SELINUS UNIVERSITY
OF SCIENCES AND LITERATURE

Location Detection System (LDS) with Delay Time Routing (DTR) and MAC for 802.11Wireless Networks

By Hassan Ftouni

A THESIS

Presented to the department of
Electronic Engineer
Program at Selinus University

Faculty of Engineering & Technology
In fulfillment of the requirement
For the degree of Doctor of Philosophy
In Electronic Engineer

2022

A Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (PhD)

Declaration: I have read and understand the PhD Thesis guidelines on plagiarism and cheating, and I certify that this submission fully complies with these guidelines.

Abstract

One of the most common problems which faces the 802.11 wireless LAN is the hidden terminal problem. The main issue of this problem is that it effects negatively on the network performance, and the MAC protocol. In order to solve this problem, this Thesis presents two techniques: Location Detection System (LDS), and Delay Time Routing (DTR-MAC). The first technique is based on each client can detect its location. LDS solves the hidden terminal problem by diving the coverage area of the access point into four different areas then by sharing the geographical position of each client. On the other hand, the DTR-MAC uses the time division multiple access and Location detection system by dividing the area of the access point into four areas, allowing each area to access the network for a specific time. Accessing the network is each area is controlled by Common Sense Multiple Access-Collision Avoidance (CSMA-CA). The advantages of the two proposed techniques can be seen through simulation tests.

LDS improves the network's media access delay compared to the original MAC, which improves the media access delay by up to 37.5 % compared to the original MAC. With respect to the DTR it improves the following: Data Dropped, Delay, Network Load, Retransmission, Throughput, and Collision, which improves the data dropped up to 74 %, delay 20 %, Network Load 50 %, Retransmission 15%, Throughput 80 %, and collision 25 % compared to the standard 802.11 MAC.

Finally, this thesis shows us the LDS and the DTR based on the idea of dynamic delay, and time division multiple access technique. The proposed method of the LDS incorporates the Interleaving method with the time delay. It has an advantage on the original MAC, which could only achieve less media access delay. On the other hand, the proposed method of the DTR incorporates the time division multiples access technique with the four different zones, which improves the whole network performance

Acknowledgement

My appreciation should be extended to my family for their support, help, guidance, and patience through my whole work in this project. Their support is deeply appreciated.

Contents

Table of Figures

Glossary

ACK: Acknowledgement.

AP: Access-Point.

BSS: Basic Service Set.

CS: Carrier Sense.

CW: Contention Window.

DIFS: DCF Inter-Frame Space.

LDS: Location Detection System.

DCF: Distributed Coordination Function.

DSSS: Direct Sequence Spread Spectrum.

DS: Distribution System.

EDCA: Enhanced Distributed Coordination Access.

ESS: Extended Service Set.

FDD: Frequency Division Duplex.

FHSS: Frequency Hopping Spread Spectrum.

HCF: Hybrid Coordination Function.

HCCA: HCF Control Channel Access.

IAPP: Inter-Access Point Protocol.

IBSS: Independent Basic Service Set.

IEEE: Institute of Electrical and Electronic Engineers.

LAN: Local Area Network.

LLC: Logic Link Control.

MAC: Media Access Control.

MIMO: Multiple Input Multiple Outputs.

NAV: Net Allocation Vector.

NIC: Network Interface Card.

OFDM: Orthogonal Frequency Division Multiplexing.

OSI: International Standard Organization.

PCF: Point Coordination Function.

PHY: Physical.

RTS/CTS: Request to Send/ Clear to Send.

SIFS: Short Inter-Frame Space.

STA: Station.

TDD: Time Division Duplex

DTR-MAC: Delay Time Routing-Media Access Control.

TIA: Telecommunication Industry Association.

TKIP: Temporal Key Integrity Protocol.

TXOP: Transmit Opportunity.

WECA: Wireless Ethernet Compatibility Alliance.

WEP: Wired Equivalent Protocol.

Wi-Fi: Wireless Fidelity.

WPA2: Wi-Fi Protected Access 2.

1 Chapter1

Introduction

1.1 Overview

IEEE 802.11 is the most popular standard that is used by the wireless local area network (WLAN), because it’s efficient, easy to use, and low cost. It allows any client to connect to an exciting network through wireless connection [1]. It is no longer a second device, but we can say that it becomes a primary device. Most of the PDAs, laptops, phones, are built in with wireless networks [2].

IEEE 802.11 was developed by 42 different countries by some of the most expert of wireless LAN engineers. The project began in 1990’s and was chaired by Vic Hayes, it was approved in 26 of June 1997 after it passed through six versions, and the final approval of this standard was given in 1999. This standard defines only two layers, the MAC layer and the PHY layer, or the International Standard Organization (OSI) 7 layers. Also, this standard does not identify all interfaces, like when a client node roams between two cells it does not allow an access point to hand-off a connection from one access-point to another, which was a big problem for the users, because users who want to have roaming capability would need to buy the whole device from the same manufacturer. To solve these problem big companies worked together for example: Lucent, AiroNet, and Digital Ocean Collaborated and developed a new protocol name: Inter-access Point Protocol (IAPP). This developed protocol will improve the IEEE 802.11 standard by covering a higher layer such as the logic link control (LLC), and allow inter access points communication which will allow different access point from different manufacturers to communicate with each other, this development was a big step in the improvement of the IEEE 802.11 standard [3].

The purpose of this project is to add improvement to the Distributed Location Aided DLA algorithm, which is used to mitigate the hidden terminal problem in DCF 802.11 WLANs. According to the paper of DLA, the DLA algorithm suffers from delay unfairness in distributing the delay among the four different zones of the access point. The improvement to this algorithm will be in applying a fair delay algorithm to the DLA, so that the delay will be distributed evenly among the four different zones of the access point. After applying the algorithm to the DLA to eliminate the delay unfairness, we hope to have a lot of improvements on the network performance like: end-to-end delay, throughput, data drop, etc.

The Second purpose of this project is to design a new algorithm named Time Division Multiple Access (TDMA-MAC), which is designed to eliminate the hidden terminal problem. According to the TDMA-MAC it is done by splitting the coverage area into four zones, each zone is allowed to access the medium for a specific time, while the others are stopped. After applying the TDMA-MAC to the original-MAC we hope to eliminate the hidden terminal problem and see a lot of improvements on the network.

1.2 Aims and Objectives

The aim of this project is to design the Location Detection System (LDS) by dividing the coverage area of the access point into four areas and allowing each client to detect its area location, to prevent the hidden terminal problem. The results which we will obtained are to be compared with the original MAC layer of the 802.11 Wireless LAN. The second aim of this project is to design a new technique DTR-MAC to solve the hidden terminal problem, and improve the performance of the network.

The Objective of the project is to be defined as:

- Simulating the Network LDS with OPNET stimulator.
- Modifying in the original Wireless LAN MAC layer to make the both of station and clients to detect their own location in each area of the four areas of the access point to prevent the hidden terminal problem.
- The mechanism should define how to acquire the location of each node.
- Writing a new algorithm called DTR-MAC to solve the hidden terminal problem, and to improve the performance of the network.
- Modifying the original-MAC in order to do Time Division between the 4 different areas.

- The algorithm should define how to divide the time between the four different areas.

After simulation is done, we should study the results and see the improvements of the LDS, and DTR-MAC, and how the algorithm should help in improving the network performance.

1.3 Organization of the Project

Chapters one and two give us a general idea about the “802.11” standard. They talk about the architecture, distributed coordination function, RTS/CTS, and briefly explain the operation of the MAC-layer and how it works.

Chapter three is divided into two sections. Section one talks about “LDS” and gives us information about the method acquired in order to eliminate hidden terminal problem. Section two discusses the new method which we have developed to eliminate the hidden terminal problem, which name is DTR.

Chapter four shows us the results of the methods which we have acquired and compares them with the old result of the original MAC. Chapter four also shows the calculations which we have done in order to prove that the delay in "LDS" between the zones is equal.

In chapter five, we discuss the results which we have obtained in our project, and we focus especially on the network performance for example: “data dropped, delay, media access delay, network load, and throughput”.

Finally, chapter six gives us a brief conclusion about our work, and explain the further work that we should do in order to improve the performance for the LDS, and DTR.

2 Chapter 2

Literature Review

2.1 Introduction and History of IEEE 802.11

Wi-Fi a word that has spread quickly in the world of communications is ruled by the US Federal Communications Commission, which released many bands for unlicensed used. But the progress of this invention was moving too slowly since there was no common standard, which made the device of different manufacture incompatible. In 1997 the IEEE came up with a common standard which was the 802.11 [4]. The IEEE 802.11 stands for “Institute of Electrical and Electronics Engineers”, this institute works with local and metropolitan networks, the 802 can be divided into many working groups each group focuses on specific issues of the local and metropolitan networks. Here are some working groups of the 802 series:

- 802.1: works with management and bridging.
- 802.2: works with Logic Link Control (LLC).
- 802.3: works with the access method like CSMA/CA “Common Sense Multiple Access Collision Avoidance”.
- 802.4: works with access method like Token-Passing Bus.
- 802.7: works with broadband LAN.
- 802.11: works with wireless technologies.

But what is 802.11? As we mentioned before 802.11 stands for the wireless working group, it was formed in September 1990, to create a wireless LAN that works on the ISM frequency range “Industrial, Scientific, and Medical”. The 802.11 protocol was divided into two layers the MAC layer and the PHY layer. The MAC layer is responsible for handling the moving data between the link layer and the physical medium. The 802.11 can operate in three different modes: Infrared, 2.4 GHz FHSS (Frequency Hopping Spreading Spectrum), and in 2.4 GHz Direct Sequence Spread Spectrum (DSSS). These three different types of operation provide 1 Mbps, or 2 Mbps data rate. The original standard of 802.11 suffered from many problems, first the throughput delivered by the 802.11 was very low, second if an 802.11 card using DSSS it cannot communicate with an 802.11 card using FHSS. Since the original 802.11 suffers from these kinds of problems it was important to improve and develop this standard. After 2 years later in 1999 the 802.11b was released which offers higher bit rate, and operates on 2.4 GHz using the DSSS. Also 802.11b offers higher data rate than the original 802.11 which can reach 11 Mbps depending on the conditions. After facing the problem that the original 802.11 was suffering from, the wireless companies agreed to work together and created the Wireless Ethernet Compatibility Alliance (WECA). This WECA uses the 802.11b protocol. Two years later, in 2001 another standard was created which is the 802.11a, which operates on the frequency band 5 GHz providing a data rate up to 54 Mbps using Orthogonal Frequency Division Multiplexing (OFDM). Then the 802.11g was created which was between 802.11a and 802.11b standards which operates on the 2.4 GHz frequency band, and uses the OFDM technique.

These four standards 802.11a/b/g were the first standards in the 802.11. The IEEE continues to create standards till this day it reaches the 802.11n which is faster and much more complicated than the older standards operating either in 2.4 GHz or 5 GHz frequency band using OFDM technique with data rate up to 600 Mbps. In the following table we can see the development of the IEEE 802.11 standards, and how they keep on improving [5].

802.11 network standard				
802.11 protocol	Release	Frequency band	Modulation tech.	Data rate
Original standard	Jun 1997	2.4 GHz	DSSS, FHSS	1,2 Mbps
A	Sep 1999	5 GHz	OFDM	11 Mbps
B	Sep 1999	2.4 GHz	DSSS	54 Mbps
G	Jun 2003	2.4 GHz	OFDM, DSSS	54 Mbps
N	Oct 2009	2.4/5 GHz	OFDM	600 Mbps

Table 2.1: History of IEEE 802.11

2.2 IEEE 802.11 Architecture

IEEE 802.11 architecture has many advantages, it is very tolerant of faults, very flexible, and supports both small and large networks. This architecture is designed to support a network where most of the information is distributed to the STA. It can be divided into many components,

- Station (STA).
- Access-Point (AP).
- Wireless Medium.
- Basic Service Set (BSS).
- Distribution system (DS).
- Extended Service Set (ESS) [6].

2.2.1 *Station (STA)*

The station is a device which connects to the access point. It consists of two points: a Medium Access Control (MAC) layer and a Physical (PHY) layer. The station connects to the network through a Network Interface Card (NIC), or a network adapter. This station can be two types: mobile or fixed. All stations must support several services like Authentication, de-authentication, and data delivery [6].

2.2.2 *Basic Service Set (BSS) and Independent Basic Service Set (IBSS)*

A basic service set is a group of stations that are connected to the same access-point forming a network. This BSS can be called Independent Basic Set Service (IBSS) when all the station in the BSS is mobile. This IBSS is a short live network designed for a special purpose like to exchange data inside a conference, or in a presentation, in this type of network the STA can communicate directly with each other. The difference between the IBSS and the BSS is the access point, when an access point is found the network cannot be considered as IBSS [6]. In figures 2.1 and 2.2 below we can see the design of the BSS and IBSS

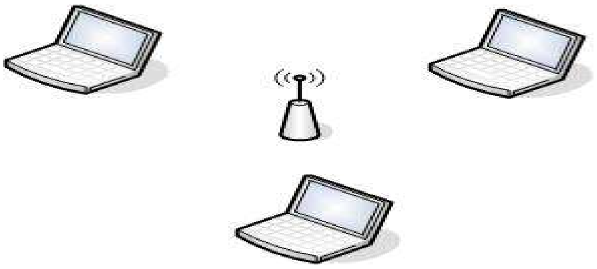


Figure 2-1: Basic Set Service (BSS)

figure 2.2 shows us the IBSS (independent Basic Set Service) which forms a small network without the use of an access point.

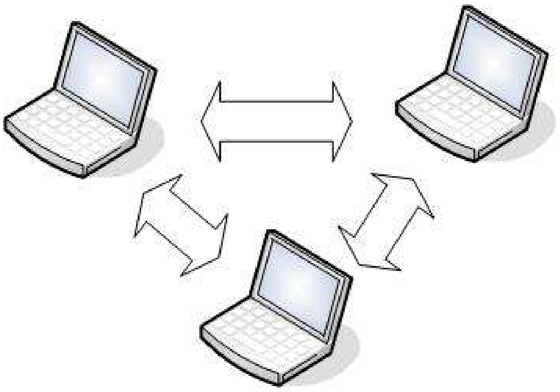


Figure 2-2: Independent Basic Set Service (IBSS)

2.2.3 *Extended Service Set*

Extended Service Set (ESS) is a set of BSSs combined together to extend their range, where the access point communicates with each other to transfer data from one BSS to another. It also makes the movement of the station easier when trying to move from one BSS to another. The communication between the access points is done using a Distribution System (DS), This DS is a backbone that connects the BSS to each other. It can be a wired connection or wireless connection. This ESS hides the mobility of the stations from the networks outside the ESS, which is good since it helps the WLAN which has a lot of mobility to operate correctly with a network that has no concept of mobility [6]. Figure 2.3 shows us how as ESS is formed and how data move from one BSS to another.

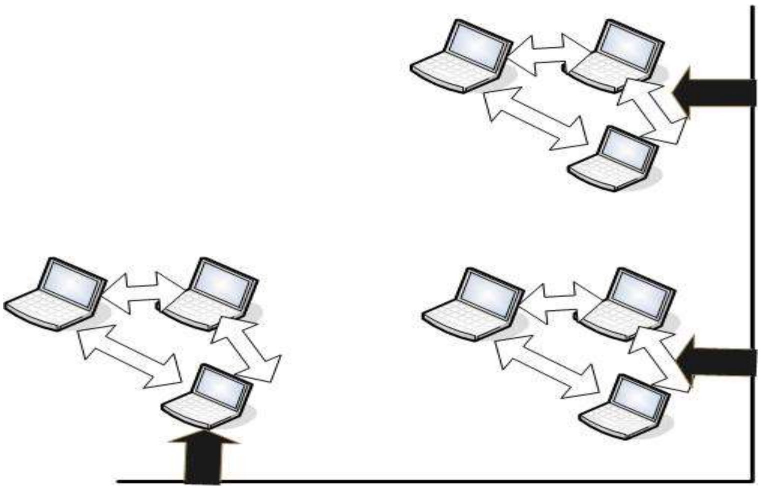


Figure 2-3: Extended Service Set (ESS)

2.2.4 *Distribution System (DS)*

Distribution System (DS) is a backbone that connects BSS with each other to allow the transfer of data from one access-point to another, or to allow transfer of data to a wired network. IEEE 802.11 describes the DS by two things: first DS does not always mean a network, second there is no restriction on how this DS is implemented, but there is a restriction on the service that this DS should provide. The DS can be two types wired or wireless [6].

2.3 Access Protocol in IEEE 802.11

2.3.1 *Introduction*

IEEE 802.11 employs two protocols the Distributed Coordination Function (DCF), and Point Coordination Function (PCF) for medium access control based on CSMA/CA (Carrier Sense Multiple Access Collision Avoidance). The DCF is done by a binary exponential back-off counter (EB). Also, the DCF provides the RTS/CTS mechanism which is not obligatory. With respect to PCF is only used with infrastructure networks [7].

2.3.2 *Distributed Coordination Function (DCF)*

DCF is the main access method in IEEE 802.11, where each station in the medium has an acceptable chance in accessing the medium. This is the main aim of the Distribute Coordination function (DCF) which is known as Carrier Sense Multiple Access-Collision Avoidance (CSMA/CA). This CSMA-CA based DCF works as follows. First a station must check if the medium is free, if the medium is free for a specific time the station can transmit immediately, otherwise the station should wait and not send the packet, in this way the collision is avoided. In order to prevent this collision, the station chooses a random number which is called back-off time and waits for the back-off time to reach zero then it transmits [8].

2.3.2.1 Carrier Sense Mechanism

IEEE 802.11 divides the carrier sensing into two parts, virtual carrier sensing, and physical carrier sensing. The main function of the carrier sensing is to indicate if the station is either busy or free. The virtual carrier sensing uses the net allocation vector (NAV) that save the time of the station that is using the medium, based on the NAV the station can know if the medium is busy or free. On the other hand, the operation responsible for physical sensing is called clear channel assessment (CCA). [9]

2.3.2.2 DCF Inter-Frame Space (DIFS)

DIFS (DCF Inter-Frame Space) is always used by the DCF access protocol, when a station is trying to transmit a data frame or a management frame. The main function of the DIFS is that the DCF uses it to control the transmission of the station, the station must check if the medium is idle for a DIFS time before transmitting to the network, if the medium is busy for the DIFS duration the station cannot transmit [10]. DIFS is shown in figure 2.4 below.

2.3.2.3 Short Inter-Frame Space (SIFS)

SIFS (Short Inter-Frame Space) is a time space that begins at the end of the last symbol of the previous frame to the first symbol of the subsequent frame that should always be used between the acknowledgement and the data frame, or it can be used for any type of frame that is used by the PCF. SIFS is used when the station needs to keep the duration of the frame sequence to be performed. Using SIFS prevents other devices from using the medium while they are waiting for the medium to become idle, by doing this SIFS allow the exchanged frame to be completed successfully [10]. SIFS is shown in figure 2.4 below.

2.3.2.4 Random Back-off Time

When a station wants to send data, it should first check if the medium is free, if it is free the station can transmit directly. But if the medium is busy the station should wait for another

DIFS value and a random back-off counter, then the station starts to decrement the back-off until it reaches zero where it can transmit, if the back-off is interrupted by another station the back-off counter is stopped. We can see back-off in figure 2.4 below. But how is the back-off calculated? Back-off is calculated by using the following equation 2.5 [10]:

$$- \; Back - off = Random\ (\quad) * \ aSlottime \qquad 2.5$$

Where the Random ( ) and $aSlottime$ is defined as follow:

$Random\ (\quad)$: It is called a Pseudo random integer which is chosen randomly form the CW (Contention Window) which should be between the following intervals [0, CW].

$aSlottime$: "*The value of the correspondingly named PHY characteristic.*" [10].

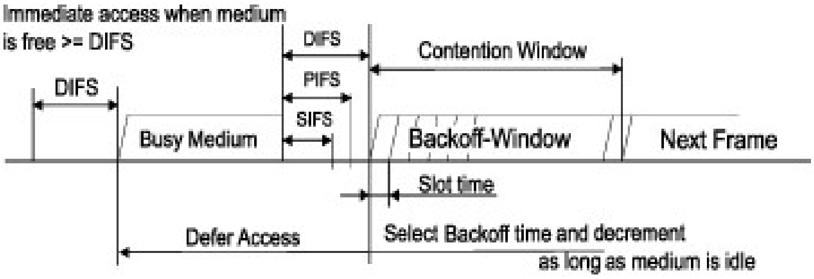


Figure 2-4: DIFS, SIFS, and Back-off [10]

2.3.2.5 *Back-Off Procedure for DCF*

We must know that the back-off procedure is applied when we are using the DCF. If the CS mechanism indicates that the medium is busy, or the transmission was not successful then back-off procedure should be applied. The first step of the back-off procedure is that the station should set the back-off timer according to the equation which we have mentioned before (2.5), then the station starts to decrement the back-off counter by aSlottime, while the station doing this it should check that the medium is not busy by using the physical or virtual CS mechanism, but if the medium was found to be busy, the back-off procedure should stop or be suspended [10]. Figure 2.5 shows us the back-off procedure.

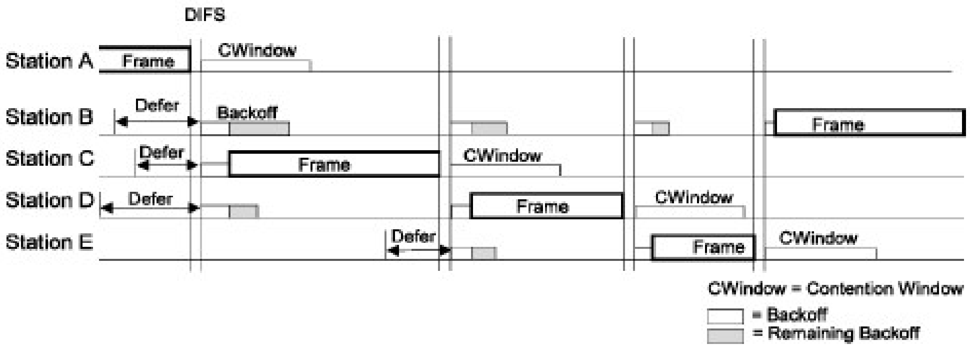


Figure 2-5: Back-off procedure [10]

2.3.3 *RTS (Request to Send)/CTS (Clear to Send)*

To understand the RTS/CTS we should first talk about the hidden terminal problem. A hidden terminal is a problem that occurs when two transmitters cannot hear each other. It happens when two transmitters cannot detect each other while using the carrier sensing. Such a problem will lead to collision of the transmitting information from the two stations since they are sending at the same time. Figure 2.6 shows us the hidden terminal problem.

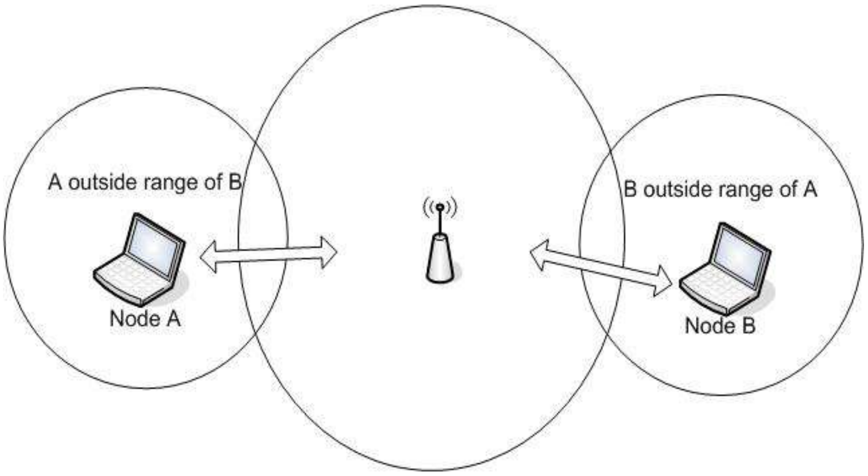
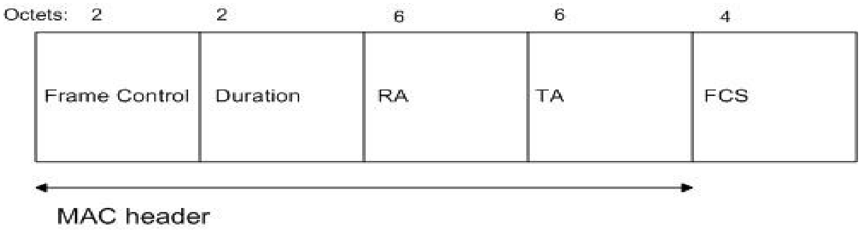


Figure 2-6: Hidden Terminal Problem

After we have look to the hidden terminal problem, we should look to the process that eliminates this problem which is the RTS/CTS. The DCF define the RTS/CTS to help in eliminating the hidden terminal problem, it works as follows: first a station that wants to transmit it sends an advertising frame called RTS (Request to Send) that tells the other station that this station wants to send data. If it receive a CTS (Clear to Send) then the station can starts transmitting [11].

2.3.3.1 RTS Frame Format

In the following figure 2.7 below we can see the RTS frame format.



Octets: 2	2	6	6	4
Frame Control	Duration	RA	TA	FCS

Figure 2-7: RTS Frame Format

As we see the RTS frame format is divided into many fields which are: frame control, duration, RA, TA, FCS. Whereas the frame control, duration, RA, and TA are to be considered as the MAC header.

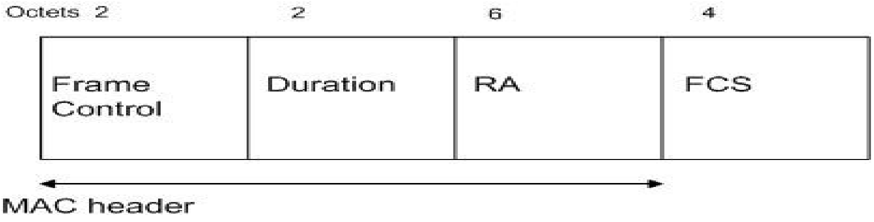
First let us look to the RA field; the RA field is the address of the device, while TA is the address of the station that is transmitting the RTS frame.

The time required for the RTS can be calculated as follows:

The duration value (microseconds), plus one CTS frame, plus one ACK frame, plus three SIFS intervals [10].

2.3.3.2 CTS Frame Format

In the following figure 2.8 below we can see the CTS frame format:



Octets 2	2	6	4
Frame Control	Duration	RA	FCS

Figure 2-8: CTS Frame Format

As we see the CTS frame format is divided into many fields which are: Frame control, duration, RA, and FCS. The first three fields are known as the MAC header. Let us look to the functionality of them.

RA: RA field of the CTS is the same as the TA field of the RTS frame in which the CTS frame copies it from the RTS frame.

Duration value: the duration value of the CTS is the duration value of the RTS, but we should subtract from it the time needed to transmit the CTS frame and the SIFS intervals, the unit of time is in microseconds [10].

2.3.3.3 ACK Frame Format

In the following figure 2.9 below we can see the ACK frame format:

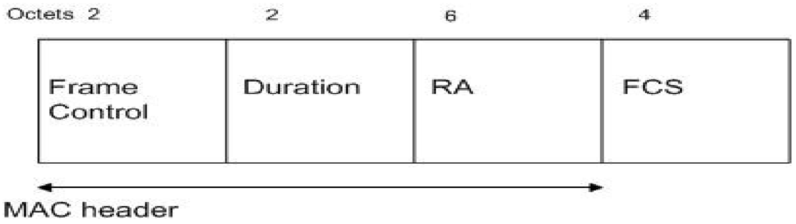


Figure 2-9: ACK Frame Format

As we look to the ACK frame format we can see it is divided into many subfields which are: frame control, duration, RA, and the FCS. Also the first three fields represent the MAC header. Let us look at the functionality of some of these fields.

RA: RA field is taken form the address 2 filed of the BlockAckReq contorl, data, management, BloakAck control.

Duration Filed: it is also obtained from the duration field of the CTS frame, minus the time required to transmit a frame, and it's SIFS [10].

2.3.4 *PCF (Point Coordination Function)*

Point Coordination Function (PCF) is another access method that is used by the IEEE 802.11 that can only be used on the infrastructure network. This PCF works as follows: it uses a PC which acts as polling, which needs operating at the access point to know which of the devices has the right to transmit. The access method which is used by the PCF is a virtual carrier sensing. It controls the medium by setting the NAV for the station. The frame that is transmitted under the PCF uses an Inter Frame Space (IFS) that means it uses frames smaller than the frames sent by the DCF. Because of this, the PCF has more priority in case of overlapping BSS over the access method by the DCF. Let us move on further more to see how the Virtual Carrier Sensing operates. This virtual carrier sensing is provided by the MAC, it is known as NAV, but how does NAV operate? NAV predicts the future traffic of the medium based on the duration information of the RTS/CTS. Also this duration information can be found in the MAC header. But what CS does is combine the NAV and the station transmitter to determine whether the medium is busy or idle [10].

