LONG RANGE FOR COMMUNICATION IN WIRELESS SENSOR NETWORKS

BY

Omar Hameed

2003-NUST-BIT-21

Project Report in partial fulfillment of
the requirements for the award of
Bachelor's degree in Information Technology (BIT)

In

NUST Institute of Information Technology (NIIT)
National University of Sciences and Technology (NUST)
Rawalpindi, Pakistan
(2007)

CERTIFICATE

Certified that the contents and form of thesis entitled "**Long Range Communication in Wireless Sensor Networks**" submitted by Omar Hameed have been found satisfactory for the requirement of the degree.

Advisor: ___________________________
Dr. Arshad Ali

Co-Advisor: __________________________________
Dr. Ashiq Anjum

Member: __________________________________
Dr. Ali Khayam

Member: __________________________________
Mr. Ali Sajjad

DEDICATION

In the name of Allah, the Most Gracious, the Most Merciful

To my Dear family

Acknowledgements

I am extremely grateful to Dr. Ali Khayam for his valuable supervision and support to perform such a valuable research work and studies in the field of wireless communication technologies. I truly admire his dedication to my project and for all the help and suggestions and guidance that he has offered. A great deal of my project has been possible because of him.

I would like to express my gratitude to Dr. Ashiq Anjum my co-advisor and Mr. Ali Sajjad (Faculty member) for their valuable implications and comments to improve the dissertation.

I am highly thankful to all of my teachers whom had been guiding me through out my course work. Their knowledge, guidance and training enabled me to carry out this research work.

I am also thankful to Prof. Dr.Arshad Ali my Advisor for being such nice to me at times when I needed it most. I am thankful to him for his support that was always there during my project despite his busy schedule

Declaration

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institute or elsewhere.

Table of Contents

List of Figures

List of Tables

Abstract

In this project research work has been conducted for comparison of MAC layers of both 802.11e and 802.16e. More specifically Quality of service and reliability issues of both the protocol standards are compared, with the help of simulations. 802.11e and 802.16e are compared because they are only available standards that provide built-in quality of service and mobility. This comparison is done for a doctor patient application where patients have sensors which communicate with the base station located at the hospital. Sensors are used for the transmission of patient's condition at any instant of time. So starting from the MAC layer issues of both WiMAX and WiFi, future work can be conducted on physical layer so that pros and cons for enabling WiMAX protocol for WSN can be discovered.

Chapter 1

INTRODUCTION

1.1 WIRELESS SENSOR NETWORKS

Use of sensors has been amplified significantly in the past few years in different industrial, medical, military, tracking and control applications, by real time awareness and organization of resources. Sensors are anticipated to become a chief spring of information in the near future [100], it can be said that sensor networks will ultimately permeate the physical world [8]. Extensive research work and advancements are pragmatic in Wireless Sensor Networks so ultimately they will be a means to true ubiquitous computing [6].

1.2 WIRELESS COMMUNICATION TECHNOLOGIES

From the very start of wireless communication a melee between diverse wireless technologies like RFID, Bluetooth, WiFi, WiMAX, 3G and 4G was started. But as time passed it revealed that all of these technologies have their own expanse of functioning. For instance, RFID is used in PANs (Personal Area Network), Bluetooth and WiFi used for LANs (Local Area Network) while that of WiMAX and 3G are available for MANs (Metropolitan Area Network) and WAN (Wide Area Network) respectively. So it is expected that all of these technologies will co-exist in the prospect and none of them can vanish completely [9].

1.3 WIRELESS APPLICATIONS

Wireless technologies are being tremendously used in different types of applications nowadays. Doctor Patient application is one the key example where different patients are connected to the doctor wirelessly and can communicate with their doctors. Patients don't need to be in hospital for their routine checkup. More notably doctor can be reported about the patients' critical situation at any time. At the same time sound and consistent transfer of information is very important. So reliability and Quality of Service of wireless technology for such applications can't be ignored.

1.4 PROBLEM STATEMENT

For the doctor-patient application, which existing wireless technology will provide better Reliability and Quality of Service? So in this project comparison is done for a doctor patient application where patients have sensors which communicate with the base station located at the hospital. Sensors are used for the transmission of patient's condition at any instant of time. For such comparison 802.11e and 802.16e are the two communication technologies that selected specifically because these are the only available standards that provide built in support for Quality of Service and Reliability. Both have differences too in case of data rate, bandwidth, channel access mechanisms etc. But these both are the only technologies that can meet critical quality of service and reliability issues of doctor patient application. So starting from the MAC layer issues of both WiMAX and WiFi, future work can be conducted on physical layer so that pros and cons for enabling WiMAX protocol for WSN can be discovered.

1.5 DOCTOR PATIENT APPLICATION

In wirelessly communicating applications Doctor patient application is one of the most emergent applications. Doctor patient application is assumed to function using both WiMAX and WiFi protocols over wireless sensors network.

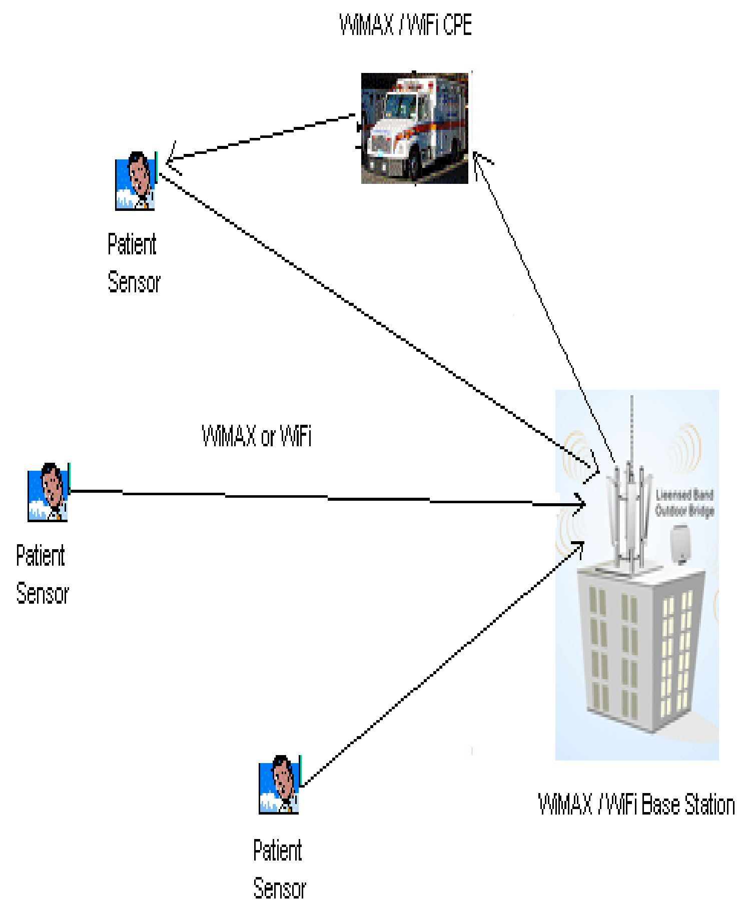

Figure 1: Application Architecture

Sensors will be with patients who can communicate with doctor through WiMAX or WiFi base station fixed over the hospital. These sensors can transmit specific information to the base station whenever needed and that information can be received by doctor in the hospital. There can be different patients which need to send different type of information to the hospital. This information is based upon different medical condition of a single patient or different medical conditions of different patients. For that purpose these patients are categorized on the basis of their priority. For instance a heart patient sending his/her information will have higher priority over a routine checkup patient. These categories are:

a. Critical/Emergency
    - Heart attack, Epilepsy Fits etc
b. Non-Critical
    - Cough, Flu etc
c. Routine Checkup
    - Heart Beat, Blood Pressure etc
d. Continuous Examining
    - To record patients history

1.6 MOTIVATION

Sensors have mixture of applications nowadays in variety of areas. We can see that use of sensors is classified to only those regions strictly where there is communication detachment gap not more than few hundred feet, this is because of their slighter power and hence range [1]. Numerous boundaries and problems are there in sensor networks like restricted battery life, partial communication bandwidth, some degree of processing power, limited range, high rate of packet loss, poor communication links, packet collision and fading of signal strength but their importance in ubiquitous computing cant be ignored [1].

We see that WiFi is now being used in sensor networks, extensively [2][10]. It provides added range as compared to other few other communication technologies like Bluetooth and RFID. Thus we can say that range of sensors is inhibited to the range of

WiFi. So the setback is, if WiFi can be used with sensors then why not WiMAX. Although WiFi and WiMAX have differences but similarities can't be ignored.

Starting from the comparison of WiFi's and WiMAX's, reliability and Quality of Service which are very basic issues, exact to the Doctor Patient application, we can decide about enabling WiMAX protocol for sensor networks.

1.7 AIMS AND OBJECTIVES

Our aim is to compare reliability and Quality of Service provided by 802.11e (WiFi) and 802.16e (WiMAX) so that to achieve finest reliability and services for a doctor patient application. Only then future work can be carried out for enabling WiMAX for sensor networks once it becomes obvious that which protocol ensures reliable communication and provides best quality services.

Chapter 2

LITERATURE REVIEW

2.1 INTRODUCTION TO WiMAX

This standard was paved a new path for entrance of broadband wireless access as a major new tool in the effort to link homes and business to core telecom networks, worldwide. Particularly, WiMAX was proposed as an alternative to cables, DSL, fiber optics, etc because wireless systems have capacity to address wide geographic areas in a cost effective manner [100] . WiMAX is used to provide point to multipoint wireless access to internet and other networks [1300]. By using WiMAX, costly infrastructure development required in deploying cabled access networks can be abandoned. Wireless MAN uses exterior antennas and radio base stations for communications. These antennas are usually mounted on the buildings and central base stations can communicate with antennas on different buildings.

Wireless MAN brings network to the users within the building with conformist LANs or Ethernet. However, the elementary design of the standard may in due course allow for the efficient extension of the Wireless MAN networking protocols openly to the individual user. WiMAX provides broad bandwidth distribution and QoS mechanisms but the details of scheduling and reservation management are left un standardized and endow with an essential mechanism for vendors to distinguish their equipment.

Although, WiMAX provides high data rates and fast access to a wide geographic areas but still there are some limitations. These limitations are similar to DSL that if distance increases, bandwidth decreases and vise versa [100].

2.1.1 WiMAX Forum

The WiMAX Forum is "the exclusive organization dedicated to certifying the interoperability of BWA products, the WiMAX Forum defines and conducts conformance and interoperability testing to ensure that different vendor systems work seamlessly with one another" [100]. WiMAX forum is a non-profit in association comprised of key industry players for example Intel, Fujitsu, Alcatel, Motorola, China Mobile Telecom, British Telecom, France Telecom, Dell, Cisco Systems etc and founding members include Nokia, WiLAN and Ensemble Communications.

2.1.2 WiMAX System

WiMAX organization uses Base Stations to present high speed data connections that can be used for voice, data, video and audio services at distances equal to 30 km. A characteristic WiMAX system consists of two parts:

- A WiMAX tower (Base Station)
- A WiMAX Receiver (Subscriber Station/Customer Premises Equipment)

WiMAX tower is just like a gigantic cell phone tower. It provides superior coverage area; around up to 3000 sq. KMs. Receiver can be an antenna or en suite PCMCIA (Personal Computer Memory Card International Association) card. These antennas and cards converse with the tower in the same way as WiFi access is performed [11] [12].

