station transmits in a randomly chosen slot time is expressed now. As any transmission occurs when the back off time counter is equal to zero, regardless of the back off stage, it is

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{b_{0,0}}{1-p} = \frac{2(1-2p)}{(1-2p)(W+1) + pW\left(1-(2p)^{m}\right)} \tag{7}$$

As a side note, it is interesting to highlight that, when m = 0, i.e., no exponential back off is considered, the probability τ results to be independent of p, and (7) becomes the much simpler one independently found in [8] for the constant back off window problem

$$\tau = \frac{2}{W+1} \qquad (8)$$

However, in general, τ depends on the conditional collision probability p ,which is still unknown. To find the value of p it is sufficient to note that the probability p that a transmitted packet encounters a collision, is the probability that, in a time slot, at least one of the n-1 remaining stations transmit. The fundamental independence assumption given above implies that each transmission "sees" the system in the same state, i.e., in steady state. At steady state, each remaining station transmits a packet with probability τ . This yields

$$p = 1 - (1 - \tau)^{n-1} \qquad (9)$$

Equations (7) and (9) represent a nonlinear system in the two unknowns τ and p , which can be solved using numerical techniques. It is easy to prove that this system has a unique solution. In fact, inverting (9), we obtain τ *(p) = 1 - $(1-p)^{\frac{1}{(n-1)}}$. This is a continuous and monotone increasing function in the range p $\in$ (0,1) , that starts from τ *(0) = 0 and grows up to τ *(1) = 1. Equation τ (p) defined by (7) is also continuous in the range p $\in$ (0,1) : continuity in correspondence of the critical value p =1/2 is simply proven by noting that τ (p) can be alternatively written as

$$\tau(p) = \frac{2}{1+W+pW\sum_{i=0}^{m-1}(2p)^i} \qquad (10)$$

and, therefore, τ (1/2) = 2 / (1+W+mW/2). Moreover, τ (p) is trivially shown to be a monotone decreasing function that starts from τ (0) = 2/(W+1) and reduces up to τ (1) = 2/(1+2^{m}W). Uniqueness of the solution is now proven noting that τ (0) > τ *(0) and τ (1) < τ *(1).

4.3 Throughput

Let S be the normalized system throughput, defined as the fraction of time the channel is used to successfully transmit payload bits. Let P_{tr} be the probability that there is at least one transmission in the considered slot time. Since n stations contend on the channel, and each transmits with probability τ

$$P_{tr} = 1- (1- \tau)^n \tag{11}$$

The probability P_s that a transmission occurring on the channel is successful is given by the probability that exactly one station transmits on the channel, conditioned on the fact that at least one station transmits, i.e.,

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n} \tag{12}$$

We are now able to express S as the ratio

$$S = \frac{E[payload \inf ormation\, transmitted\ in\ a\ slot\ time\ ]}{E[length\ of\ \ a\ slot\ time]} \tag{13}$$

Being E[P] the average packet payload size, the average amount of payload information successfully transmitted in a slot time is P_{tr} P_s E[P] , since a successful transmission occurs in a slot time with probability P_{tr} P_s . The average length of a slot time is readily obtained considering that, with probability 1- P_{tr} ,the slot time is empty; with probability P_{tr} P_s it contains a successful transmission, and with probability P_{tr} (1- P_s) it contains a collision. Hence, (13) becomes

$$S = \frac{P_s P_{tr} E[P]}{(1-P_{tr})\sigma + P_{tr} P_s T_s + P_{tr}(1-P_s)T_c} \tag{14}$$

Here, T_s is the average time the channel is sensed busy (i.e., the slot time lasts) because of a successful transmission, and T_c is the average time the channel is sensed busy by each station during a collision. σ is the duration of an empty slot time.

Consider a system in which each packet is transmitted by means of the RTS/CTS Access mechanism.

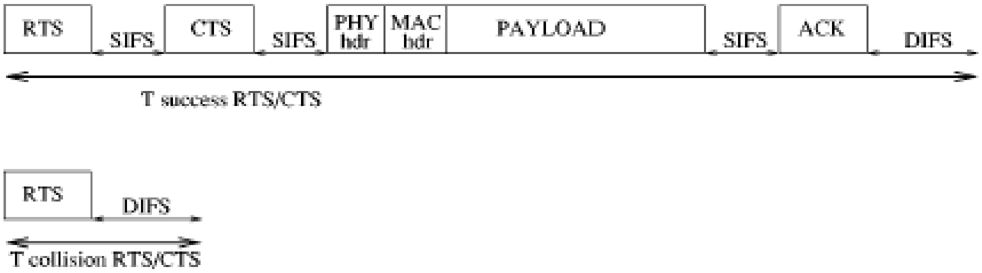

Figure 4.2 T_s and T_c for RTS/CTS mechanism

As, in such a case, collision can occur only on RTS frames, it is

$$T_s = RTS + SIFS + \delta + CTS + SIFS + \delta + H + E[P] + SIFS + \delta + ACK + DIFS + \delta$$

$$T_c = RTS + DIFS + \delta \qquad (15)$$

where, $H = PHY_{hdr} + MAC_{hdr}$ is the packet header, and δ be the propagation delay.

and the throughput expression depends on the packet size distribution only through its mean.

4.4 Maximum Saturation Throughput

Now, saturation throughput can be achieved easily. Rearranging (14) we get,

$$S = \frac{E[P]}{T_s - T_c + \frac{\sigma(1-P_{tr})/P_{tr} + T_c}{P_s}} \qquad (16)$$

As T$_s$, T$_c$, E[P] and σ , are constants, the throughput S is maximized when the following quantity is maximized:

$$\frac{P_s}{(1-P_{tr})/P_{tr} + T_c/\sigma} = \frac{n\tau(1-\tau)^{n-1}}{T_c^* - (1-\tau)^n (T_c^* - 1)} \qquad (17)$$

where $T_c^* = T_c/\sigma$ is the duration of a collision measured in slot time units σ . Taking the derivative of (17) with respect to τ , and imposing it equal to 0,after some simplifications, the following equation is obtained :

$$(1-\tau)^n - T_c^* \{n\tau - [1 - (1-\tau)^n]\} = 0 \qquad (18)$$

Under the condition τ << 1

$$(1-\tau)^n \approx 1 - n\tau + \frac{n(n-1)}{2}\tau^2 \qquad (19)$$

holds, and yields the following approximate solution:

$$\tau = \frac{\sqrt{[n + 2(n-1)(T_c^* - 1)]/n} - 1}{(n-1)(T_c^* - 1)} \approx \frac{1}{n\sqrt{T_c^*/2}} \qquad (20)$$

Equation (19) and its approximate solution (20) are of fundamental theoretical importance. In fact, they allow to explicitly compute the optimal transmission probability τ that each station should adopt in order to achieve maximum throughput performance within a considered network scenario (i.e., number of stations). In other words, they show that (within a PHY and an access mechanism, which determine the constant value) maximum performance can be, in principle, achieved for every network scenario, through a suitable sizing of the transmission probability τ in relation to the network size.

However, (7) and (9) show that depends only on the network size and on the system parameters m and W. As n is not a directly controllable variable, the only way to achieve optimal performance is to employ adaptive techniques to tune the values m and W (and consequently) on the basis of the estimated value of n .

Moreover, the maximum throughput is practically independent of the number of stations in the wireless network. This is easily justified by noting that the throughput formula can be approximated as follows. Let K = $\sqrt{T_c^*/2}$, and let us use the approximate solution τ = 1/(nK). For n sufficiently large

$$P_{tr} = 1 - (1-\tau)^n = 1 - \left(1 - \frac{1}{nK}\right)^n \approx 1 - e^{\frac{-1}{K}} \qquad (21)$$

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} \approx \frac{n}{(nK-1)(e^{\frac{1}{K}}-1)} \approx \frac{1}{K(e^{1/K}-1)} \qquad (22)$$

The maximum achievable throughput S_{max} can thus be approximated as

$$S_{max} = \frac{E[P]}{T_s + \sigma K + T_c(K(e^{\frac{1}{K}}-1)-1)} \qquad (23)$$

which results to be independent of n .