2.3.5 *HCF (Hybrid Coordination Function)*

In 2007 a new coordination function was added to IEEE 802.11 which is known as Hybrid Coordination Function (HCF). This new function combines the work of the both previous function which are: DCF, and PCF, and improve them to create two channel access methods which are known as: Enhanced Distributed Channel Access (EDCA), and HCF Controlled Channel Access (HCCA).

The main difference between the HCF and both of DCF and PCF, is that the DCF and PCF can only transmit one frame, while the HCF is capable enough to send multiple frames, the multiple frames are sent during a period of time called Transmit Opportunity (TXOP). This operation of sending multiple frames is known as frame burst. The frame burst includes Short Inter-frame Space (SIFS) to make sure that the other station is transmitting during the frame burst [12].

2.3.6 *PHY layer*

Different PHYs are defined in the IEEE 802.11 standard. Each PHY consists of two protocols functions:

1. A PHY Media Dependent (PMD) system that defines the characteristics and the method of transmitting and receiving data frame through the wireless medium amongst STAs.

2. A PHY layer convergence protocol (PLCP), which defines a method of mapping the IEEE 802.11 MPDUs into a framing format suitable for sending and receiving user data and management information between the STAs using the associated PMD system. A reference model of 802.11 architecture showing the interaction between the PHY, MAC and higher layers as shown in the figure 2.10 below:

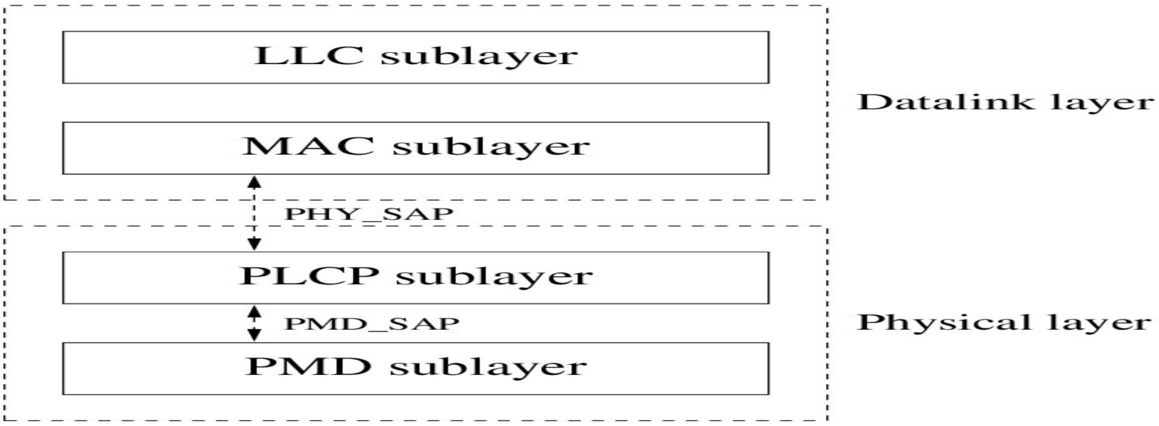


Figure 2-10: Datalink layer, and Physical layer

In order to transmit frames, PLCP forms what has been transferred from the MAC layer into PLCP protocol data unit (PPDUs) from. The PPDU format consists of three parts: a PLCP preamble, a PLCP header, and a PSDU. The PLCP preamble field allows the synchronization and defines the frame start. The PLCP header is used to specify the length of the whitened PSDU field and provide PLCP management information. The PLCP preamble and PLCP header are transmitted at 1 Mbps, while the PSDU can be transmitted at any supported transmission rate. The fields of the PLCP frame are depicted in Figure 2-11

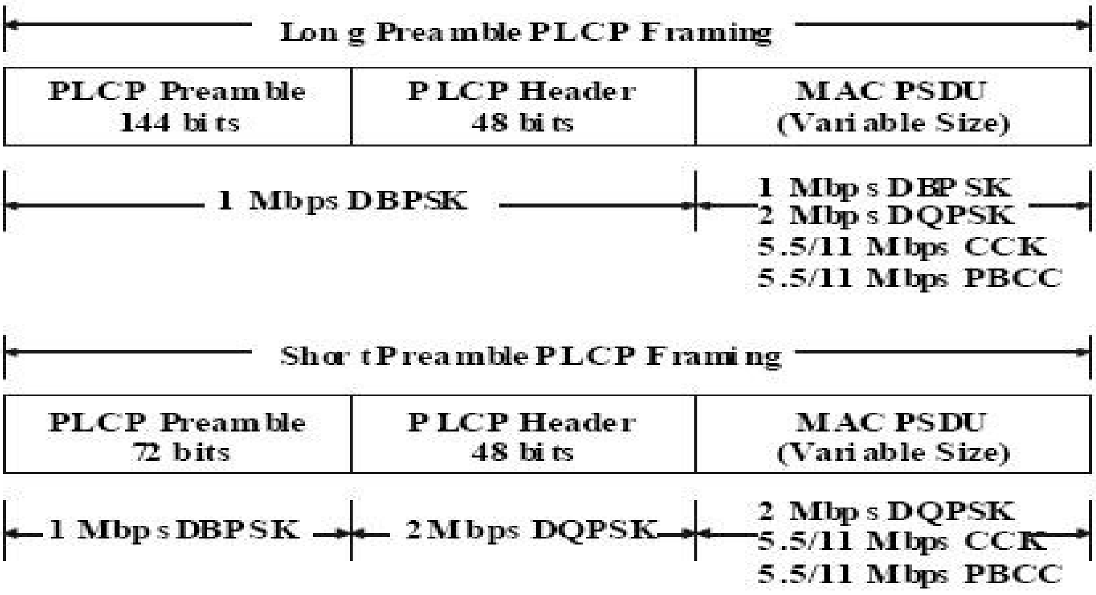
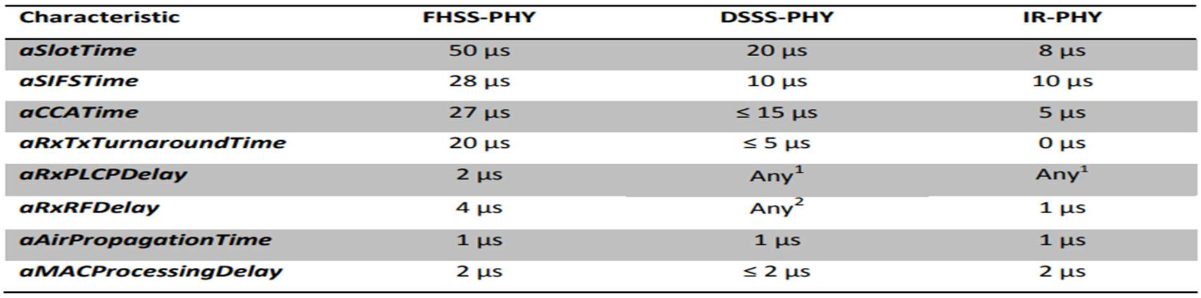


Figure 2. ERP-DSSS/CCK PHY layer PPDU framing

Figure 2-11: the PPDU packet format

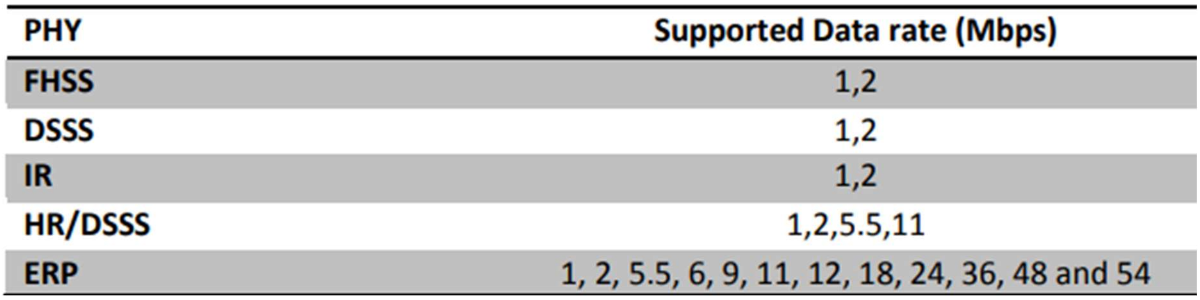
Three different types of PYHs are defined in the original 802.11 standard including Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS) and Infrared (IR). The static characteristics of FHSS-PHY, DSSS-PHY and IR-PHY are given in table 2.1



Characteristic	FHSS-PHY	DSSS-PHY	IR-PHY
aSlotTime	50 µs	20 µs	8 µs
aSIFSTime	28 µs	10 µs	10 µs
aCCATime	27 µs	≤ 15 µs	5 µs
aRxTxTurnaroundTime	20 µs	≤ 5 µs	0 µs
aRxPLCPDelay	2 µs	Any^1	Any^1
aRxRFDelay	4 µs	Any^2	1 µs
aAirPropagationTime	1 µs	1 µs	1 µs
aMACProcessingDelay	2 µs	≤ 2 µs	2 µs

Table 2-1: the timing characteristics

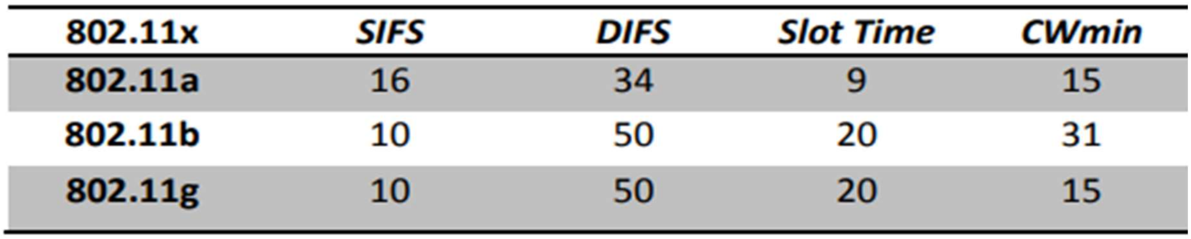
In addition, various extensions of the previously mentioned PHYs have been identified, in order to increase the supported data transmission rate. The high rate DSSS (HR/DSSS) is an extension of the DSSS system, which is designed to support higher payload transmission data rates at 5.5 and 11 Mbps. The Extended rate PHY (ERP), which makes use of the Orthogonal Frequency Division Multiplexing (OFDM) PHY, is developed to provide a data transmission rate of up to 54 Mbps. Table 2-2 illustrates the various PHYs and their supported data rates, taking into consideration the 2.4 GHz ISM band.



PHY	Supported Data rate (Mbps)
FHSS	1,2
DSSS	1,2
IR	1,2
HR/DSSS	1,2,5.5,11
ERP	1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48 and 54

Table 2-2: the supported data rates

Various 802.11 sub-standards have been defined based on the different PHYs specifications and the frequency band used. The 802.11a operates in the 5GHz frequency band and uses the OFDM PHY to support a data rate of up to 54 Mbps. The 802.11b and 802.11g operate in the same 2.4GHz frequency band, however, the 802.11g PHY is based on OFDM to provide high data rate of up to 54 Mbps. Table 2-3 depicts the values for the MAC parameters of the various IEEE 802.11 standards.



802.11x	*SIFS*	*DIFS*	*Slot Time*	*CWmin*
802.11a	16	34	9	15
802.11b	10	50	20	31
802.11g	10	50	20	15

Table 2-3: Some MAC parameters

Figure 2-12 depicts the wireless channels of the 2.4GHz frequency band allocated to the 802.11 standard showing the three non-overlapping channels.

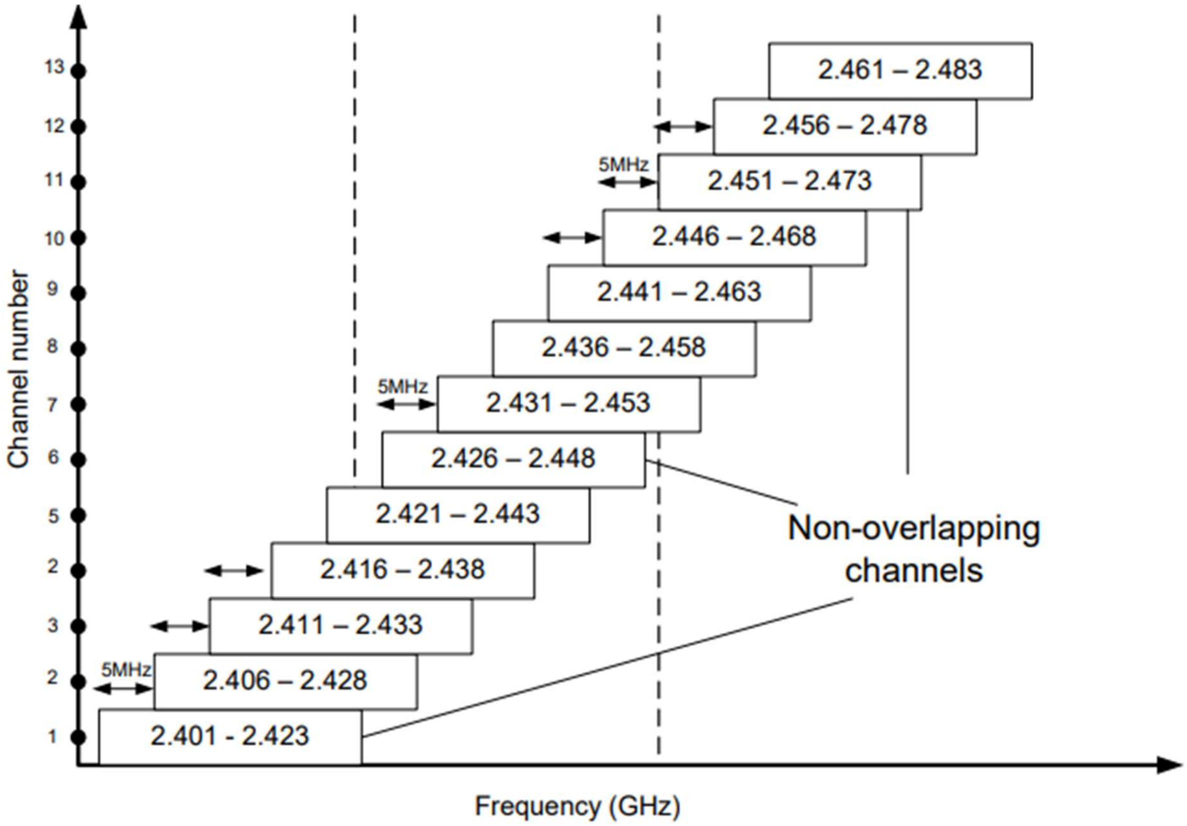


Figure 2-12: The wireless channels

2.4 Mobile Ad-Hoc Networks (MANETs)

Mobile Ad-hoc Networks (MANETs) are a collection of mobile devices (nodes) moving within a geographical area to form a self-healing, self-configuring wireless network. Such type of a wireless network lacks the existence of any sort of infrastructure or centralised entity. Figure 2-13 illustrates an instance of MANET, where a connection to external networks (local area network, internet, etc) can be acquired through one or several devices. MANET configuration is suitable for networks that require rapid deployment such as meeting rooms, sport stadiums, search and rescue and disaster recovery. It can be applied where the deployment of a wired network is impossible such as in battlefield and maritime scenarios.

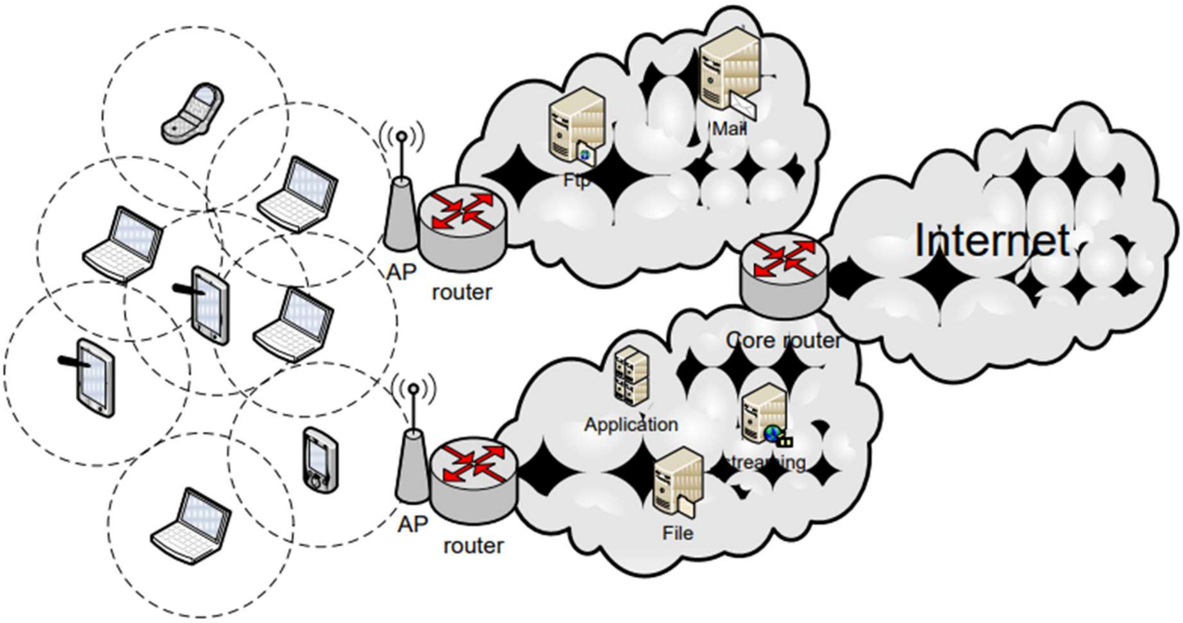


Figure 2-13: An instance of a MANET connected to external networks

A MANET node is able to communicate with all the other nodes within its transmission range via a direct connection. In case the desired destination node is beyond the transmission range of the source, a multi-hop connection, composed of a set of intermediate nodes, is established to maintain the source-destination communication. In addition to the multi-hop communication capability, the mobility support is another principal feature that MANET provides, allowing the wireless nodes to freely move within the defined area. However, the link breakage that is caused by the movement of the mobile nodes changes the network topology very frequently, imposing, therefore, serious challenges on the protocol design, in general, and on the routing algorithms, in particular.

2.5 Characteristics and advantages

Mobile Ad-hoc Network (MANET) shares common characteristics with wireless networks and preserves for itself some others that distinguish it as a special,

- Wireless medium: The MANET nodes communicate with each other wirelessly by sharing the same medium.
- Multi-hop communications: The multi-hop feature of MANET allows the MANET node to communicate with any destination in the network, regardless whether the destination is within its radio range or not. Therefore, each node acts as a router to enable information routing between the source and destination.

- Autonomous and infrastructure-less: MANET does not rely on any sort of infrastructure. Hence, the network administration and control are performed in a distributed manner, where each node acts as an independent router.
- Mobility and dynamic topology: The mobility support in MANET permits the node to move around without interrupting the active communications. The mobility causes link breakage and topology variation, which in turns changes the connection patterns between the mobile nodes.

The above-mentioned characteristics give MANETs important advantages over the other types of wireless networks in terms of moving while communicating, multi-hop communication ...etc. On the other hand, several challenges in the protocol design may rise, especially in developing routing protocols that has to cope with the nodes' mobility and the dynamic topology of the network.

2.6 Routing in MANETs

Various criteria can be considered to classify MANET routing protocols. Such criteria include the way that the routes to the destination nodes are established (reactive or proactive), the topology structure (flat or hierarchical), the routing method (hop by hop or source routing) and the type of information that the protocol relies on to perform the routing process (link or position). Based on the latter, routing in MANETs is classified into topology-based and position based.

2.7 Topology-Based Routing

Topology-based routing approach performs routing based on links' information. The protocols of this category maintain a routing table, where they store topological information that will be used in the routing process. Three sub-classes may be distinguished: reactive on demand, proactive or table driven and hybrid routing protocols. The difference amongst them lies on when the information about the routes is obtained. While proactive protocols periodically exchange topological changes to maintain routes to all the possible destinations, reactive protocols discover routes when needed and maintain the routes that are in active communication. From a routing performance's perspective, reactive protocols reduce the injected routing overhead by discovering routes when needed. However, data transmission is deferred for an additional period of time delay, waiting on the completion of the route discovery process. Hybrid protocols aim to combine the advantages of both previously mentioned schemes. Accordingly, the network area is divided into zones. The communication within the zone (intra-zone) is performed in a proactive manner, while routing between the zones (inter-zone) is performed reactively.

2.7.1.1 Ad-Hoc On-Demand Distance Vector

AODV [7] is a reactive hop-by-hop routing protocol that discovers routes to the destination when needed. It does not require the maintenance of routes to destinations that are not in active communication. When an AODV source node has data to send, it initiates the route discovery process if no valid entry is found in the routing table. The discovery process is started by broadcasting a Route Request (RREQ) message to propagate in the entire network until it reaches either the destination node or an intermediate node with a valid route to the destination. Upon receiving a RREQ message, an intermediate node creates or updates a route to the previous sender of the RREQ. The received RREQ is discarded if the node has received a RREQ with the same originator and RREQ ID within, at least, the last PATH_DISCOVERY_TIME. The node then checks whether a valid entry for the destination exists in its table. If this is the case, a Route Reply (RREP) message is unicasted back to the originator of the RREQ by using the already created reverse path. The same procedure is followed if the RREQ reaches the destination node. In case no valid entry is found in the table, the RREQ message is rebroadcasted after incrementing the hop count value by one. Moreover, AODV uses a sequence number field in its control messages to determine the freshness of the information acquired from the originating node. When the source node receives multiple RREP, the route with the lowest hop count value is selected. The dissemination of the RREQ is controlled by an expanding ring search technique. The originator of the RREQ sets the Time-To-Live (TTL) of the IP header to TTL_START and waits for a RING_TRAVERSAL_TIME before attempting to broadcast the RREQ with an incremented TTL. This continues until the TTL of the RREQ reaches TTL_THRESHOLD, after which a TTL = NET_DIAMETER is used for each subsequent attempt. In addition, the nodes of an active route monitor the link status of the next hops. If a link breakage is detected, a Route Error (RERR) message is flagged to notify the other nodes and to indicate the destinations that are no longer reachable through the broken link. Finally, as a part of an active route, the mobile node periodically broadcasts HELLO messages. The broadcasting of HELLO messages is restricted to the one hop neighborhood.

2.7.1.2 Distance Source Routing

DSR [8] is a reactive source routing protocol. The entire route to the destination is discovered and consequently made known to the source node prior to data transmission. Similar to AODV, the discovery process is initiated when a source node attempts to transmit data to a destination node with an unknown route. The source node broadcasts a RREQ message throughout the network until the requested destination node or an intermediate node with a valid route to the destination is reached. The DSR RREQ packet is different than AODV's as the former contains the entire discovered route. Upon receiving a RREQ, the node checks its cache for a valid route to the requested destination. If no route is found in the cache, the node adds its address to the RREQ and rebroadcasts it further. If, however, the node has a valid route to the RREQ destination, the complete route (the route included in the RREQ + the cached route) is copied to a RREP message and is sent back to the source node. Finally, when the destination node is reached, it simply sends a RREP to the originator of the RREQ by reversing the route recorded in the RREQ. DSR introduces the concept of route salvaging, according to which an intermediate node uses an alternative route from its cache to the packet's destination, when the next hop link along the packet's route is detected as broken. Therefore, the node salvages the packet rather than discarding it by replacing the original source route of the packet with the route of its cache. In addition, DSR uses route shortening mechanism, which is applied when one or more intermediate nodes become unnecessary to the route.

2.7.2 Position Based Routing

2.7.2.1 Existing Position Based Routing Protocols

Several working efforts aim to enhance the routing performance in MANETs by introducing location information into the algorithm have been proposed. A survey of position- based routing algorithms is extensively discussed in [13] and [14]. Below, we highlight several location-based routing protocols, which are related to our work.

A. Greedy forwarding schemes:

Algorithms that use greedy forwarding strategy, which selects the neighbor that satisfies specific criterion as the next hop relaying node, are proposed in [15], [16], [17], [18] [19], [20], [21] and [23]. Random progress method is proposed in [15] according to which packets destined toward a destination node D are routed with equal probability towards one neighboring node that makes progress in the direction of D. The source node will select among the (n) neighbors one terminal located in the direction of the destination D as all neighbors having the same probability (1/n). Progress is defined as the distance separating the transmitter and the receiver projected onto the line joining the transmitter and the final destination. In [16] a variant of random progress method called Cartesian Routing is proposed. Progress in Cartesian Routing is defined as the distance between the transmitter (Xt ,Yt) and the final destination (Xd,Yd). According to this, packets are forwarded to any direct neighbor (Xi ,Yi) for which the distance [(Xi ,Yi) to (Xd,Yd)] is less than the distance [(Xt ,Yt) to (Xd,Yd)]. In case of no direct neighbor closer to the destination is found, conditioned the network is n-Cartesian regular, a search of no farther than (n-1) hops will lead to a node that makes progress. According to [16], a network is called n-Cartesian regular if for any transmitter node T and any destination node D, some other node Ni exists within n-hopes of T and closer to D. Takagi and Kleinrock [17] proposed the Most Forward within Radius (MFR) routing algorithm. MFR forwards the packet to the next neighbor that maximizes the progress. The progress is defined as the distance between the transmitted node and the neighboring node projected onto the line joining the transmitter node and the final destination. In MFR strategy, a case might arise where the selected neighbor having the maximum progress is farther from the destination. Nearest with Forward Progress (NFP) routing algorithm is introduced in [18] where the nearest neighbor with forward progress is selected as the next hop node. Furthermore, greedy forwarding schemes are characterized by routing data packet relying on positions of one-hop neighbors only. However, there are topologies in which some of these schemes fail to deliver the packet to the destination even though a route exists, e.g., a topology where the node itself is closer to the destination than any of its neighbors. This case is referred to as local maxima. Greedy Perimeter Stateless Routing (GPSR) algorithm proposed in [19] maintains information about its direct neighbors' positions to make a routing decision. It consists of two methods of packet forwarding: greedy forwarding and perimeter forwarding. GPSR header includes a field indicating whether the packet is in greedy mode or perimeter mode. Upon receiving a packet for forwarding, a node applies greedy scheme and searches for the neighbor which is geographically closest to the destination. When no neighbor is closer to the destination than the node itself, the packet is marked into perimeter and will be forwarded using simple planar graph traversal.

Stojmenovic and Lin proposed in [20] two hop flooding GEDIR, two hop flooding MFR and two hop flooding DIR, modifications of GEDIR [21], MFR and compass routing schemes to avoid packet dropping. The proposed algorithms are referred to as 2-f- GEDIR, 2-f- MFR and 2-fDIR respectively. The main idea behind these variants is that the transmitter nodes choose the closest terminal to the destination among the first and second hop neighbors except concave node that floods the packet to all its neighbors. A node is called concave if it is the only neighbor of the selected node for forwarding, closer to the destination. Greedy Routing with Anti-Void Traversal (GAR) is introduced [22] to solve the void problem of greedy forwarding scheme by exploiting the boundary finding technique fort the unit disk graph (UDG). Rollingball UDG boundary traversal (RUT) technique is further proposed in [22] to solve the boundary finding problem. Liu and Feng developed the Largest Forwarding Region (LFR) [23] routing protocol, which selects the neighbor that possesses the largest Extended Forwarding Region (EFR). EFR is associated with every neighbor contains both the distance and the direction information related to the destination. Note that the forwarding region is defined as the area including the closest nodes to the destination. Furthermore, Backward Constraint (BC) and Dead-End Recovery (DER) mechanisms are defined to resolve backward loops and dead ends problems in the network. Although LFR resolves the problem of void in the network by transmitting the packet back to the concave node, it would be more efficient not to consider at all the nodes that lead to void which will be shown in this paper.