2.1.3 WiMAX Versions

WiMAX endow with two forms of wireless services:

- Non-Line of Sight service is identical as that of WiFi. Inferior frequency ranges 2 GHz to 11 GHz are used so as to reduce the dwindling because inferior frequency transmissions are not effortlessly disrupted as a result of obstructions. For example a tiny antenna on any machine like computer, television etc is attached that communicates with the tower or base station [11].

- Line of Sight transmission is used when bunch of data is to be sent at abridged interference. Superior frequencies up to 66 GHz are used for line of sight transmissions. Line of sight can be achieved by insertion of dish antenna in front of the tower for without any obstruction. And lot more bandwidth can be achieved [11].

2.1.4 WiMAX Types

Types of WiMAX depend upon there mobility. These are two.

- Fixed WiMAX
- Mobile WiMAX

Every technology has its own meticulous strengths for particular applications in which they are used [13]. Mobile WiMAX will bring technology to the portable devices such as PDAs, Laptops, and mobile phones whereas that of fixed WiMAX has brought broadband connectivity for suburban and business users. Basically the difference depends upon how these both deal with the channel characteristics. More the user gets mobile; more it gets difficult to deal with the channel because then channel starts varying more rapidly. Although WiMAX forum (Evolutionary Task Group) tried a smooth transition between the two standards that is Fixed WiMAX (IEEE 802.16-2004) and Mobile WiMAX (IEEE 802.16e), for fixed WiMAX, OFDM 256 mode was adopted while for mobile WiMAX scalable OFDMA was preferredFully Mobile WiMAX probable in 2008. And it is anticipated that 802.16e will grind down the 802.16-2004 market share entirely as it provides together the fixed and mobile services [14]. As far as compatibility between the two standards may be achieved by introducing dual mode chips with the intention that both the versions turn out to be interoperable.

Here is the table defining the two technologies very comprehensively.

A definition of fixed and mobile access

Definition	Devices	Locations/ Speed	Handoffs supported	802.16-2004 WiMAX	802.16e WiMAX
Fixed access	Outdoor and indoor CPEs	Single/ Stationary	No	Yes	Yes
Nomadic access	Indoor CPEs, PCMCIA cards	Multiple/ Stationary	No	Yes	Yes
Portable access	Laptop PCMCIA or mini cards	Multiple/ Walking speed	Hard handoffs	No	Yes
Simple mobile access	Laptop PCMCIA or mini cards, PDAs or smartphones	Multiple/ Low vehicular speed	Hard handoffs	No	Yes
Full mobile access	Laptop PCMCIA or mini cards, PDAs or smartphones	Multiple/ High vehicular speed	Soft handoffs	No	Yes

Source: WiMAX Forum

Figure 2: Fixed and Mobile access

2.1.5 WiMAX Protocol Layers

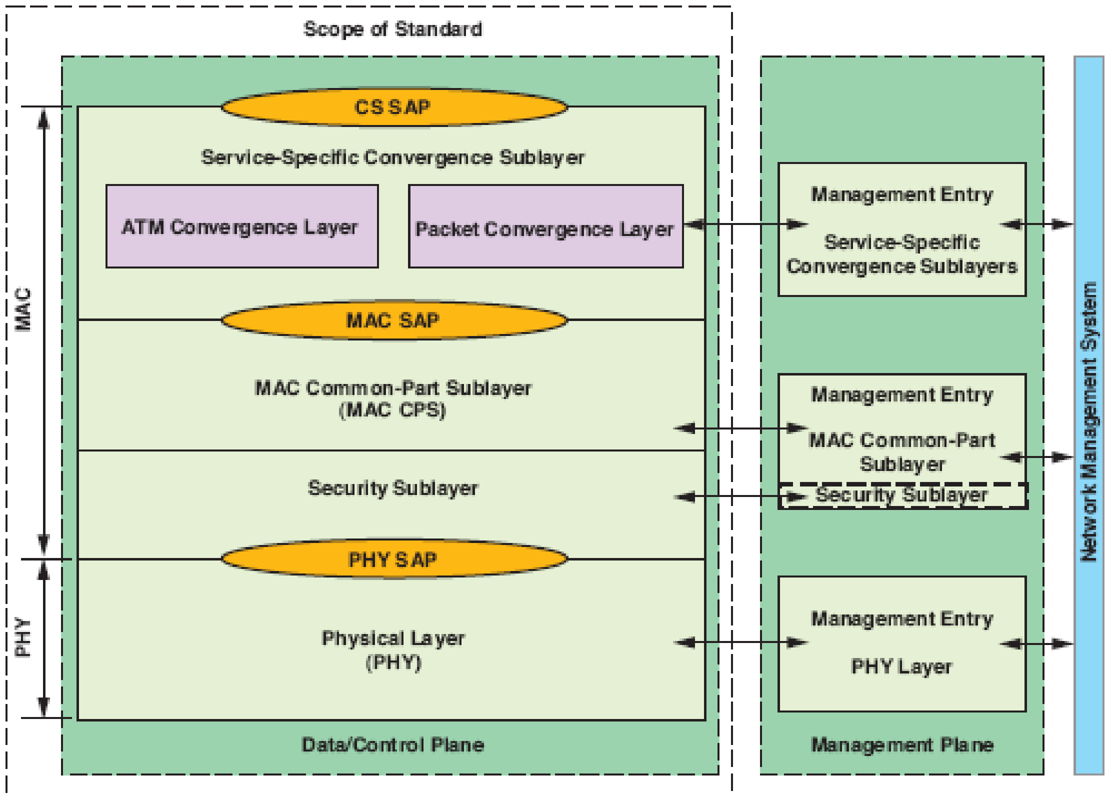

Figure 3: WIMAX Protocol Layers

2.1.6 MAC Layer

The 802.16 protocols are extremely adaptive, and they facilitate subscriber terminals to signal their wants while at the same time allowing the base station to regulate operating parameters and power levels to gather subscriber needs [5]. MAC layer of 802.16 uses Scheduling Algorithm it could be polling, as an alternative of contention procedure. For which the subscriber has to compete once, and thus a slot is being allocated to subscriber to access base station. This slot can expand or contract as the subscribers needs changes. But slot assigned to one of the subscriber by base station cannot be used by other subscribers. This Scheduling algorithm enables base station to control Quality of Service by balancing the time slot assignments among the subscribers, on the basis of application needs of subscriber stations. Because of scheduling algorithm, MAC layer of 802.16 is proficient of supporting overload and excess of subscriptions, contrasting 802.11[100].

Protocol Data Unit of variable length is used to increase efficiency of the stack. Multiple MAC PDUs can be concatenated, with multiple MAC SDUs (Service Data Unit) in each MAC PDU to save physical header overhead and MAC header overhead respectively. Larger SDUs can be fragmented to send them across the network to ensure Quality of Service. And redundancy caused in SDUs can be easily removed by using Payload Suppression Headers. Another important feature of 802.16 MAC layer is that it can overcome the problem of delay in acknowledgements by using a self correcting bandwidth request/grant scheme [4].

MAC layer's Security sub layer uses DES (Data Encryption Standard) in CBC (Cipher Block Chaining) mode for data encryption and decryption. Basically payload fields of transport layer and secondary management connections are encrypted. RSA public key methods and x.509 certificates are used to transfer keys between subscriber stations and base stations. This is done by PKM (Personal Knowledge Management) protocol. Also PKM provides certificate based authorization to subscriber station [4].

Once any device enters the network, various tasks are performed for authentication and synchronization of both the subscriber and base station. When downlink signal of base station is synchronized the UCD (uplink channel descriptor) is used to get the initial timing parameters and initial ranging contention slot. For authorization secured secondary management connection is established with the help of PKM protocol. Then registration process is performed by Base Station.

MAC layer of 802.16 is specific for IP traffic. And we see that commercially IP based services signify the best approach for operators and service providers. 802.16 MAC layer provides both modes of traffic that is bursty and incessant but it depends upon the mode that is provided by the connection oriented protocol [5].

2.1.7 Working Scenario of 802.16e

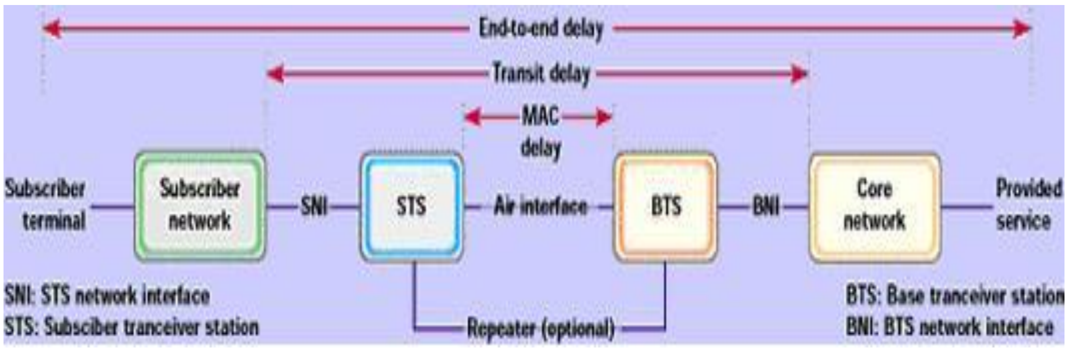

Figure 3: Working Scenario of 802.16

Reference: students.csci.unt.edu/~sf0040/WiMAX%20Protocol.ppt

2.1.8 Industrial Review

2.1.8.1 Deployment of two mobile WiMAX networks

Samsung has revealed to deploy two mobile WiMAX networks nation wide in North America. Fixed WiMAX provides broadband access by eliminating cables and DSL. While mobile WiMAX is supposed to provide another choice to existing mobile networks [16] [17].

2.1.8.2 Deployment of WiMAX on Oil rigs in Gulf of Mexico

The project is being carried out by Redline Communications. Basically 3.4 GHz multipoint WiMAX will be finally deployed in order to provide broadband access to on board 11 offshore oil rig platforms in Gulf of Mexico [18].

2.1.8.3 Accton and Alvarion in joint venture to produce WiMAX products

Accton, the Taiwanese network and Internet communications OEM, has entered a partnership with Alvarion, the Israeli wireless technology company, to produce WiMAX base station equipment and consumer premise equipment (CPE) products. The products are expected to enter the market in 2007.

Also different companies like Intel and Fujitsu are making efforts to come with mobile WiMAX. Also Nortel is playing an important role in bringing mobile WiMAX in market.

2.2 INTRODUCTION TO WiFi

WiFi when launched was initially intended for commercially cashed register systems. These days it provides wireless broadband access headed for any user with wireless connectivity technology within a diminutive range. Characteristically, a WiFi signal has a highest range of 150 feet inside and 300 feet in the open.