CHAPTER-5

SIMULATION AND RESULTS

SIMULATION AND RESULTS

5.1 NS - Network Simulator

Ns[7,9] is one of the most commonly used simulators today, in the networking research community, mostly because of its open-source. The simulator is a discrete event simulator originally developed at LBL (Lawrence Berkeley Laboratory) at University of Berkeley, within the VINT (Virtual InterNetwork Testbed) project.

Berkeley released the initial code that made wireless network simulations possible in *ns*. That code provided some support to model wireless LANs, but was fairly limited. As a result of the Monarch project at Carnegie Mellon University[11] the simulator was extended with support for *node mobility*, *a realistic physical layer*, *radio network interfaces* and an implementation of this work was presented as a part of a larger study of performance for different ad-hoc routing protocols[10]. It was this contribution that made it possible to perform real wireless simulations with *ns*.

NS-2 has many and expanding uses including:

- To evaluate the performance of existing network protocols.
- To evaluate new network protocols before use.
- To run large scale experiments not possible in real experiments.
- To simulate a variety of ip networks.

5.2 Design of NS-2

NS is an Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network set-up (plumbing) module libraries (actually, plumbing modules are implemented as member functions of the base simulator object).

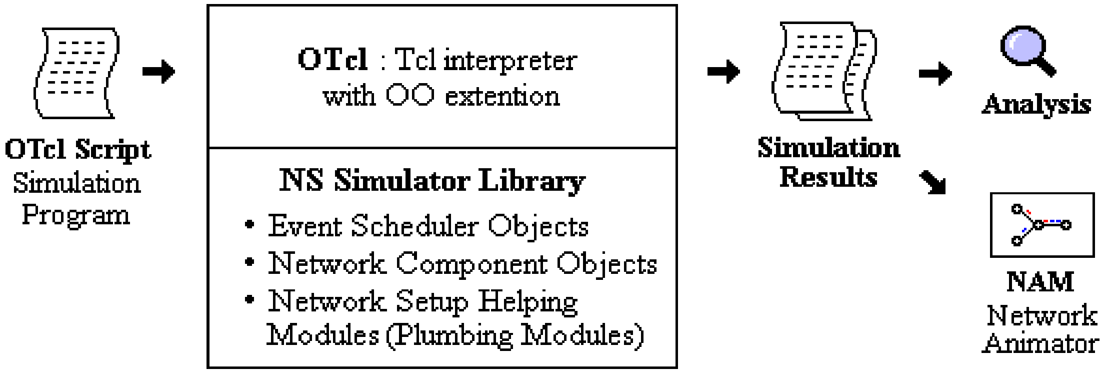

Figure 5.1 User's view of NS-2

To use NS-2, a user programs in the OTcl script language. An OTcl script will do the following:

- Initiates an event scheduler.
- Sets up the network topology using the network objects.
- Tells traffic sources when to start/stop transmitting packets through the event scheduler.

The term "plumbing" is used for a network setup, because setting up a network is plumbing possible data paths among network objects by setting the "neighbor" pointer of an object to the address of an appropriate object. When a user wants to make a new network object, he or she can easily make an object either by writing a new object or by making a compound object from the object library, and plumb the data path through the object. The power of NS comes from this plumbing. A user can add OTcl modules to NS-2 by writing a new object class in OTcl. These then have to be compiled together with the original source code.

Another major component of NS beside network objects is the event scheduler. An event in NS is a packet ID that is unique for a packet with scheduled time and the pointer to an object that handles the event. In NS, an event scheduler keeps track of simulation time and fires all the events in the event queue scheduled for the current time by invoking appropriate network components, which usually are the ones who issued the events, and let them do the appropriate action associated with packet pointed by the event. Network components communicate with one another passing packets, however this does not consume actual simulation time. All the network components that need to spend some simulation time handling a packet (i.e. need a delay) use the event scheduler by issuing an event for the packet and waiting for the event to be fired to itself before doing further action handling the packet. For example, a network switch component that simulates a switch with 20 microseconds of switching delay issues an event for a packet to be switched to the scheduler as an event 20 microseconds later. The scheduler after 20 microseconds dequeues the event and fires it to the switch component, which then passes the packet to an appropriate output link component.

Another use of an event scheduler is timer. For example, TCP needs a timer to keep track of a packet transmission time out for retransmission (transmission of a packet with the same TCP packet number but different NS packet ID). Timers use event schedulers in a similar manner that delay does. The only difference is that timer measures a time value associated with a packet and does an appropriate action related to that packet after a certain time goes by, and does not simulate a delay.

Depending on the user’s purpose for an OTcl simulation script, simulation results are stored as trace files, which can be loaded for analysis by an external application:

- A NAM trace file (file.nam) for use with the Network Animator tool.

- A Trace file (file.tr) for use with Xgraph and TraceGraph.

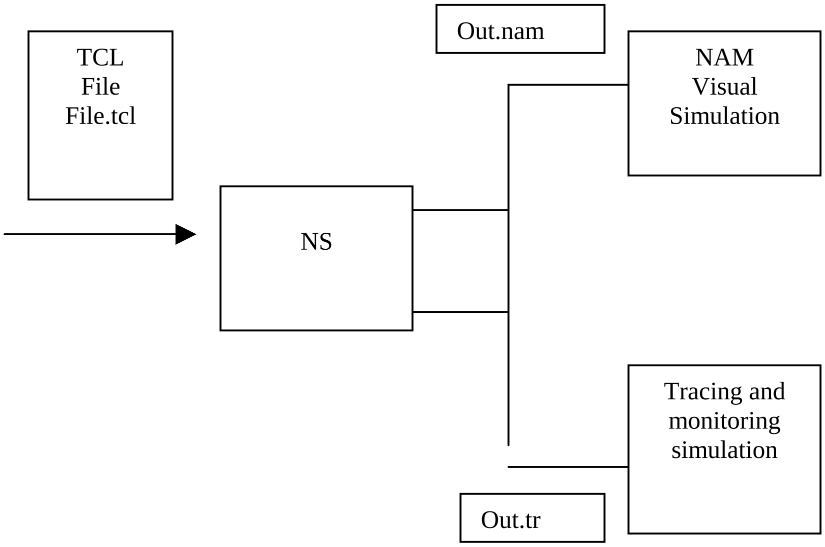

Figure 5.2 Flow of events for a Tcl file run in NS

5.3 C++ / OTcl Linkage

NS-2 is written in C++ with OTcl interpreter as a front end. For efficiency reason, NS separates the data path implementation from control path implementations.

Languages[19] used with NS-2:

- Split-language programming is used

  1. Scripting language (Tcl- Tool Command Language and pronounced ‘tickle’).

  2. System Programming Language (C/ C++).

- Ns is a Tcl interpreter to run Tcl Scripts.

- By using C++/OTcl, the network simulator is completely Object-oriented.

In terms of lines of source code, NS-2 was written with 100k of C++ code, 70k lines of Tcl code and 20k of documentation.

5.3.1 The Tcl Interpreter

TclCL is the language used to provide a linkage between C++ and OTcl[9]. Toolkit command language (Tcl/OTcl) scripts are written to set up network topologies. TclCL provides linkage for class hierarchy, object instantiation, variable binding and command dispatching. OTcl is used for periodic or triggered events.

The event scheduler and basic network component objects are written and compiled with C++.