B. Directional Routing Scheme.

Directional routing methods that rely on the direction of the destination to select the next forwarding node algorithms are discussed in [24], [25], [26], [27] and [28]. In Compass Routing presented in [24], the transmitter node T (source or intermediate node) forwards the packet to its closest neighbor N to the destination D that minimizes the angle (TND). The same procedure is applied at every intermediate node until the packet reaches the destination. Ko and Vaidya in [25] demonstrate with their Location Aided Routing (LAR) protocol how the utilization of location information can improve the flood mechanism of route discovery messages and hence reduce the routing overhead. In LAR, the source node defines the expected zone where the destination is expected to be, based on the location information of the destination and the speed that the destination can reach. The source node only broadcasts the discovery request within the request zone which is the smallest rectangle formed by the expected zone and the source node's position. Two algorithms of LAR also presented in *22]; LAR scheme-1 and LAR scheme-2 which differ in the manner that the request zone is specified in the request message. In the scheme-1, the zone is specified explicitly by the source node while in scheme-2 it is implicitly specified, where the source includes in the request message additional information about the destination coordinates and its distance to the destination. Although LAR reduces routing overhead as it reactively discovers a route to the destination, it still requires maintaining an explicit path between every source and destination prior to data transmission. In [26], a challenge of Location Aided-Routing algorithm is discussed and an improved version of the protocol is presented. Although during the route discovery phase, the destination node receives the request from different routes, it only responds to the earliest request received. Therefore, any later route breakage will lead to a new route discovery process. The author proposed to select a backup route to be used as a secondary route in case of any failure in the primary route. Location Aided Knowledge Extraction Routing (LAKER) [24] utilizes a combination of caching strategy in Dynamic Source Routing and limited flooding in Location-Aided Routing [25]. The idea of LAKER is to learn the topological characteristics of the network and use this information to guide the route discovery more precisely in the request zone. Simulation results show that LAKER saves up to 30% broadcast messages as compared to LAR. A variant of LAR protocol is Multipath Location Aided Routing in 2D and 3D, referred to as MLAR [28] which is designed to work efficiently in 3 dimensions by using alternate path caching strategy. MLAR caches several paths although one path is used at a time and the others are alternate routes to be used when the primary path fails. A close work to FORTEL is Distance Routing Effect Algorithm for Mobility (DREAM) proposed by Basagni et al. [29]. DREAM represents an all-to-all location service that disseminates and updates nodes' location throughout the entire network. The frequency of updates is determined based on the distance between the nodes and the mobility rate. Data packets are transmitted to all the one-hop neighbors that lay in the direction to the destination represented by the angular range that includes the node's position, the destination’s position and the zone that the destination is expected to be. The same procedure is applied at every node until the destination had been reached. Although, transmitting data packets through multiple paths may increase the probability of reaching the destination, the protocol lacks scalability due to the communication overhead and data message redundancy.

C. Hierarchical routing schemes:

Hierarchical approach is discussed in [30] and [31]. GRID protocol discussed in [30] exploits location information in route discovery, packer forwarding and route maintenance. It considers the MANET as 2D logical grids controlled by grid gateways. Packet routing is performed gird-by-grid manner and the gateway hosts are responsible of discovering, maintaining the routes and forwarding data packets to the neighboring grids. Blazevic et al. proposed in [31] the Terminode routing that combines location-based routing and link state routing. Location routing referred as Terminode Remote Routing (TRR) is used when the destination node is far, while link state routing referred to as Terminode Local Routing (TRL) is used when the destination is up to two hops away. Moreover, the concept of anchors, which represent imaginary geographical locations installed in the packet header to assist in the routing process, is introduced. In Position and Neighborhood based Routing (PNR) [32], the networks is represented by a set of quadrants. The quadrants are organized in a hierarchical meaner, where each higher-level quadrant is divided into four lower-level quadrants. PNR requires each node to initiate an initial flooding as a startup phase. Any node moves more than a pre-defined distance must send an update packet. The dissemination of the update packets is optimized using the concept of quadrant. Accordingly, when receiving an update packet, the node maintains the exact location of the packet originator if they are in the same quadrant or it stores the quadrant that the originator belongs otherwise. The routing is based on the shortest path using the concept of greedy forwarding.

D. Other schemes

GPS/Ant-Like Routing Algorithm (GPSAL) routing protocol is described in [30]. The key point of GPSAL is the mobile software agents modeled on ants used to disseminate and collect nodes’ location information more rapidly. An ant holds a routing table and is transmitted to a specific destination. Upon receiving an ant packet, older entries are updated by the current host and the ant is passed to another node carrying the most updated routing table. The same procedure is followed until the ant has reached its destination at which point is sent back to the node that created it. Zeng et al. introduced in [34] Geographic On Demand Disjoint Multipath routing protocol to be used instead of blind flooding of route discovery in the network. Every node knows the position of its one-hop neighbors. Before transmitting route request (RREQ) message, the source node selects the k nearest neighbors to the destination and includes their addresses in the packet. Upon receiving RREQ, only intermediate nodes having their addresses stated in the packet forward the request after selecting a new list of nearest neighbors to the destination. This is repeated until the destination has been reached which in turn transmits a route reply (RREP) message back to the source. In addition, the authors described two schemes: Geographic Node-disjoint–paths routing and Geographic Edge-disjoint-paths routing. The difference between these schemes lies in the processing of the duplicate RREQ messages. While the first scheme drops all the duplicate RREQ, edge-disjoint routing may forward duplicate RREQ having been received from a different neighbor. Recent work is presented in [35], [36] and [37], where different geographic routing algorithms are developed. Predictive Mobility and Location Aware Routing (PMLAR) [35] predicts the movement behavior of the mobile nodes to assist the routing operation. PMLAR is designed in a way that the source node predicts the current and the future location of the destination to increase the routing efficiency. The prediction is based on a precious location update of the destination acquired through a location service. To transmit data packets, the source node determines the predicted zone, which is expected to include the potential future position of the destination. The route discovery process is then initiated to establish a valid route to the destination. During the discovery phase, the intermediate nodes apply the Velocity-Aided Routing (VAR) mechanism to ensure that the RREQ is forwarded by the nodes that are moving toward the destination along their connecting lines. In [36], Location-Aware Routing for Delay tolerant networks (LAROD) is proposed, which is a beacon-less routing protocol designed for intermittently connected MANETs that combines the store-carry-forward technique with the geographical position. LAROD consists of an enhanced location service and a location dissemination service to update the nodes' location information. Finally, PredictionBased Routing (PBR) protocol for vehicular ad hoc networks is proposed in [37]. PBR takes the advantage of the predictable mobility pattern of vehicles on highways to predict the route lifetimes and pre-emptively create new routes before existing ones fail.

2.7.2.2 Distance Routing Effect Algorithm for Mobility (DREAM)

DREAM [29] is a hop-by-hop position-based routing protocol, specifically designed for mobility that proactively disseminates the location information across the network. Each mobile node maintains a Location Table (LT), which contains the location information of all the other nodes. Therefore, when a source node wants to transmit data to a specific destination, it refers to the LT to select all its one-hop neighbors in the direction of the destination that will be the next hop forwarding nodes. The same process is applied at every intermediate node until the destination is reached. The direction of the destination, as shown in Figure 2-14, is defined as the sector formed by the source node and the zone in which the destination node is expected to be located.

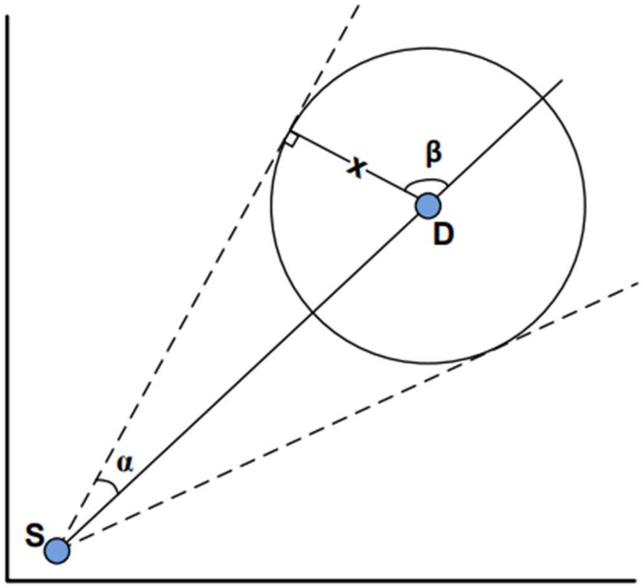


Figure 2-14: Direction of Destination Node

Each node, periodically, broadcasts a control packet containing its own coordinates. To control the routing overhead injected in the network, DREAM uses the distance effect, according to which the further apart the two nodes are, the slower they appear to be moving in respects to each other and subsequently their LTs need updating less often. Therefore, an age parameter is associated with every control message to limit the distance that the message travels from the sender. Besides, DREAM introduces a mobility rate factor to determine the frequency at which the control packets are transmitted. Accordingly, the faster the node moves, the more often it must communicate its location. Furthermore, DREAM supports two types of control messages: short lived and long lived. Every node broadcast, periodically, a short-lived control message that is meant to be delivered to all the nodes whose Euclidean distance to the originator is less than a predefined distance (K grid units). Following the transmission of a specific number (ρ) of short-lived messages, one long lived control message is disseminated throughout the network. To further control the frequency of transmitting the control messages, DREAM uses a mobility rate, which allows the node to self optimise its dissemination frequency. Accordingly, the faster a node moves, the more often its updates its location information.

2.7.2.3 Location Aided Routing.

LAR [24] protocol is a position-based routing protocol that discovers routes to destinations reactively. It uses location information to reduce the routing overhead caused by the route discovery process. Its main concept is to confine the propagation area of the route request (RREQ) messages to the geographical zone that leads to the destination node. For this reason, LAR defines two zones: expected zone and request zone. The expected zone, illustrated in Figure 2-15, is the circle where the destination node is expected to be located.

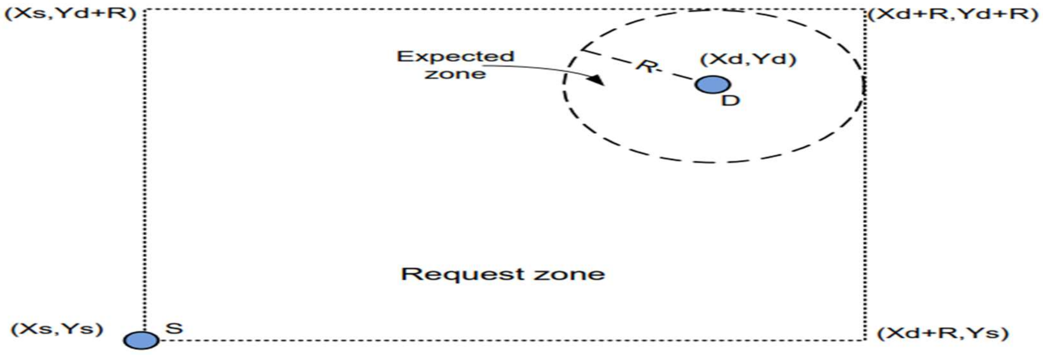


Figure 2-15: Request and Expected Zone.

The source node, only, broadcasts the discovery request within the request zone, which is the smallest rectangle formed by the expected zone and the source node's position. Furthermore, LAR defines two schemes: scheme-1 and scheme-2. The difference resides in the way the request zone is specified within the request message. In the scheme-1, the source node explicitly specifies the request zone by including the coordinates of the zone's four corners in the RREQ. The receivers located outside the specified rectangle discards the RREQ. On the other hand, in scheme-2, the source node includes in the RREQ the destination's coordinates as well as its distance, Dists, to the destination. The receiving nodes will then calculate their distance to the destination node, and only the nodes whose distance is greater than Dists will forward the RREQ.

2.8 IEEE 802.11 Standards

Since 1997 as the first standard was issued, a lot of IEEE 802.11 standards have been developed according to the original IEEE 802.11. These are some of the standards:

- 802.11a: it is the extension of the original IEEE 802.11 it uses a data rate up to 54 Mbps, with an OFDM (Orthogonal Frequency Division Multiplexing), and a frequency 5 GHz. This standard is newer than the IEEE 802.11b.
- 802.11b: this standard is called 802.11 high rate, or Wi-Fi. It uses a data rate up to 11 Mbps with a frequency 2.4 GHz, and DSSS (Direct Sequence Spreading Spectrum) modulation.
- 802.11i: this protocol is also known as Wi-Fi protected Access 2 (WPA2) this standard has developed to overcome on the security problems that the Wired Equivalent Protocol (WEP) suffers from. This protocol also uses the Temporal Key Integrity Protocol (TKIP) which rotates key periodically. The importance of this standard in mainly for improving the Wireless security.
- 802.11j: this protocol was issued to use in Japan, for wireless networks using frequency of 4.9 GHz- 5 GHz.
- 802.11n: this protocol is known as Multiple Input Multiple Output (MIMO) it was issued to help increasing the throughput at a higher speed, and provides more coverage range of up to 400 meters.
- 802.11r: this protocol is used in applications which need fast roaming when moving from one access-point to another, this standard is applied for the application that needs low latency and high quality [39].

2.9 Time Division Multiple Access (TDMA)

TDMA is a technology, where the bandwidth of the users is divided among them according to the time. TDMA is most used by the Second-Generation System (GSM). TDMA split the band into several time slots which form a frame, the stations which is using TDMA each one is assigned a slot number for sending and receiving data, each station knows the slot number which it is using in order to know how much time it should wait. The transmission of TDMA can either occur by using FDD-TDMA or by TDD-FDMA, where each one operates in a different way from another [40].

2.9.1 *Time Division Multiple Access-Frequency Division Duplex (TDMA-FDD)*

In TDMA-FDD is done by using two different channels for the Uplink and the Downlink, it can be done in both directions. The Base station multiplexes the data and sends it to all stations, while the stations use the TDMA to send it back to the base station. By using two different frequencies to communicate with the base station, this operation is known as TDMA-FDD [41]. Figure 2.16 below shows us TDMA-FDD operation.

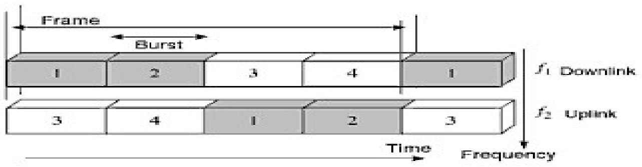


Figure 2-16: TDMA-FDD [42].

2.9.2 *Time Division Multiple Access-Time Division Duplex (TDMA-TDD)*

In TDMA-TDD is achieved by using same frequency with alternative time slot in the frame to get a full duplex connection. In TDMA-TDD the time slots in the frame are divided to forward link and reverse link in which each channel use 50% of the time slots. It is often used in Pico-cell or Micro-cells [43]. Figure 2.17 below shows us the structure of TDMA-TDD

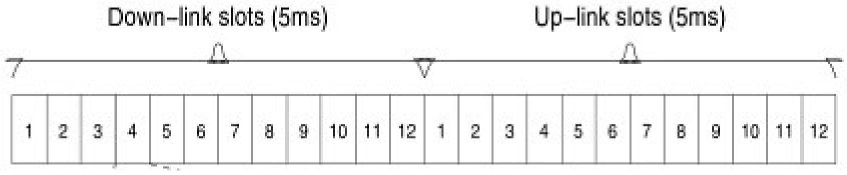


Figure 2-17: TDMA-TDD [44].

2.9.3 *TDMA standards*

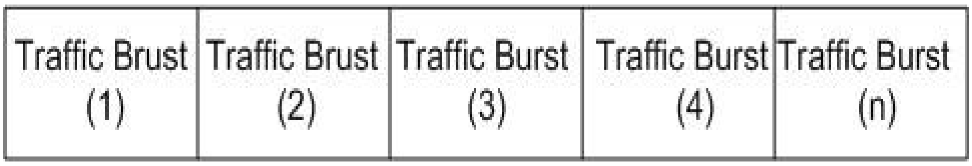
The telecommunication industry association (TIA) introduced the first TDMA standard in 1988-1989, which was named IS-54. The feature set which was provided by this standard was authentication, calling number-ID, and voice privacy.

- In 1994 a new standard was introduced named IS-136.

- Another standard which name is IS-136A was introduced, which use the same frequency used by IS-136 (800 and 1900 MHz), but with some additions like: Over-the-air-activation and programming services.
- Another standard named IS-136B was also introduced which has more features like: broadcast SMS, packet data and other [45].

2.9.4 *TDMA Frame Structure*

The frame structure of TDMA is made of many components. First it starts with a reference burst which is transmitted by the station followed by a traffic burst. Between traffic bursts come guard bands which are used to separate the traffic burst that comes from different stations. The Reference burst must be synchronized to the traffic burst. Figure 2.18 below shows us the frame structure of the TDMA [46].



Traffic Brust (1)	Traffic Brust (2)	Traffic Burst (3)	Traffic Burst (4)	Traffic Burst (n)

Figure 2-18: TDMA frame structure

But we should note here that according to the type of communication, the TDMA frame structure changes, this means TDMA frame structure used by satellite differs from the TDMA frame structure used by GSM.

2.10 Conclusion

This chapter gives an overview on the IEEE 802.11 Mobile Ad-Hoc Networks. The main objectives are to outline the fundamentals of the WLAN technology by highlighting the basic operations of its MAC and PHY layers, and to explain the principles and the characteristics of MANETs. A detailed study on the routing approaches in MANET is then presented, especially the position-based type of them, which forms the basis of the related discussion in chapter 3.

References

[1] Natarajan Meghanathan, Brajesh Kuamr Kaushik, Dhinaharan Nagamalai, *Advances in Networks and Communications Part 2*, Germany: Springer-Verlag, 2011, pp. 243

[2] Beny Bing, *Emerging technologies in wireless LANs: theory, design, and deployment,* New York: Cambridge University Press, 2008, pp. 13

[3] Garret T. Okamoto, *Smart antenna system and wireless LANs*, Norwel (Massahusetts) : Kluwer Academic Publishers,199, pp.18

[4] "Wi-Fi." Encyclopedia Britannica. Encyclopedia Britannica Online, 2011. Web. 25 Jun. 2011. Available: http://www.britannica.com/EBchecked/topic/1473553/Wi-Fi

[5] Bruce Potter, Bob Potter, *802.11 security,* United States of America: O'Reilly & Associated , Inc., 2003, pp. 9-11

[6] Bob O'hara, Al Petrick, *IEEE 802.11 handbook: a designer's companion*", New York, USA: The Institute of Electrical and Electronic Engineering, Inc., 2005, pp. 5-9.

[7] Hangyi Wu, Yi Pan, *Medium Access Control in Wireless Networks*, Nova Science Publishers, Inc, 2008, pp. 300

[8] John Terry, JuhaHeiskala*, OFDM Wireless LANs: A theoretical and practical guide*, United States of America: Sams Publishers, 2002, pp. 231

[9] Dhiman Deb Chowdhury, *High speed LAN technology handbook*, San Jose, USA: Springer-Verlag. 2000, pp. 336

[10] IEEE Computer Society, *IEEE Standard for Information Technology Telecommunication and information exchange between system Local and metropolitan area networks Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, New York: Institute of Electrical and Electronic Engineers, 12 June 2007, pp. 256-267

[11] SamratGangult, SudeeptBhatnagar, *VoIP: wireless, P2P and New Enterprise voice Over IP,* England: John Wiley & Sons Ltd., 2008, pp. 141-142.

[12] Javvin, *Network Dictionary*, USA: Javvin Technologies Inc., 2007, pp. 9

[13] M. Mauve, A. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks". IEEE Network, Vol. 15, No. 6, pp. 30-39, 2001.

[14] I. Stojmenovic, "Position-based routing in ad hoc networks," IEEE Communications Magazine, July 2002.

[15] R. Nelson and L. Kleinrock, “The spatial capacity of a slotted ALOHA multihop packet radio network with capture,” IEEE Transsactions on Communications, vol. 32, no. 6, pp. 684–694, Jun. 1984.

[16] G.G. Finn, “Routing and Addressing Problems in Large Metropolitan-Scale Internetworks”, Research Report ISU/RR-87-180, Inst. For Scientific information, Mar. 1987.

[17] H. Takagi and L. Kleinrock, “Optimal transmission ranges for randomly distributed packet radio terminals,” IEEE Transactions on Communications, vol. 32, no. 3, pp. 246–257, March 1984.

[18] Ting-Chao Hou Victor Li , “Transmission Range Control in Multihop Packet Radio Networks,” IEEE Transactions on Communications, vol. 34, no. 1, pp. 38- 44, Jan 1986.

[19] B. Karp and H. T. Kung, “GPRS: Greedy perimeter stateless routing for wireless networks,” in ACM/IEEE International Conference on Mobile Computing and Networking, pp. 243-254, 2000.

[20] I. Stojmenovic and X. Lin, “Loop- hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 10, pp. 1023 – 1032, October 2001.

[21] I. Stojmenovic and X. Lin, “GEDIR: Loop-free location based routing in wireless networks”, IASTED Int. Conf. on Parallel and Distributed Computing and Systems, pp, 1025-1028, 1999.

[22] W.J Liu, K.T Feng, “Greedy Anti-Void Routing Protocol for Wireless Sensor Networks”, IEEE Communications Letters, vlo. 11, no. 7, pp. 562-564, July 2007.

[23] Wen-Jiunn Liu; Kai-Ten Feng, Largest Forwarding Region Routing Protocol for Mobile Ad Hoc Networks, Proc. IEEE GLOBECOM, pp. 1 – 5, 2006.

[24] E. Kranakis, H. Singh, and J. Urrutia, “Compass routing on geometric networks”, Canadian Conference on Computation Geometry (CCCG), pp. 51-54, 1999.

[25] Young-Bae Ko , Nitin H. Vaidya, “Location-aided routing (LAR) in mobile ad hoc networks”, Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, pp.66-75, 1998.

[26] S. Kalhor, M. Anisi, A.T. Haghighat, “A New Position-Based Routing Protocol for Reducing the Number of Exchanged Route Request Messages in Mobile Ad-hoc Networks”, Proc. ICSNC, pp. 13 – 13, 2007.

[27] Jian Li; Mohapatra, P.v, “LAKER: location aided knowledge extraction routing for mobile ad hoc networks”, IEEE Wireless Communications and Networking Conference (WCNC), pp. 1180 – 1184, 2003.

[28] S. Nanda, R.S Gray, “Multipath location aided routing in 2d and 3d”, IEEE Wireless Communications and Networking Conference (WCNC), pp. 311-317, 2006.

[29] S. Basagni et al., “A Distance Routing Effect Algorithm for Mobility (DREAM)”, Proc. of ACM MOBICOM’98, pp.76-84, 1998.

[30] W.H. Liao, Y.C. Tseng, J.P. Sheu, “GRID: a fully location-aware routing protocols for mobile ad hoc networks”, Telecommunication Systems 18 (1–3), pp. 37–60, 2001.

[31] L. Blazevic et al., ”Self-organization in Mobile Ad Hoc Networks: The Approach of Terminodes,” IEEE Commun.Mag., pp. 166–75, 2001.

[32] Hossein Ashtiani, Shahpour Alirezaee, S. mohsen mir hosseini and Hamid Khosravi, “PNR: New Position based Routing Algorithm for Mobile Ad Hoc Networks“, Proceedings of the World Congress on Engineering, Vol 1, 2009.

[33] Daniel Câmara and Antonio A.F. Loureiro, "A Novel Routing Algorithm for Ad Hoc Networks," Hawaii International Conference on System Sciences, vol. 8, pp.8022 2000.

[34] Kai Zeng Kui Ren, Wenjing Lou, “Geographic On-Demand Disjoint Multipath Routing in Wireless Ad Hoc Networks”, Proc IEEE Military Communications Conference MILCOM, pp. 1-7 , 2005.

[35] E. Kuiper and S. N. Tehrani, “Geographic Routing With Location Service in Intermittely Connected MANETS”, IEEE Transaction on Vehicular Technology, Vol. 60, No. 2, pp. 592-604, 2011.

[36] K.T. Feng, C.H. Hsu and T.E. Lu, “Velocity-Assisted Predictive Mobility and LocationAware Routing Protocols for Mobile Ad Hoc Networks, IEEE Transactions on Vehicular Technology, Vol. 57, No. 1, pp. 448-464, 2008.

[37] V. Namboodiri and L. Gao, “Prediction-Based Routing for Vehicular Ad Hoc Networks”, IEEE Transactions on Vehicular Technology, Vol. 56, No. 4, pp. 2332-2345, 2007.

[38] Bernhard H. Walke, Stefan Mangold and Lars Berlemann, IEEE 802 Wireless Systems, West sussex, England, John Wiley & Sons, 2006. [36] S. Basagni et al., Mobile Ad Hoc Networking, IEEE Press and John Wiley & Sons, 2003.

[39] IEEE Computer Society, *IEEE Standard for Information Technology Telecommunication and information exchange between system Local and metropolitan area networks Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, New York: Institute of Electrical and Electronic Engineers, 12 June 2007, pp. 256-267

[40] SamratGangult, SudeeptBhatnagar, *VoIP: wireless, P2P and New Enterprise voice Over IP,* England: John Wiley & Sons Ltd., 2008, pp. 141-142.

[41] Javvin, *Network Dictionary*, USA: Javvin Technologies Inc., 2007, pp. 9

[42] P. Nicopolitidis, M. S. Obaidat, G. I. Papadimitriou, A. S. Pomportsis, *Wireless Network*, England: Johns Wiley& Sons Ltd.,2003, pp. 56

[43] K. V. K. K. Prasad, *Principles of Digital Communication Systems and Computer Networks,* Hingham, Charles River Media, Inc., 2003, pp. 95

[44] Simon R. Saunders, Alejandro Aragon-Zavala, *Antenna and propagation for wireless communication systems,* England, Wiley & Sons Ltd., 2007, pp. 18

[45] T L Singal, *Wireless Communication*, New Delhi: Tata McGraw Hill, 2010, pp. 260

[46] LajosHanzo, Peter J. Cherriman, JurgenStreit, *Video Compression and Communication From basics to H.261, H.263, H.264, MPEG for DVB and HSDPA-Style Adaptive Turbo-Transceivers*, England: John Wiley & Sons Ltd, 2007, pp. 364

3 Chapter 3

Methods

3.1 Introduction

As we know the Location Detection System (LDS) algorithm is applied on the network to eliminate the hidden terminal problems on the network. The main ideas of the LDS are built on performing time delay between the four different zones of the access point, which will give each zone a different delay form the other zone so that collisions won't happen. But to achieve delay fairness we should apply an algorithm that distribute the time delay in equal intervals. Below we can see how we divide the time delay for each zone:

- $Zone\ 1 = 0 * delay.$
- $Zone\ 2 = 1 * delay.$
- $Zone\ 3 = 2 * delay.$
- $Zone\ 4 = 3 * delay.$

As we see above this mechanism is the one that will be done to split the time intervals between the four zone.

3.2 Explanation of the method acquired (LDS)

In the LDS algorithm the time delay will be divided equally between the four zones as follows:

$$Zone\ 1 = 0 * delay.$$

$$Zone\ 2 = 1 * delay.$$

$$Zone\ 3 = 2 * delay.$$

$$Zone\ 4 = 3 * delay.$$

Figure 3.1 below shows us how the delay will be divided in LDS algorithm.

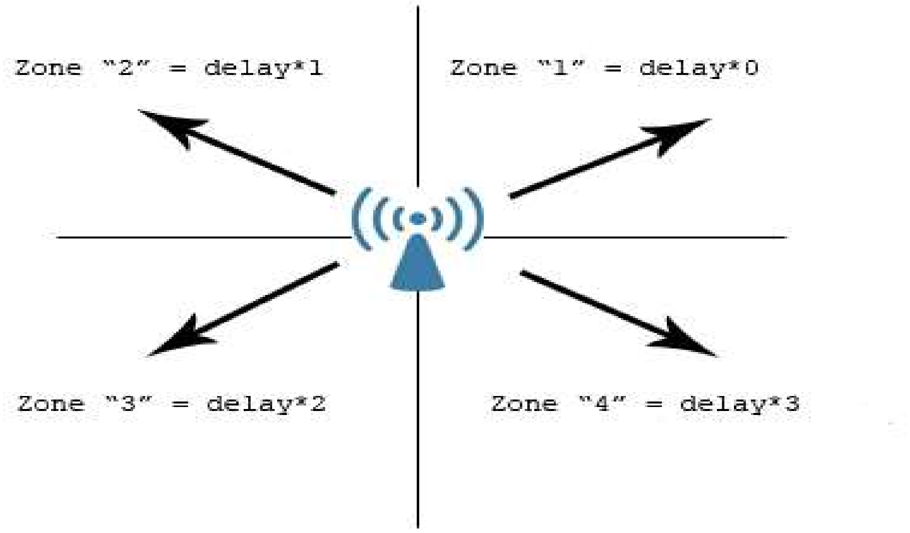


Figure 3-1: delay sequence in LDS algorithm

According to the mechanism, time delay will be changed in a fair way between the four areas so each area will get a fair amount of delay during the simulation. So, as we see the time will be divided into equal intervals between the four different areas.

To do the mechanism for LDS. First, we should assign a fix value which we will name it time-interleaving "α", after a specific time, the time-interleaving "α" will be rearranged. By looking at the example below we can see how the time interleaving works. Let us say that we set up "α" which to be 5 seconds so after every 5 seconds the delay will be rearranged in a different way.

In the first 5 seconds the delay among the four zones will be as follows:

- $Zone\ 1 = 0 * delay$
- $Zone\ 2 = 1 * delay$
- $Zone\ 3 = 2 * delay$
- $Zone\ 4 = 3 * delay$

By looking to the figure 3.2 below shows us the delay during the first five seconds.