WiFi is an ellipsis for wireless fidelity. Wi-Fi enabled device can be connected to the network if there is any access point available within the specified range of WiFi. Hotspot is the region or area covered by several access points so WiFi enabled equipment is assumed to work in hotspots. Hotspot can be small similar to one in a single office or can as large as to coverthe whole building.[21]

2.2.1 Working Of WiFi

A Wi-Fi net is consisted of access points which make a hotspot. Each AP broadcasts is Service Set Identifier (SSID), by means of beacons, the broadcast rate of beacons is 100 ms. The beacons are transmitted for a petite duration and are transmitted at 1 Mbit/s. The client can decide whether to connect to a specific AP or not depending upon the pre known settings. Client has an option to choose any connection to any AP depending upon the signal potencies of the APs. [20]

2.4 GENERAL COMPARISON OF WiFi AND WiMAX

Table2-1: Comparison of WiFi and WiMAX

	WiFi	**WiMAX**
Deployment	Use of cables is eradicated	Use of cables is eradicated
Cost	Cost is reduced by eradicating cables.	Cost is reduced by eradicating cables. But WiMAX is more costly then WiFi because of hardware differences.
Forums	WiFi Alliance	WiMAX Forum
Mobility	Roaming is supported.	Roaming is somehow supported in portable WiMAX. But not yet fully implemented in the form of mobile WiMAX.
Network interruption	WiFi is signal is interrupted more as it works is in frequency range of 2.4 GHz and most of devices also us same frequency range for signals transmission.	WiMAX signal is not interrupted by ordinary home appliances as it works in larger frequency ranges, even interruption is also not very effective in

		unlicensed frequcncy range.
Security	Transmissions are not very much secure, as it use Wireless Equivalent Privacy (WEP), that is easily breakable.	Transmissions are far more secure because enhanced encryption techniques are used.
Congestion	Congestion occurs frequently as many access points access the same channel.	Congestion is not a problem as each access point can have its own different channel.
Range	Up to 300 feet.	Up to 30 miles.
Interference	Greater interference due to overlapping channels	Lesser interference
Network	Focuses on LAN	Focuses on MAN
Standard	802.11	802.16
Bandwidth	Fixed: 20 MHz /52 sub carriers [28]	Variable: 1-28 MHz /256 sub carriers [28]
Spectral efficiency	2.7 mbits/s/hz	3.1 to 3.8 mbits/s/hz
Duplexing	TDD	TDD, FDD, HFDD
Tx dynamic range	Tx power fixed	50 dB range
Spectrum	Unlicensed	Licensed and unlicensed
EVM requirements	-25 dB	-31 dB
Rx noise figure	10 dB maximum	7 dB maximum
Guard interval	Fixed at ¼ * symbol time	Variable ranges from 1/32 to ¼ * symbol

		time
Availability	WiFi equipment is available in market due to its lower costs.	WiMAX equipment is not as much common as that of WiFi because of its higher costs. Also this can be because WiMAX is still in development stages.

2.5 COMPARATIVE ANALYSIS

WiFi stands for Wireless Fidelity while that of WiMAX stands for Wireless Microwave Access. Prior is IEEE 802.11 standard and focuses on LAN while WiMAX is IEEE 802.16 standard and focuses on WMAN (Wireless Metropolitan Area Network). WiFi basically provides backhaul while that of WiMAX provides last mile access. With respect to wireless sensor networks, if we compare both the standards, WiFi protocol is being enabled on sensor networks while that of WiMAX is not. Reason could be that WiFi was introduced before WiMAX and was intended for wireless local area networks where mostly sensors are in use. Although there are other standards like Zigbee and Bluetooth that are specifically designed to work for sensors, but at the same time WiFi is being deployed and can be deployed for sensor networks to increase range and there where efficient communication is required. In this regard from [22] we can see first IP powered wireless sensor networks using WiFi. Also reference to [23], WiFi is being used by WSN for different monitoring and controlling purposes.

If we come to WiMAX, it was introduced to provide last mile access and eradicate the use of cables, because wireless access is more cost effective in providing communication means to remote areas, where cables either cannot be used or can be very expensive. Now if we see there are currently three standards for three different types of networks that is Zigbee (802.15.4) used for WPANs, WiFi (802.11) used for WLANs and WiMAX (802.16) used for WMANs. Zigbee and WiFi both are used for sensor networks but WiMAX is not. The question arises why not, if we see both the standards that is WiFi and WiMAX we will come to know both have the same job but with different areas of operations. In both standards there are little differences between there signal to noise ratio and EVM requirements, WiFi uses TDD which can also be used by WiMAX, equipment is different but TeleCIS announced there single chip solution that will support both WiMAX and WiFi, WiFi operates in unlicensed frequency range and WiMAX can also operate in unlicensed frequency range. But in presence of all these little differences WiMAX provides very larger range than that of WiFi. WiMAX can provide range upto 300 miles while that of WiFi can provide only 300 ft. So if there are not much difference between WiFi and WiMAX then why not WiMAX can be enabled on sensors? There could be a reason that sensors are very small in size and WiMAX protocol is much bigger to implement on sensors. But if we ignore such constraints like size and power of sensors than it may be possible to WiMAX protocol on sensors.

Chapter 3

SIMULATIONS AND PARAMETERS

We have compared the two most important constraints of Doctor Patient Application these are Quality of Service and Reliability. So for comparing these parameters we are have calculated their throughput and delay.

3.1 SIMULATION TOOL

For simulating different scenarios in this project, MATLAB 7.0 is used. MATLAB is used because of its better support for graph generations.

3.2 THROUGHPUT

It is the amount of data per time unit that is delivered to the receiver. If we have n number of bits to be transmitted and r is the number of error correcting bits padded with the original data then total bits transmitted would be equal to:

$$\text{total length of packet} = n+r$$

Let k be the number of transmissions that occurred while sending 'p' number of packets then throughput would be calculated as:

$$\text{throughput} = \frac{(n+r)}{(n+r)*k+(n+r)}$$

Where k = Number of transmissions

Let suppose we need to send a packet from mobile station to base station and the packet is not acceptable by the destination or either is not received at the destination and it is retransmitted then k would be equal to 1 and throughput will start decreasing as retransmissions increase and if k=0 that means no transmission was required so in that case through put would be maximum that is 1.

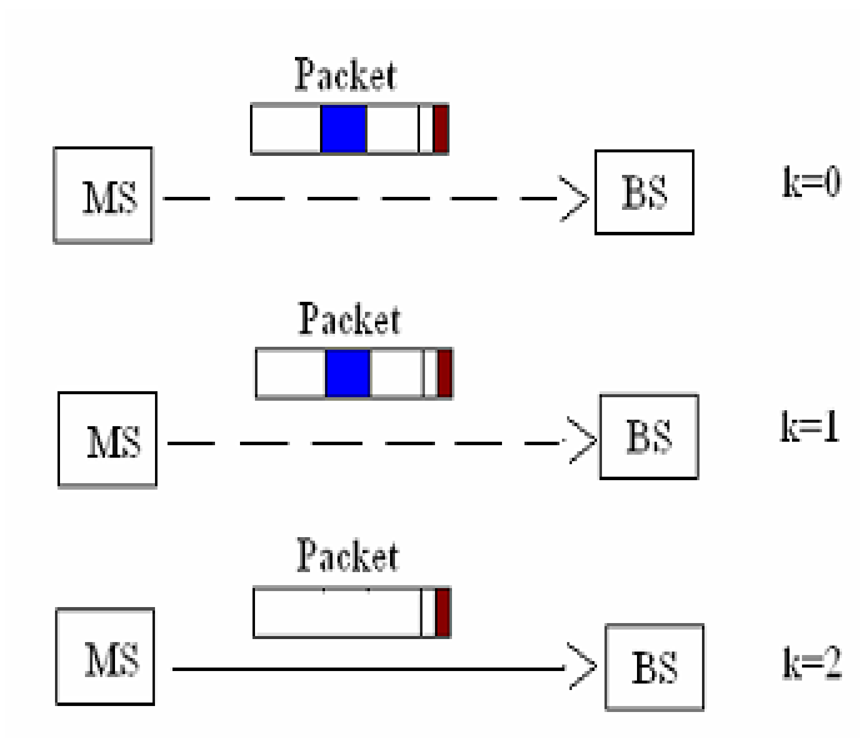

Figure 4: Throughput

3.3 DELAY

Delay is the measure of time a packet or data takes to reach its destination. Delay is directly proportional to the number of retransmissions.

Delay α number of retransmissions

As errors increase delay becomes more noticeable. We can calculate total delay of transmitting a particular amount of data using formula:

Total Delay = RTO * k +1

Where RTO = Return Time Out
And k = Number of transmissions

Let suppose we need to send a packet from mobile station to base station and the packet is not acceptable by the destination or either is not received at the destination and it is retransmitted then k would be equal to 1 and delay will start increasing as retransmissions increase and if k=0 that means no transmission was required so in that case through put would be maximum that is 1.

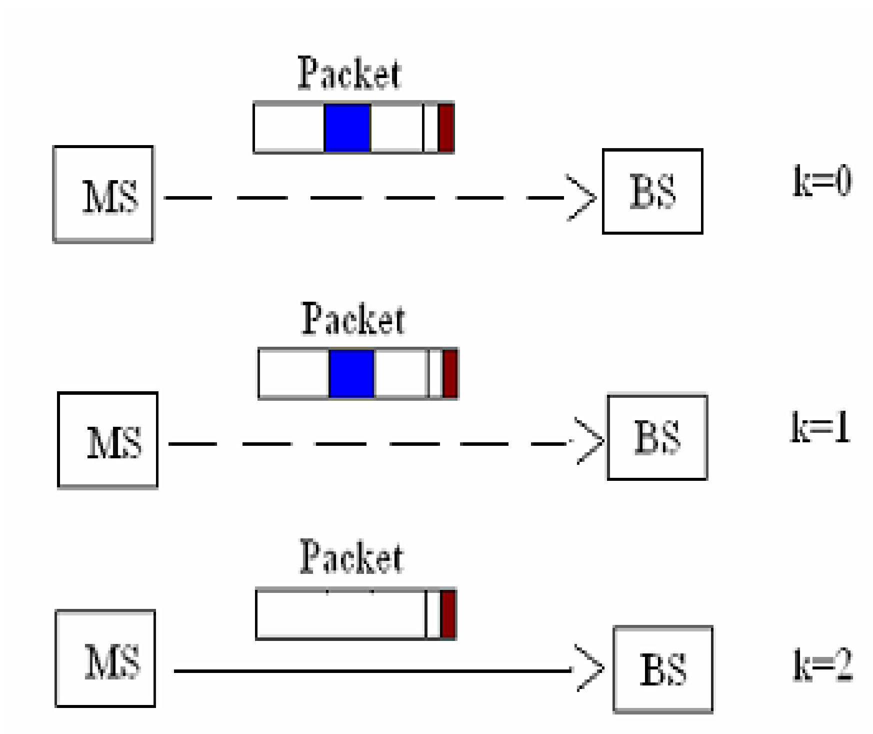

Figure 5: Delay

3.4 RELIABLILITY

Reliability is one the most important factor in wireless communication which affects the performance of a network. We have compared reliability of WiFi and WiMAX on the basis of throughout achieved and delay observed when certain probability of errors is in attendance. Both the standards i.e. 802.11e and 802.16e have different approaches to achieve a reliable communication which is discussed in the later sections. Both throughput and Delay are measured against probability of errors and then we compare results.

3.4.1 Reliability in 802.11e

Reliability is achieved in 802.11e by using Selective Repeat method for retransmissions and Convolution encoding as a forward error correction technique

3.4.1.1 Sliding Window Protocol (Selective Repeat method)

In selective repeat method a window of data packets is maintained with the sender. It keeps on sending the packets till the window is empty. After the window is being emptied it waits for the receiver’s acknowledgment. Receiver sends acknowledgement informing sender about the packets which where either not received or received corrupted and are not correctable. Then sender increases its window size to the new size and includes those packets to the window which were are to be retransmitted.