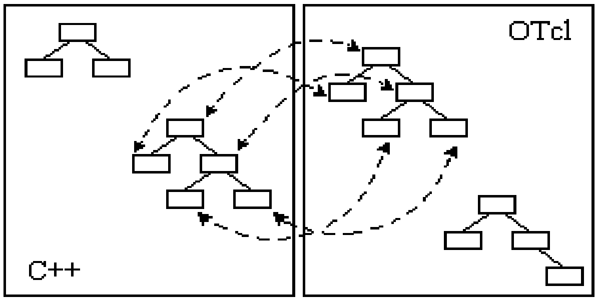

Figure5.3 C++ and OTcl

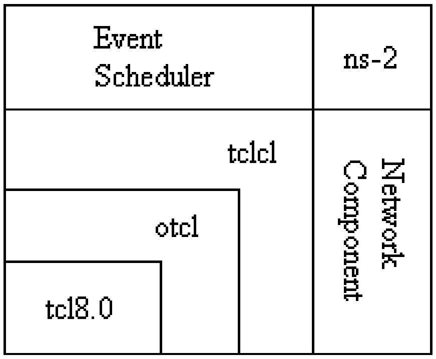

Figure 5.4 Architectural view of NS

These compiled objects are made available to the OTcl interpreter through an OTcl linkage that creates a matching OTcl object for each of the C++ objects and makes the control functions and the configurable variables specified by the C++ object act as member functions and member variables of the corresponding OTcl object. It is also possible to add member functions and variables to a C++ linked OTcl object.

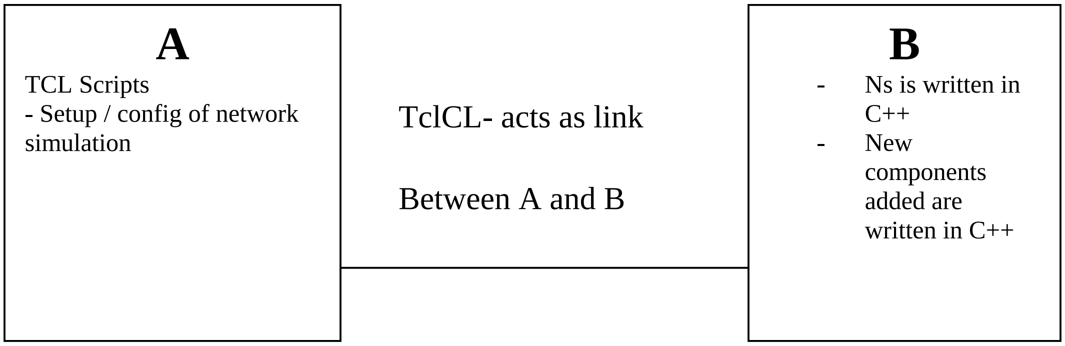
A
TCL Scripts
- Setup / config of network simulation
TclCL- acts as link
Between A and B
B
- Ns is written in C++
- New components added are written in C++

Figure 5.5 TclCL provides the linkage between C++ and OTcl

5.4 Characteristics of NS-2

NS-2 implements the following features:

- Router queue management techniques Drop Tail, RED, CBQ.
- Multicasting.
- Simulation of wireless networks

  i. Developed by Sun Microsystems +UC Berkeley.

  ii. Terrestrial (cellular, adhoc, GPRS, WLAN, BLUETOOTH) satellite.

  iii. IEEE 802.11 can be simulated, Mobile-IP, and adhoc protocols such as DSR, TORA, DSDV and AODV.

- Traffic source behavior - www, CBR, VBR.
- Transport agents - UDP/TCP.
- Routing.
- Packet flow.
- Network Topology.
- Applications - Telnet, FTP, Ping.
- Tracing packets on all links / specific links.

5.5 Node Architecture in NS

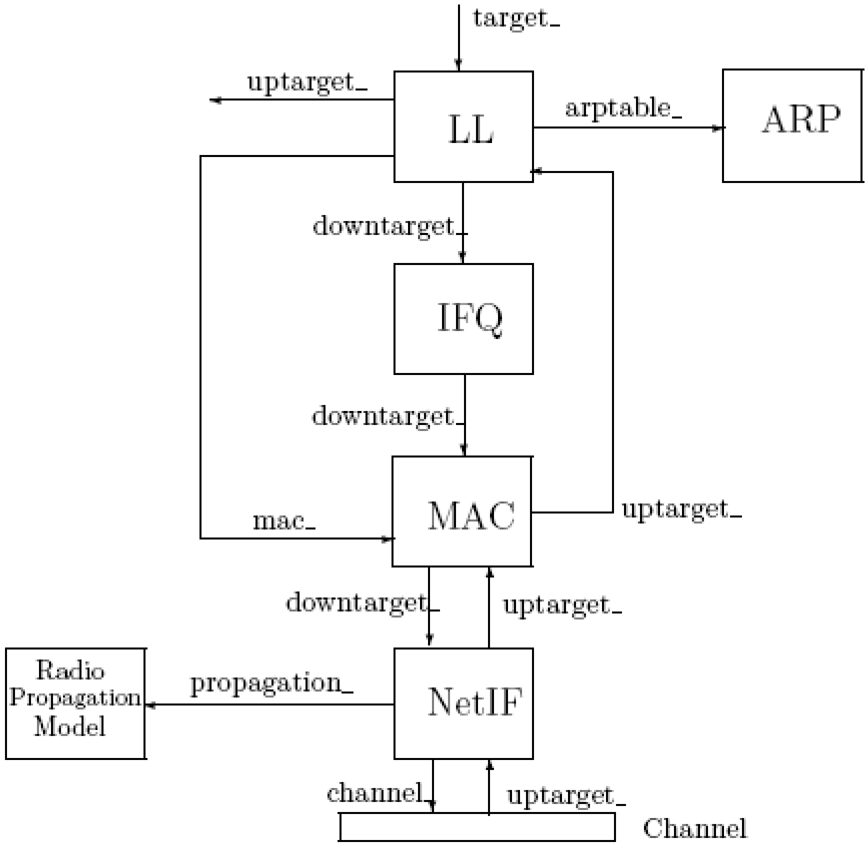

Figure 5.6 Architecture of NS Node

The Figure 5.6 shows the architecture of a NS mobile node below the Link Layer. The outgoing packets after being processed by the routing layer are handed over to the link layer (LL object) through the target interface. The link layer is connected to the ARP module (ARP object) through the interface arptable. After processing the packet and resolving MAC address through ARP, the link layer hands over the packet to the Interface Queue (IFQ object) through the down target interface. The interface queue is connected to the MAC layer (MAC object) through interface down target, and the packets are pulled off by the MAC when required. The link layer also contains a reference to MAC through the Mac interface. After the MAC has acquired the medium, it sends it to the physical layer (NetIF object) through the interface down target. The physical layer sends this packet over the channel (Channel object) through the channel interface and the packet reaches to the physical layer of the station at the other end of the link. The outgoing packet is handed over by Channel to the physical layer through the up target interface. The physical layer determines the received power levels through propagation model (propagation) and processes the packet. The packet after processing is handed over to the MAC layer through up target interface. The MAC layer processes the packet and handles over to the link layer through the interface up target.

5.6 Simulation Setup

This describes the simulation of 802.11 DCF. The simulations are done by using the public domain simulator NS-2. The following assumption are made in the simulation:

- The effect of propagation delay on the model are neglected. This is fairly realistic considering the fact the area in which stations are present is limited to 500mx500m and inter-node distance is of the order of few hundred feet.

- The effect of channel errors is ignored in the simulations.

- No stations are operating in power save mode.

A finite buffer is maintained at each station. If the buffer fills, the newly generated packets are simply dropped. All the packets in the DCF mode are sent using RTS/CTS exchange. FTP traffic is employed here at application layer. At transport layer , Transmission control protocol(TCP) is used. The routing protocol used is DSDV[23] at routing layer. The reason for choosing DSDV protocol for routing is that it provides constant routing overhead in case of static and less mobile networks. Here Saturation Throughput is calculated as each station has always a packet to transmit. This is a fundamental performance figure defined as the limit reached by the system throughput as the offered load increases, and represents the maximum load that the system could carry in stable conditions.