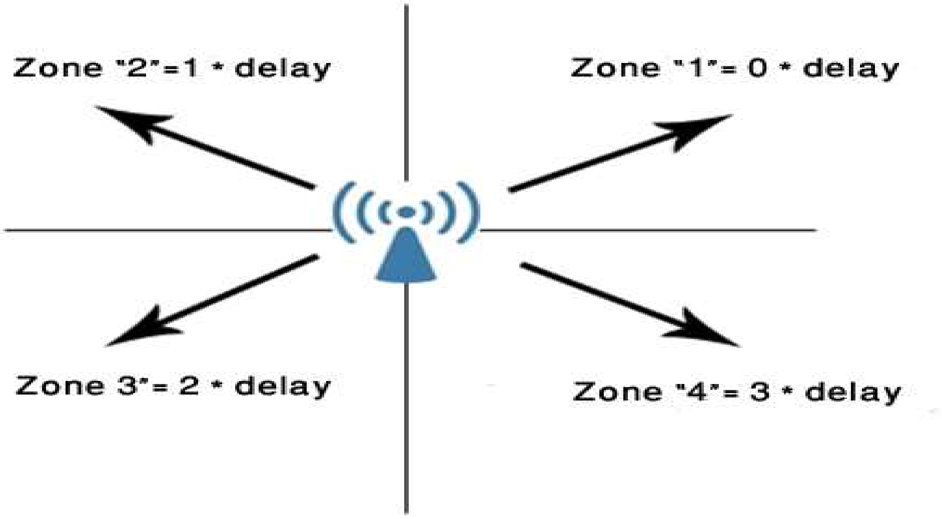


Figure 3-2: delay during first five seconds

Now during the second five seconds the time delay among the four zones will be different, the time interleaving will rearrange the time delay among the four zones in such a way.

- $Zone\ 1 = \ 3 * delay$
- $Zone\ 2 = 0 * delay$
- $Zone\ 3 = 1 * delay$
- $Zone\ 4 = 2 * delay$

Figure 3.3 below shows us how the time delay will be rearranged during the second five seconds:

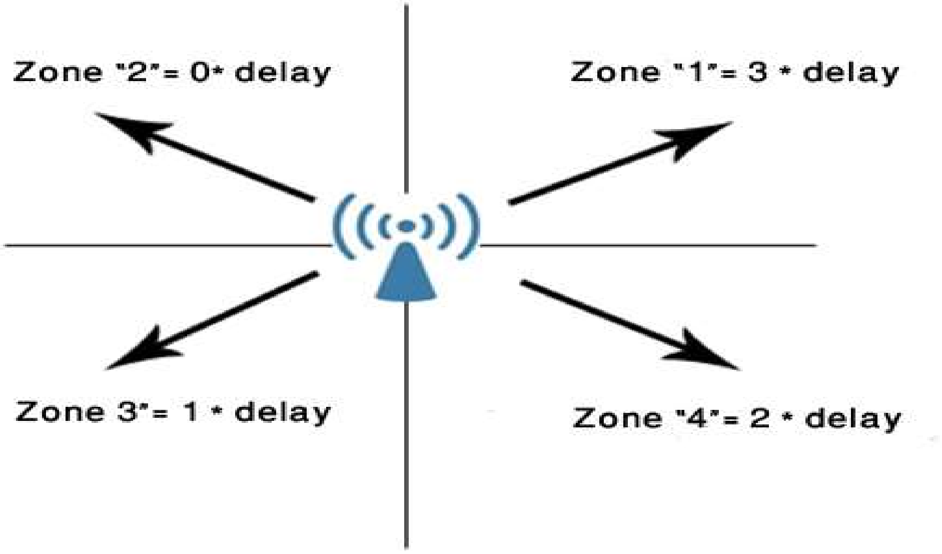


Figure 3-3: the delay during the second five seconds

After finishing from the second five seconds the delay interleaving will move into the third five seconds and change the sequences of the delay and we will get the following:

- $Zone\ 1 = 2 * delay$
- $Zone\ 2 = 3 * delay$
- $Zone\ 3 = 0 * delay$
- $Zone\ 4 =\ 1 * delay$

Figure 3.4 shows us how the time delay will be rearranged during the thrid five seconds:

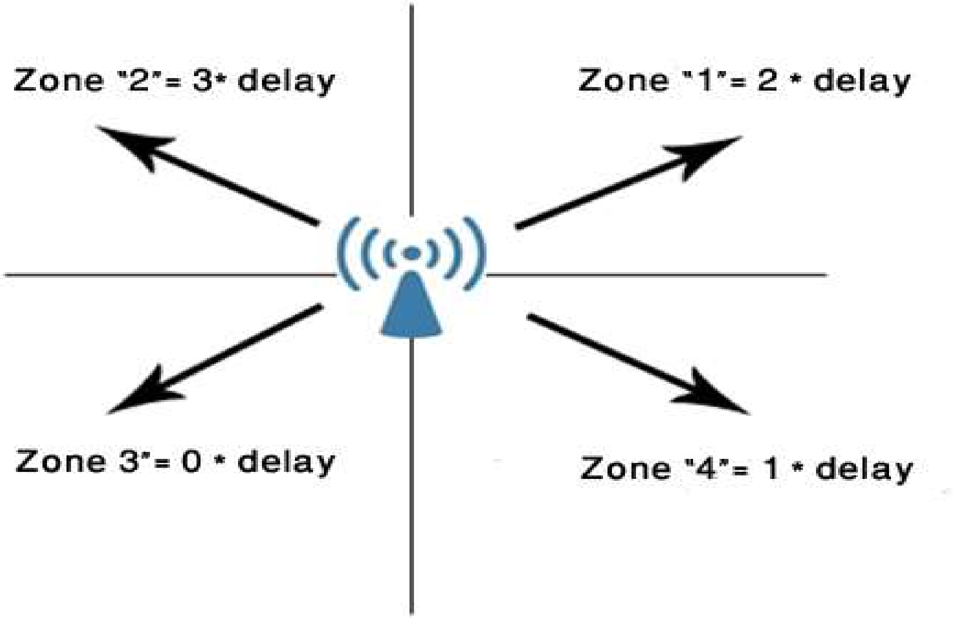


Figure 3-4: the delay during the third five seconds.

Finally in the fourth five seconds, the delay interleaving will rearrange the sequence of delay in another way which will be as follows:

- $Zone\ 1 = 1 * delay$
- $Zone\ 2 = 2 * delay$
- $Zone\ 3 = 3 * delay$
- $Zone\ 4 = 0 * delay$

Figure 3.5 below shows us how the delays are rearranged in the fourth five seconds:

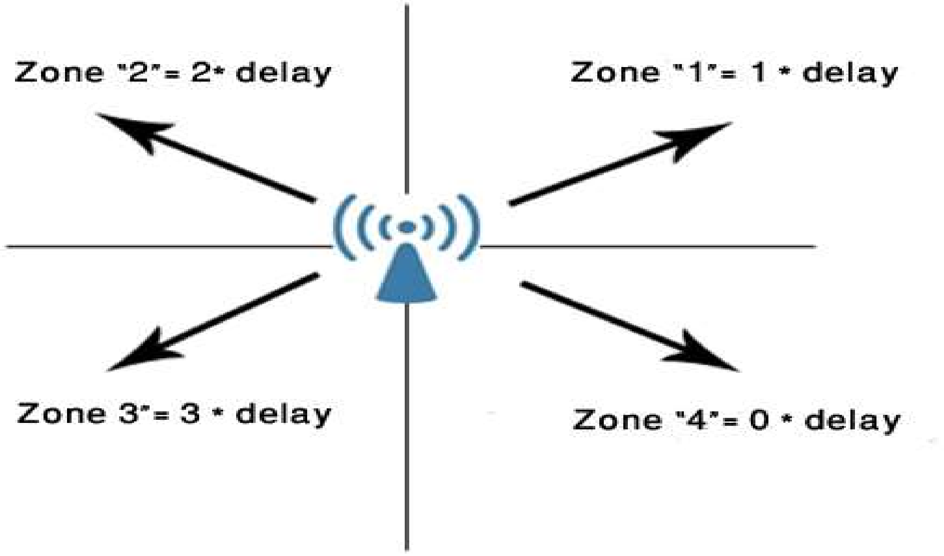


Figure 3-5: the delay in the fourth 5 seconds.

And so on the interleaving will be changing after each 5 seconds so the time-delay will be distrubuted equally among the different areas. If we analize to the time-delay among the areas we can see that each area have been delayed in the same time the other area have been delayed. In another word we can say that the time delay will be equal in each zone. To be more clear we can see the delays which each zone has after 20 seconds:

- $Zone\ 1\ get: \quad 0 * delay, 1 * delay, 2 * delay, and\ 3 * delay$
- $Zone\ 2\ get: \quad 0 * delay, 1 * delay, 2 * delay, and\ 3 * delay$
- $Zone\ 3\ get: \quad 0 * delay, 1 * delay, 2 * delay, and\ 3 * delay$
- $Zone\ 4\ get: \quad 0 * delay, 1 * delay, 2 * delay, and\ 3 * delay$

3.3 Explanation of the LDS using OPNET

3.3.1 *Introduction*

What is OPNET? OPNET is used to design and analyse the network as it functions as a stimulator. It allows us the to compare the stimulated networks, and also to study the modification which we have done to the network. Since our scenarios is focused on the MAC layer, we need a well-known stimulator like OPNET which supports simulation and analysis of MAC layer.

3.3.2 *MAC Layer in OPNET*

In the figure below 3.6 the MAC layer that is used in the OPNET, this MAC layer is the same MAC layer defined in the IEEE 802.11 standard. The MAC layer is formed up by combining many transitions to each other. Below we can see those transitions and the functionality of each one:

- INIT: this initializes the process model which is the MAC layer model, where all of the variables and attributes are load at this transition.
- BSS_INIT: this will initialize the BSS variables like: determining the final MAC address, initialize the address list, get the station type, record the station address of the access point, and many other functions are done inside.
- IDLE: this transition waits the packet to receive from the higher layer or from the lower layer, if the medium is idle for the DIFS time it transmits directly, but if the medium is busy for the DIFS time it goes for the DEFER transition.
- DEFER: this transition waits for the medium to become free, when the medium is free it moves to the other transition state which is the BACK-OFF
- BACK-OFF NEEDED: this state just checks if a back-off is needed or not, if a back-off is needed it moves to the BACK-OFF state and if not, it moves to the transmit state.
- BACK-OFF: this state performs back-off counter to the packet using equation 2.5. When the back-off counter reaches zero it moves to the transmit state, but if the back-off counter was interrupted it returns back to the DEFER state.
- SCAN: this state performs scanning to the medium if there is a station or any access-point around.
- FRM_END: after each frame is sent, this state check for the next state.

As we see these are the function of each state in the MAC layer of 802.11 [1], to be clearer figure 3.6 below shows us the MAC layer in OPNET.

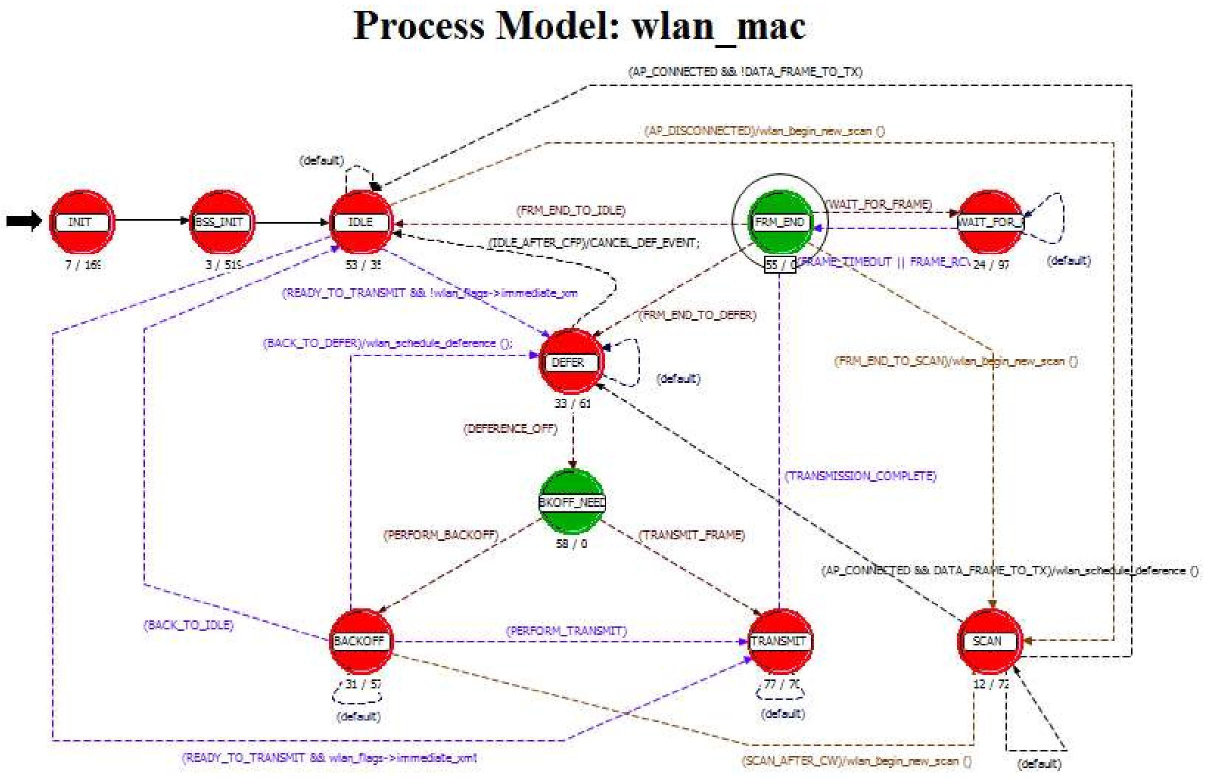

Process Model: wlan_mac
(AP_CONNECTED && !DATA_FRAME_TO_TX)
(AP_DISCONNECTED)/wlan_begin_new_scan ()
(default)
INIT
7 / 169
BSS_INIT
3 / 519
IDLE
53 / 35
(FRM_END_TO_IDLE)
(IDLE_AFTER_CFP)/CANCEL_DEF_EVENT;
FRM_END
55 / 0
(WAIT_FOR_FRAME)
WAIT_FOR_F
24 / 97
(default)
(FRAME_TIMEOUT || FRAME_RCV
(READY_TO_TRANSMIT && !wlan_flags->immediate_xm
(FRM_END_TO_DEFER)
(BACK_TO_DEFER)/wlan_schedule_deference ();
DEFER
(default)
33 / 61
(FRM_END_TO_SCAN)/wlan_begin_new_scan ()
(DEFERENCE_OFF)
(TRANSMISSION_COMPLETE)
BKOFF_NEED
58 / 0
(PERFORM_BACKOFF)
(TRANSMIT_FRAME)
(AP_CONNECTED && DATA_FRAME_TO_TX)/wlan_schedule_deference ()
(BACK_TO_IDLE)
BACKOFF
(PERFORM_TRANSMIT)
TRANSMIT
SCAN
31 / 57
(default)
77 / 70
(default)
12 / 72
(READY_TO_TRANSMIT && wlan_flags->immediate_xmt
(SCAN_AFTER_CW)/wlan_begin_new_scan ()
(default)

Figure 3-6: MAC layer of 802.11 in OPNET [2]

3.3.3 *"LDS" MAC Layer*

Figure 3.7 below shows us the LDS MAC layer which has been modified with algorithm to acquire the locations and divide the time interval between the zones in a fair way

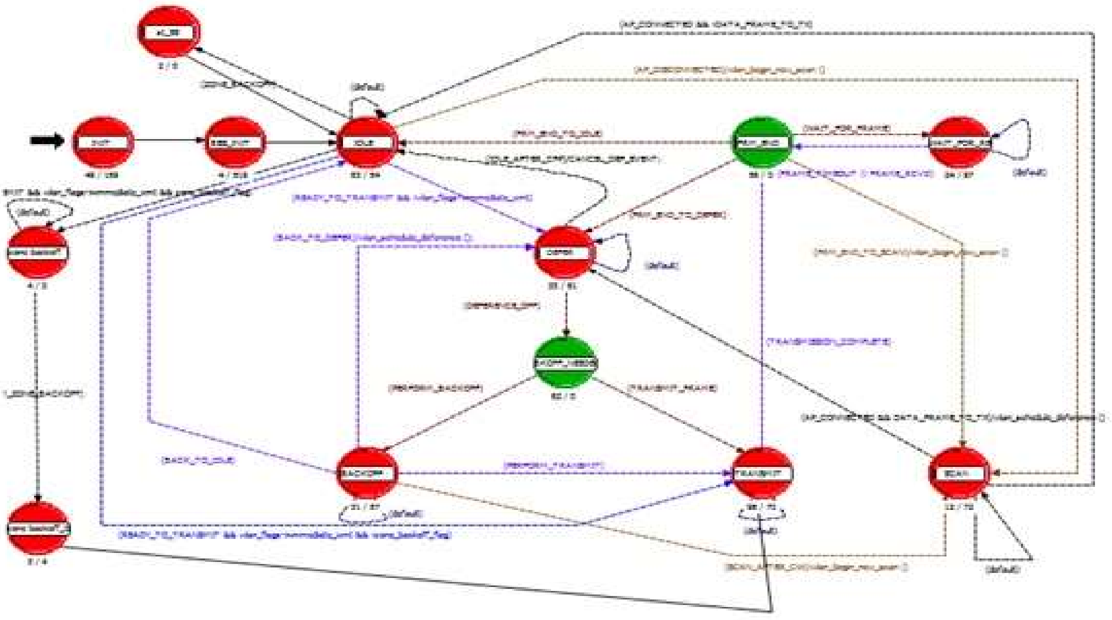


Figure 3-7: MAC layer of LDS algorithm [12].

The MAC layer has been modified to divide the network into 4 areas, and then perform fair zone delay for each area depending on the position of each node.

The modification was to add three transition state and to connect this transition state to the IDLE transition, these transitions are:

- Zone back-off
- Zone back-off
- Delay inter-leaver

The code below shows us how the LDS algorithm obtain the location x, and y. also it shows us how the access point makes a fixed delay on each zone, but this code misses delay fairness between the four zones. Later on, we will see how to avoid the problem of delay unfairness. The code is like the following:

```
op_ima_obj_attr_get(my_node_objid,"x position",& x);
op_ima_obj_attr_get(my_node_objid,"y position",& y);
zone_delay = zone_delay_get(x,y);
if(zone_delay == 0)
        {
        //                  sprintf (str, "zone backoff zero");
        //          op_prg_odb_print_major (str, OPC_NIL);
        zone_backoff_flag = OPC_TRUE;
        }
/* Checking whether zone delay backoff is needed or not.                    */
if((zone_backoff_flag == OPC_FALSE) && (AP == 1))
        {
        //sprintf (str, "zone backoff block :%f",op_sim_time());
        //op_prg_odb_print_major (str, OPC_NIL);
        zone_backoff_flag = OPC_TRUE;
        }
else
        {
        /* end modifications */
        //zone_backoff_flag = OPC_FALSE;
        }
if(zone_delay > 0)////
                        {
                        Requested_time = op_sim_time() + zone_delay;
                        zone_backoff_flag = OPC_TRUE;////
```

```
                }
        else
                zone_backoff_flag = OPC_FALSE;////
```

after identifying the location x, and y for each client in the network whether it's located to zone 1, zone 2, zone 3, or zone 4, and after applying the fixed delay now we should specify an interleaving value "α" which we have discussed before, that applies delay fairness between the four zones.

3.3.4 *"LDS-Dynamic" MAC Layer*

The modification which we applied to the LDS algorithm to provide a dynamic delay (DLA-Dynamic) can be shown in figure 3.8 below:

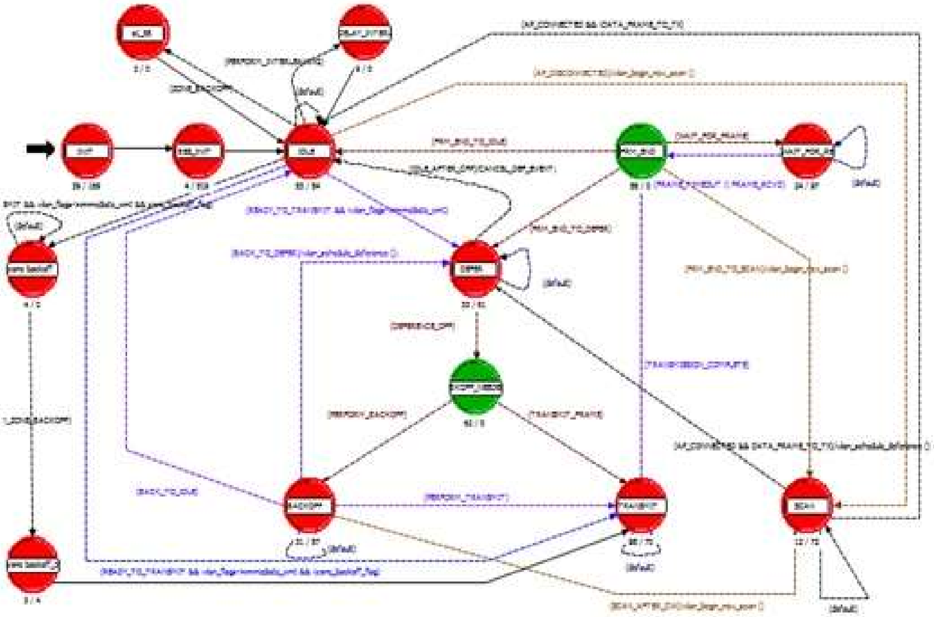


Figure 3-8: LDS-Dynamic MAC-layer

The modification was to setup a transition state and to connect this transition state to the IDLE transition, and specify a dynamic value "α" which we have discussed before. According to this dynamic value "α" it will rearrange the delay of each zone (ex. If "α" was 10 seconds the dynamic delay will rearrange the delay in each zone every 10 seconds) this means that the dynamic delay will occur every 10 seconds to rearrange the delay among the areas. Let us say in the first 10 seconds zone1 was (0* delay), zone2 (1*delay), zone3 (2*delay), and zone4 (3*delay) when the first 10 seconds pass, the interrupter will occur and perform dynamic delay which will change the sequence of the zone delay such that it will become: zone1 (3*delay), zone2 (0*delay), zone3 (1*delay), zone4 (2*delay). The interrupter will continue to change the sequence every 10 seconds. By doing this operation, the delay will keep on changing and this will offer us a dynamic delay (equal delay) between the zones, which will help in eliminating the delay unfairness, and split the delay among the zones in a fair way.

Figure 3.9 below will show us the DYNAMIC-DELAY transition which we have added to the LDS MAC-layer.

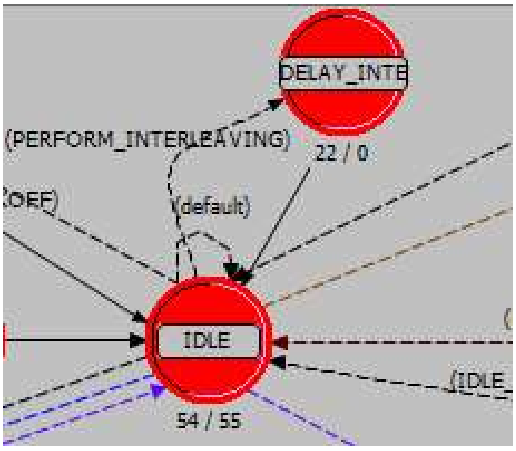


Figure 3-9: DYNAMIC-DELAY

Also code 3.1 below shows us the code which we have written in the DYNAMIC-DELAY.

```
/** Perform DYNAMIC DELAY **/
Interleaver ++;
Interleaver = Interleaver %4;
zone_delay = Interleaver*delta;

        sprintf(str, "New zone delay %f and %d %f", zone_delay, Interleaver, op_sim_time());
        op_sim_message(str,OPC_NIL);

op_intrpt_schedule_self (op_sim_time () + 5, 100);
```

Code 3-1: Dynamic-delay

By analysing this code, we can see how the dynamic-delay of the LDS works:

First, we increase the Dynamic delay, which is an number every time the simulator enters the Dynamic-Delay transition. But this will lead us to a large number of zones, and we only need between "zero" and “3”. In order to avoid this, we modulus the dynamic delay by 4 “%4”, so we will only have from “zero” to “3”.

Then we calculate the zone delay which is the multiplication of the dynamic delay with delta (delta is:$\triangle > TxRx(data\,frame) + TxRx(ACK) + SIFS$ [12]. This will give us the required zone delay which is: “0, 0.0001, 0.0002, and 0.0003”. As we can see, there is also another function”

$$op\_intrpt\_schedule_{self}(op_{sim_{time(\ )}} + 10, 100)”.$$

This function is used to interrupt the simulator every 10 seconds in order to perform Dynamic delay.

3.4 Explanation of the method acquired (DTR MAC)

3.4.1 *Introduction*

DTR MAC is a method that we have created according to the idea of Time Division Multiple access (TDMA). The DTR MAC provides connection to the access point by using the TDMA and Common-Sense Multiple Access Collision Avoidance (CSMA/CA) mechanism in order to solve the problem of the hidden terminal node. According to the method used before LDS used to mitigate the hidden terminal problem but without using time division multiple access between the zones. So, the main idea of the DTR MAC is to split time between zones, in which each zone is giving a specific amount of time to access the medium, and the other zones must stop. The difference between the DTR MAC and the LDS method is that only one zone is allowed to access the medium, while the others should stop for a specific time. Unlike the old methods, where the 4 zones were trying to access the medium at the same time.

3.4.2 *Explanation of DTR-MAC Method*

DTR-MAC is a method formed from the combination of two methods which are: TDMA, and CSMA/CA to eliminate the hidden terminal problem between different zones. The TDMA method divides the coverage area of the access point to 4 zones, and splits the connection between them according to time, where each zone has a specific time to transmit. This time is 0.001 seconds, meaning one zone is access the media for 0.001 seconds while the other zones are not allowed to access the media, waiting for their turn to transmit. While CSMA/CA do access method in each zone. In the following figures we can see how the operation of the DTR-MAC is done. Figure 3.10 below shows us the first zone of the access point that is active.

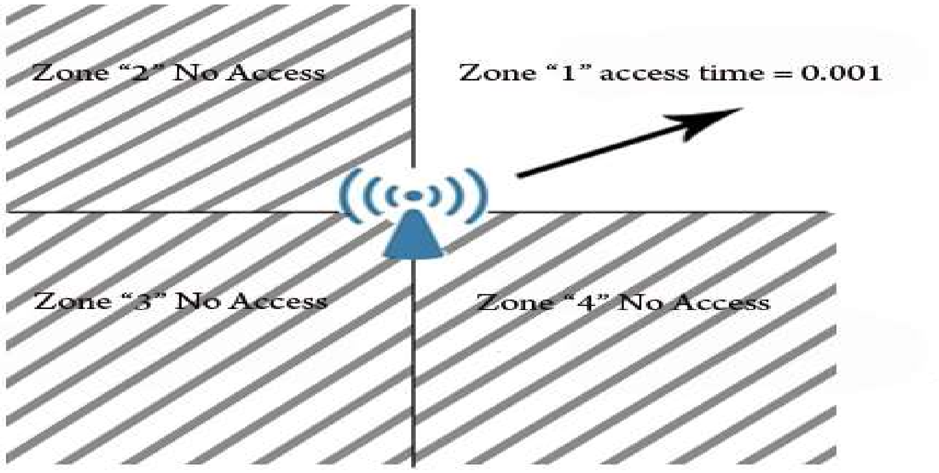


Figure 3-10: Zone 1 access-time

According to figure 3.10 above we can see that zone "1" have an access to the medium which is 0.001 seconds, and all other zones are stopped.

- $Zone$ "1" access time = 0.001 seconds
- $Zone$ "2" access time = No Access
- $Zone$ "3" access time = No Access
- $Zone$ "4" access time = No Access

Figure 3.11 below shows us the access time for zone "2", where it has an access time of 0.001 seconds, and all other zones are not allowed to access the medium.

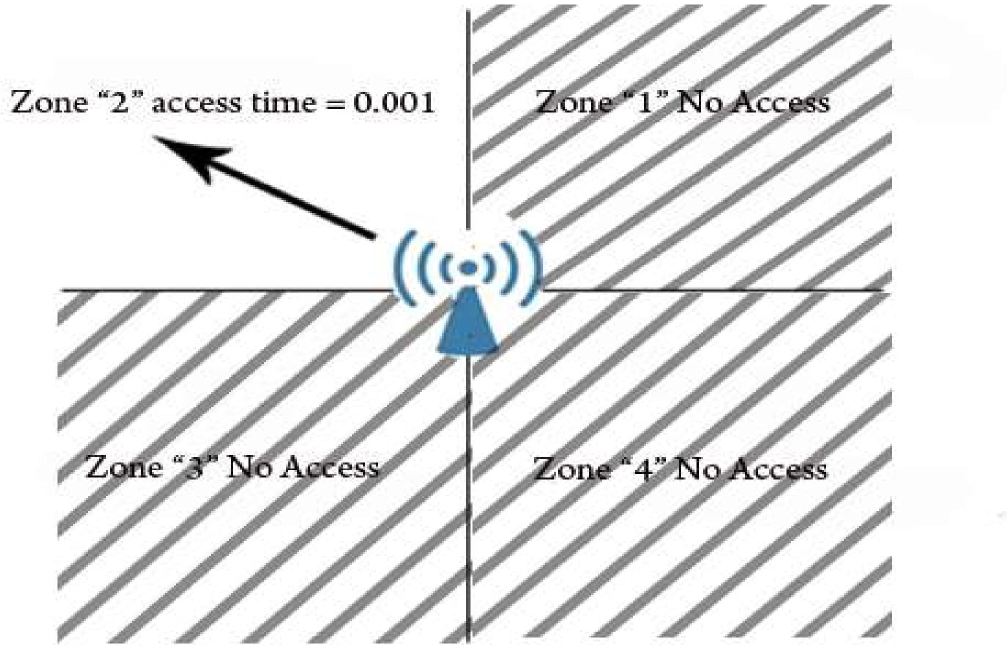


Figure 3-11: Zone 2 access-time

According to figure 3.11 above we can see that the access time has changed from zone ”1” to zone “2”, and now zone “2” has the right to access the medium and all other zones must be stopped.