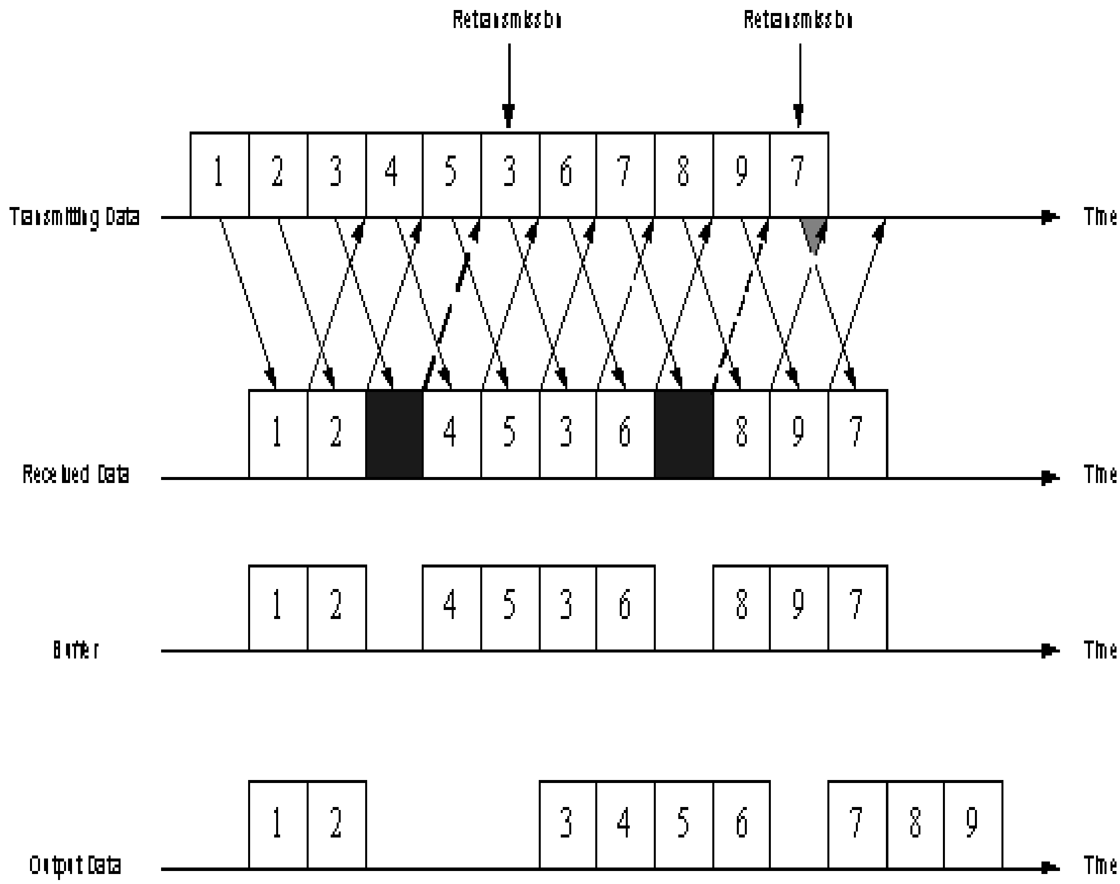

Figure 6: Selective Repeat Protocol
Reference: Simulation of ARQ from www.ececs.uc.edu

3.4.1.2 Convolution Codes

Convolution codes are used to correct errors up to infinite number. These codes first calculate the hamming distance and then correct half the number of errors detected. The number of errors that can be detected and corrected depends upon the decoder on the receiving side. We have assumed and tested our simulations for the decoders that can correct 16 errors. This is known as correcting capability‘t’ of convolution codes. It is represented as:

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor.$$

Where d = hamming distance

Hamming distance is calculated by XORing the input received and predetermined values.

3.5.1 Reliability in 802.16e

In WiMAX reliability is achieved using HARQ (Hybrid Automatic Repeat request). HARQ is basically a combination of ARQ and FEC (Forward Error Correction). Both are combined to work together. ARQ is used for requesting retransmissions of data while FEC is achieved using Reed Solomon Codes. Hybrid ARQ is a MAC layer retransmission strategy i.e. it does not involve higher layers for retransmissions so reduced delay is observed while using HARQ.

In hybrid ARQ both the data block and error correction code is encoded prior to transmission. When data is received, error correction block is decoded first and if all the errors are correctable, receiver corrects them and acknowledgment is sent. And if errors are not correctable then retransmission is requested.

3.5.1.1 Stop and Wait Protocol:

In ARQ data is sent to the receiver and sender waits for acknowledgement from the receiver. If positive acknowledgment is received by the sender then it sends the next frame, else if negative acknowledgment is received it retransmits the previous frame. If acknowledgment is not received than it retransmits the previously sent frame again after the time-out period which it equal to the round trip time.

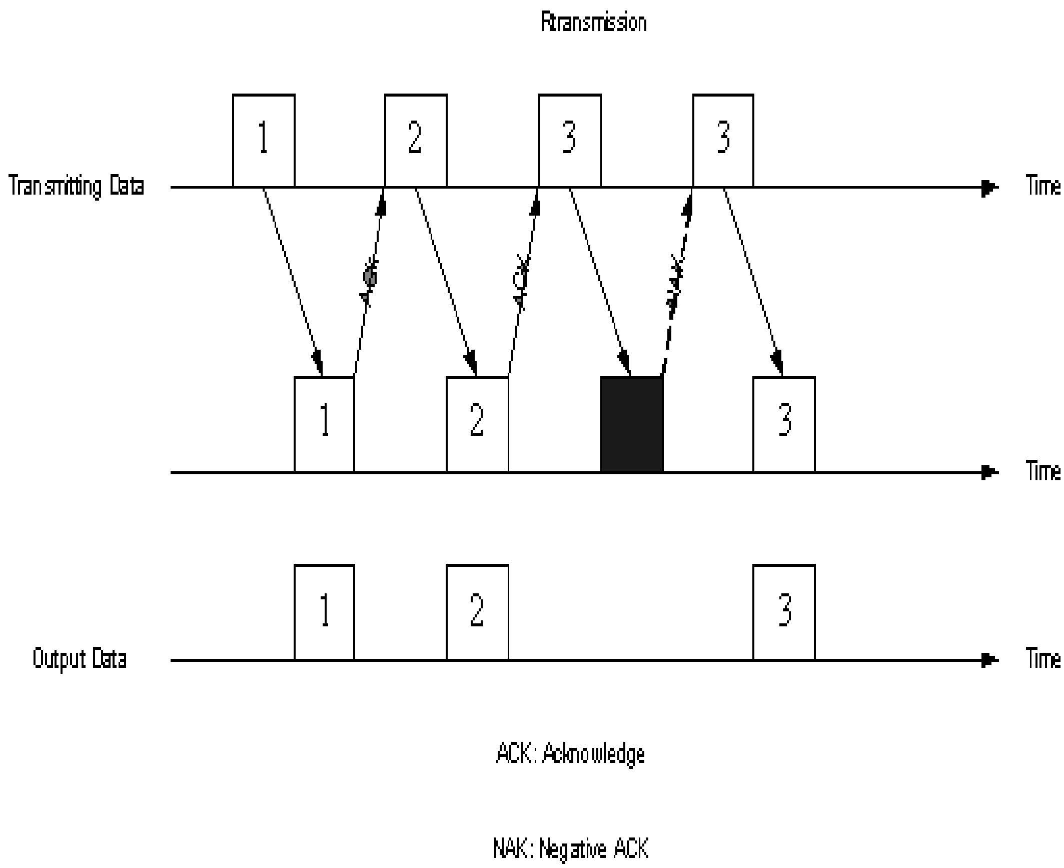

Figure 7: Stop and Wait Protocol
Reference: Simulation of ARQ < www.ececs.uc.edu>

3.5.1.2 Reed Solomon Encoding:

Reed Solomon encoding is used as method for forward error correction in 802.16e. As shown in figure 8.

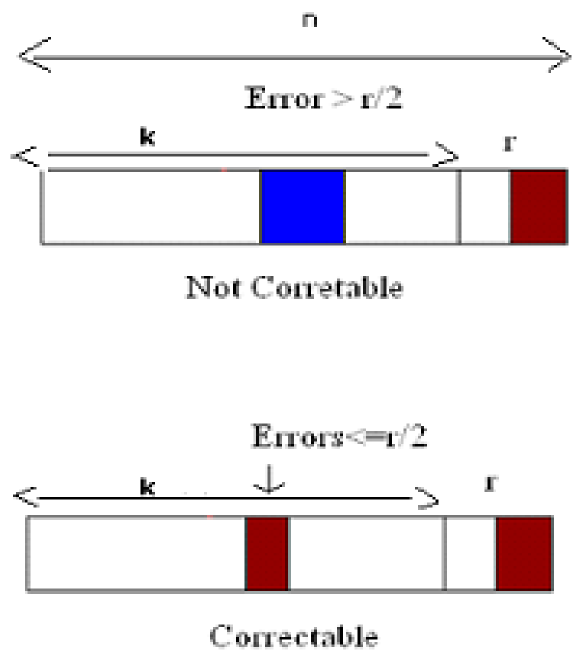

Figure 8: Reed Solomon Encoding

As in above figure if we have 'n' number of total bits sent and 'k' is the significant number of bits then error correcting capability of Reed Solomon Code is determined by:

$$\frac{n-k}{2}$$

In Reed Solomon encoding operating on 8 bit symbol has:

$$n = 2^8 - 1 = 255 \text{ symbols per block}$$

A commonly used code encodes $k = 223$ 8-bit data symbols plus 32 8-bit parity symbols in an $n = 255$-symbol block; this is denoted as a $(n,k) = (255,223)$ code, which is capable of correcting up to 16 symbol errors per block.

3.6 QUALITY OF SERVICE

As doctor patient application requires high level of prioritization because sensors with higher priority need to be serviced earlier then those with lower priorities. Both 802.11e and 802.16e provide enough capability to support prioritization in context to quality of service.

3.6.1 Quality of service in 802.11e:

802.11e uses Hybrid co-ordination function to share the medium between multiple stations. Hybrid co-ordination function can be either EDCA (Enhanced Distributed Channel Access) or HCCA (HCF controlled channel access). With respect to our application EDCA provides prioritization capability. Quality of service is also provided in HCCA mode but as we have predefined priorities of patients in our application so we will use EDCA.

3.6.1.1 Enhanced Distributed Co-ordination function:

EDCF is DCF with some of the elements of the MAC parameterized per-TC. With EDCA traffic of high priority has lesser chance to wait for accessing the channel. In EDCA each user is assigned a bounded time interval called transmit opportunity (TXOP) based upon its priority. Higher the priority greater would be the transmit opportunity. Each sender sends as many frames as it can in this interval of time. The use of TXOPs reduces the problem of low rate stations gaining an inordinate amount of channel time in the legacy 802.11 DCF MAC.

In EDCA 8 different priorities for different types of traffic categories are provided. In our application we have 4 different types of patients so EDCA can easily solve the problem of prioritization. Traffic categories in EDCA are represented as TC.

3.6.2 Quality of service in 802.16e:

802.16e provides optimal scheduling of time resources on frame by frame basis. Each station provides it's some quality of service parameters to the base station prior to communication on the basis of which time slots are assigned to different categories of stations. Each time frame is send is send to the base station, quality of service parameters are also provided. These quality of service parameters determine the priority of user and also the time for which the slot is needed and for what type of traffic will be sent. Scheduling is done for both uplink and downlink connections. In WiMAX quality of service is achieved more as each connection is associated with specific type of service. A service flow database is always managed by bas station in order to know have information about available services and track different types of services provided to different stations.