5.7 Different scenarios and results

For the simulation, various Tcl scripts are written. A sample script has been shown in Appendix A. The results are obtained in two files namely Out.tr and Out.nam. The snapshots of both are shown in Appendix B and C. Nam file is used for visual animation only and all the data for the graphs are interpreted from the out.tr file. Data is interpreted i.e. saturation throughput is calculated by using script written in perl (Appendix D). Then, the numerical values obtained are plotted using Mat lab[15]. Now, the various scenarios and their results are described.

Firstly, the saturation output of varying number of stations keeping the window size fixed is shown. The window size has been fixed to 32 and 64 for two different scenarios. Their throughputs are shown in the following graph for the five different network sizes i.e. number of stations n equal to 5, 10,15,20 and 25:

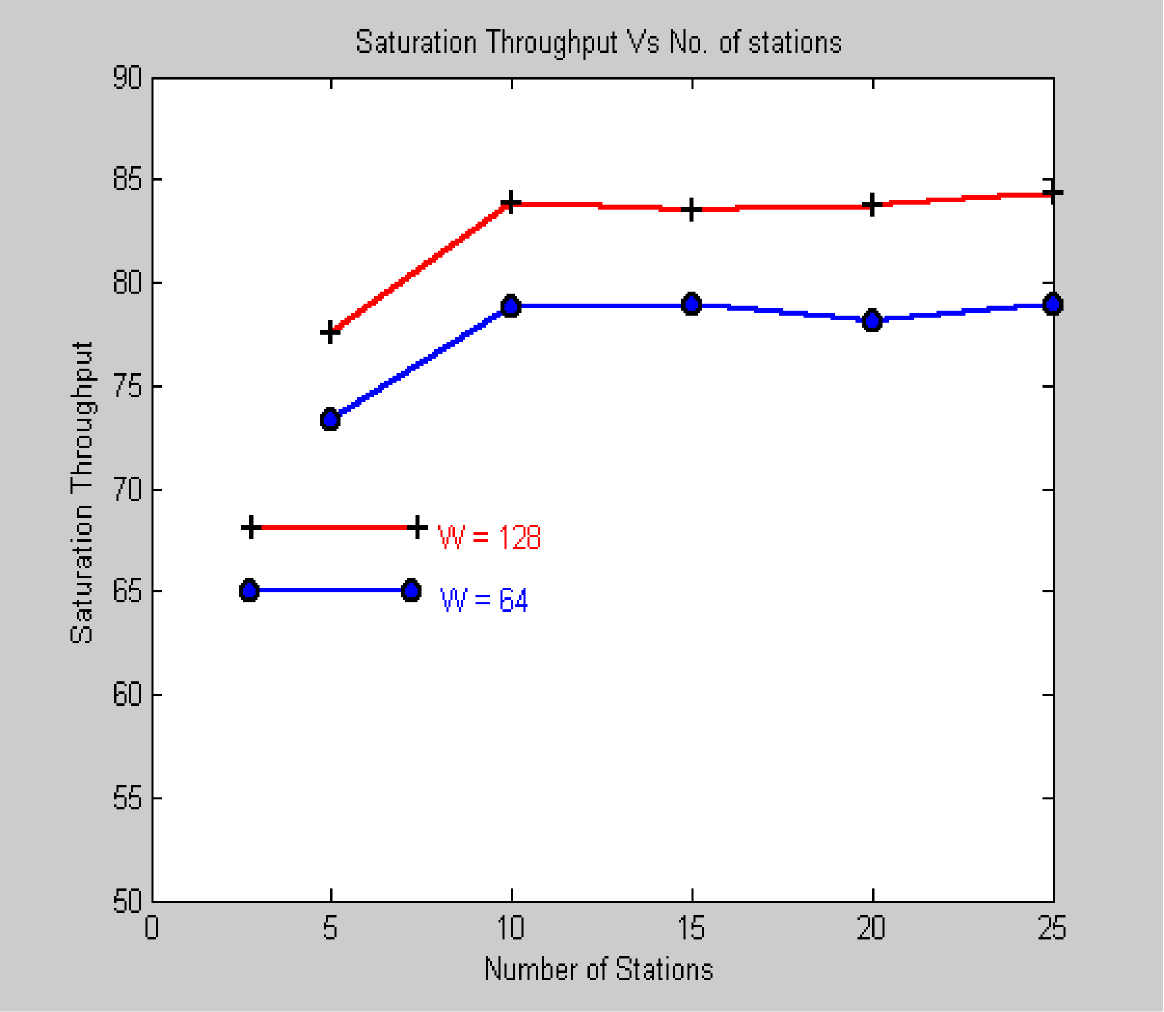

Figure 5.7 Saturation throughput Vs No. of stations

The higher throughput is obtained when the window size is more as there will be less collisions.

To show the dependency of the throughput from the initial contention window size W, the graph is shown below, the saturation throughput versus the value W. The graph reports two different network sizes, i.e. number of stations n equal to 5 and 20.