- $Zone$ "1" access time = No access
- $Zone$ "2" access time = 0.001 second$s$
- $Zone$ "3" access time = No access
- $Zone$ "4" access time = No access

Figure 3.12 below shows us the access time for zone “3”, where it has an access time of 0.001 seconds, and all other zones are not allowed to access the medium.

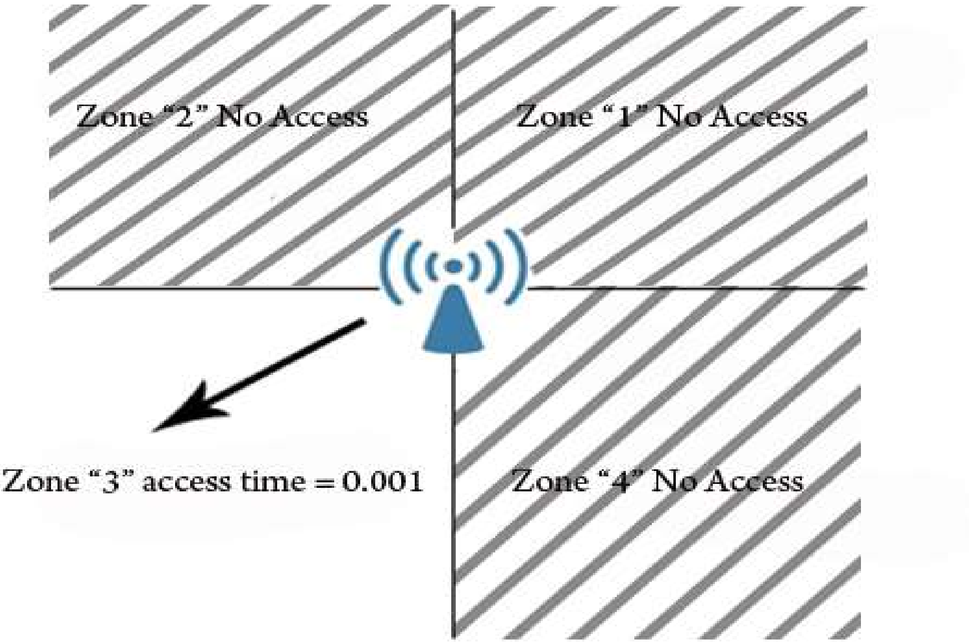


Figure 3-12: Zone 3 access-time

According to figure 3.12 above we can see that the access time has changed from zone ”2” to zone “3”, and now zone “3” has the right to access the medium while all other zones must be stopped.

- $Zone$ "1" access time = No access
- $Zone$ "2" access time = No access
- $Zone$ "3" access time = 0.001 seconds
- $Zone$ "4" access time = No access

Figure 3.13 below shows us the access time for zone “4”, where it has an access time of 0.001 seconds, and all other zones are not allowed to access the medium.

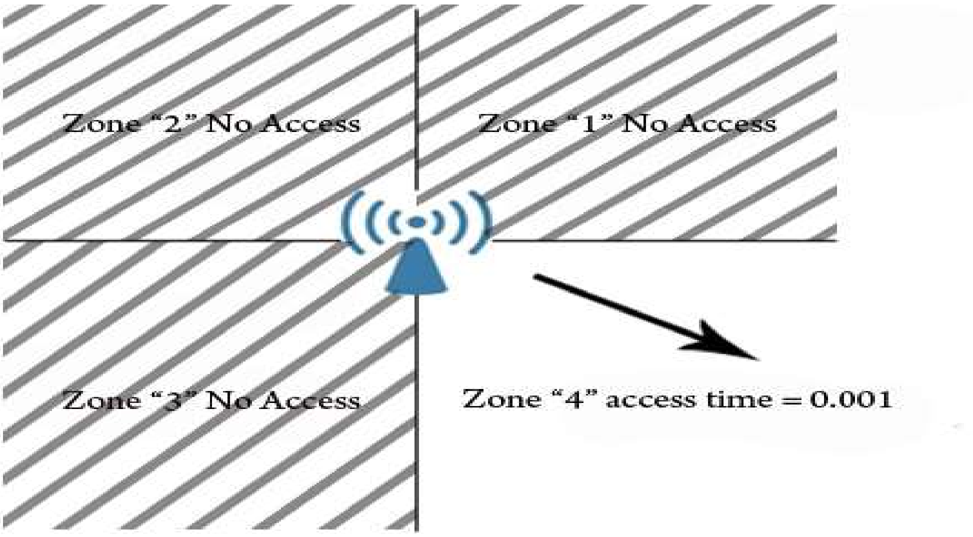


Figure 3-13: Zone 4 access-time

According to figure 3.13 above we can see that the access time has changed from zone ”3” to zone “4”, and now zone “4” has the right to access the medium and all other zones must be stopped.

- $Zone$ "1" access time = No access
- $Zone$ "2" access time = No access
- $Zone$ "3" access time = No access
- $Zone$ "4" access time = 0.001 seconds

3.5 Explanation of the DTR-MAC in OPNET

3.5.1 *Introduction*

In order to test our TDR-MAC method we should use a simulator to see whether this new mechanism will give us a good results or bad results, and what will it effects the network. For this purpose, we will use the OPNET 14.5 simulator which is the best choice. After a lot of long study and experiments, the important part which we should modify in it in order to perform the DTR-MAC is the MAC-INTERFACE.

3.5.2 *MAC-Interface in OPNET*

In figure 3.14 below the MAC-Interface that is used in the OPNET. This MAC-Interface is formed up by combining many transitions to each other. Below we can see those transitions and the functionality of each one:

- INIT: the purpose of this state is to initialize the state variables, and to allow the lower layer which is the MAC-layer to connect to it.
- INIT2: the purpose of this state is to do a self-interrupt in order to wait for a lower layer which is the MAC-layer to finish it job.
- WAIT: the main purpose of this state is to obtain the information of the MAC layer for the local MAC, and also it has another type of functionality.
- IDLE: this state only has two purposes which are interrupts. The types of interrupts which this state does are: stream interrupt from the MAC layer, or from the application layer.
- APPL LAYER ARRIVAL: this state has many purposes but the main purpose of this layer is that it forwards the packet which received from the APPLICATION-layer to the MAC-layer.
- MAC LAYER ARRIVAL: passes the packets that have arrived from the MAC layer [2].

Figure 3.14 below shows us the WLAN MAC INTERFACE.

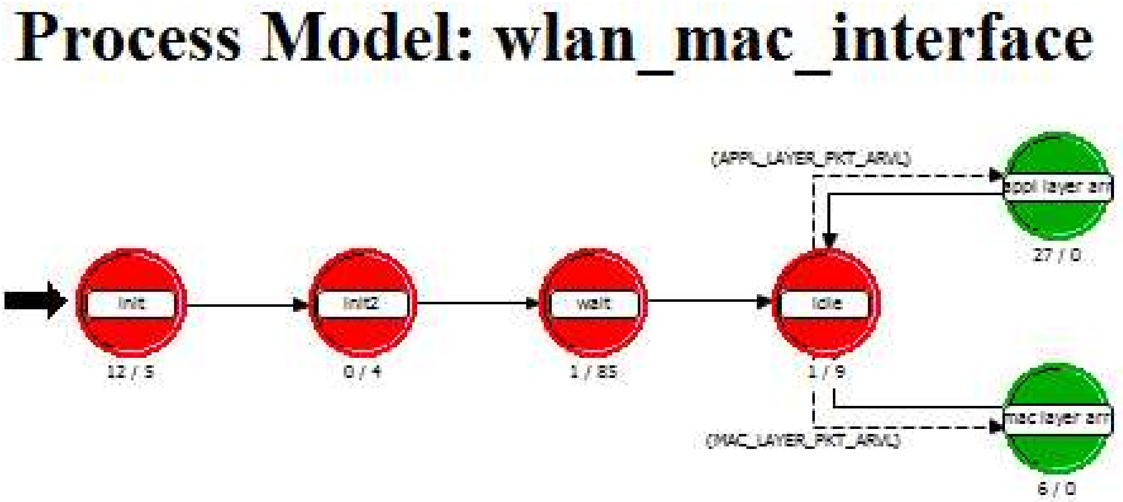


Figure 3-14: WLAN MAC INTERFACE process model [2]

3.5.3 *MAC-Interface for DTR-MAC*

The DTR-MAC INTERFACE was modified to apply time division multiple access on the four areas of the access-point, where the access time of each area is fixed and equal to 0.001 seconds.

The modification which we applied to the MAC-INTERFACE to provide time division multiple access which can be shown in figure 3.15 below:

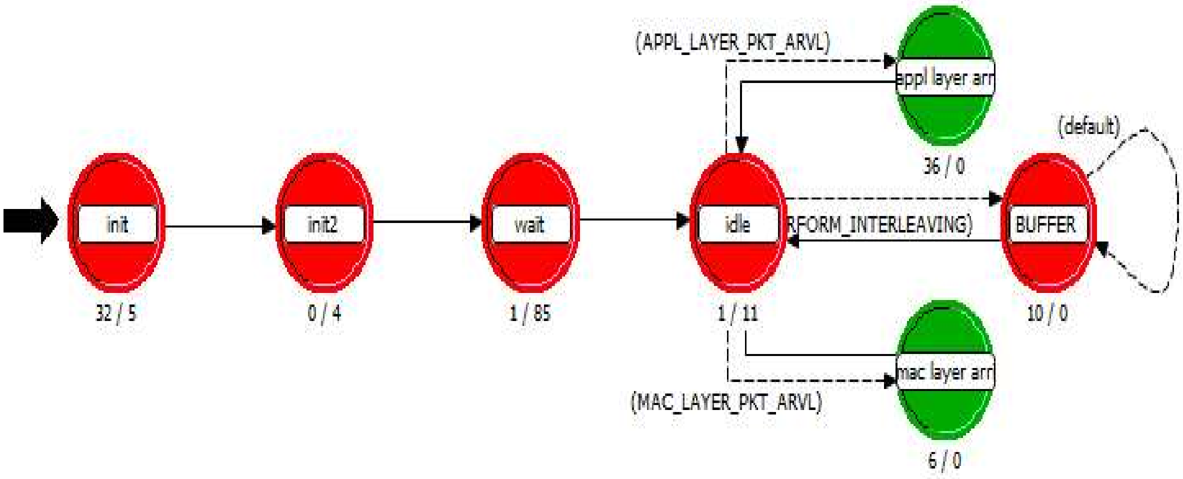


Figure 3-15: TDMA MAC INTERFACE

The modification to the MAC-INTERFACE was done by creating a new transition state, which we named it BUFFER. The idea under this BUFFER state is to manage the work of the DTR by letting each packet that does not eligible to the active zone to stand by there till its turn comes. But what is the difference between active zone and current zone? Active Zone is the zone that have access to the medium for 0.001 seconds, while the Current Zone is the zone that have access to the medium at this moment. This scenario will show us how the operation of current zone and active zone is done. (Ex. Let us say that current zone is zone 1 which is sending, but the active zone is "2" which has the right to use the media. That means zone 1 will not be allowed to transmit since they are not the same zone. But if the current zone was zone 2 then it is allowed to transmit since they are equal). But how can we distinguish which zone is the current zone and which zone is active zone? To know the current zone and the active zone the following algorithm was added to the MAC-

INTERFACE in order to identify the current zone and the active zone. The code 3.2 below shows us the code written to identify the current and the active zone.

```
Inttdm_wlan_zone(double x, double y)
        {
        //char str[100];
        FIN(tdm_wlan_zone(<args>));
        if((x >AP_x) && (x <AP_x + RADIO_RANGE))
                {
                if((y >AP_y) && (y <AP_y + RADIO_RANGE))
                {
                FRET(0);
                }
                else if((y >AP_y - RADIO_RANGE) && (y <AP_y))
                {
                FRET(3);                        }
                }
        if((x >AP_x - RADIO_RANGE) && (x <AP_x))
                {
                if((y >AP_y) && (y <AP_y + 97))
                {
                FRET(1);
                }
                else if((y >AP_y - RADIO_RANGE) && (y <AP_y))
                {
                FRET(2);
                }
                }
        FRET(-1);
                }
```

Code 3-2: Code to identify the current and the active zone

After identifying the current zone and the active zone, we should divide the time between the four different zones in order for each zone to have an access time of 0.001 seconds. To do that code 3.3 shows us how to divide the time between the zones.

```
static void tdm_change_active_zone()
        {
        FIN(tdm_change_active_zone());
        if(op_sim_time() >= start_time + 0.01)
                {
                Interleaver ++;
                active_zone = Interleaver %4;
                start_time = op_sim_time();
                }
        FOUT;
        }
```

Code 3-3: Code to divide the time among the zones

Now, after we have known which zone is the active zone and which zone is the current zone, and after we have done the time division between the four zones, we should do of the buffer. The main purpose of buffer is to compare the active zone with the current zone. If the active zone is the same as the current zone, then there is no need to buffer. The packets are sent directly to the MAC LAYER, but if the active zone is not the same as the current zone, this means that the buffer must prevent the packet until the active zone and the current zone are the same. The importance of this buffer is to allow the packets that have been prevented from transmitted to wait in it. Figure 3.16 and code 3.4 shows us the buffer state and the code needed to do buffering.

(default)
BUFFER
10 / 0

Figure 3-16: Buffer state

Code 3.4 below shows us the code written to do the buffering and the comparison between the active zone and the current zone.

```
tdm_change_active_zone();

if((current_zone == active_zone)&&(prg_list_size(buffer) >0))
        {
        while (op_sim_time() <start_time + 0.001)
                {
                pk_to_mac_send();
                }
        }
op_intrpt_schedule_self (op_sim_time() + 0.001, 100);
```

Code 3-4: code to perform buffering and comparison

After applying all of these changes to the MAC INTERFACE we have implemented the TDMA-MAC in order to divide the time access between the zones.

3.6 Explanation of the Method used to change the DTR Access-Time

3.6.1 *Introduction*

As we know the main idea of the DTR algorithm is about switching the access-time between the 4 areas of the access point, and this is done by giving each area a specific access-time. The default access-time time was 0.001 seconds based on mathematical and experimental studies. But what will happen if we slightly changed the access-time? By increasing or decreasing the access time, network performance may change. To know the effect of changing the access-time we have tested different access-times which are: 0.1 seconds, 0.001seconds, and 0.00001 seconds. These times were simulated using OPNET.

3.6.2 *Changing the Access-Time of DTR*

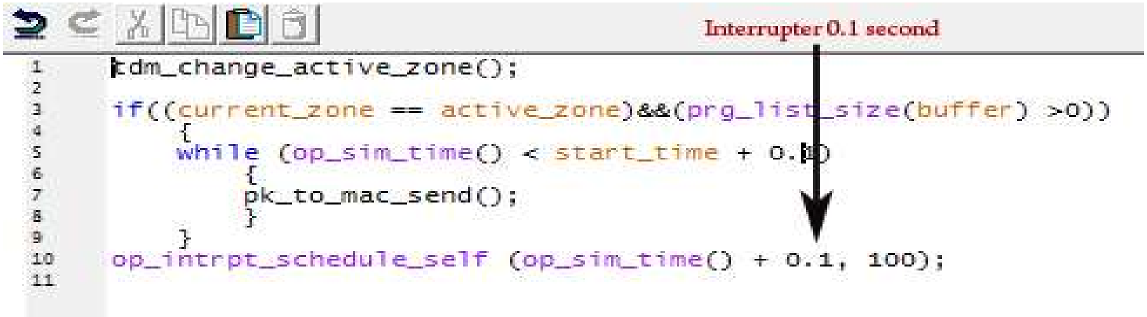
An interrupter which is a cut-off is required in order to apply changes to the DTR access times, but what is an interrupter? An interrupter is a function in OPNET used to interrupt (time, packet stream). In our case we need to interrupt time in order to perform DTR. In the function below 3.2 shows us how to write an interrupter.

$(op\_intrpt\_schedule\_self\ (op\_sim\_time\ ()\ 0.001, 100)$ 3.2

This function works as follow: it interrupts the zones for 0.001 seconds which is the time for access. This means that when an interrupter occurs it will prevent three zones from transmitting for 0.001 seconds allowing only one zone to transmit. In the scenarios where we implemented the interrupter, the time used were as follows: "0.1, 0.001, and 0.00001". We can see the functions below:

- $op\_intrpt\_schedule\_self\ (op\_sim\_time\ ()\ +\ 0.1, 100)$ 3.3
- $op\_intrpt\_schedule\_self\ (op\_sim\_time\ ()\ +\ 0.001, 100)$ 3.4
- $op\_intrpt\_schedule\_self\ (op\_sim\_time\ ()\ +\ 0.00001, 100)$ 3.5

As can be seen above in these functions, each one we should change the time to that which we want the interrupter to perform. Figure 3.17 and 3.18 show a sample of the two access-times. Since all of the three are the same, we need only to change the interrupter time. Figure 3.17 shows the interrupter set to 0.1 seconds:



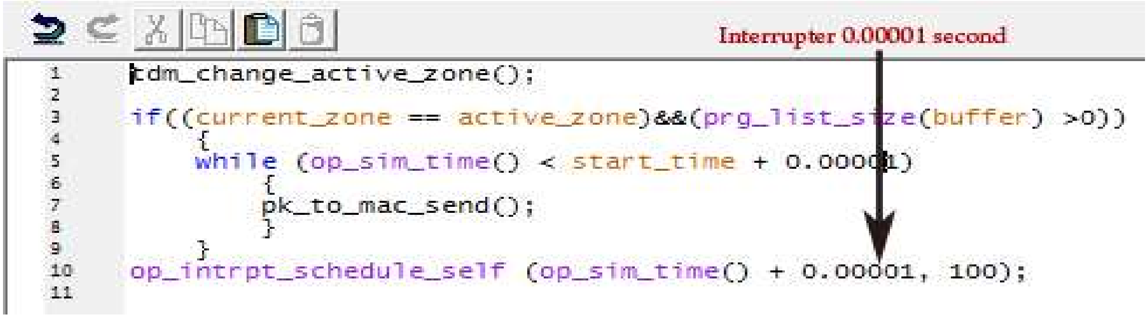
Interrupter 0.1 second

```
tdm_change_active_zone();

if((current_zone == active_zone)&&(prg_list_size(buffer) >0))
    {
    while (op_sim_time() < start_time + 0.1)
        {
        pk_to_mac_send();
        }
    }
op_intrpt_schedule_self (op_sim_time() + 0.1, 100);
```

Figure 3-17: interrupter for 0.1 second

And figure 3.18 below shows the interrupter set to 0.00001 seconds:



Interrupter 0.00001 second

```
tdm_change_active_zone();

if((current_zone == active_zone)&&(prg_list_size(buffer) >0))
    {
    while (op_sim_time() < start_time + 0.00001)
        {
        pk_to_mac_send();
        }
    }
op_intrpt_schedule_self (op_sim_time() + 0.00001, 100);
```

Figure 3-18: interrupter for 0.00001 second.

3.7 Explanation of the Method used to change the DTR Access-Time based on number of nodes (DTR+).

3.7.1 *Introduction*

As we know the main idea of the DTR algorithm is about switching the access-time between the 4 areas of the access point, and this is done by giving each area a specific access-time. But let us suppose that in each area the number of nodes or clients are different, let us say zone 1 has 10 clients connected, zone 2 has 20 clients connected, zone 3 has 30 clients connected, and zone 4 has 40 clients connected. This is unfair to give each zone the same amount of time the time should be dynamic, as there is a greater number of nodes the zone should has more time to access. So, by increasing or decreasing the number of nodes, network performance may change. To know the effect of changing the access-time we have tested different node number with different access-time which are:

0.1 seconds for 40 nodes.

0.001seconds for 30 nodes.

0.00001 seconds for 20 nodes.

0.000001 seconds for 10 nodes.

These times, and nodes were simulated using OPNET

3.7.2 *Changing the Access-Time of DTR based on the number of nodes (DTR+)*

A counter which is used required in order to count the number of clients connected in each zone. but what is a Counter? A Counter is a function in OPNET used to count (nodes). In our case we need a counter in order to perform DTR based on the number of nodes. In the function below 3.5 shows us how to write a counter.

$(op\_count\_detect\ (op\_count\_time\ ()\ 40, 0.1)$ 3.2

```
/** Perform delay interleaving **/
Interleaver ++;
Interleaver = Interleaver %4;
zone_delay = Interleaver*delta;
op_intrpt_schedule_self (op_sim_time () + 5, 100);
/** Perform node check **/
int counter;
op_count_detect (op_count_time, () counter, interleaver);
if (current_zone1 > current_zone2 < current_zone3 < current_zone4)
        return curent_zone1 = start_time 0.1;
                  current_zone2 = start_time 0.001;
                  current_zone3 = start_time 0.0001;
                  current_zone4 = start_time 0.00001;
end;
```

Code 3-5: code to perform DTR+

This function works as follow: it counters of the zones detects that the number of nodes in current zone is greater than the one before or after it will assign it the highest access time which is 0.1. This means that when an interrupter occurs it will prevent three zones from transmitting for 0.1 seconds allowing only one zone to transmit. In the scenarios where we implemented the counter, the nodes were used as follows: “40, 50, and 60”. We can see the functions below:

- $(op\_count\_detect\ (op\_count\_time\ ()\ 40, 0.1)$ 3.3
- $(op\_count\_detect\ (op\_count\_time\ ()\ 50, 0.1)$ 3.4
- $(op\_count\_detect\ (op\_count\_time\ ()\ 60, 0.1)$ 3.5

As can be seen above in these functions, each one we should give the highest access time to that which we want the counter to perform.

4 Chapter 4

Results of Dynamic LDS

4.1 Introduction

LDS has been developed to eliminate the hidden terminal problem that is caused by two stations outside the range of each other trying to send information. As we know, the concept of the LDS algorithm involves dividing the area of the access-point into 4 areas, and then delaying each area for a specific time of delta seconds. This time should not be less than the transmission, and the receiving time of the data frame, plus the transmission and the receiving time of the acknowledgement frame, plus the SIFS time. In equation 4.1 we can see the equation of time delay “delta” [12]:

$$\triangle > TxRx(data\,frame) + TxRx(ACK) + SIFS \qquad 4.1$$

As we have mentioned the time is divided between the 4 zones and each node should know the geographical position which also should give in position to the other clients beside it. The main idea of LDS is combining the time division to the carrier sense.

In this chapter we are going to view the results of the LDS interleaving algorithm, and compare it with the original results of the standard 802.11 MAC. We use commercial software which is OPNET®, to modify and stimulate the network in order to obtain the results of the LDS algorithms.

For this purpose, the design of the network which on we need to perform the simulation and the experiment can be seen in figure 4.1 below. It was configured to have 12 nodes, and one access-point. All of these nodes were to have the same configuration: start time = constant (1), ON state-time = constant (1), OFF state time = constant (0). Packet generation was configured as follows: inter-arrival time = constant (0.1), Packet size (1024). The interleaving time was chosen to be 5 seconds as default meaning that each five seconds, the time delay between the 4 different areas will be rearranged. Most important was the distance between the different 12 nodes which we have placed; they needed to be at a distance out of range of each other so that the hidden terminal problem could be presented. We set the range of transmission of each node at 97 meters, and we placed each node at a distance less than 97 meters from the access point and more than 97 meters from the nodes of other zones, so as to ensure that the distance between any two nodes is more than 97 meters. With respect to the wireless configuration, it was as follows: Physical characteristic: Extended Rate PHY (802.11 g), Data Rate 24 mbps, transmit power = 0.001 (W), packet reception-power thresholds = -80.

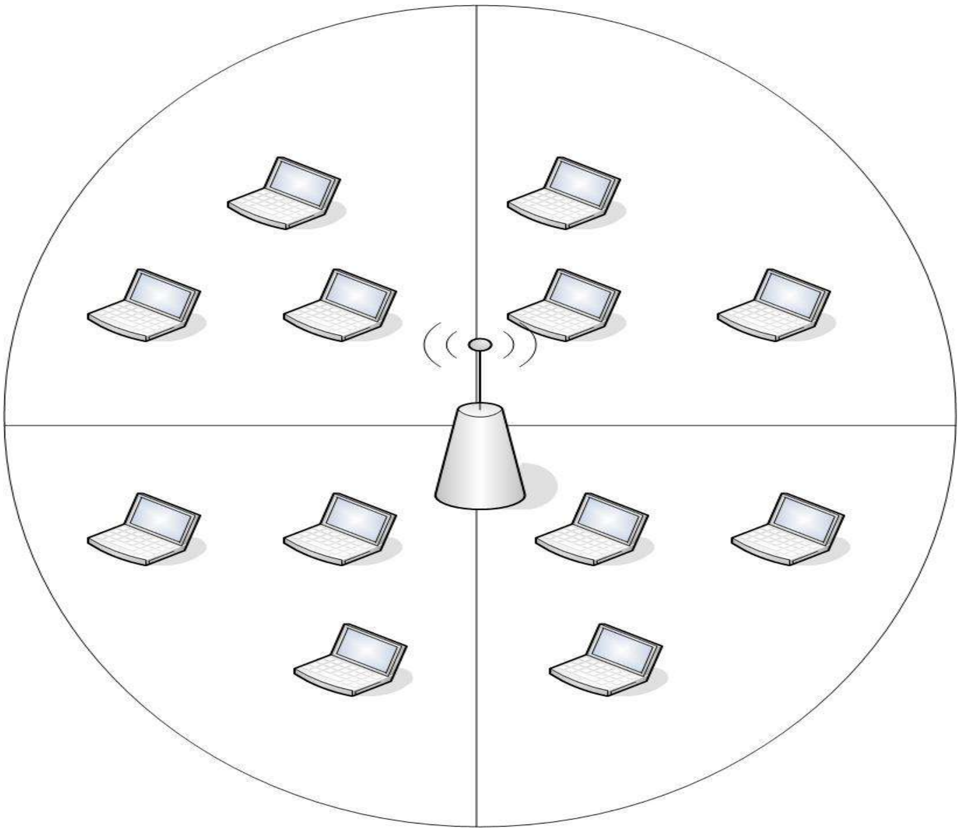


Figure 4-1: design of the network

4.2 Results of LDS compared with Results of Original Mac 802.11.

4.2.1 *Data Dropped*

Figure 4.2 below shows us the data dropped for both of the LDS and the Original 802.11 MAC with respect to the number of nodes, the blue line represents the data dropped for the LDS, and the red line represents the data dropped for the Original 802.11 MAC.