3.6.2.1 Different Service Classes

- Unsolicited Grant Services (UGS): UGS is designed to support Constant Bit Rate (CBR) services, such as Voice Over IP (VoIP). *(CBR)*
- Real-Time Polling Services (rtPS): rtPS is designed to support real-time services that generate variable size data packets on a periodic basis, such as MPEG video or VoIP.
- Non-Real-Time Polling Services (nrtPS): nrtPS is designed to support non-real-time services that require variable size data grant burst types on a regular basis.
- Best Effort (BE) Services: BE services are typically provided by the Internet today for Web surfing. *(UBR)*

Chapter 4

ALGORITHMS

4.1 ALGORITHM FOR RELIABILITY OF 802.11e

Both forward error correction and selective repeat method are used in 802.11e standard to achieve reliability. We have assumed time period for sending a packet and receiving acknowledgment for that packet to be 2 sec. Then 15 packets are transmitted with a window size of 5. After sending every five packets, sender waits for acknowledgment and is informed about the packets which need to be retransmitted. Then the new window includes those packets which are to be retransmitted. But the window size remains same. On the other side when packet is received they are checked for the errors which corrected using forward error correction. Errors can be corrected maximum up to 8, 16 or 32. Here are the algorithms for both sending and receiving sides.

Pseudo Code:

Sender Side

```
Function sender ( )
String  data= "Data Packet"
While ( datalength  isnotequalto null )
        Send ( data )
                If windowsize==0
                        Receive( ack, packets )
                        If packets isnotequalto 0
```

```
            Resend ( packets )
            Datalength+1;
        Else
            Movewindowforward()
        End
    Else
        Windowsize-1
    End
    Datalength-1
End
End
```

Receiver Side

```
Function Receiver ( )
Receive( data )
Checkerrors( data )
If errors are correctable
    Correcterrors ( )
Else
    Send ( ack, packets)
End
Calculatethroughput ( )
Calculatedelay ( )
```

End

4.2 ALGORITHM FOR RELIABILITY OF 802.16e

802.16e uses a combination of ARQ and FEC. We have assumed time period for sending a packet and receiving acknowledgment for that packet to be 2 sec. Then 15 packets are transmitted with a window size of 5. After sending every packet, sender waits for acknowledgment and is informed whether retransmission of previous packet is required or not. The decision of retransmission is made by receiver after checking number of errors occurred if all errors are correctable no retransmission is required retransmission would be requested. Sender can also make decision to retransmit packet if time-out period is over. Here are the algorithms for both sending and receiving sides.

Pseudo Code

Sender Side

Function Sender()

String block= data + error correcting code

While (datalenght isnotequalto zero)

Send (block);

Receive(ack);

If (ack isequalto No retransmit)

Datalenght+1;

Else

Resend(block);

Datalength = Datalength ;

End

End

Receiver Side

```
Function Receiver ()
Receive (block)
    For I =1: blocklength
    Checkerrors ();
        If errors > 16
            Ack = positive;
            Send (Ack, nextblock);
         Else if
            Errors <16
            Ack = negative;
            Send (ack);
        End
    End
End
```

4.3 ALGORITHM FOR 802.16e QOS:

Four different types of stations are present each user has different priority based on the service flow ID it sends to the base station. Time slot is being allocated to users on request and this time slot can grow or shrink as the user needs changes. Users request base station for allocating time slot and it is allocated to the user with higher priority, if time slot is available and if its not then user will have to wait. Waiting time will be more for lower priority users if channel is already occupied by higher priority users. When higher priority user requests for channel access then it will have to wait for the maximum of 5 ms. Because the maximum frame time in 802.16e is 5 ms and a slot can be extended maximum up to this time.

Pseudo Code

Base Station Side

```
ReceiveRequest (ServiceFlowID, ConnectionID)
        If SlotAvailable = yes
                AssignSlot (SFID)
        Else SlotAvailable = No
                Priority = CheckPriority (SFID)
                        If Priority = High
                                Wait (minimum slot time);
                                PreviousSFID = ChannelAccessCheck ( );
                                StopCommunication (PreviousSFID);
                                AssignSlot(SFID)
                                        If ChannelNeed(SFID) = Over
                                                StartCommunication(PreviousSFID);
                                        End
                        Else if Prioriy = Low
                                Wait;
                        End
```

End

End

4.4 ALGORITHM FOR 802.11e QOS:

Stations are randomly generated with different traffic categories. Then these stations are prioritized on the basis of their traffic categories. Each station occupies medium for it's transmit opportunity. For instance if any channel is being pre-occupied by the lower priority station and a station with higher priority needs any service from base station then it waits the maximum up to TXOP of lower priority station or the random backoff time that is lesser for high priority users and greater for low priority users. All other lower priority stations wait their backoff time that is much greater until incoming higher priority channel is not served by the base station.

Pseudo Code

Base Station Side

RecieveRequestforService(TC, StationID, TXOP)

If ChannelAvailable = Yes

CheckPriority(StationID)

If Priority = high

AssignChannel(StationID, TC)

Else

WaitRandomBackoff(StatioID)

End

Else if ChannelAvailable= No

CheckPriority(TC)

AssignChannel Where Priority = High

End

End

Station Side

SendRequest(StationID, TC)

If ChannelAvalable = Yes

StartTransmitting

Else

Wait Random Backoff Time;;

End

End

Chapter 5

RESULTS AND ANALYSIS

5.1 RELIABILITY

Reliability techniques of both 802.11e and 802.16e are simulated and then results are shown by the graphs in which probability of errors or Bit Error Rate is taken along x-axis while that of delay and through put are taken along y-axis simultaneously. And then with the help of graphs results are analyzed.

5.1.1 At Constant Bit Error Rate

5.1.1.1 Through put

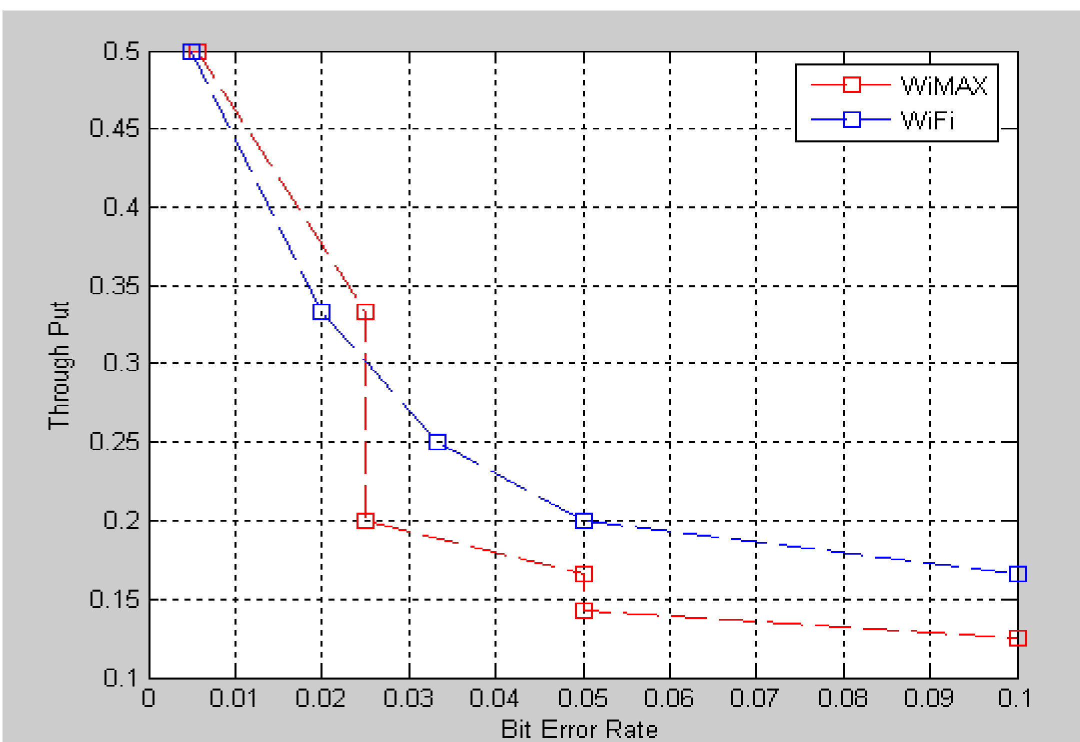

Figure 9: Throughput Graph

In the above results we have two lines generated red for the WiMAX and blue for WiFi. Squares on these graphs shows the points at which some of the packets are transmitted without errors and then if error occurs throughput starts decreasing with the increase in the error rate. While in case of WiFi squares represent a complete window transmitted.

5.1.1.2 Delay

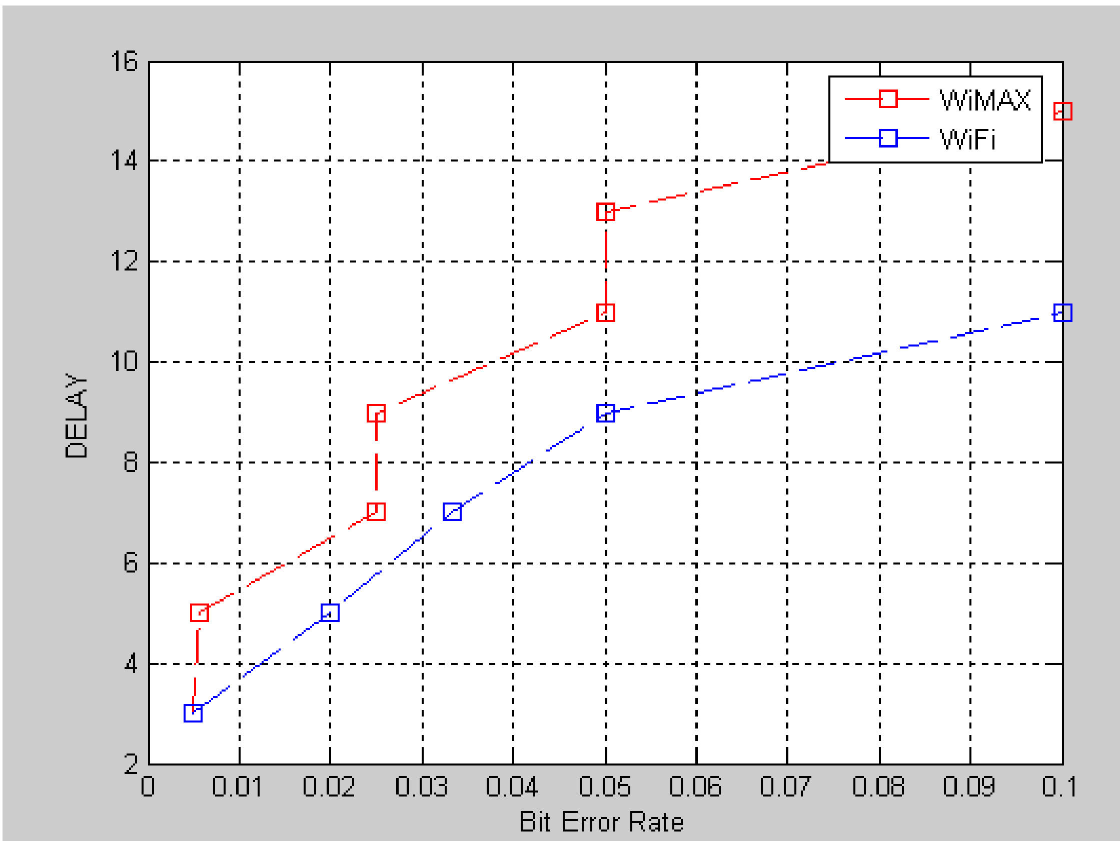

Figure 10: Delay Graph

Again squares on these graphs shows the points at which some of the packets are transmitted without errors and then if error occurs delay starts increasing with the increase in the error rate. While in case of WiFi squares represent a complete window transmitted.