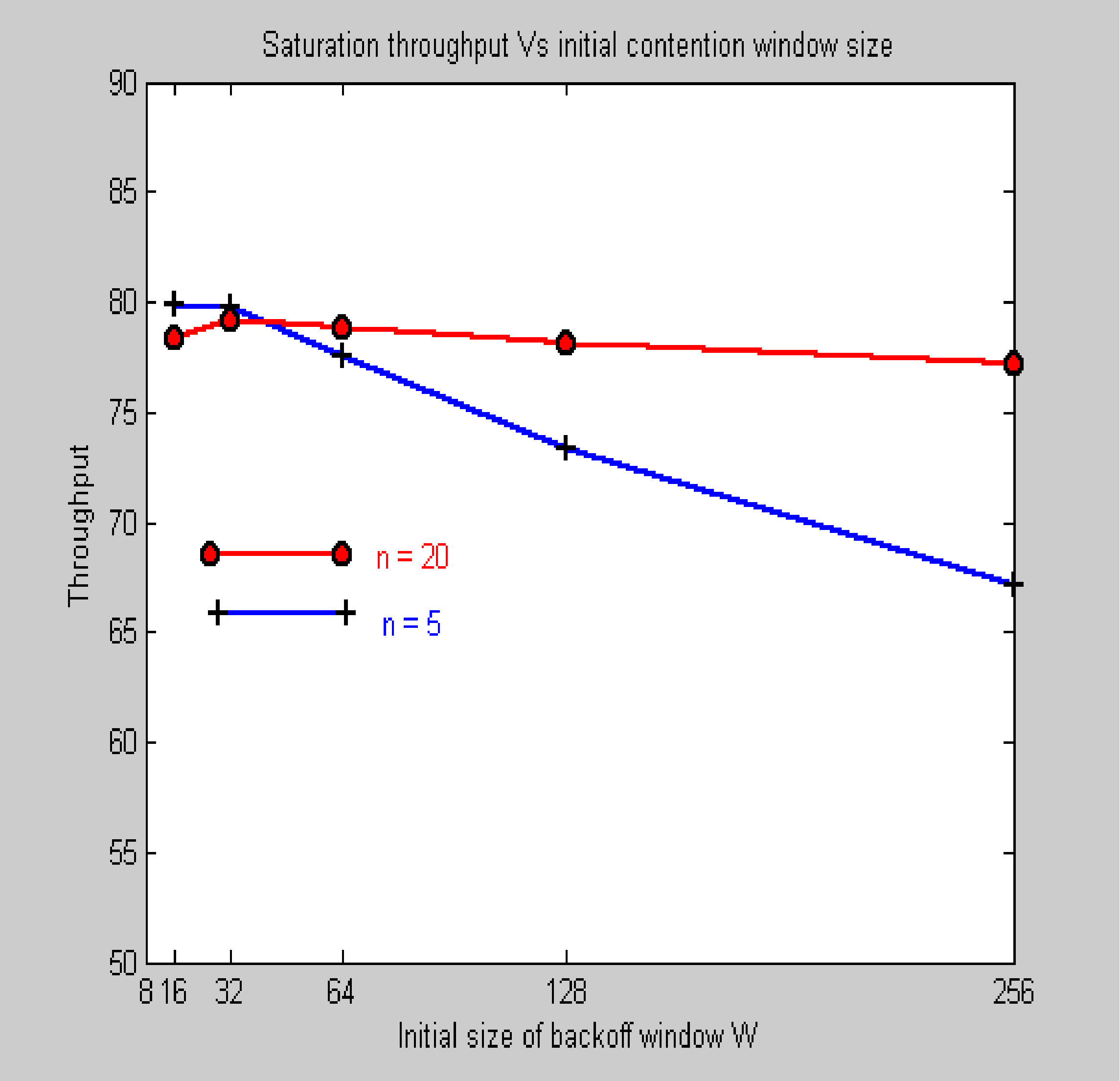

Figure 5.8 Saturation throughput versus initial contention window size

Fig.5.8 shows that the throughput of the RTS/CTS mechanism highly depends on the window size and the optimal value of W depends on the number of terminals in the network. For example, a high value of W (e.g., 256) gives good throughput performance in the case of 20 contending stations, while it drastically penalizes the throughput in the case of small number (e.g., 5) of contending stations.

CHAPTER-6

QUALITY OF SERVICE ENHANCEMENT FOR 802.11e

QUALITY OF SERVICE ENHANCEMENT FOR 802.11e

6.1 Introduction

People are now requiring to receive high-speed video, audio, voice and Web services even when they are moving in offices or traveling around campuses. However, multimedia applications require some quality of service (QoS)[26] support such as guaranteed bandwidth, delay, and jitter and error rate. Guaranteeing those QoS requirements in 802.11 WLAN is very challenging due to the QoS unaware functions of its medium access control (MAC) layer and the noisy and variable physical (PHY) layer characteristics.

There are several ways to characterize QoS in WLAN such as *parameterized* or *prioritized* QoS. Generally, QoS is the ability of a network element (e.g. an application, a host or a router) to provide some levels of assurance for consistent network data delivery.

Parameterized QoS is a strict QoS requirement that is expressed in terms of quantitative values, such as data rate, delay bound, and jitter bound. In a Traffic Specification (TSPEC), these values are expected to be met within the MAC data service in the transfer of data frames between peer stations (STAs).

Prioritized QoS is expressed in terms of relative delivery priority, which is to be used within the MAC data service in the transfer of data frames between peer STAs. In prioritized QoS scheme, the values of QoS parameters such as data rate, delay bound, and jitter bound, may vary in the transfer of data frames, without the need to reserve the required resources by negotiating the TSPEC between the STA and the AP.

6.2 QoS limitations of DCF

According to the definitions of QoS above, the QoS limitations of IEEE 802.11 DCF is described. DCF can only support best-effort services, not any QoS guarantees. Typically, time-bounded services such as Voice over IP, or audio/video conferencing require specified bandwidth, delay and jitter, but can tolerate some losses. However, in DCF mode, all the STAs in one BSS compete for the resources and channel with the same priorities. There is no differentiation mechanism to guarantee bandwidth, packet delay and jitter for high-priority STAs or multimedia flows. There is no way to guarantee the QoS requirements for high-priority audio and video traffic unless admission control is used.

6.3 802.11e MAC Enhancements

The current MAC has no means of differentiating traffic streams or sources. All data is treated equally. As a result, no consideration can be made for the service requirements of the traffic on the channel. For example, low priority bursty traffic may choke out a long running critical video feed thereby destroying the user’s experience.

The two new MAC modes, EDCF[26] and HCF[20], being defined under 802.11e, support up to eight priority traffic classes.

6.3.1 Enhanced Distributed Coordination Function (EDCF)

EDCF uses different mechanisms to provide service differentiation. The minimum contention window for the back off is different for different priority classes. This will in turn reflect on the higher priority classes getting more transmission time than lower priority classes. Further different inter frame spaces can be used for different priority classes.

The EDCF is operative only during the *Contention Period* (CP). The various streams are classified into *Traffic Categories* (TCs). During the CP, each TC within the stations contends for a T*ransmission Opportunity* (TXOP) independently. Each TC starts a back off after detecting channel to be idle for a time interval equal to *Arbitration Inter Frame Space* (AIFS). The value for AIFS is dependent on the traffic category the traffic belongs to. The back off is set to a counter, which is a random number from the interval [1, CW+1]. As in DIFS contention for each collision of the frames the CW value is increased. The initial value of CW is set to CWmin. The value of CWmin is also dependent on the traffic category of the stream.

After a collision is detected the CW is increased as follows

newCW [TC] = (oldCW [TC] +1) * PF [TC] . 1

Here PF is the Persistence Factor, which is also a traffic category dependent parameter. PF determines the degree of increase of the Contention Window when collisions occur. Higher priority traffic will have lesser PF value than lower priority traffic PF value. Thus when collisions occur the higher priority traffic flows. CW value will increase by a lesser value than lower priority traffic's CW value.

6.3.2 Hybrid Coordination Function (HCF)

The polling scheme of PCF is extended in 802.11e by using the Hybrid Coordination Function (HCF). In this scheme, there is a hybrid coordinator (HC) usually co-located with the access point. The HC may allocate TXOPs to itself to initiate frame transmission after waiting for a time equal to PIFS, which is shorter than DIFS and any AIFS. Thus the HC gets priority over other nodes to transmit frames.

The HCF is operative during both the CP and CFP durations. During the CP each station gets its TXOP either when the medium is determined to be available under the EDCF rules or when the station receives a QoS CF-Poll frame from the HC. During the CFP, the starting time and maximum duration of each TXOP is specified by the HC using CF-Poll frames. As the name (contention free period) denotes, stations cannot contend among themselves for TXOP during the CFP. The CFP ends either at the time specified in the beacon frame or by a CFEnd frame sent by the HC.

The 802.11e also uses another mechanism by which the stations send update information to the HC. This includes which stations need to be polled, polling time and duration of transmissions. The mechanism used is called controlled contention in which the HC allocates a number of controlled contention opportunities separated by SIFS. This is done so that stations with high priority traffic need not contend with other EDCF traffic for transmitting the request information. The HC also sends out a filtering mask containing the TCs in which resource requests may be placed. Each station chooses one opportunity interval and transmits a resource request frame containing the requested TC and TXOP[14] duration. The HC also sends out an acknowledgment control frame so that requesting stations can detect collisions during controlled contention.

CHAPTER-7

CONCLUSIONS AND FUTURE DIRECTIONS

CONCLUSIONS AND FUTURE DIRECTIONS

7.1 Conclusions

This dissertation aimed at the analysis of the results obtained by analytical model and simulation runs but the exact value of some parameters used in the analytical model of IEEE 802.11 DCF protocol are not known in the simulation environment. So their results could not be tallied. A detailed study of analytical model and performance analysis of IEEE 802.11 Wireless LAN is done using ns-2 simulator.

The IEEE 802.11 DCF protocol is known to exhibit some form of instability. It is noted that the throughput increases with the increasing offered load initially and attains a maximum value called ‘ maximum throughput’ but as the load increases heavily the throughput decreases and it attains an almost constant value called ‘saturation throughput’. So, instead of maximum throughput, saturation throughput is considered in this dissertation.

From the simulations, it is clear that the increase in the window size increases the throughput for different network sizes. Secondly, it is also shown that a high value of window gives good throughput performance for large network size, but it penalizes the throughput in case of small network sizes.

Also, the discussed DCF protocol does not support the traffic streams with different priorities, a brief description of EDCF mechanism is given which incorporates this feature.

7.2 Future Directions

The future scope of this work is summarized below:

- Results obtained from analytical model and simulation model can be tallied for the correctness of analytical model.

- An analytical model can be developed with variable collision probability p as a function of m where m is the collision stage.

- An Enhanced model can also be developed where instead of one access category, four access categories can be incorporated (as in 802.11e). In the same vein, EDCF protocol can be incorporated in ns-2.