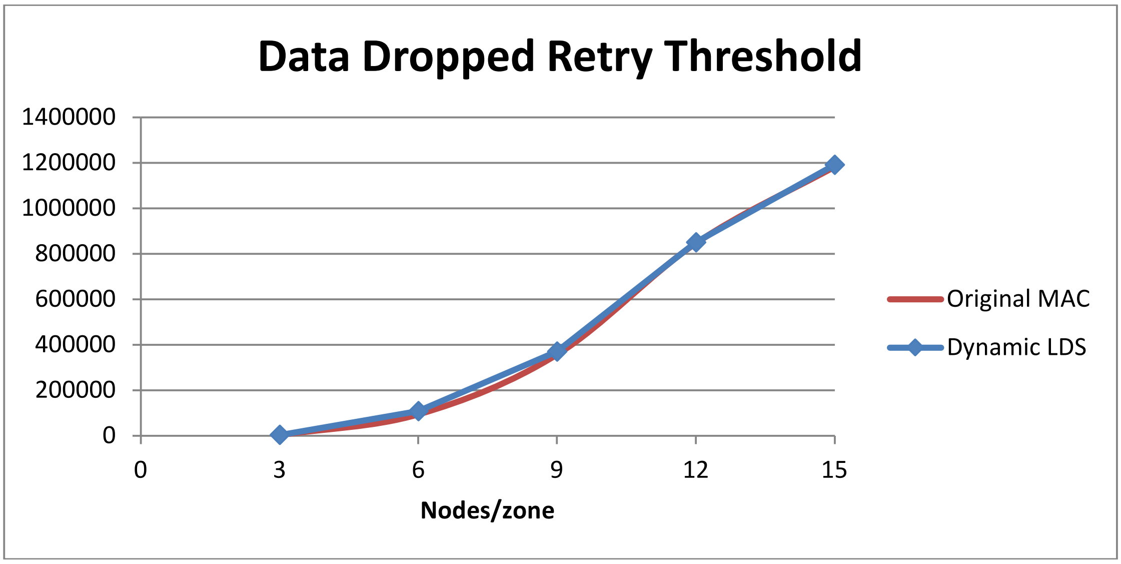


Figure 4-2: data dropped for LDS and Original 802.11 MAC

4.2.2 *Delay*

Figure 4.3 below shows us the delay for both the LDS and the Original 802.11 MAC with respect to the number of nodes. The red line represents the delay for the Original 802.11 MAC, and the blue line represents the delay for the LDS;

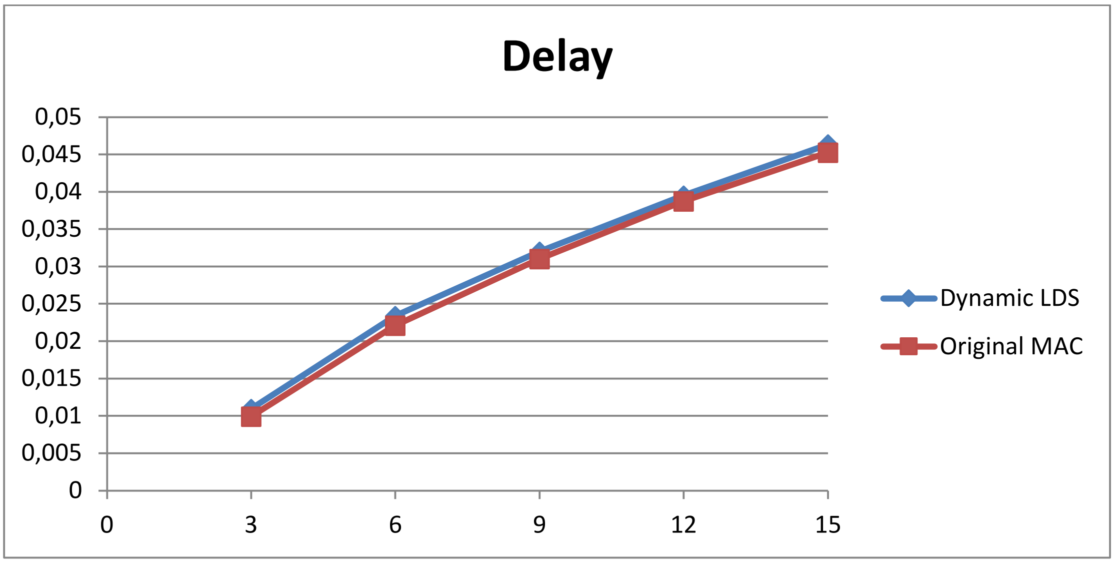


Figure 4-3: Delay for LDS and Original 802.11 MAC

4.2.3 *Media Access Delay*

Figure 4.4 below shows us the Media Access Delay for both of the LDS and the Original 802.11 MAC with respect to the number of nodes. The blue line represents the media access delay for the LDS, and the red line represents the media access delay for the Original 802.11 MAC:

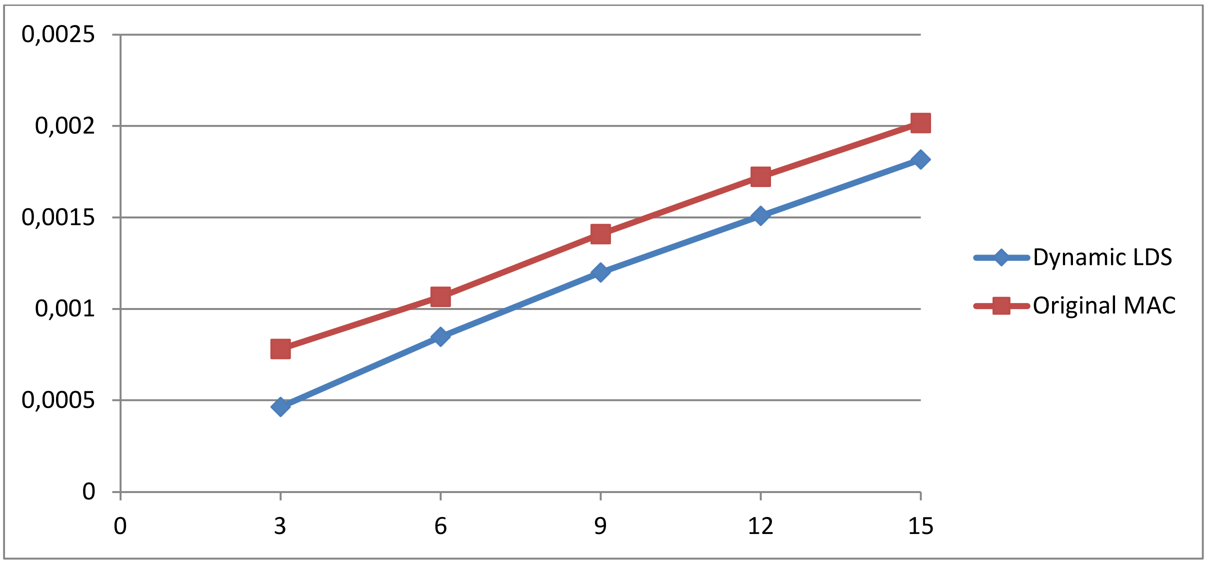


Figure 4-4: Media Access Delay for LDS and Original 802.11 MAC.

4.2.4 *Network Load*

Figure 4.5 below shows us the Network Load for both of the LDS and the Original 802.11 MAC with respect to the number of nodes. The blue line represents the network load for the LDS, and the red line represents the network load for the Original 802.11 MAC:

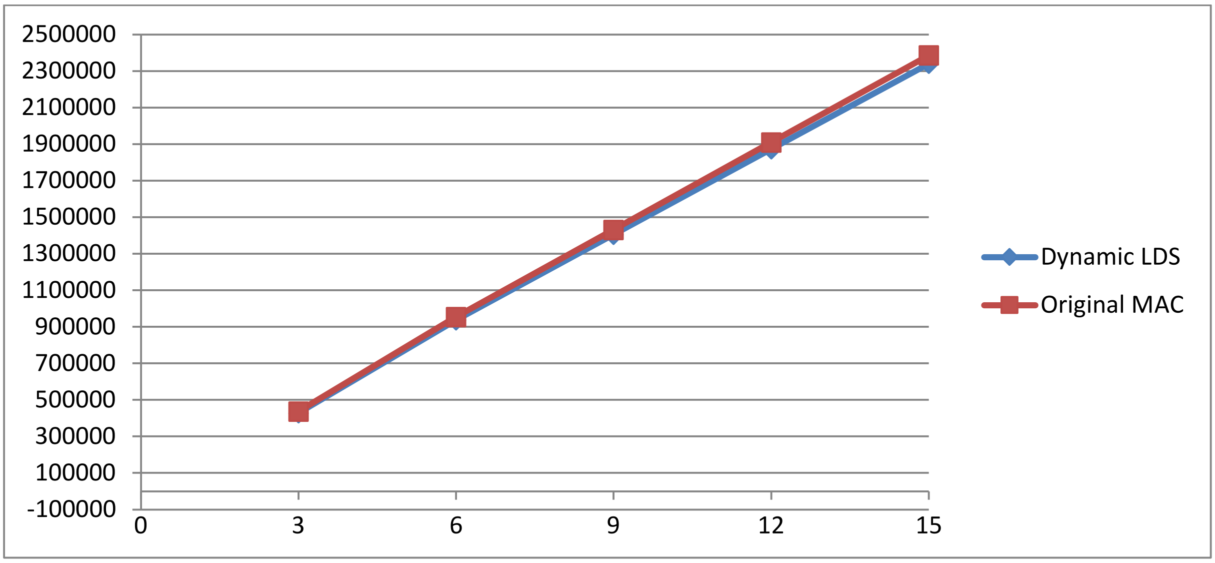


Figure 4-5: Network load for LDS and Original 802.11 MAC.

4.2.5 *Retransmission*

Figure 4.6 below shows us the Retransmission for both of the LDS and the Original 802.11 MAC with respect to the number of nodes. The blue line represents the retransmission for the LDS, and the red line represents the retransmission for the Original 802.11 MAC:

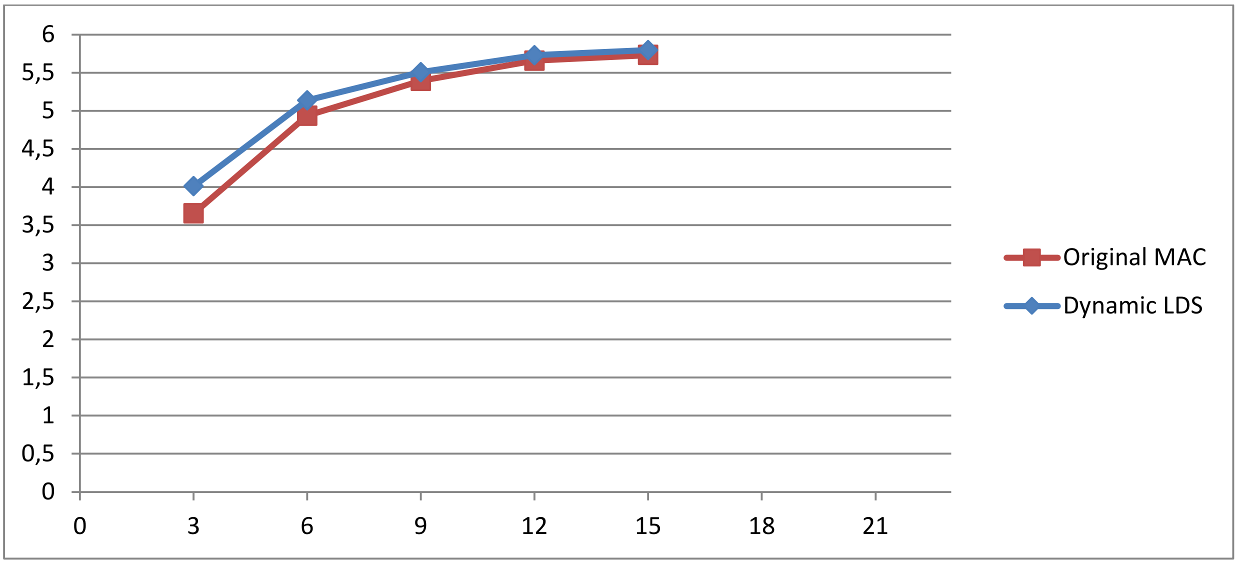


Figure 4-6: Retransmission for LDS and Original 802.11 MAC for node (3 and 15)

4.2.6 *Throughput*

Figure 4.7 below shows us the Throughput for both of the LDS and the Original 802.11 MAC with respect to the number of nodes. The blue line represents the throughput for the LDS and the Red line represents the throughput for the Original 802.11 MAC:

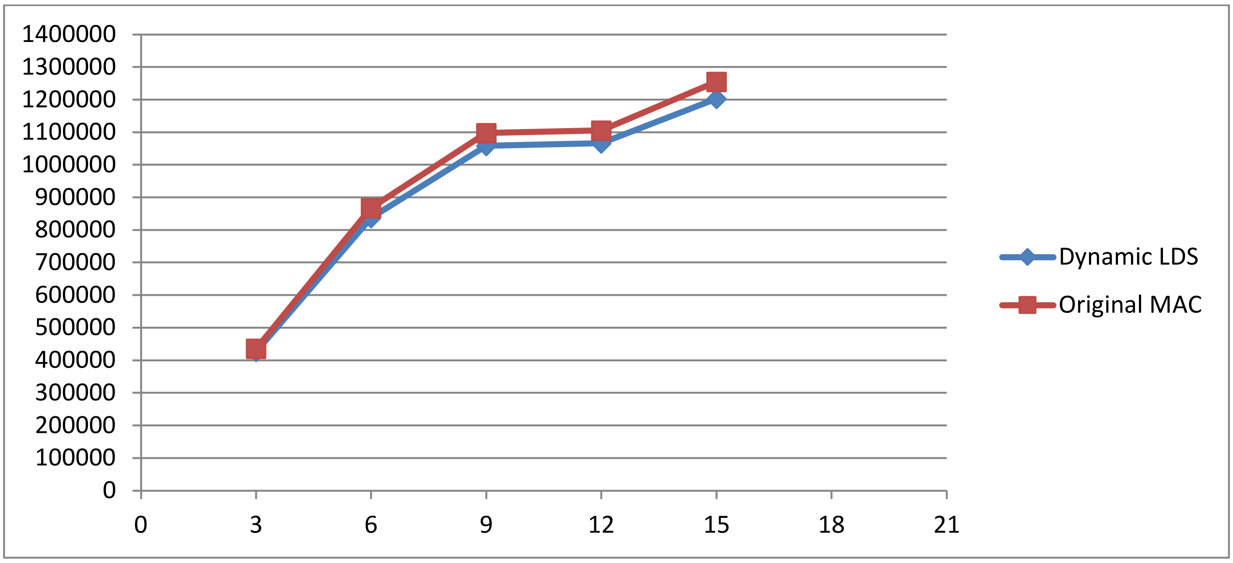


Figure 4-7: Throughput for Original 802.11 MAC and LDS.

4.3 Results of DTR

DTR is another way that is similar to LDS and RTS/CTS that have been developed to eliminate the hidden terminal problem that is caused by station that are outside the range of other station trying to send information to the receiver. To get the results of DTR we going to use the OPNET simulator, the following network has been built and set up with the following parameters: 12 stations and one access-point (Note*: Number of nodes differs in each scenario). All of these nodes have the same configuration: start time = constant (1), ON state-time = constant (1), OFF state time = constant (0). Packet generation was configured as follows: inter-arrival time = constant (0.1), Packet size (1024). The access time was chosen to be 0.001 seconds as default depending on mathematical calculation, meaning that each 0.001 seconds only one zone is allowed to access the medium. Also, we should be careful for placing the nodes inside the zones since we need to place them in a situation that included a hidden node. To do this we have defined the radio range of the node and the access point 97 meters, and placed the nodes in each zone at a distance less than 97 meters from the access point and more than 97 meters from the nodes in the other zones. In that way we could present a hidden terminal problem. With respect to the wireless configuration, it is as follows: Physical characteristic: Extended Rate PHY (802.11 g), Data Rate 24 mbps, transmit power = 0.001 (W), packet reception-power thresholds = -80. In figure 4.8 below shows us the network:

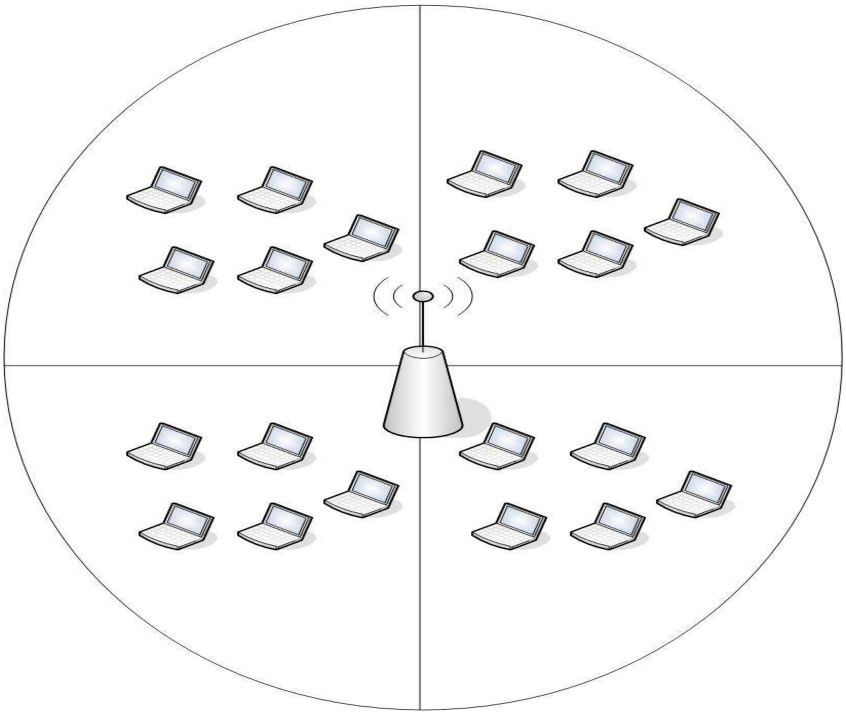


Figure 4-8: Network for DTR

4.4 Results of DTR compared with Results of Original MAC

4.4.1 *Data Dropped*

Figure 4.9 below shows us the data dropped for both of the DTR and MAC with respect to the number of nodes. The blue line represents the data dropped for the MAC, while the red line represents the data dropped for the DTR:

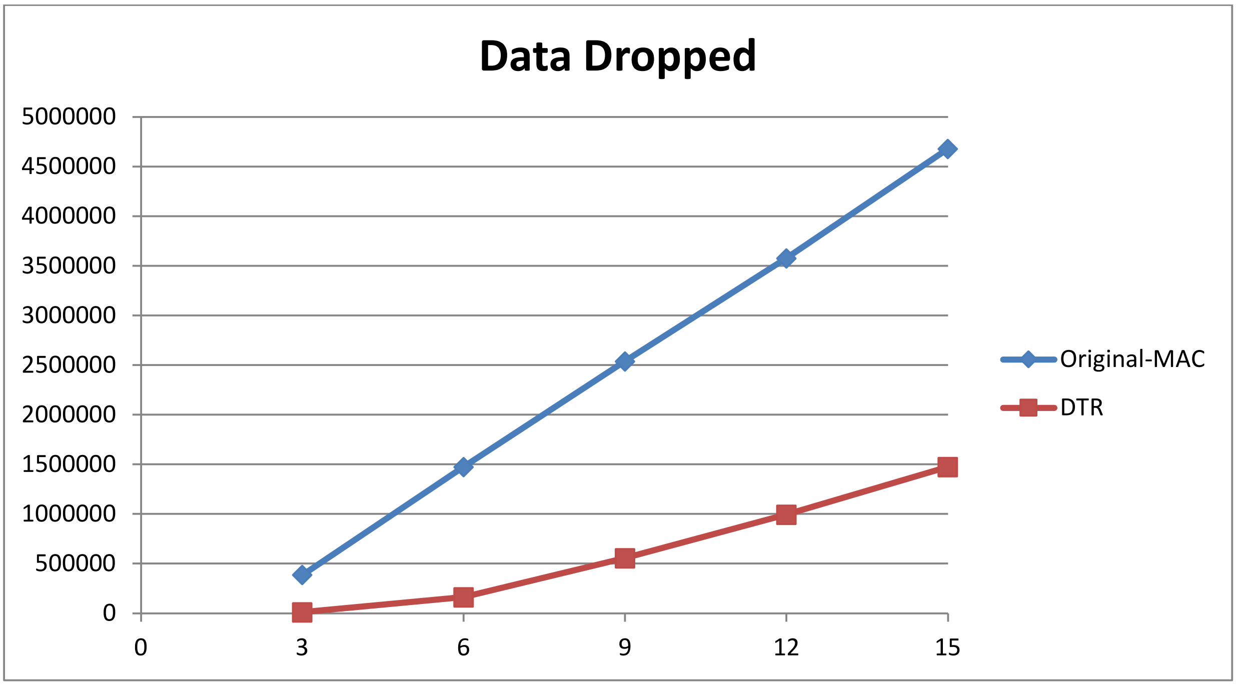


Figure 4-9: Data Dropped for DTR and MAC

4.4.2 *Delay*

Figure 4.10 below shows us the delay for both of the DTR and MAC with respect to the number of nodes. The blue line represents the delay for the MAC, while the red line represents the delay for the DTR:

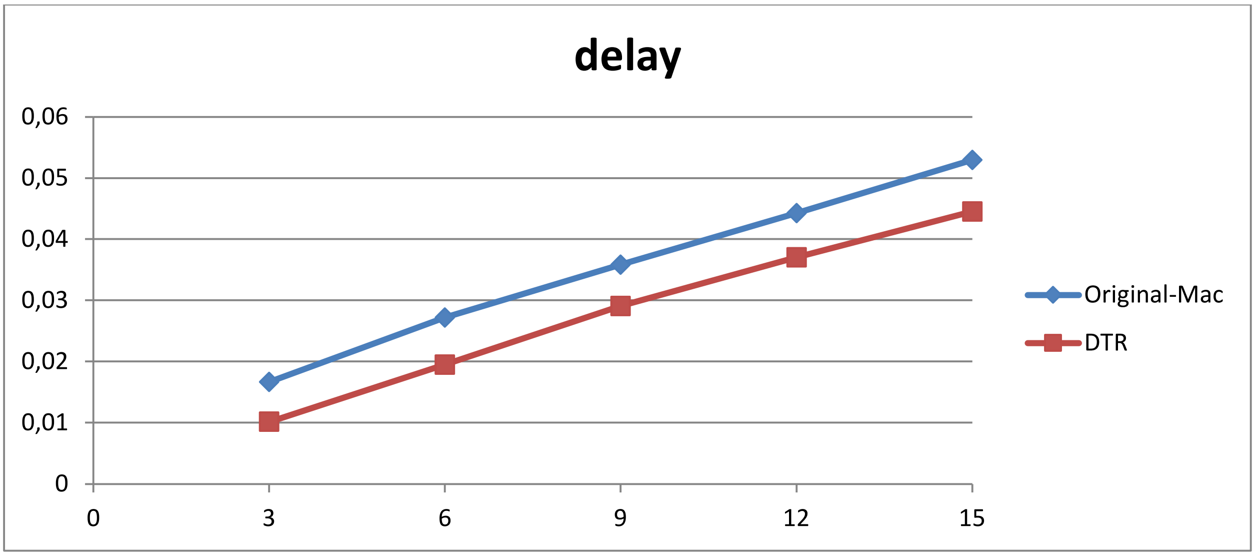


Figure 4-10: Delay for DTR and MAC

4.4.3 *Network Load*

Figure 4.11 below shows us the Network load for both of the DTR and MAC with respect to the number of nodes. The blue line represents the network load for the MAC, while the red line represents the network load for the DTR:

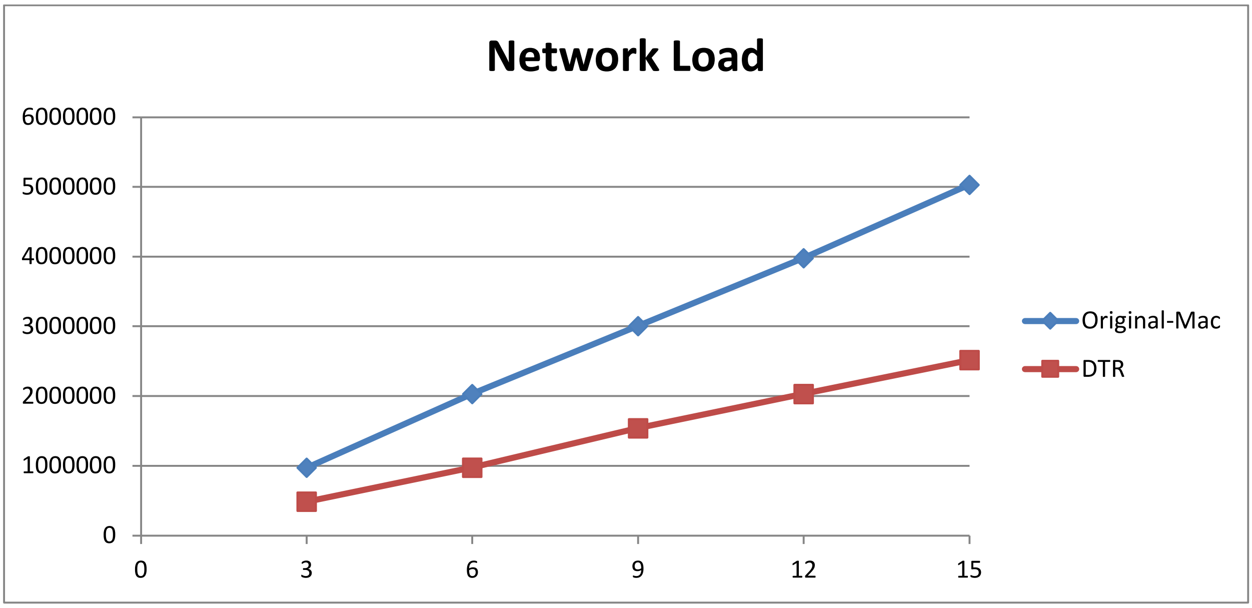


Figure 4-11: Network load for DTR and MAC

4.4.4 *Retransmission*

Figure 4.12 below shows us the retransmission for both of the DTR and MAC with respect to the number of nodes. The blue line represents the retransmission for the MAC, while the red line represents the retransmission for the DTR:

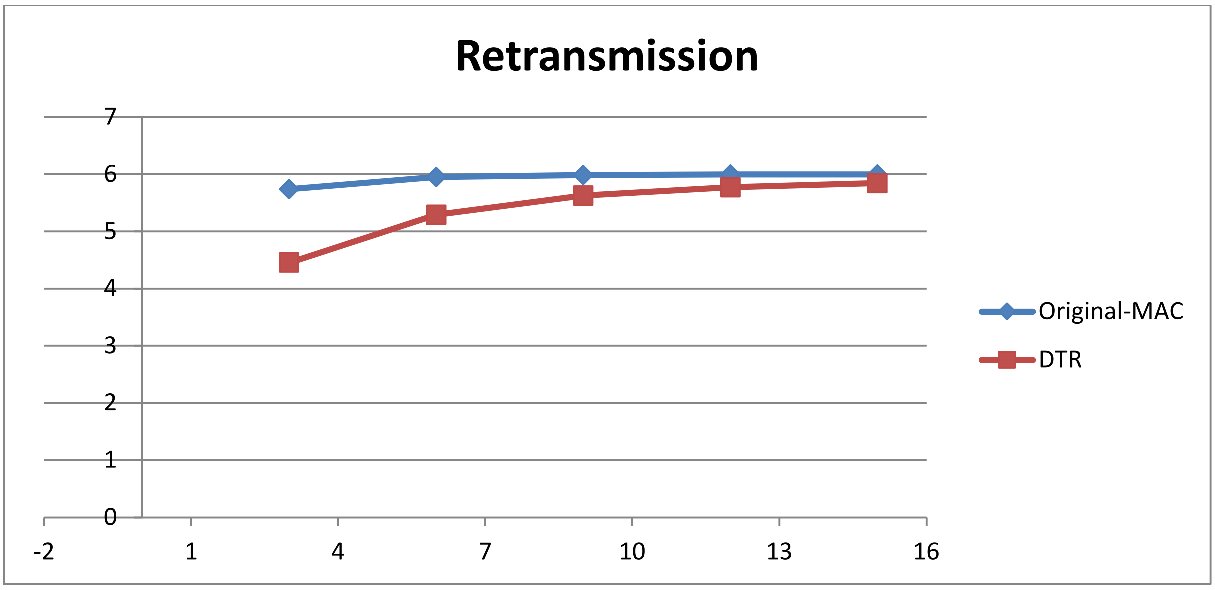


Figure 4-12: Retransmission for DTR and MAC

4.4.5 *Throughput*

Figure 4.13 below shows us the throughput for both of the DTR and MAC with respect to the number of nodes. The blue line represents the throughput for the MAC, while the red line represents the throughput for the DTR:

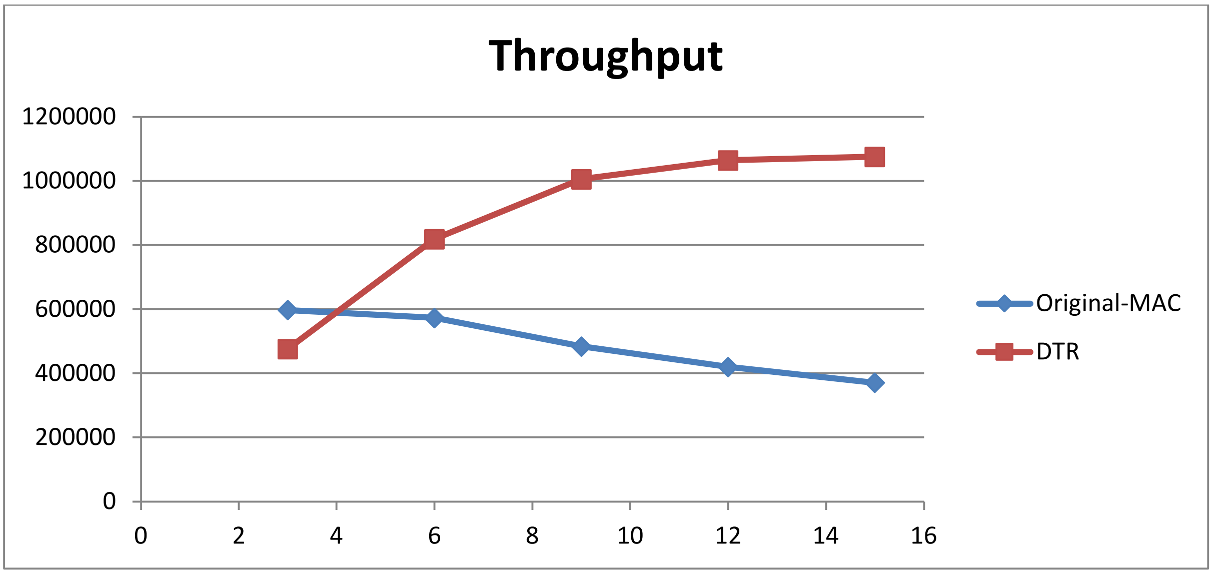


Figure 4-13: throughput for DTR and MAC

4.4.6 *Collision*

Figure 4.14 below shows us the collision for both of the DTR and MAC with respect to the number of nodes (3 and 6 nodes). The blue line represents the collision for the MAC, while the red line represents the collision for the DTR:

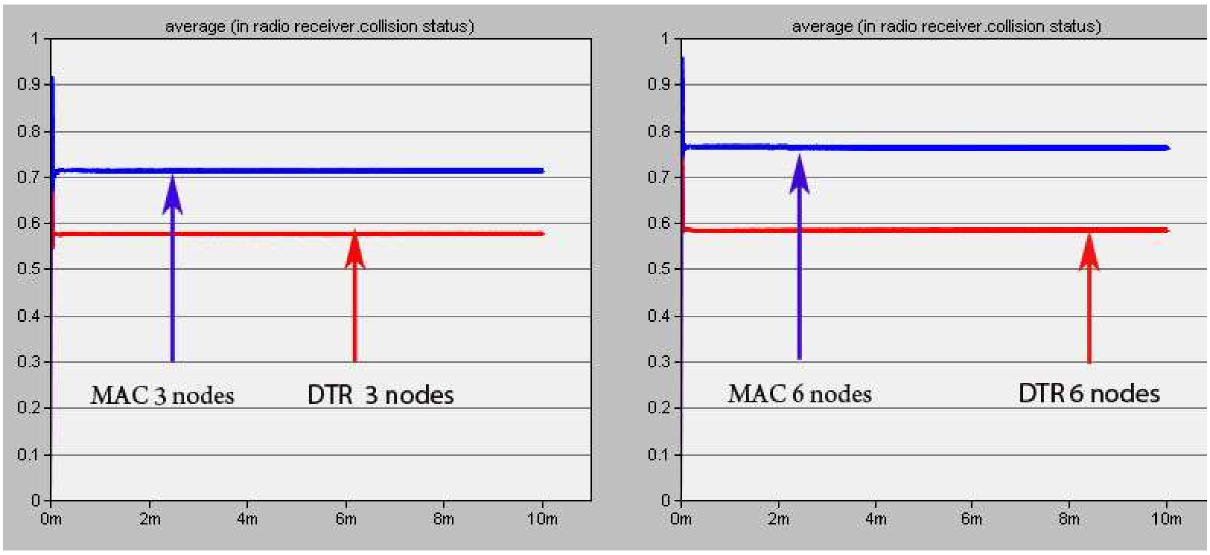


Figure 4-14: Collision for DTR and MAC.