5.1.2 With different probabilities of errors

5.1.2.1 Through put

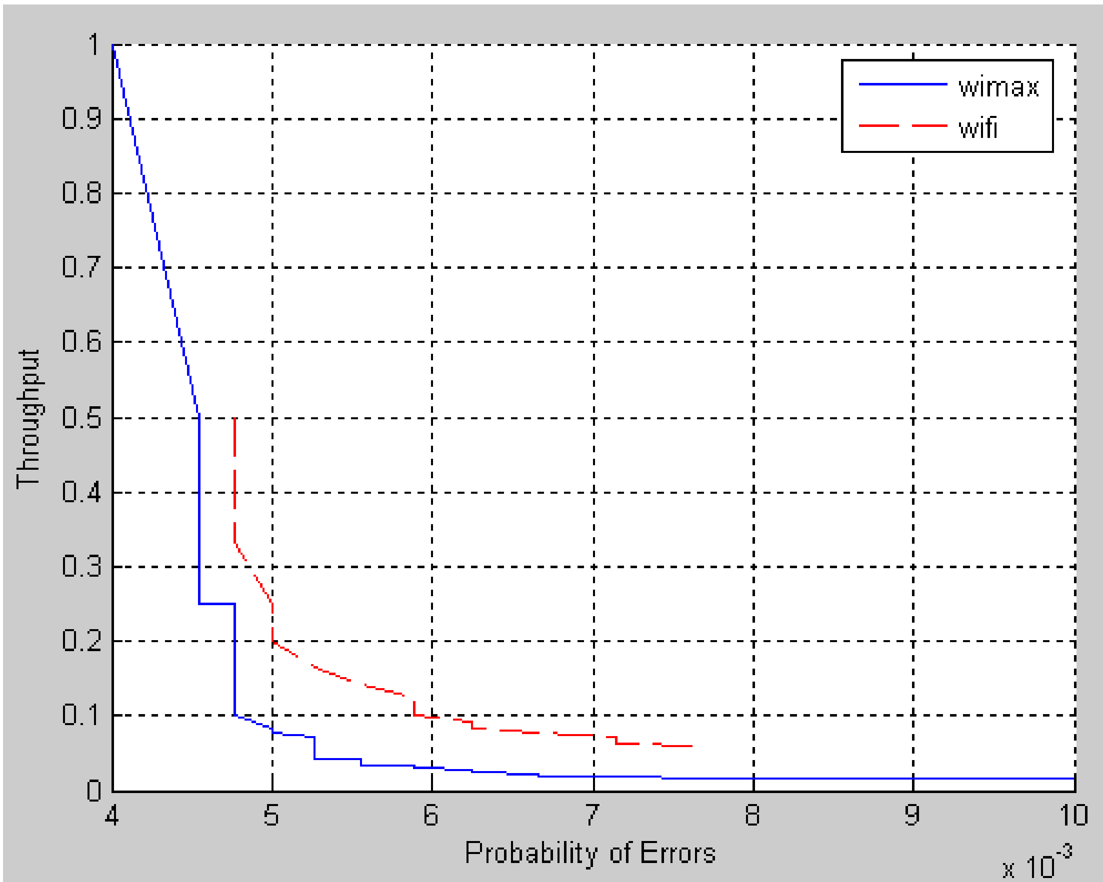

Figure 11: Throughput Graph

In this graph averaged dashed line represents WiFi while that of averaged continuous line represents WiMAX so when we compared results for these both at different error rate again we see that WiFi is providing more overall through put than that of WiMAX.

5.1.2.2 Delay

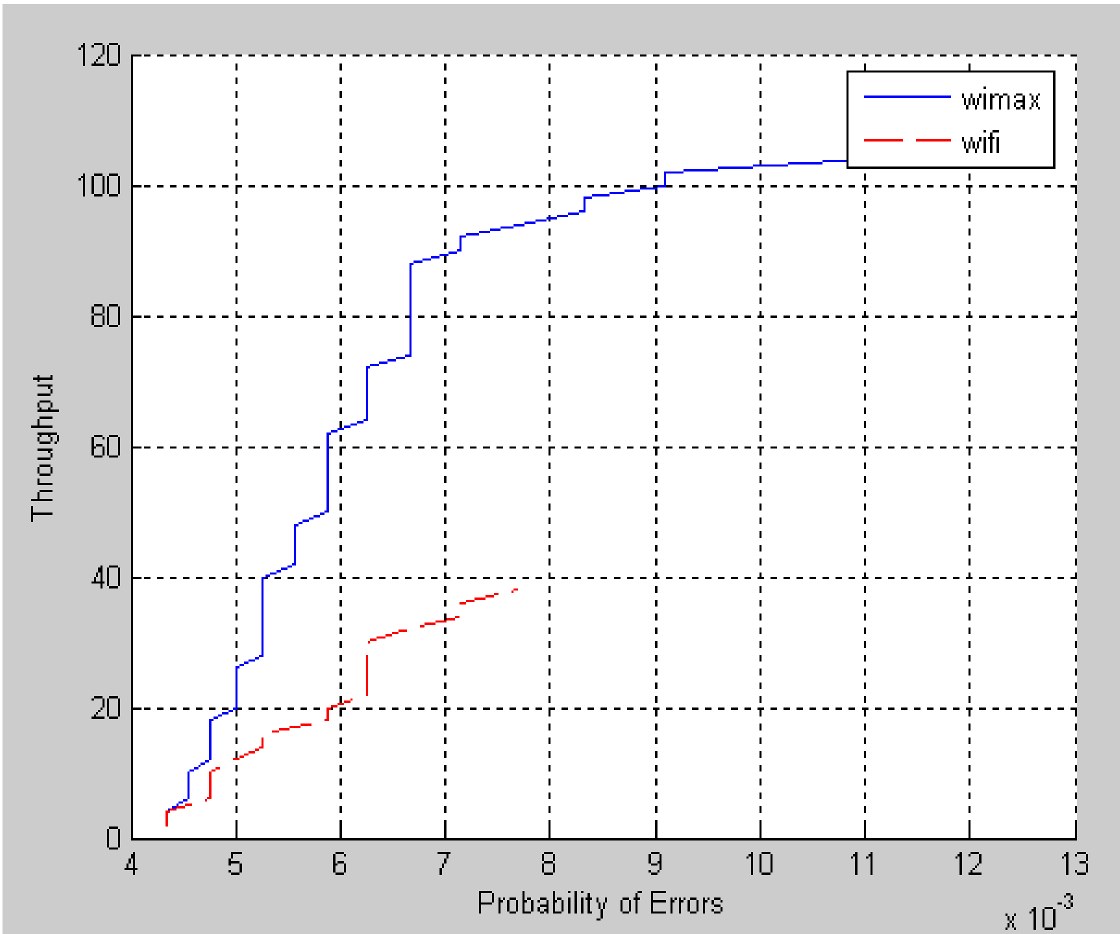

Figure 12: Delay Graph

In this graph averaged dashed line represents WiFi while that of averaged continuous line represents WiMAX so when we compared results for these both at different error rate again we see that WiFi is providing less delay than that of WiMAX.

5.1.3 Analysis

If we see both of these graphs we come to know that by using WiFi lesser delay and more through put can be achieved. This is because the difference in the ARQ method used by both the standards. As in ARQ of 802.16e sender has to wait for the acknowledgment of each and every packet and because of this waiting time sender observes greater delay than that of WiFi and also lesser through put. While in that of 802.11e sender does not wait for each and every packet it sends, it only waits for acknowledgment to be received after a complete window is sent. That's why 802.11e has lesser delay than that of 802.16e.

5.2 QUALITY OF SERVICE

Basically for analyzing quality of service a 3d graph is generated by the help of which we can determine that at what time which priority user gets access to the channel.

5.2.1 QoS results of 802.11e

When some high priority or low priority user sends a request when channel is already occupied by some other user then we predict the behavior of that user with the help of this 3D graph in which priority is shown on x-axis, turn is shown on y-axis and delay is taken along z-axis. The priority decreases along x-axis while delay and turn increase along their respective axis.

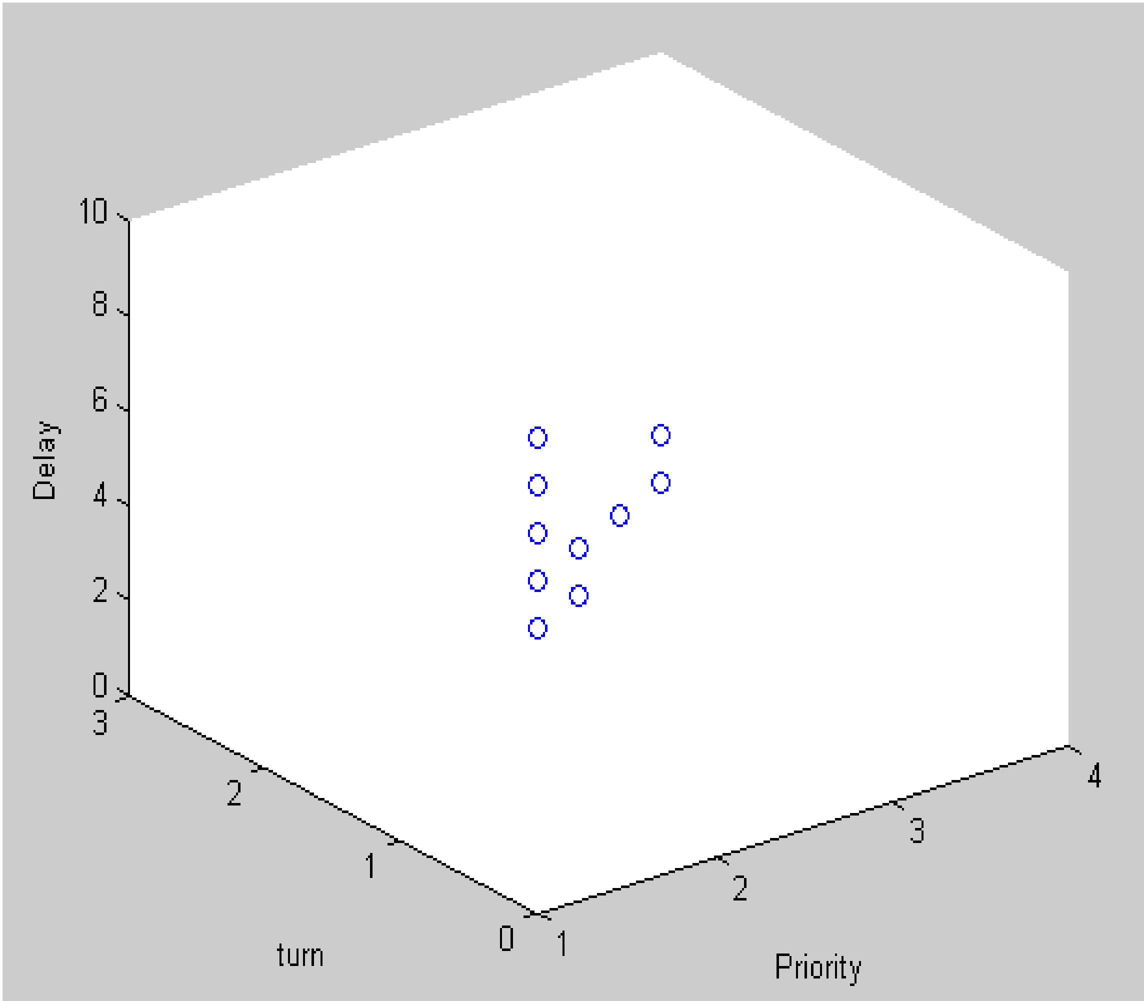

Figure 13: WiFi QoS

5.2.2 QoS results of 802.16e

When some high priority or low priority user sends a request when channel is already occupied by some other user then we predict the behavior of that user with the help of this 3D graph in which priority is shown on x-axis, turn is shown on y-axis and delay is taken along z-axis. The priority decreases along x-axis while delay and turn increase along their respective axis.