- The aim of this dissertation is focused on the DCF. So, PCF can also be studied and simulated.

REFERENCES

[1] IEEE 802.11 WG, Reference number ISO/IEC 8802-11:1999(E) IEEE Std 802.11, 1999 edition. International Standard [for] Information Technology - Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific Requirements - Part 11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999.

[2] L. Kleinrock and F. Tobagi, "Packet switching in radio channels, Part II—The hidden terminal problem in carrier sense multiple access and the busy tone solution," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1417–1433, Dec. 1975.

[3] The Institute of Electrical and Electronics Engineers, Inc. *IEEE Std 802.11 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, 1999 edition.

[4] Hiperlan/2 Global Forum (web page).
URL: *http://www.hiperlan2.com/web/*

[5] HomeRF Wireless LAN (web page).
URL: *http://www.homerf.org*

[6] K. C. Huang and K. C. Chen, "Interference analysis of nonpersistent CSMA with hidden terminals in multicell wireless data networks," in *Proc. IEEE PIMRC*, Toronto, Canada, Sept. 1995, pp. 907–911.

[7] S. McCanne and S. Floyd. ns network simulator version 2.1b6, August 2000.
URL: *http://www.isi.edu/nsnam/ns/*

[8] T. S. Ho and K. C. Chen, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LAN's," in *Proc. IEEE PIMRC*, Taipei, Taiwan, Oct. 1996, pp. 392–396.

[9] K. Fall and K. Varadhan. *The ns Manual*. VINT Project, UC-Berkeley and LBNL, 2000.

[10] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In

Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking. ACM, October 1998.

[11] CMU Monarch Project (web page).

URL: *http://www.monarch.cs.cmu.edu*

[12] D-J. Deng and R-S. Chang. A priority scheme for IEEE 802.11 DCF access method. *IEICE Transactions on Communications*, E82-B(1), January 1999.

[13] IEEE 802.11 TGe, "HCF Channel Access Rules", TR- 02/015r1, January 2002.

[14] IEEE 802.11 TGe, "HCF Ad Hoc Group Recommendation – Normative Text to EDCF Access Category", TR-02/241r0, March 2002.

[15] Stephen J.Chapman, *Matlab Programming for Engineers: Second ed.* Thomson Books/Cole.

[16] **URL:** *http://www.pearl.com*

[17] IEEE 802.11 WG, Draft Supplement to STANDARD FOR Telecommunications and Information Exchange Between Systems-LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), IEEE 802.11e/Draft 4.2, Februray 2003.

[18] IEEE WG, 802.11a, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer in the 5Ghz Band, Sep. 1999.

[19] John Ousterhout. Scripting: Higher-level programming for the 21st century. *IEEE Computer*, 31(3):23–30, March 1998.

[20] IEEE 802.11 TGe, "Hybrid Coordination Function (HCF) – Proposed Updates to Normative Text of D0.1", TR-01/110r1, March 2001

[21] Andreas Kopsel, Jean-Pierre Ebert, and Adam Wolisz, A Performance Comparision of Point and Distributed Coordination Function of an IEEE 802.11 WLAN in the presence of Real-Time Requirements, Proc. of 7th Intl. Workshop on Mobile Multimedia Communications (MoMuC2000), October 23-26, 2002.

[22] Mangold S, Choi S, May P, Klein O, Hiertz G, and Stibor L. IEEE 802.11e wireless LAN for quality of service, *Proc. of European Wireless (EW2002)*, Florence, Italy, February 2002.

[23] Charles E. Perkins, Pravin Bhagwat, Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers, ACM SIGCOMM Computer Communication Review, v.24 n.4, p.234-244, Oct. 1994.

[24] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Nov. 1997. P802.11.

[25] Guiseppe Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function”, IEEE Journal on Selected Areas in Communications, March 2000

[26] IEEE 802.11 TGe, “EDCF Proposed Draft Text”, TR- 01/131r1, March 2001.

[27] Heegard, C. and Coffey, J. and Gummadi, S. and Murphy, P. A. and Provencio, R. and Rossin, E. J. and Schrum, S. and Shoemake, M. B., .High- Performance Wireless Ethernet., *IEEE Comm. Magazine*, vol. 39, no. 11, Nov. 2001.

[28] G. Bianchi, L. Fratta, and M. Oliveri, “Performance analysys of IEEE 802.11 CSMA/CA medium access control protocol,” in *Proc. IEEE PIMRC*, Taipei, Taiwan, Oct. 1996, pp. 407–411.

[29] K. Pahlavan and A. H. Levesque, “Wireless data communications,” *Proc. IEEE*, vol. 82, pp. 1398–1430, Sept. 1994.

[30] A. De Simone and S. Nanda, “Wireless data: Systems, standards, services,” *J. Wireless Networks*, vol. 1, no. 3, pp. 241–254, Feb. 1996.

[31] IEEE, Draft Supplement to STANDARD 802.11 1999 Edition, “Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)”, version 2.0

[32] H. S. Chhaya and S. Gupta, “Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol,” *Wireless Networks*, vol. 3, pp. 217–234, 1997

[33] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[34] Qiang Ni, Lamia, Thierry Turletti, “A survey of QOS Enhancements for IEEE 802.11 Wireless LAN”, Wireless comm. & mobile Computing, vol.4, pp. 547-566, 2004