Figure 4.15 below we can see the collision for both of the DTR and MAC with respect to the number of nodes (12 and 15 nodes). The blue line represents the collision for the MAC, while the Red line represents the collision for the DTR:

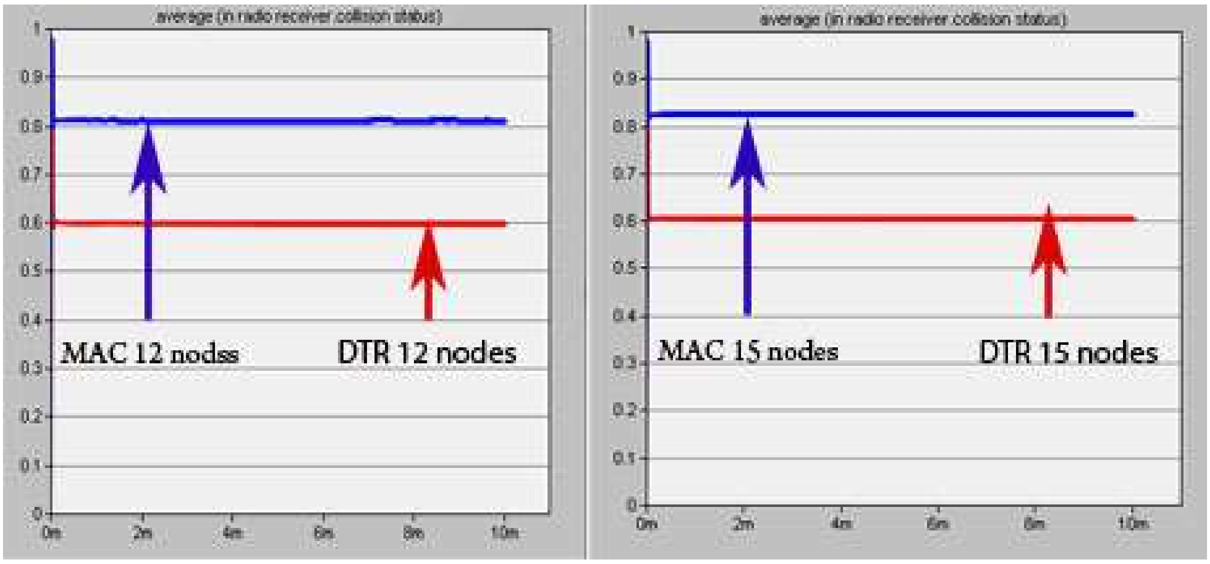


Figure 4-15: Collision for DTR and MAC.

4.5 Results of changing the "Access-Time" for DTR

4.5.1 *Introduction*

In this part we are going to see the results of changing the time for the "Access-Time". The default value was 0.001 seconds based on mathematical and experimental studies. The access-time is set as follows: 0.1, 0.001, and 0.00001 seconds.

4.5.2 *Data Dropped (Retry Threshold Exceeded)*

Figure 4.16 below shows us the data dropped for changing the Access-Time which is (0.1, 0.001, and 0.00001 seconds) with respect to the number of nodes:

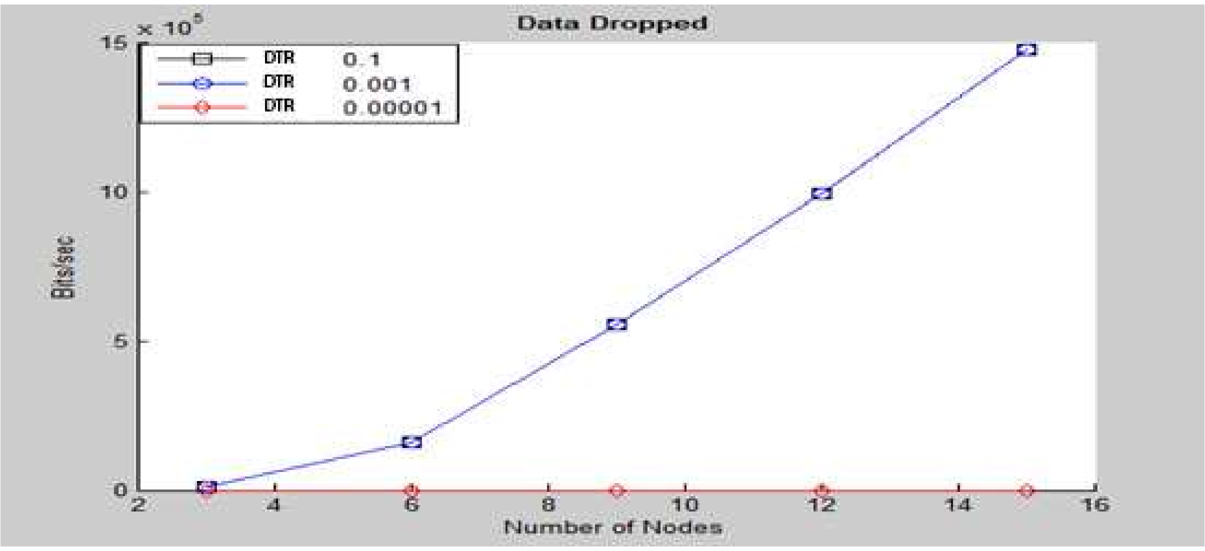


Figure 4-16: data dropped

4.5.3 *Delay*

Figure 4.17 below shows us the delay for changing the Access-Time which is (0.1 seconds, 0.001 seconds, 0.00001 seconds) with respect to the number of nodes:

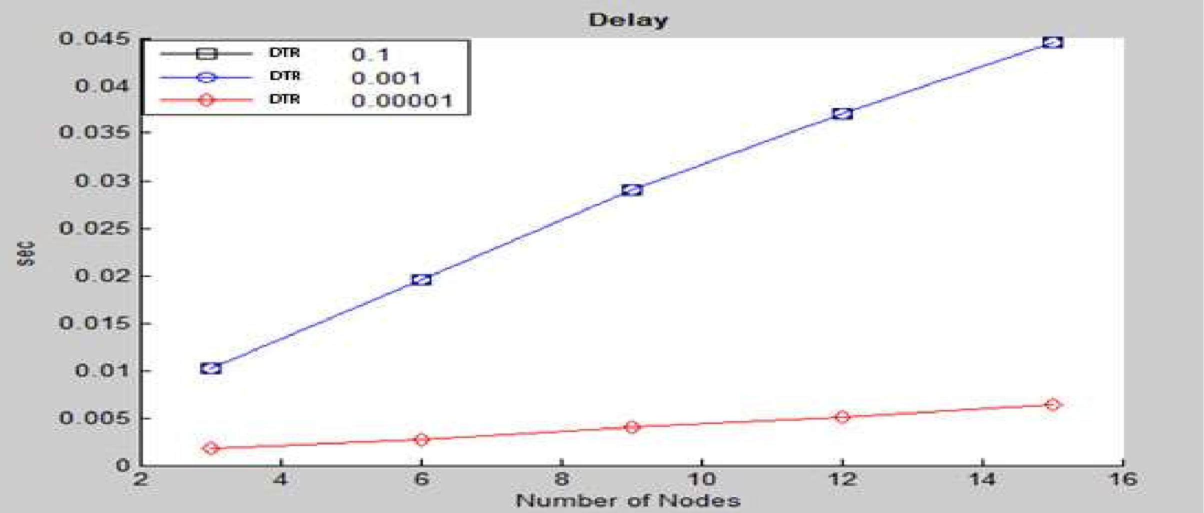


Figure 4-17: delay

4.5.4 *Network Load*

Figure 4.18 below shows us the network load for changing the Access-Time which is (0.1 seconds, 0.001 seconds, 0.00001 seconds) with respect to the number of nodes:

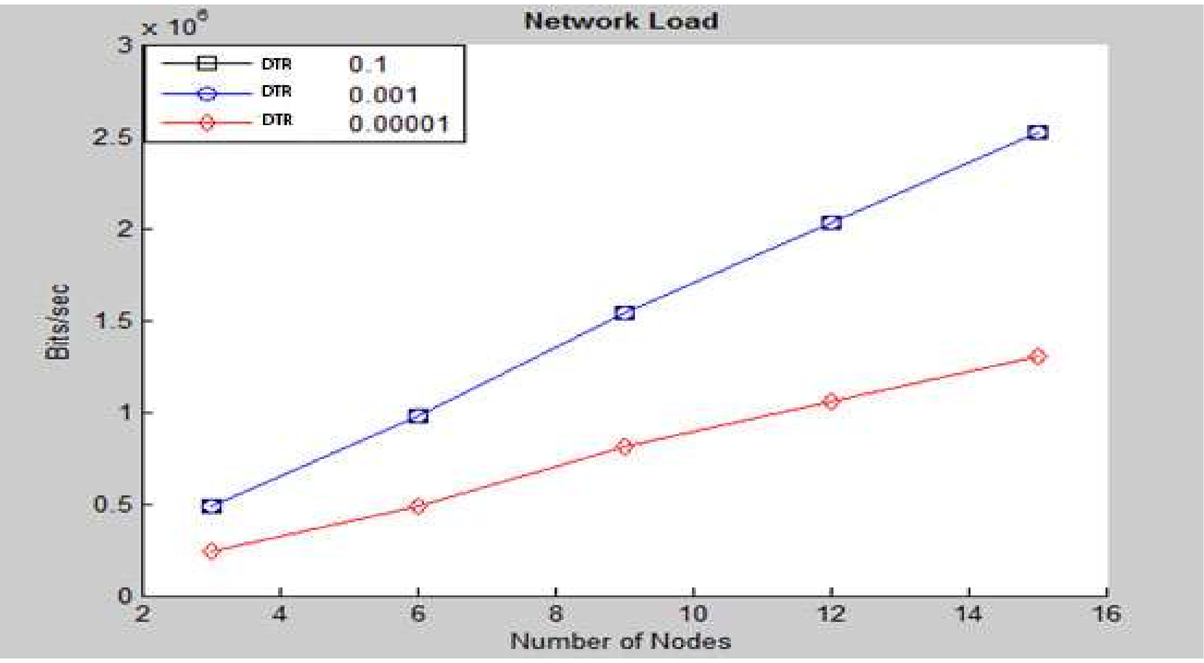


Figure 4-18: Network load

4.5.5 *Retransmission*

Figure 4.20 below shows us the retransmission for changing the Access-Time which is (0.1 seconds, 0.001 seconds, 0.00001 seconds) with respect to the number of nodes:

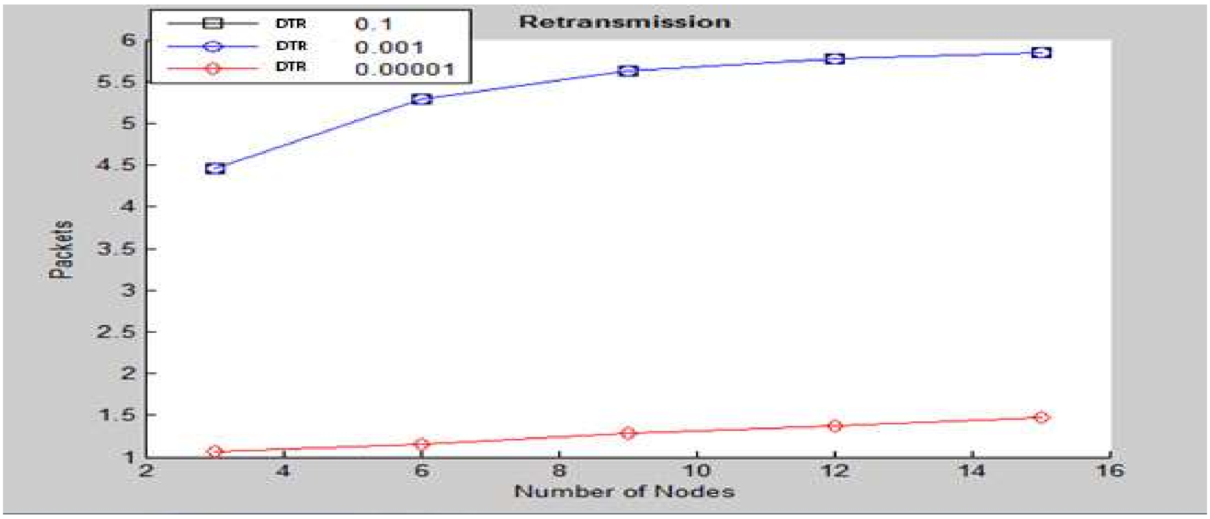


Figure 4-20: retransmission

4.5.6 *Throughput*

Figure 4.21 below shows us the Throughput for changing the Access-Time which is (0.1 seconds, 0.001 seconds, 0.00001 seconds) with respect to the number of nodes:

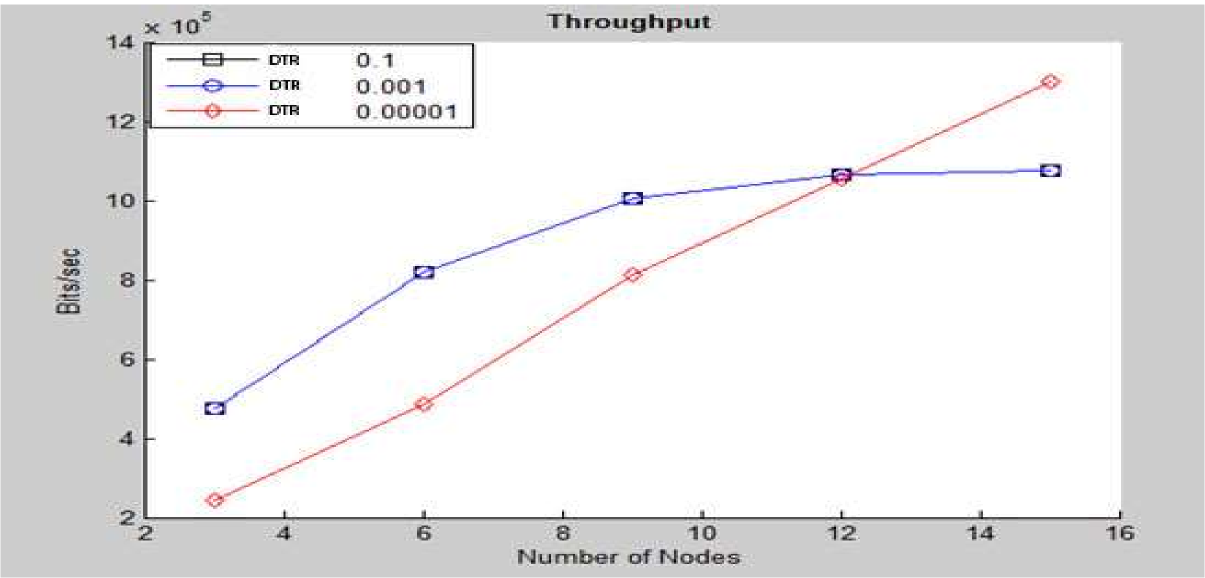


Figure 4-21: throughput

4.6 Results of changing the Access-Time based on the number of node (DTR+)

4.6.1 *Introduction*

In this part we are going to see the results of changing the time for the "Access-Time", but with respect to the number of nodes in each zone. The default value was 0.001 seconds based on mathematical and experimental studies. The access-time is set as follows, the more number of nodes in zone means the high access time for the zone which are: 0.1, 0.001, and 0.00001 seconds.

4.6.2 *Data Dropped*

Figure 4.22 below shows us the data dropped for changing the Access-Time based on the number of nodes which is (0.1, 0.001, and 0.00001 seconds) with respect to the number of nodes:

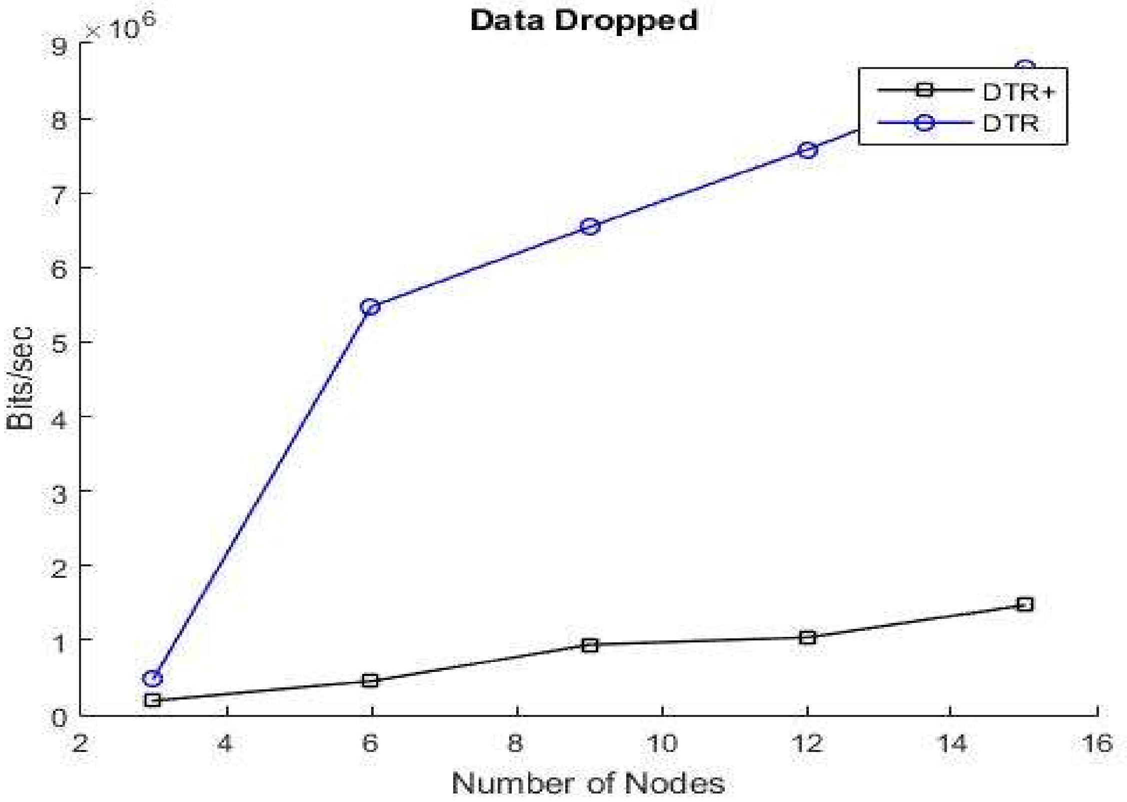


Figure 4-22: Data Dropped

4.6.3 *Delay*

Figure 4.23 below shows us the delay for changing the Access-Time based on the number of nodes which is (0.1 seconds, 0.001 seconds, 0.00001 seconds) with respect to the number of nodes:

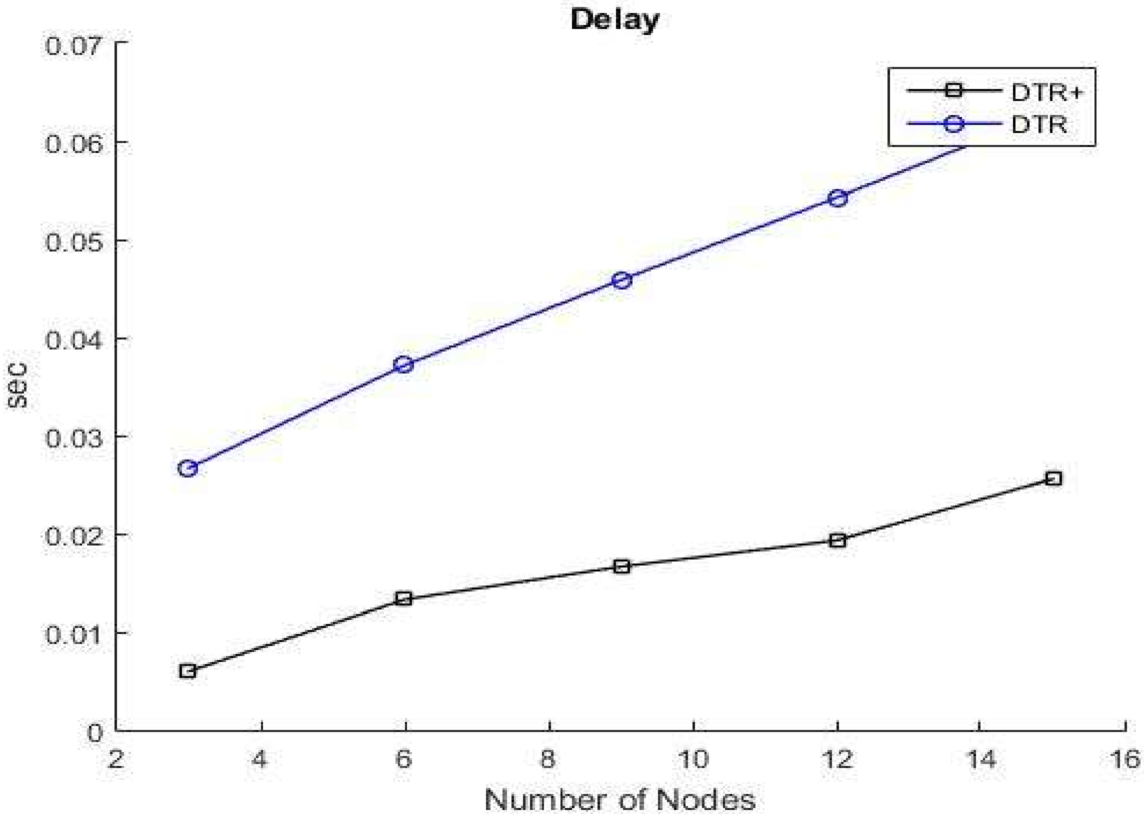


Figure 4-23: delay

4.6.4 *Retransmission*

Figure 4.24 below shows us the retransmission for changing the Access-Time based on the number of nodes which is (0.1 seconds, 0.001 seconds, 0.00001 seconds) with respect to the number of nodes:

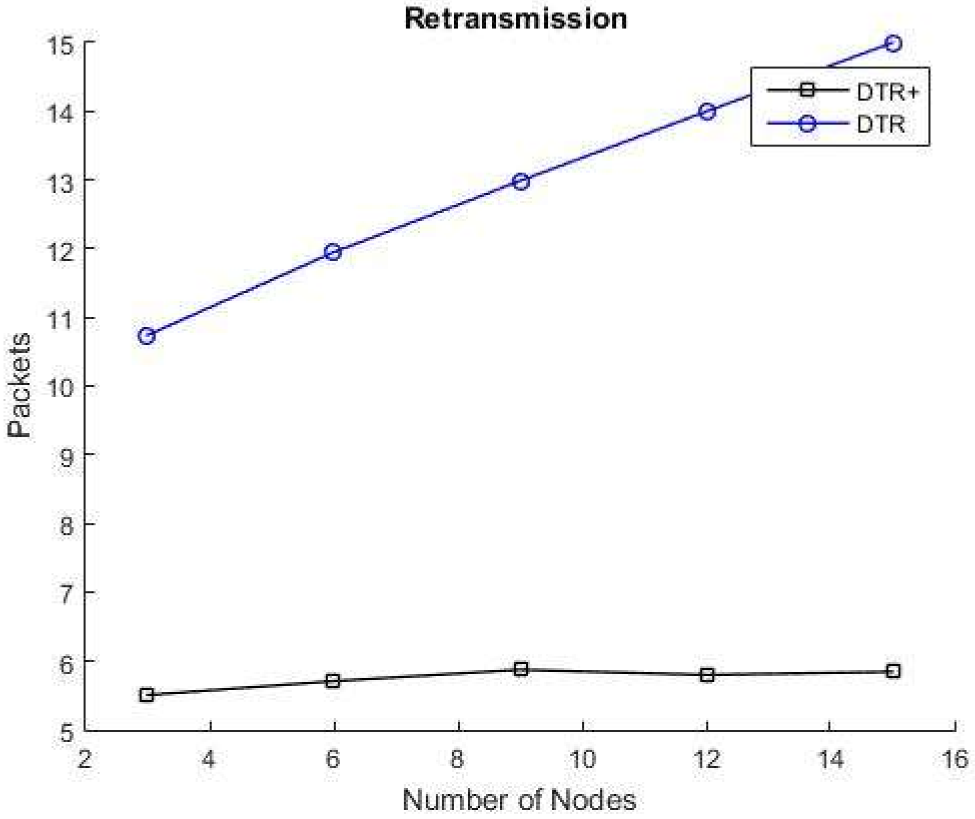


Figure 4-24: Retransmission

4.6.5 *Throughput*

Figure 4.25 below shows us the Throughput for changing the Access-Time based on the number of nodes which is (0.1 seconds, 0.001 seconds, 0.00001 seconds) with respect to the number of nodes:

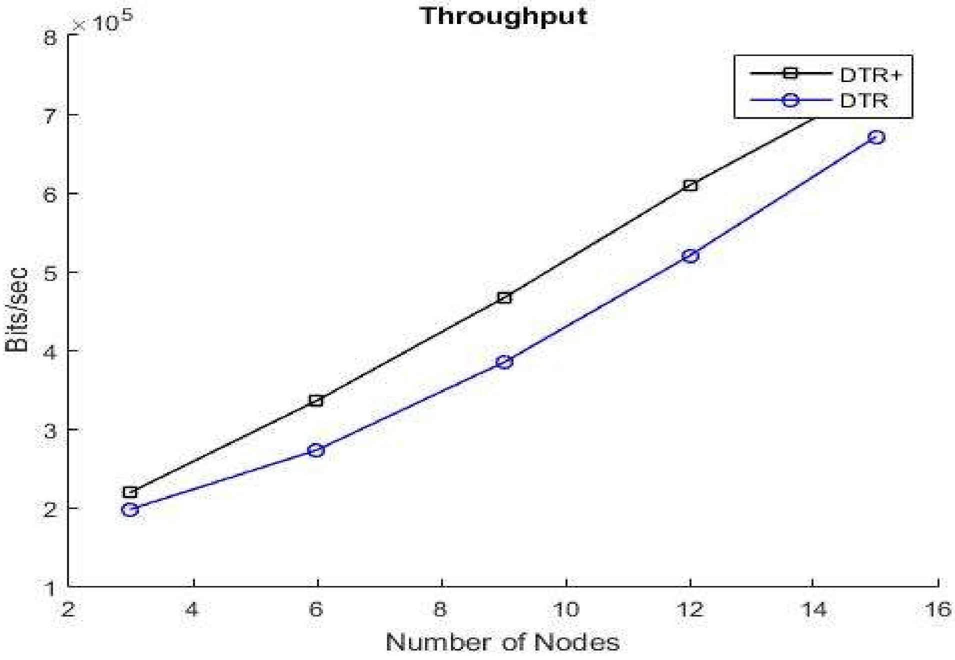


Figure 4-25: Throughput

5 Chapter 5

Analysis

5.1 Analysis of network performance for LDS Time Interleaving

After simulating the Interleaving LDS with different numbers of nodes using OPNET, we have compared the results which we got with the original MAC to see how the performance of the network changed. Below is the performance of the network compared with the original MAC.