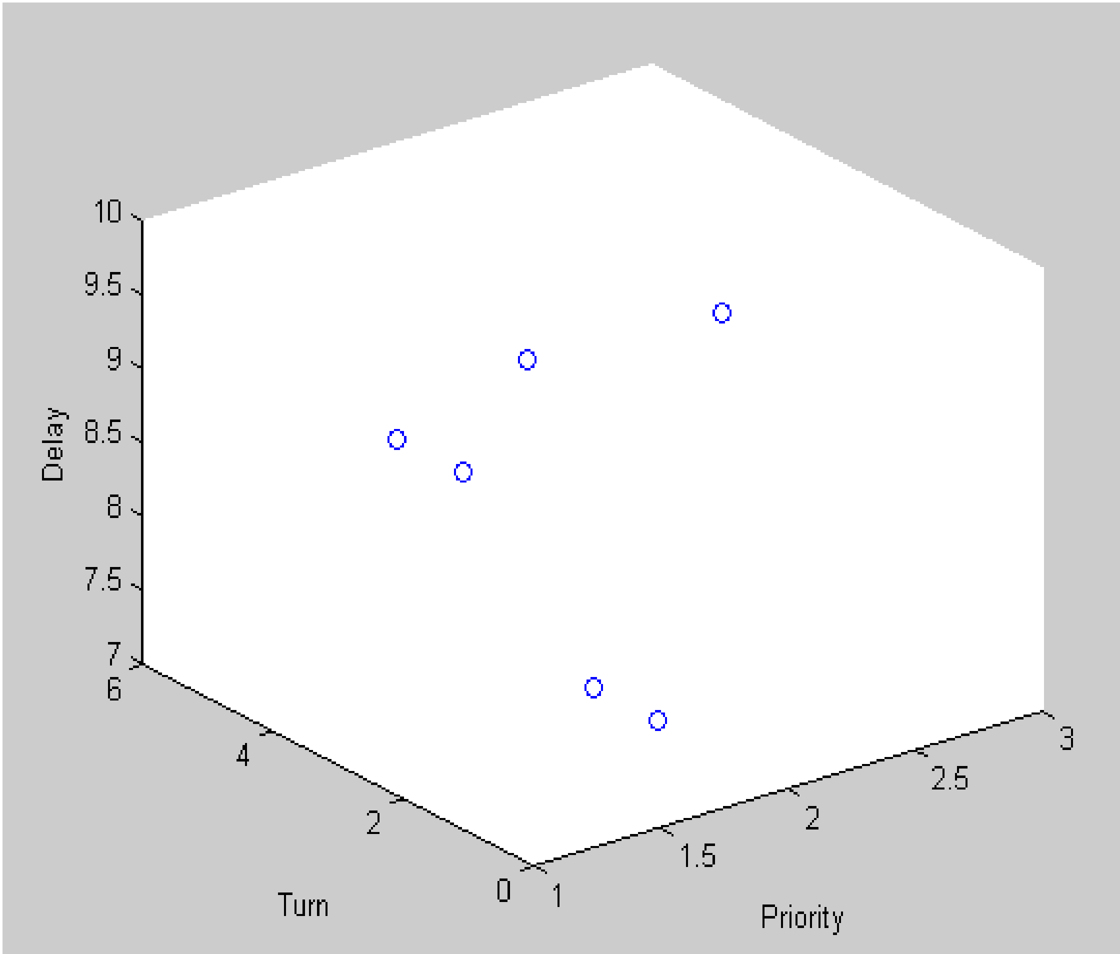

Figure 14: WiMAX QoS

5.2.3 Analysis

By analyzing both of the graphs for QoS of 802.16e and 802.11e we can conclude that 802.16e has better quality of service than that of 802.11e because in 802.11e not only low priority user has to wait for high priority user but it is also possible that high priority user might be waiting for some low priority user. But in 802.16e although low priority user might wait for the high priority traffic but high priority traffice would never wait for the low priority traffic.

Chapter 6

CONCLUSION AND FUTURE DIRECTIONS

6.1 CONCLUSION

After comparing the most significant parameters of Doctor Patient application i.e. Quality of Service and reliability. We have seen that reliability of 802.11e is more than that of 802.16e but at the same time 802.16e has better support for quality of service, which is the far most requirement of our application. Because it would never be acceptable that a patient in a critical condition waits for a patient in a non critical condition to have channel access. So we can say that for a doctor patient application 802.16e would be more suitable because of its better QoS support than 802.11e.

6.2 FUTURE DIRECTIONS

After the comparison of MAC layer, this project can be continued for the comparison of different security and authentication schemes used by both of the standards. Also different physical layer issues like their modulation schemes can be compared. More over different hardware related issues pertaining to sensors can be studied with respect to Doctor Patient application where sensors should be able to communicate from longer distances but also keeping in mind the different constraints of sensors like their power, energy and battery issues.

REFERENCES

1. Archana, B. Vijay, A. S. P. “Sensor Networks: An Overview” Department of Computer Science, University of California, Davis, CA 95616 Email: {bharathi, ponduru}@cs.ucdavis.edu
2. David, J. Tim, S. Russ, F. Dan, F. Rob, R. Jay, L. {April 8, 2005},“TrueMobile: A Mobile Robotic Wireless and Sensor Network Testbed”, University of Utah University of Utah Flux Group Technical Note FTN–2005–02
3. Katayoun, S. Jay, G. Vishal, A. Gregory, J. “Protocols for Self-Organization of a Wireless Sensor Network”, Electrical Engineering Department,UCLA Box 951594, Los Angeles, California, 90095-1594 {sohrabi, gao, vishal, pottie} @ ee.ucla.edu
4. Amit, D. “WiMAX Delivering on the promise of wireless broadband”, April 21st 2007, Director, Infrastructure Vertical Markets. Xilinx, Inc. amit.dhir@xilinx.com <http://www.xilinx.com/publications/xcellonline/xcell_57/xc_pdf/p010-013_57-wimax.pdf>
5. Daniel, S. “WiMAX Operator’s Manual; Building 802.16 Wireless Networks” 2nd Edition
6. Dr. Syed-Amr, El-H. President and CEO WiLAN “WiMAX: What’s all the Fuss about”, April 28th 2007 <www.wi-lan.com/library/WiMAX_Intro_CES_2005.pdf>
7. “The 802.16 WirelessMAN™ MAC: It’s Done, but What Is It?”, June 18th 2007 <http://www.ieee802.org/16/docs/01/80216-01_58r1.pdf>
8. “WLAN-WMAN Evaluation”, January 21st 2007 <www.elet.polimi.it/upload/antlab/RESEARCH/WIRELESS/WLAN-WMAN/EVALUATION/WIMAX/index.html>
9. “How WiMAX Works”, January 26th 2007 <http://computer.howstuffworks.com/WiMAX1.htm>
10. “Wireless Sensor Network News”, March 12th 2007 <http://www.ece.ucdavis.edu/~lpszumel/wsn/wsnn.html>
11. Working of WiMAX, February 7th 2007 <http://computer.howstuffworks.com/WiMAX1.htm>

12. “Personal Computer Memory Cards Association”, February 15th 2007 <http://en.wikipedia.org/wiki/PCMCIA>

13. “The WiMAX Fixed to Mobile Evaluation”, June 11th 2007 <http://www.wi-fiplanet.com/columns/article.php/3577391>

14. “Fixed or Mobile WiMAX”, April 20th 2007 <http://www.researchandmarkets.com/feats/download_sample.asp?report_id=316593&file_name=Fixed%20or%20Mobile%20WiMAX&file_ext=pdf>

15. “WiMAX Protocol Stack”, May 20th 2007 <http://www.tutorialreports.com/wireless/WiMAX/protocol.php?PHPSESSID=7a226a1ce5f95bc15ecfea6e882f9eb4>

16. “WiMAX providing broadband Services”, July 21st 2007 <http://www.tek.com/products/communications/technologies/broadband/WiMAX.html>

17. “North America’s first mobile WiMAX deployment”, January 4th 2007 <http://www.3g.co.uk/PR/April2006/2948.htm>

18. “Communication Solutions”, January 4th 2007 <http://www.tmcnet.com/enews/e-newsletters/Communication-Solutions/20060927/2811-redline-system-be-used-WiMAX-deployment-oil-rigs.htm>

19. “Accton and Alvarion in joint venture to produce WiMAX products”, January 13th 2007 <http://investintaiwan.nat.gov.tw/en/news/200611/2006112701.html>

20. “Wireless Networks”, December 19th 2006 <computer.howstuffworks.com>

21. “WiFi (Wireless Fidelity)”, November 11th 2007 <en.wikipedia.org/wiki>

22. “First-ip-powered-wireless-sensor-network-arrives”, May 3rd 2007 <www.fierceWiFi.com/story>

23. “Wireless Sensor Networks”, November 19th 2006 <www.ece.ucdavis.edu/~lpszumel/wsn/wsnn.html>

APPENDICES

Appendix A

Function ARQ( ):

```
function [PE,DE,tp]=arq()
string =
'abcd5efgh5ijkl5mnop5qrst5uvwx5yzab5cdef5ghij5klmn5opqr5stuv5wxyz5abcd5efgh
5abcd5efgh5ijkl5mnop5qrst5uvwx5yzab5cdef5ghij5klmn5opqr5stuv5wxyz5abcd5efg
h5'
count=0;
max=1;
limit=5;
len=0;
check_code=1;
n=0;
val=0;
q=0;
ct=1;
%persistent next_element;
next_element=1;
while (len~=length(string))

 if limit>length(string)
     remainder=limit-length(string);
     length_remainder=5-remainder;
     new_len=length(string)-length_remainder+1;

     for j=new_len:length(string)

         pdu(j)=string(j);
         count=count+1;
     end
     count;
     pdu
     val=val+1;

%n=n;
 %    next_element=next_element+1;
% next_element=next_element+1;

%  if count>16
```

```
%      n=n+1;
%  end
 error=error_array(next_element);
 next_element=next_element+1;
     [ack,count,throughput]=arq_reciever(pdu(j),check_code,n,error)
     delay=n*2;

     %n=n+1;

%   n=n+1;
%   n=val;
  if count>16
     n=n+1;
   end
     for j=1:n
    tp(n)=throughput;
    PE(n)=1/(10*count);
    DE(n)=delay;

     end



     delay1=val*2+1;
     throughput1=(223+32)/((223+32)*val+223+32);

     for j=1:val

           tp1(val)=throughput1;

           PE1(val)=1/(10*count);

           DE1(val)=delay1;


     end


     break;


 else

     for i=max:limit
```

```
        pdu(i)=string(i);


      end


  end

  val=val+1;
  pdu

  %next_element=next_element+1;

  error=error_array(next_element);
  next_element=next_element+1;

  [ack,count,throughput]=arq_reciever(pdu(i),check_code,n,error)
 %  n=n+1;
  if count>16
       n=n+1;
  end

  delay=n*2;
  %n=n+1;

  for j=1:n

      tp(n)=throughput;

      PE(n)=1/(10*count);

  end


  for o=1:n

      DE(n)=delay;

  end

 delay1=val*2+1;
 throughput1=(223+32)/((223+32)*val+223+32);


  for j=1:val
```

```
    tp1(val)=throughput1;

    PE1(val)=1/(10*count);

end


for o=1:val

    DE1(val)=delay;

end


%extraaaaaaaaaaaaaaaaaaaaa
 % for o=1:q
   %tp(n)=throughput;

   %PE(q)=1/count;
   %DE(q)=delay;
  %end


  if (ack==0)
   max=max;
   limit=limit;
   pdu='';
   next_element=next_element;
   %n=n+1;

    ' val isssssssssssssssss'
      val
   'n isssssssssssssssss'
   n

   limit
   'Packet is not recieved '

       %DE(n)=delay;

   %break;

  else
```

```
        pdu= '' ;

        max=max+5;

        limit=limit+5;
        next_element=next_element;
        %n=n+1;
%if limit>length(string)
 %  remainder=limit-length(string);
  % new_len=len+remainder
   %for j=len:new_len
    %   string(j)='';
   %end

%end


%len=len+1;

   end

   len=len+1;

end


PE=PE;
DE=DE;
tp=tp;
%DE
%x=sort(PE,'ascend');
%plot(x,tp,'o');
%grid on
%xlabel('Bit Error Rate');
%ylabel('Through Put');
%figure;

%x=sort(PE,'ascend');
%plot(x,tp,'--rs');
%grid on
%figure;


%x=sort(PE,'ascend');
```

```
%plot(x,DE,'o');
%grid on

%figure;
%x=sort(PE,'ascend');
%plot(x,DE,'--rs');
%grid on




%figure;
% x1=sort(PE1,'ascend');
% plot(x1,tp1);
% grid on
% figure;
%
% x1=sort(PE1,'ascend');
% plot(x1,DE1);
% grid on
```