[35] **URL:** http://www.cs.toronto.edu/~wbiao/ns2/traceformats/

APPENDIX A
TCL SCRIPT

A sample Tcl script has been shown below:

```
# Define options
# ============================================================

set val(chan)   Channel/WirelessChannel          ;# channel type
set val(prop)   Propagation/TwoRayGround         ;# radio-propagation model
set val(netif)  Phy/WirelessPhy                  ;# network interface type
set val(mac)    Mac/802_11                       ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue          ;# interface queue type
set val(ll)     LL                               ;# link layer type
set val(ant)    Antenna/OmniAntenna              ;# antenna model
set val(ifqlen)             50                   ;# max packet in ifq
set val(nn)                 15                   ;# number of mobilenodes
set val(adhocRouting)        DSDV                ;# routing protocol
set val(x)                  500
set val(y)                  500

#============================================================

# create simulator instance
set ns_   [new Simulator]
#ns_ use_newtrace
set tracefd  [open 15nod.tr w]
set namtrace [open 15nod.nam w]
```

```
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
# Create topography object
set topo   [new Topography]

# define topology
$topo load_flatgrid 500 500

# create God
create-god $val(nn)

$ns_ node-config -adhocRouting $val(adhocRouting) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -propType $val(prop) \
                 -phyType $val(netif) \
                 -channelType $val(chan)\
                 -topoInstance $topo \
                 -agentTrace OFF \
                 -routerTrace OFF \
                 -macTrace ON \
                -movementTrace OFF \

for {set i 0} {$i < $val(nn)} {incr i}
{
   set node_($i) [$ns_ node]
   $node_($i) random-motion 0
   $node_($i) set Z_ 0.0
}
   $node_(0) set X_ 2.0
   $node_(0) set Y_ 5.0
   $node_(1) set X_ 4.0
   $node_(1) set Y_ 7.0
   $node_(2) set X_ 7.0
   $node_(2) set Y_ 5.0
   $node_(3) set X_ 13.0
   $node_(3) set Y_ 18.0
   $node_(4) set X_ 8.0
   $node_(4) set Y_ 6.0
   $node_(5) set X_ 3.0
   $node_(5) set Y_ 9.0
   $node_(6) set X_ 6.0
   $node_(6) set Y_ 8.0
   $node_(7) set X_ 4.0
   $node_(7) set Y_ 6.0
   $node_(8) set X_ 4.0
   $node_(8) set Y_ 6.0
   $node_(9) set X_ 4.0
   $node_(9) set Y_ 6.0
   $node_(10) set X_ 4.0
   $node_(10) set Y_ 6.0
   $node_(11) set X_ 4.0
   $node_(11) set Y_ 6.0
   $node_(12) set X_ 4.0
   $node_(12) set Y_ 6.0
   $node_(13) set X_ 4.0
   $node_(13) set Y_ 6.0
   $node_(14) set X_ 4.0
   $node_(14) set Y_ 6.0
```

```
$ns_ at 10.0 "$node_(0) setdest 250.0 250.0 100.0"
$ns_ at 10.0 "$node_(1) setdest 250.0 10.0 100.0"
$ns_ at 10.0 "$node_(2) setdest 280.0 30.0 100.0"
$ns_ at 10.0 "$node_(3) setdest 300.0 50.0 100.0"
$ns_ at 10.0 "$node_(4) setdest 320.0 70.0 100.0"
$ns_ at 10.0 "$node_(5) setdest 340.0 90.0 100.0"
$ns_ at 10.0 "$node_(6) setdest 360.0 130.0 100.0"
$ns_ at 10.0 "$node_(7) setdest 360.0 160.0 100.0"
$ns_ at 10.0 "$node_(8) setdest 400.0 200.0 100.0"
$ns_ at 10.0 "$node_(9) setdest 370.0 230.0 100.0"
$ns_ at 10.0 "$node_(10) setdest 400.0 250.0 100.0"
$ns_ at 10.0 "$node_(11) setdest 300.0 280.0 100.0"
$ns_ at 10.0 "$node_(12) setdest 250.0 300.0 100.0"
$ns_ at 10.0 "$node_(13) setdest 220.0 340.0 100.0"
$ns_ at 10.0 "$node_(14) setdest 100.0 200.0 100.0"

set tcp1 [new Agent/TCP]
$tcp1 set class_ 2
set sink1 [new Agent/TCPSink]
$ns_ attach-agent $node_(1) $tcp1
$ns_ attach-agent $node_(0) $sink1
$ns_ connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns_ at 20.0 "$ftp1 start"

set tcp2 [new Agent/TCP]
$tcp2 set class_ 2
set sink2 [new Agent/TCPSink]
$ns_ attach-agent $node_(2) $tcp2
$ns_ attach-agent $node_(0) $sink2
```

```
$ns_ connect $tcp2 $sink2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ns_ at 20.0 "$ftp2 start"

set tcp3 [new Agent/TCP]
$tcp3 set class_ 2
set sink3 [new Agent/TCPSink]
$ns_ attach-agent $node_(3) $tcp3
$ns_ attach-agent $node_(0) $sink3
$ns_ connect $tcp3 $sink3
set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp3
$ns_ at 20.0 "$ftp3 start"

set tcp4 [new Agent/TCP]
$tcp4 set class_ 2
set sink4 [new Agent/TCPSink]
$ns_ attach-agent $node_(4) $tcp4
$ns_ attach-agent $node_(0) $sink4
$ns_ connect $tcp4 $sink4
set ftp4 [new Application/FTP]
$ftp4 attach-agent $tcp4
$ns_ at 20.0 "$ftp4 start"

set tcp5 [new Agent/TCP]
$tcp5 set class_ 2
set sink5 [new Agent/TCPSink]
$ns_ attach-agent $node_(5) $tcp5
$ns_ attach-agent $node_(0) $sink5
$ns_ connect $tcp5 $sink5
```

```
set ftp5 [new Application/FTP]
$ftp5 attach-agent $tcp5
$ns_ at 20.0 "$ftp5 start"

set tcp6 [new Agent/TCP]
$tcp6 set class_ 2
set sink6 [new Agent/TCPSink]
$ns_ attach-agent $node_(6) $tcp6
$ns_ attach-agent $node_(0) $sink6
$ns_ connect $tcp6 $sink6
set ftp6 [new Application/FTP]
$ftp6 attach-agent $tcp6
$ns_ at 20.0 "$ftp6 start"

set tcp7 [new Agent/TCP]
$tcp7 set class_ 2
set sink7 [new Agent/TCPSink]
$ns_ attach-agent $node_(7) $tcp7
$ns_ attach-agent $node_(0) $sink7
$ns_ connect $tcp7 $sink7
set ftp7 [new Application/FTP]
$ftp7 attach-agent $tcp7
$ns_ at 20.0 "$ftp7 start"

set tcp10 [new Agent/TCP]
$tcp10 set class_ 2
set sink10 [new Agent/TCPSink]
$ns_ attach-agent $node_(10) $tcp10
$ns_ attach-agent $node_(0) $sink10
$ns_ connect $tcp10 $sink10
set ftp10 [new Application/FTP]
```

```
$ftp10 attach-agent $tcp10
$ns_ at 20.0 "$ftp10 start"

set tcp11 [new Agent/TCP]
$tcp11 set class_ 2
set sink11 [new Agent/TCPSink]
$ns_ attach-agent $node_(11) $tcp11
$ns_ attach-agent $node_(0) $sink11
$ns_ connect $tcp11 $sink11
set ftp11 [new Application/FTP]
$ftp11 attach-agent $tcp11
$ns_ at 20.0 "$ftp11 start"

set tcp12 [new Agent/TCP]
$tcp12 set class_ 2
set sink12 [new Agent/TCPSink]
$ns_ attach-agent $node_(12) $tcp12
$ns_ attach-agent $node_(0) $sink12
$ns_ connect $tcp12 $sink12
set ftp12 [new Application/FTP]
$ftp12 attach-agent $tcp12
$ns_ at 20.0 "$ftp12 start"

set tcp14 [new Agent/TCP]
$tcp14 set class_ 2
set sink14 [new Agent/TCPSink]
$ns_ attach-agent $node_(14) $tcp14
$ns_ attach-agent $node_(0) $sink14
$ns_ connect $tcp14 $sink14
set ftp14 [new Application/FTP]
$ftp14 attach-agent $tcp14
```

```
$ns_ at 20.0 "$ftp14 start"

set tcp15 [new Agent/TCP]
$tcp15 set class_ 2
set sink15 [new Agent/TCPSink]
$ns_ attach-agent $node_(0) $tcp15
$ns_ attach-agent $node_(3) $sink15
$ns_ connect $tcp15 $sink15
set ftp15 [new Application/FTP]
$ftp15 attach-agent $tcp15
$ns_ at 20.0 "$ftp15 start"

for {set i 0} {$i < $val(nn) } {incr i}
{
$ns_ at 120.0 "$node_($i) reset";
}
$ns_ at 120.0 "stop"
$ns_ at 120.01 "$ns_ halt"
#Define a 'finish' procedure
proc stop {} {
        global ns_ tracefd
        $ns_ flush-trace
              #Close the trace file
        close $tracefd
              #Execute nam on the trace file
        exec nam 15nod.nam   &
        exit 0
       }
$ns_ run
```