5.1.1 *Media Access Delay*

According to the definition of the OPNET simulator, the media access delay is the delay of total contention and the queue delay that is received by a higher layer [2]. So, if the media access delay represents the queue and the contention delay, we can conclude that the media access delay represents the transmission delay due to the IFS. But her let us concentrate on the transmission delay. In our case the transmission delay is the main reason which is affecting on the Media access delay. we can see the media access delay for the LDS which is the blue line, while the original MAC is the red line. It is clear that the media access delay in the LDS is lower than the media access delay of the DLA. The difference in the delay increases as the number of nodes increase, this means the media access delay in LDS is much better than the media access delay of the Original MAC, since the lower the media access delay the better it is. As we said before, the media access delay depends on the transmission delay. In Original MAC, as we know, the access time for the packet is depends on the CSMA/CA. as we said the hight the transmission the hight the CA is, this means that the packet trying to access the medium must be delayed for a long time with depends on the transmission of nodes. This will therefore increase the transmission delay and the media access delay. With respect to the LDS, however, the delay time is dynamic (changes all the time there are no fixed values), which means the packet is not delayed as much as the packet of the original MAC. Let us take the same scenario which is 0.0003 seconds, that means the delay will be changing since it is dynamic, it can be 0.0003 or 0.0002 or 0.0001 or 0. Sometimes small delay and sometimes high delay so that will make the average delay less, therefore the transmission delay will be less leading to less media access delay. Also, if we took the other scenarios, it is the same.

Another main reason that is affecting on the media access delay is the queue delay, let us go back to the definition of OPNET that says the media access delay is the amount of contention and queue delay. As we know in original MAC algorithm the delay is depends on the CSMA/CA, that means in the zones which we have longer delay time, this will allow high number of packets to be transmitted by the other zones, since it is giving them more time for using the medium, which will lead to high queue delay and congestion, therefore leading to more medium access delay. On the other hand, where we have LDS the delay in each zone is dynamic, that means the delay will not be fixed for the whole time of simulation, leading to less average delay which will decrease the time allowed for the other zones to use the medium, thus leading to less queue delay and congestion, and that will decrease the media access delay.

Thus, the media access delay of LDS is less than the media access delay of the DLA algorithm.

5.1.2 *Data dropped, delay, network load, retransmission, throughput*

By looking to the data dropped, network load, retransmission, and throughput we can clearly see that the values of the results between the Original MAC and LDS are identical, or there is a slight difference between them which is negligible. This is true, since by distributing the time in a fair way between the zones that will only effect on the media access delay since the LDS is otherwise working in the same way that the Original MAC. Thus, the other results like data dropped, delay, network load, retransmission or throughput will not be affected.

5.1.3 *Conclusion of LDS*

As a final conclusion we can say that the LDS reached it goals of providing a fair way in distributing the delay between the zones and of giving each zone the same amount of delay. The other goal which we have reached by applying the LDS is improving the media access delay, and not affecting the other network performance like data dropped, delay, load, network load, retransmission, and throughput. This can be clearly seen where we can see that the amount of media access delay in the LDS has dropped in a significant way from the original MAC, while the other characteristics of the network retain identical values.

5.2 Analysis of DTR

The DTR is another mechanism used to eliminate the hidden terminal problem. After simulating the DTR and comparing it with the original MAC, the results of DTR show a better performance than the original-Mac, meaning that the design of the DTR has also reached its desired objective of eliminating the hidden terminal problem, and improving the performance of the MAC.

5.2.1 *Data dropped of DTR vs. ORIGINAL-MAC*

we can see the blue line which represents the data dropped of the original-MAC, while the black line shows us the data dropped of the DTR. According to the lines we can see that the data dropped of the DTR is much lower than the amount of data dropped of the original-MAC. The data dropped in the DTR starts from zero at 3 nodes and increases to 1,000,000 at 15 nodes. The data dropped in the original-MAC starts from 500,000 at 3 nodes and increases to 5,000,000 at 15 nodes. This is because, as we know the hidden terminal problem causes data to collide at the access-point, and since in the original-Mac all the nodes are sending at the same time while they are outside the range of each other's, this will cause a hidden terminal problem. So, a lot of packets will colloid at the access-point and that will lead to high amount of data dropped. But by looking at the DTR we can see the amount of data dropped is less than the amount of data dropped in the original-MAC meaning that the hidden terminal problem is solved. This is because as we divided the range of the access-point to 4 zones, and allowed each single zone to access the access-point for a specific time. This means the hidden terminal problem that is caused by the other zones is solved, because only one zone is being activated and the other zones are stopped, so leading to less collision, and less data dropped.

Another main reason that causes data drop is the congestion at the access-point. In DTR, as we know, only one zone is accessing the access-point at a time, which means fewer stations are using the access-point at each moment of time, and this leads to less congestion on the access-point which decrease the amount of data dropped. With respect to the original-MAC, there are no zones meaning that more stations are connected to the access-point, which will increase the congestion on the access-point, and this will lead to an increase in the amount of data dropped. In conclusion the DTR has solved the hidden terminal problem, since the data dropped (which was caused by the hidden terminal) and congestion has decreased significantly.

5.2.2 *Delay of DTR vs. ORIGINAL-MAC*

According to the definition of the OPNET program the delay represents the end-to-end delay of the packets [2]. The packet delay is the delay that is caused by summation of the queuing, processing, and transmission delay. Equation 5.3 below shows us how to calculate the delay [2].

$$- delay = N\left(d_{proc} + d_{trans} + d_{qeue}\right) \qquad 5.3$$

Where (d_{proc}): is the processing delay, (d_{trans}): transmission delay, (d_{pro}): propagation delay, d_{qeue}: queue delay, N: number of links.

Back to figure which shows us the delay difference between the DTR and the MAC with respect to the number of nodes; the blue line shows us the delay of the original-MAC, while the black line shows us the delay of the DTR. In DTR the delay starts from 0.01 seconds at 3 nodes and increases to 0.045 seconds at 15 nodes, while the delay in the original-MAC starts from 0.016 seconds at 3 nodes and increase to 0.055 seconds at 15 nodes. We can see a clear difference between them, the DTR has a lower delay than the delay of the original-MAC and this is because of the queue delay. If we look at equation, we can see that there is a (d_{qeue}) which is the queue delay, and as we know the definition of the queue delay is the time needed for the disk to become free. In other words, the waiting time depends on the load [3]. In DTR the queuing delay will be much lower than the queuing delay in the original-MAC, because DTR shows us less network load than the original-MAC this means the load and congestion on the access-point in DTR will be less than the load and congestion on the access-point in original-MAC, and the reason behind this is that in DTR, only one zone will access the access-point at a specific moment of time. This means that a lower number of nodes connected to the access-point, causing lower load and congestion on the access-point, which will decrease the queue and queue delay on the access-point since there is a relation between the load, congestion and the queue, which is the less the load, less the congestion and the queue, less the queuing delay. Also, by decreasing the number of nodes connected to the access-point we are decreasing "d_{proc}, d_{trans}.". As a result, if we decrease these three different types of delay "$d_{proc} + d_{trans} + d_{qeue}$", we will decrease the whole delay. On the other hand, with respect to the original-MAC, all the nodes are connected to the access-point since there is are no zones, which will offer more load, and more congestion meaning that more queue and more queuing delay, and more transmission and processing delay. This is the main reason that makes the delay in DTR is less than the delay of original-MAC.

5.2.3 *Network load of DTR vs. ORIGINAL-MAC*

According to the definition of OPNET network load means: *"the total data traffic received by the entire WLAN BSS from higher layers of the MAC"* [2]. In figure we can see the network load between the DTR and the original-MAC. The black line represents the DTR, while the blue line represents the original-MAC. It is clearly seen from the figure that the network load of the DTR is much less than the network load of the original-MAC. DTR starts from 0.2 bits/sec to 2.2 bits/sec, while the original-MAC starts from 1 bits/sec to 5.2 bits/sec. The reason behind is that DTR splits the network into 4 zones, and as we said before only one zone is allowed to access to the access-point. This means that the numbers of nodes that are connected to the access-point are reduced, which means less traffic and less network load. According to our scenario let us take the case where we have 3 nodes per zone. In case of original-MAC the 12 nodes are transmitting through the access-point since there are no zones, which will increase the traffic leading to an in increase in network load. On the other hand, in the case of DTR, when we divide the network into 4 zones, and each zone accesses the access-point for a specific time, we are effectively splitting the number of nodes that are connected to the access-point. Let us take the scenario where we have 3 nodes per zone. In this case not all of the 12 nodes are using the access-point; only 3 nodes are using the access-point for a specific time. And this will decrease the traffic leading to a decrease in network load. This is the main cause that makes the traffic "network load" of DTR less than the traffic "network load" of original-MAC.

5.2.4 *Retransmission of DTR vs. ORIGINAL-MAC*

According to the definition of OPNET retransmission means: *"Total number of retransmissions attempts by all WLAN MACs until the packet is totally transmitted or discarded as a result of reaching short or long retry limit"* [2]. In figure we can see the retransmission for both the DTR and the original-MAC; the back line represents the retransmission of the DTR, while the blue line represents the retransmission of the original-MAC. It is clearly shown by the figure that the retransmission of the DTR is less than the retransmission of the original-MAC. The DTR starts from 4.5 to 5.8 packets, while the original-MAC starts from 5.8 to 6 packets. There are two possible reasons for this. First, reducing the number of nodes that are accessing the access-point for a specific time, second reason, is that the DTR can solve the hidden terminal problem. Let us discuss the first reason: as we said before the number of nodes which are using the access-point for a specific time is reduced in the DTR, so by reducing the number of nodes that are communicating with the access-point this will reduce the number of packets being sent to the access-point. This will therefore it will reduce the possibility of packet collision which will lead to less retransmission. In the case of the original-MAC, the number of nodes that are communicating with the access-point is not being divided. This means more packets are being received by the access-point, which will increase the possibility of collision, leading to more retransmission. Now let us move to the second reason, which is the hidden terminal problem. As we know the hidden terminal problem causes the packets to collide with each other at the receiver (access point), which will increase the number of retransmissions. In our scenario, the hidden terminal problem is presented between the zones. In the original-MAC the hidden terminal problem is presented, which suffers from a high number of collisions at the receiver leading to an increase in the number of retransmissions. On the other hand, the DTR is designed to terminate the hidden terminal problem by dividing the zones into 4 zones and only allowing one zone to communicate with the access-point at a specific time, which means the other zones which are causing hidden terminal problem are being deactivated. So, by terminating the hidden terminal problem, we are reducing the number of collisions that is caused by it, therefore reducing the number of retransmissions. These are the main two reasons that make the number of retransmissions in DTR less than the number of retransmissions in the original-MAC.

5.2.5 *Throughput of DTR vs. ORIGINAL-MAC*

According to the definition of OPNET throughput means: *"Total number of bits/secs forwarded from wireless LAN layers to higher layers in all WLAN nodes of the networks"* [2]. In other words, throughput is the number of correct messages that are received by the network. In figure we can see that the throughput between the DTR and the original-MAC;

the black line represents the DTR throughput, while the blue line represents the original-MAC throughput. It is clearly seen from the figure that the throughput of the DTR is much higher than the throughput of the original-MAC. The DTR throughput starts from 490,000 bits/sec to 1,100,000 bits/sec while the original-MAC starts from 600,000 bits/sec to 350,000 bits/sec. This due to many reasons including: collision, traffic and delay. As we know, DTR is much better than original-MAC in collision, traffic, and delay. With respect to collisions, as we know, the DTR is used to remove the hidden terminal problem. By removing the hidden terminal problem, we are decreasing the number of collisions, since the main problem that hidden terminal problem cause is collision of packets. This increases the throughput. In the original-MAC the hidden terminal problem is still present, that meaning more collisions occur decreasing the throughput. Second reason that makes the throughput of DTR more than the throughput of original-MAC is the traffic. In DTR, as we have seen before, the network-load (which is the network traffic) is much less than the original-MAC, because DTR divides the zone of the access-point into 4 zones and each zone is only allowed to access the access-point at a specific time meaning less nodes are connected to the access-point at one time, therefore this causes less network-load and less traffic, which will lead to an increase in throughput. With respect to the original-MAC the 4 zones are acting as a one zone, meaning more nodes are connected to the access-point, causing higher network-load and traffic, leading to less throughput. Let us move to the final reason, which is that the delay of the DTR is less than the delay of the original-MAC, since only one zone is allowed to access the access-point that means less load, and when we say less load this means less queuing delay, which will lead to less delay. In original-MAC all the nodes are connected to the access-point, which increase the load and the queuing delay which will lead to high delay. These are the main reasons which make the throughput of the DTR more than the throughput of the original-MAC.

Note*: If we look at the 3 nodes for the DTR and the original-MAC we can see that the DTR throughput is less than the original-MAC throughput. Where the DTR throughput at 3 nodes is 490,000 bits/sec while the throughput at 3 nodes in original-MAC is 600,000 bits/sec, why? First, we should know that throughput increases as the number of nodes increases. In DTR, as we know, only one zone is allowed to access the access-point. For example, at 3 nodes per zone, only 3 nodes are allowed to connect to the access-point. But in the original-MAC there are no zones. This means that for the scenario which has 3 nodes per zone, there are 12 nodes connected to the access-point, this larger number of nodes increase the throughput. Because of this, the throughput at 3 nodes per zone in the original-MAC is higher than the throughput at 3 nodes per zone in DTR (Note*: we should know that is graphically true, but analytically is not true, since in analysis DTR should give better throughput than original-MAC at 3 nodes). We must, however, pay attention that this only works for the scenario where there are 3 nodes per zone. For the other scenarios where the number of nodes 6, 9, 12, and 15 nodes per zone, it is different. Since the collision, traffic, and delay become the dominant on the network and that will effect on the throughput, but since DTR perform better in the case of collision, traffic, and delay, it will deliver more correct packets leading to higher throughput than original-MAC.

5.2.6 *Collision of DTR vs. ORIGINAL-MAC*

According to OPNET definition, collision means: *"a Boolean value (0 or 1) that reflects the collision of packets at the receiver channel"* [2]. we can see the collision of the DTR and original-MAC for 3, 9, 12, and 15 nodes; the blue line represents the original-MAC, while the red line represents the DTR. It is clearly seen from the figure that the number of collisions of the DTR is less than the number of collisions of the original-MAC. This is because, as we know, the DTR is used in order to solve the hidden terminal problem, which causes collision at the receiver (access-point). Because of that the number of collisions is less. With respect to the original-MAC, the hidden terminal problem is an issue. There is a higher number of the collision at the receiver (access-point). That is the main reason that the number of collisions at DTR is less than the number of collisions at the original-MAC.

5.3 Analysis of changing the access-time in DTR

In our project the access-time was chosen as 0.001 seconds for two reasons. Firstly, in order to satisfy the transmission time according to network specification. Secondly, in order to maintain QoS (Quality of Service). With respect to the first reason, as we know, we should give a specific time greater than the minimum time which is required to transmit a packet. The equation below calculates the time required for transmission:

$$t_{trans} = \frac{Packet\ size\ (bits)}{Data\ rate\ (bps)}$$

According to the network configuration the network

1. Data rate is 24 Mbps which is 24,000,000 bps

2. Packet size is 1024 byte which is 8,192 bits

$t_{trans} = \frac{8192\ bits}{24{,}000{,}000\ bps} = 0.00034$ seconds.

For the second reason as we know to have a good quality of service, the time waiting for the other zones should not be too long. Due to these two reasons we should chose the minimum transmission time. After studies we found 0.001 seconds is the best time since it satisfies the minimum time required for transmission, with respect to the quality of service 0.001 seconds is an acceptable time for the other zones to wait.

These are the two main reasons which made us chose the access-time 0.001 seconds. Now let us change the access-time and see the effects of these changes on the network performance.

5.3.1 *Data dropped*

we can see the data dropped for the DTR for different access-times. The blue line and the black line represent the access-time for 0.1 seconds and 0.001 seconds respectively, while the red line represents 0.00001 seconds. It can be clearly seen from the figure that the amount of data dropped of 0.00001 seconds is much better than the amount of data dropped of 0.1 seconds and 0.001 seconds. That because the time allowed using the medium is less that means less congestion leading to less data dropped. But theoretically this is not true since the minimum time required to transmit is 0.00034 seconds and by choosing 0.00001 that means the station would not be sending the full packet, and no more than one station will be able to use the medium, which will decrease the congestion on the access-point leading to less data dropped. This is the main reason that makes the amount of data dropped to decrease. With respect to 0.1 seconds which is the black line we find exactly the same result as with 0.001 seconds, and that because it is above the minimum time required transmitting the packet size.

5.3.2 *Delay*

we can see the delay for DTR for different access-times where the blue and the black line represent the delay for 0.1 seconds and 0.001 seconds respectively, while the red line represents the delay for 0.00001 seconds. It can be clearly seen from the figure that the 0.00001 seconds is much better than 0.1 seconds and 0.001 seconds, and that because the time allowed for transmitting is much lower which means less$\left(d_{trans}, d_{qeue}, d_{pro}\right)$ and less delay. But theoretically this is not true as we said before 0.00001 seconds is much lower than the minimum time required for transmitting the packet, which means the nodes is not transmitting the full packet and not all of the station inside the zones are able to transmit. This is the main reason that makes the delay in 0.00001 seconds much lower. With respect to 0.1 seconds, we find the same results as 0.001 seconds since it is above the minimum time required for transmitting.

5.3.3 *Network Load*

we can see the network load for DTR for different access-times. The blue line and the black line represent the access-time for 0.1 seconds and 0.001 seconds, while the red line represents the 0.00001 seconds. It is clearly seen from the figure that network load of 0.00001 seconds is better than that of 0.1 seconds and 0.001 seconds, and this is because the time for transmitting of each zone is less, that means less data traffic and congestion, and as a result there will be less network load. But theoretically this is not true, because by reducing the access-time below the minimum time required to complete transmission that means the node would not be transmitting the full packet. Also, not all of the stations can transmit, which will decrease the network load. With respect to 0.1 seconds, we find the same results because it is above the minimum time required for transmitting.

5.3.4 *Retransmission*

we can see the retransmission for DTR for different access-times. The blue and the black line represent the retransmission for 0.1 seconds and 0.001 seconds respectively, while the red line represents the retransmission for 0.00001 seconds. We can see from the figure that the 0.00001 seconds is much better than 0.1 seconds and 0.001 seconds, and this is because as we said before that the data dropped is much lower, meaning that the retransmission will be much lower. But also, theoretically that is not true since the node cannot transmit the full packet, and not all of the stations are capable to transmit because the time for transmission is less than the minimum time required for the station to transmit. With respect to 0.1 seconds, we see the same results as with 0.001 seconds since it is above the minimum time required to transmit a packet.

5.3.5 *Throughput*

we can see the throughput for DTR for different access-times; the blue line and the black line represent the throughput for 0.1 seconds and 0.001 seconds respectively, while the red line represents the throughput for 0.00001 seconds. It is clearly seen from the figure that the throughput of 0.1 seconds and 0.001 seconds is much better than the throughput of 0.00001 seconds. But as we know 0.00001 seconds has much better values in data traffic, and delay, which means that the throughput in 0.00001 seconds should be much better than 0.1 seconds and 0.001 seconds. But the simulation shows us different results. Why? The results which we calculated are totally correct, but we need to go back to the definition of the throughput. What does throughput mean? Throughput means according to the definition of the OPNET: *"Total number of bits/secs forwarded from wireless LAN layers to higher layers in all WLAN nodes of the networks"* [2]. In other words, throughput is the number of correct packets that are received. Since the network at 0.00001 seconds is not performing correctly because the transmission time is not enough so some of the station in each zone is not able to send data. Also the node itself is not able to transmit a full packet. As results, the packet will not be delivered correctly to the access-point and the number of correct packets will be low, which will decrease the throughput. These are the main reason which makes the throughput at 0.00001 seconds less than at 0.1 seconds and 0.001 seconds.

Note*: By looking at the last part of the results and specifically at 15 nodes we can see that the throughput at 0.1 seconds and 0.001 seconds is lower than the throughput at 0.00001 seconds, because as we know the throughput is affected by data traffic and delay, and by increasing the number of nodes to 15 the data traffic and delay will increase significantly, which will increase the probability of collision. As we know, when the collision becomes the dominant, the throughput will decrease and this is the main reason that makes the throughput decrease at 0.1 seconds and 0.001 seconds. With respect to 0.00001 seconds, we should take in consideration that the network at 0.00001 seconds is not working properly. As a result, we cannot say that the throughput is better.

5.4 Analysis of changing the access-time in DTR based on the number of nodes (DTR+)

5.4.1 *Data dropped*

we can see the data dropped for the DTR and DTR+. The blue line and the black line represent the DTR and DTR+ respectively. It can be clearly seen from the figure that the amount of data dropped of DTR+ is much better than the amount of data dropped of DTR.

That because the zones now are accessing the access point according to the bandwidth and access time, which means the time is fairly distributed according to the number of nodes leading to less congestion and less data dropped. This is the main reason that makes the amount of data dropped to decrease.

5.4.2 *Delay*

we can see the delay for DTR and DTR+ for different access-times according to the number of nodes in each zone where the blue and the black line represent the delay for DTR and DTR+ respectively. It can be clearly seen from the figure that DTR+ is much better than DTR, and that because the time allowed for transmitting is much lower which means less$\left(d_{trans}, d_{qeue}, d_{pro}\right)$ and less delay and that because now less time for a smaller number of nodes, and more time for a greater number of nodes. This is the main reason that makes the delay in DTR+ much lower than the delay in DTR.

5.4.3 *Retransmission*

we can see the retransmission for DTR and DTR+ for different access-times and number of nodes. The blue and the black line represent the retransmission for DTR and DTR+ respectively. We can see from the figure that the DTR+ is much better than DTR, and this is because as we said before that the data dropped is much lower, meaning that the retransmission will be much lower, and that is the main reason which makes the DTR+ is much better than DTR.

5.4.4 *Throughput*

we can see the throughput for DTR and DTR+ for different access-times and number of nodes; the blue line and the black line represent the throughput DTR and DTR+ respectively. It is clearly seen from the figure that the throughput of the DTR+ is better than the throughput of the DTR. Because DTR+ has much better values in data traffic, and delay, which means that the throughput of DTR+ should be much better than DTR.

6 Chapter 6

Conclusions and Further works

6.1 Conclusion

Our goal was to create a new protocol for 802.11 WLANS which we called it Location Detection System (LDS) to eliminate the hidden terminal problem, and also to design another method using the concept of time division multiplexing (TDMA), which we named DTR in order to eliminate the hidden terminal problem in 802.11. In a way our project was successful, because we made the LDS algorithm to eliminate the hidden terminal problem. Also, the new design of TDMA-MAC was successful since we were able to eliminate the hidden terminal problem, and improve significantly on the performance of 802.11.

This report contains five main chapters in addition to this chapter. In the first chapter, we introduced the subject of this thesis. In the second chapter, we discussed the literature review of 802.11 we focus in the literature review on the operation of 802.11 especially on the MAC-layer like Back-off procedure. In the third chapter, we talk about the methods that were to be followed in order to design the LDS and DTR using programming codes like C, and C++ language, in addition to mathematical proofs. In chapter 4 we obtained the results of the LDS like (data dropped, delay, media access delay, network load, retransmission, and throughput) and compared them to the original-MAC. With respect to the DTR we also got the results like (data dropped, delay, network load, retransmission, throughput, and collision) and compared them to the results of the original-MAC. Chapter 5 was the most important chapter in this thesis since it concluded all the work we have done. First, we have concluded that LDS algorithm will improve the performance of the network and especially the media access delay, without affecting on any other part of the network. As we have seen from the results, the media access delay of the LDS is much better than the media access delay of the original MAC. Also, another conclusion from the LDS algorithm is that it removed the hidden terminal problem. The second conclusion that we have acquired from the thesis is that by applying the DTR operation to the 802.11 we can eliminate the hidden terminal problem and improve the network performance by decreasing the data dropped by dividing the zones into 4 and allowing only one zone to access the media at a specific moment in time. Also, we have decreased the delay by decreasing the queuing delay. Further, we have improved the network by decreasing the network load since fewer stations would be connecting to the network at the same time. Additionally, by increasing the throughput, since we have decreased the collision, data traffic, and delay. The third important conclusion that we have derived is that the DTR time access cannot be chosen randomly, it must be chosen according to specific calculations related to packet length, transmission rate, number of nodes and quality of service. As we have seen in our results, if we decrease the access-time below the minimum time for transmission a packet, a lot of packets loss will occur which will degrade the throughput. On the other hand, if we increase the access-time too much that means we would be increasing the stop time for other zones without any reason, and that will degrade the quality of service. As a final word, we should carefully choose the access-time depending on the scenarios which we will set up.

6.2 Further work

6.2.1 *Thesis further research*

There are some issues which haven't been solved in this thesis, which required further study.

6.2.1.1 Improving LDS

As we know LDS is used in order to achieve delay fairness between the four zones of the access-point, but it is not used to achieve delay and bandwidth fairness together. Suppose we have different number of users in each zone as shown in figure 6.1 below:

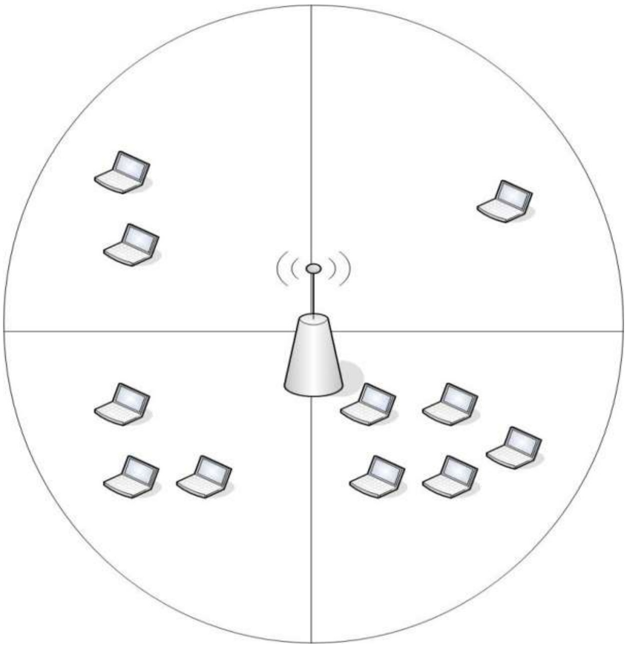


Figure 6-1: different number of stations in each zone

This would mean that the bandwidth used by each zone is different from each other, so we should find a dynamic delay algorithm to ensure that the delay time is distributed fairly according to bandwidth.

References

[1] Vern A. Dubendrof, *Wireless data technologies,* England: John Wiley & Sons Ltd, 2003, pp.30

[2] From the official OPNET site. Available:
http:www.opnet.com/solutions/network_rd/modeler.html, and OPNET® 14.5 program.

[3] JansiMohamadZain, Wan Maseri Wan Mohd, Eysa El-Qwasmeh, *Software Engineering and Computer Systems*, London, New York: Springer-Verlag Berlin Heidelberg, 2011, pp. 402.

7 Bibliography

[1] IEEE Computer Society, *IEEE Standard for Information Technology Telecommunication and information exchange between system Local and metropolitan area networks Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, New York: Institute of Electrical and Electronic Engineers, 12 June 2007.

[2] Steve Rackley, *Wireless Networking Technologies: from principles to successful implementation*, Oxford: Elsevier's Science & Technologies, 2007.

[3] SudipMisra, Isaac Woungang, Subhas Chandra Misra*, Guide to Wireless Ad Hoc Networks,* London: Springer-Verlag London, 2009.

[4] Vern A. Dubendorf*, Wireless data technologies,* England: John Wiley & Sons Ltd., 2003.