APPENDIX B

TR FILE

A snapshot of tr file is shown here. Generally a tr file is of several MBs. So it is not possible to show a complete file. Data interpretation is done from this file i.e. graphs are made by extracting and interpreting data from this tr file say out.tr.

```
s 52.244820785 _3_ MAC  --- 0 ACK 38 [0 0 0 0]
r 52.245125472 _0_ MAC  --- 0 ACK 38 [0 0 0 0]
s 52.245274998 _1_ MAC  --- 0 RTS 44 [253e 0 1 0]
r 52.245627798 _0_ MAC  --- 0 RTS 44 [253e 0 1 0]
s 52.245952598 _1_ MAC  --- 6056 tcp 1112 [13a 0 1 800] ------
- [1:0 0:0 32 0] [236 0] 0 0
r 52.254849398 _0_ MAC  --- 6056 tcp 1060 [13a 0 1 800] ------
- [1:0 0:0 32 0] [236 0] 1 0
s 52.254859398 _0_ MAC  --- 0 ACK 38 [0 1 0 0]
r 52.255164198 _1_ MAC  --- 0 ACK 38 [0 1 0 0]
s 52.255254086 _3_ MAC  --- 0 RTS 44 [253e 0 3 0]
s 52.255254198 _1_ MAC  --- 6070 message 240 [0 ffffffff 1
800] ------- [1:255 -1:255 32 0]
D 52.255254998 _0_ MAC  COL 0 RTS 44 [253e 0 3 0]
s 52.257749165 _12_ MAC  --- 0 RTS 44 [253e 0 c 0]
r 52.258101332 _0_ MAC  --- 0 RTS 44 [253e 0 c 0]
s 52.258111332 _0_ MAC  --- 0 CTS 38 [2404 c 0 0]
r 52.258415498 _12_ MAC  --- 0 CTS 38 [2404 c 0 0]
s 52.258425498 _12_ MAC  --- 6023 tcp 1112 [13a 0 c 800] -----
-- [12:0 0:11 32 0] [317 0] 0 0
```

```
r 52.267321665 _0_ MAC  --- 6023 tcp 1060 [13a 0 c 800] ------
- [12:0 0:11 32 0] [317 0] 1 0
s 52.267331665 _0_ MAC  --- 0 ACK 38 [0 c 0 0]
r 52.267635832 _12_ MAC  --- 0 ACK 38 [0 c 0 0]
s 52.267725665 _0_ MAC  --- 0 RTS 44 [5fe 1 0 0]
r 52.268078465 _1_ MAC  --- 0 RTS 44 [5fe 1 0 0]
s 52.268088465 _1_ MAC  --- 0 CTS 38 [4c4 0 0 0]
r 52.268393265 _0_ MAC  --- 0 CTS 38 [4c4 0 0 0]
s 52.268403265 _0_ MAC  --- 5772 ack 112 [13a 1 0 800] -------
[0:0 1:0 32 1] [217 0] 0 0
r 52.269300065 _1_ MAC  --- 5772 ack 60 [13a 1 0 800] -------
[0:0 1:0 32 1] [217 0] 1 0
s 52.269310065 _1_ MAC  --- 0 ACK 38 [0 0 0 0]
r 52.269614865 _0_ MAC  --- 0 ACK 38 [0 0 0 0]
s 52.270187150 _0_ MAC  --- 0 CTS 38 [2404 6 0 0]
r 52.270491693 _6_ MAC  --- 0 CTS 38 [2404 6 0 0]
```

Depending on the simulation, different trace file formats are produced[35]. A brief description of the above shown trace file format is given below:

The first field represents the operation performed in the simulation. ‘r’ stands for receive, ‘s’ stands for send and ‘d’ stands for drop.

The second field represents simulation time of event occurrence.

The third field denotes the node number at which the operation is being performed.

The fourth field, here, always has the value ‘MAC’ because only mac trace is considered. In case of collision of RTS or CTS frames, MAC is followed by COL.

The fifth field is uid which has unique value if next field is ‘tcp’ or its ‘ack’ otherwise it is assigned ‘0’ value here.

The sixth field represents the packet type like ARP, RTS, CTS, ACK, tcp, message etc.

The seventh field represents the packet size. The other fields represent flags, ip flow identifier, unique packet identifier etc.

APPENDIX C

NAM FILE

A nam file is used for visual animation. Its size also varies in several MBs. A snap shot for the same is shown below:

```
h -t 22.666566613 -s 1 -d 0 -p tcp -e 1112 -c 2 -a 0 -i 557 -k MAC
r -t 22.675463413 -s 0 -d 0 -p tcp -e 1060 -c 2 -a 0 -i 557 -k MAC
+ -t 22.675473413 -s 0 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
- -t 22.675473413 -s 0 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
h -t 22.675473413 -s 0 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
d -t 22.675513413 -s 0 -d 1 -p ack -e 60 -c 2 -a 0 -i 567 -k IFQ
r -t 22.675778213 -s 1 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
+ -t 22.675847413 -s 0 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
- -t 22.675847413 -s 0 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
h -t 22.675847413 -s 0 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
r -t 22.676199579 -s 12 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
+ -t 22.676209579 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
- -t 22.676209579 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
h -t 22.676209579 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
r -t 22.676513746 -s 0 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
+ -t 22.676523746 -s 0 -d 12 -p ack -e 112 -c 2 -a 0 -i 361 -k MAC
- -t 22.676523746 -s 0 -d 12 -p ack -e 112 -c 2 -a 0 -i 361 -k MAC
h -t 22.676523746 -s 0 -d 12 -p ack -e 112 -c 2 -a 0 -i 361 -k MAC
r -t 22.677419913 -s 12 -d 12 -p ack -e 60 -c 2 -a 0 -i 361 -k MAC
+ -t 22.677429913 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
- -t 22.677429913 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
h -t 22.677429913 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
```

```
r -t 22.677734079 -s 0 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
+ -t 22.678024079 -s 0 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
- -t 22.678024079 -s 0 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
h -t 22.678024079 -s 0 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
r -t 22.678376246 -s 12 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
+ -t 22.678386246 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
- -t 22.678386246 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
h -t 22.678386246 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
r -t 22.678690413 -s 0 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
+ -t 22.678700413 -s 0 -d 12 -p ack -e 112 -c 2 -a 0 -i 365 -k MAC
- -t 22.678700413 -s 0 -d 12 -p ack -e 112 -c 2 -a 0 -i 365 -k MAC
h -t 22.678700413 -s 0 -d 12 -p ack -e 112 -c 2 -a 0 -i 365 -k MAC
r -t 22.679596579 -s 12 -d 12 -p ack -e 60 -c 2 -a 0 -i 365 -k MAC
+ -t 22.679606579 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
- -t 22.679606579 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
h -t 22.679606579 -s 12 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
r -t 22.679910746 -s 0 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
+ -t 22.679980746 -s 0 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
- -t 22.679980746 -s 0 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
h -t 22.679980746 -s 0 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
r -t 22.680332940 -s 11 -d -1 -p MAC -e 44 -c 2 -a 0 -i 0 -k MAC
+ -t 22.680342940 -s 11 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
- -t 22.680342940 -s 11 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
h -t 22.680342940 -s 11 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
r -t 22.680647135 -s 0 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
+ -t 22.680657135 -s 0 -d 11 -p ack -e 112 -c 2 -a 0 -i 370 -k MAC
- -t 22.680657135 -s 0 -d 11 -p ack -e 112 -c 2 -a 0 -i 370 -k MAC
h -t 22.680657135 -s 0 -d 11 -p ack -e 112 -c 2 -a 0 -i 370 -k MAC
r -t 22.681553329 -s 11 -d 11 -p ack -e 60 -c 2 -a 0 -i 370 -k MAC
+ -t 22.681563329 -s 11 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
- -t 22.681563329 -s 11 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
h -t 22.681563329 -s 11 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
r -t 22.681867524 -s 0 -d -1 -p MAC -e 38 -c 2 -a 0 -i 0 -k MAC
```

APPENDIX D
PERL SCRIPT

To calculate the throughput from the tr file, the following script has been written in perl[16]. This script is run independently from different simulation scenarios. The name of the tr file and simulation time is passed in this script of a particular scenario and it gives the throughput in Kbps for the same.

```
# type: perl throughput.pl <trace file> <granularity>   >    output file

$infile=$ARGV[0];
$granularity=$ARGV[1];

#we compute how many bytes were transmitted during time interval specified
#by granularity parameter in seconds
$sum=0;
$clock=0;

      open (DATA,"<$infile")
        || die "Can't open $infile $!";

    while (<DATA>) {
             @x = split(' ');

#column 1 is time
if ($x[1]-$clock <= $granularity)
{
#checking if the packet type is TCP
if ($x[6] eq 'tcp')
{
#checking if the event corresponds to a reception
if ($x[0] eq 'r')
{
   $sum=$sum+$x[7];
}
}
}
}
  $throughput=$sum/$granularity/1024;

   print STDOUT "$x[1] $throughput Kbps\n";
   close DATA;
exit(0);
```