Function arq_reciever ( ):

```
function [a,count,throughput]=arq_reciever(pdu,check_code,n,error)
wait_time=0;
%for m=1:n
%k(n)=n
%end
%k=[1,2,3,4,5,6,7,8,9,10];
%REED SOLOMON

throughput=(223+32)/((223+32)*n+223+32);
%delay=q*2;
%web=32;
%nd=0;
%n=rando(n);
%while (web~=0)
%nd=nd+1;
%    numb=rando(nd);
 %    n=n+1;
    %web=web-1;
    %break;
%end
```

```
%numb
%if numb<0.5
%    count=count+1;
%else
%    'no error'
%end
%a=[0.2905,0.1824,0.2989,0.3493,0.1981,0.3790,0.2257,0.4464,0.0840,0.4293,0.4548
,0.3447,0.1175,0.3211, 0.0551,0.0590,0.4869]
%for i=1:32
%a(i)=rand(1);
%a(i)=i+20;
%end
count=1;
%for zero errors
%for l=1:32
%    a(l)=0.000000;
%end

for j=1:32
if error(j)>0.5
    count=count+1;
else
'no error';
end
end

'count  isssssssssssssssssssss'


count

count;
%for zero errors
%if count<33
if count<16
t = timer('StartDelay', wait_time,'TimerFcn','disp(''Next Frame'')');
start(t);
wait_time
string1='Waiting to recieve PDU'

wait(t);
delete(t);
pdu;
%a=check_code;
%check_code=0;
```

```
check_code=1;
a=check_code;
else
    'next frame'
    check_code=0;
     a=check_code;
end

count=count;
throughput=throughput;
%delay=delay;
%plot(count,throughput)

%REED SOLOMON
%wait_time=2;


%if (check_code==1)
%t = timer('StartDelay', wait_time,'TimerFcn','disp(''Next Frame'')');
%start(t);
%wait_time
%string1='Waiting to recieve PDU'

%wait(t);
%delete(t);
%pdu;
%a=check_code;
%check_code=0;
%a=check_code;
%else
 %   check_code=0;
  %  a=check_code;


%end
```

Appendix B

Function selective_repeat ( ):

```
function  [PE1,DE1,tp1]=selective_repeat()
string = 'hellotherelatifhellotherelatif'
persistent next_element;
next_element=0;
count=0;
max=1;
limit=5;
len=0;
check_code=1;
n=0;
m=0;
q=0;
bc=1;
missed_packets=0;
while (len~=length(string))

%next_element=next_element+1;
 if limit>length(string)
     remainder=limit-length(string);
     length_remainder=5-remainder;
     new_len=length(string)-length_remainder+1;
     new_len
     pdu
     for j=new_len:length(string)

         pdu(j)=string(j);
         count=count+1;


     end
     count;
     pdu;

  for ijk=new_len:length(string)
     next_element=next_element+1;
    error=error_array(next_element);

[ack,count,throughput,missed_packets]=sel_reciever(pdu(ijk),check_code,n,missed_pa
ckets,error);
```

```
%n=n+1;

     %m=m+1;
  [through,count_errors,packet_delay]=graph_arrays(n,throughput,count);

  %n=n+1;







 end



    delay=n*2+1;
    for j=1:n
   tp(n)=throughput;
   PE(n)=count_errors(n);
   %PE(n)=1/(10*count_errors(n));
   DE(n)=delay;




    end
   n=n+1;
    missed_packets
 % missed_packets=0
    break;

else
  for i=max:limit

      pdu(i)=string(i);

  end

end
 %n=n+1;
  pdu
  for ij=max:limit
  next_element=next_element+1;

   error=error_array(next_element);
```

```
[ack,count,throughput,missed_packets]=sel_reciever(pdu(ij),check_code,n,missed_pac
kets,error);

      %n=n+1;


   %m=m+1;
   [through,count_errors,packet_delay]=graph_arrays(n,throughput,count);



   end
    delay=n*2+1;

   for j=1:n
    tp(n)=throughput;
    PE(n)=count_errors(n);



    %PE(n)=1/(10*count_errors(n));
   end

   for o=1:n
   DE(n)=delay;
   end

  % for o=1:q
    %tp(n)=throughput;

    %PE(q)=1/count;
    %DE(q)=delay;
   %end
n=n+1;

missed_packets
pdu
   %missed_packets=0
if (ack==0)
    rem=5-missed_packets;
    max=max+rem;
    limit=limit+rem;
    pdu='';
    limit
```

```
    'Packet is not recieved '
    n=n+1;
        %DE(n)=delay;
    %next_element=next_element+1;
    %break;
else
rem=5-missed_packets;
pdu= '' ;
max=max+rem;
limit=limit+rem;
%n=n+1;
%if limit>length(string)
 %  remainder=limit-length(string);
  % new_len=len+remainder
   %for j=len:new_len
    %   string(j)='';
   %end

%end
%next_element=next_element+1;
missed_packets=0;

len=len+1;


end
%len=len+1;
end

PE1=PE;
DE1=packet_delay;
tp1=through;
DE
% x=sort(count_errors,'ascend');
% plot(x,through,'o');
% grid on
% figure;
% x=sort(count_errors,'ascend');
% plot(x,packet_delay,'o');
% grid on
% figure;
%
%
% x=sort(PE,'ascend');
% x=sort(count_errors,'ascend');
```

```
% plot(x,through,'--rs');
% grid on
% figure;
%
% x=sort(PE,'ascend');
% x=sort(count_errors,'ascend');
% plot(x,packet_delay,'--rs');
% grid on
```

Function sel_reciever ( ):

```
function
[a,count,throughput,missed]=sel_reciever(pdu,check_code,n,missed_packets,error)
wait_time=0;
wait_time=0;

%for m=1:n
%k(n)=n
%end
%k=[1,2,3,4,5,6,7,8,9,10];
%REED SOLOMON
missed=missed_packets;

%for zero errors
%throughput=(223+32)/((223+32)*1+223+32);
%delay=q*2;
throughput=(223+32)/((223+32)*n+223+32);
packet_count=0;

err=error;
%for i=1:32
%a(i)=rand(1);

%end

%err=error;
%for zero errors
%for l=1:32
%    a(l)=0.000000;
%end
%a=[0.2905,0.1824,0.2989,0.3493,0.1981,0.3790,0.2257,0.4464,0.0840,0.4293,0.4548
,0.3447,0.1175,0.3211,0.0551,0.0590,0.4869,0.2905,0.1824,0.2989,0.3493,0.1981,0.3
790,0.2257,0.4464,0.0840,0.4293,0.4548,0.3447,0.1175,0.3211,0.0551]
```

```
count=1;
%err=error_array(n);

%a=error_array();
for j=1:32
if err(j)>0.5
   count=count+1;

else

   'no error';
end
end

'count  issssssssssssssssssss'


count


% for zero errors
%if count<33

if count<16

t = timer('StartDelay', wait_time,'TimerFcn','disp(''Next Frame'')');
start(t);
wait_time;
string1='Waiting to recieve PDU';

wait(t);
delete(t);
pdu
%a=check_code;
%check_code=0;
check_code=1;
a=check_code;
missed=missed;
packet_count=packet_count+1;
else
   packet_count=packet_count+1;
   ' missed packet is asdasdasdasdasdasdasd'
   pdu
   'next frame'
   check_code=1;
```

```
    a=check_code;

 missed=missed+1;
 if packet_count==5
    check_code=0;
    a=check_code;
 end
end

count=count;
throughput=throughput;
%delay=delay;
%plot(count,throughput)

%REED SOLOMON
%wait_time=2;


%if (check_code==1)
%t = timer('StartDelay', wait_time,'TimerFcn','disp(''Next Frame'')');
%start(t);
%wait_time
%string1='Waiting to recieve PDU'

%wait(t);
%delete(t);
%pdu;
%a=check_code;
%check_code=0;
%a=check_code;
%else
 %   check_code=0;
  %  a=check_code;


%end
```

Appendix C

Function MS_generator:

```
function MS_generator()
for i=1:20
ms(i)=rand(1);
end
for j=1:10
if ms(j)<0.20
   priority(j)=0;
elseif ms(j)<0.40

   priority(j)=1;

elseif ms(j)<0.60

   priority(j)=2;

elseif ms(j)<0.80

   priority(j)=3;

end

end
% with only 4 users
%priority=[3,2,1,0]
%with only 4 users

sort_priority=sort(priority,'descend')
wifi_scheduler(sort_priority)
```

Function Graph_generator ( ):

```
function graph_generator()


 for i=1:1
[PE,DE,tp]=arq();

[PE1,DE1,tp1]=selective_repeat();
```

```
x1=sort(PE,'ascend');
x2=sort(PE1,'ascend');

hold on
plot(x1,tp,x2,tp1,'--r')
grid on

xlabel('Probability of Errors');
ylabel('Throughput');

legend('wimax','wifi')
end
% x=sort(PE,'ascend');
% plot(x,tp,'--rs');
% grid on
% xlabel('Bit Error Rate');
% ylabel('Through Put');
%
% hold on;
%
% x=sort(PE1,'ascend');
% plot(x,tp1,'--bs');
% grid on
%
% legend('WiMAX','WiFi')
% figure;
%
%
% x=sort(PE,'ascend');
% plot(x,DE,'--rs');
% grid on
% xlabel('Bit Error Rate');
% ylabel('DELAY');
%
% hold on;
%
% x=sort(PE1,'ascend');
% plot(x,DE1,'--bs');
%
%
% legend('WiMAX','WiFi')
% grid on
```

Function Graph_Arrays ( ):

```
function [throughp,count_PE,packet_delay]=graph_arrays(n,throughput,count)
persistent through;
persistent count_errors;
persistent delay;

'count issssssss'
count

del=n*2;
a=1;
for i=a:n

   through(n)=throughput;
   count_errors(n)=1/(10*count);
   delay(n)=del;
end

throughp=through;
count_PE=count_errors;
packet_delay=delay;
```

Appendix D

Function WiFi_scheduler ( ):

```
function wifi_scheduler(sort_priority)

TC=[0,1,2,3,4,5,6,7]
access_cat=[0,1,2,3]
AIFS4=34*10^(-3)
AIFS3=34*10^(-2)
AIFS2=34*10^(-1)
AIFS1=5
channel=0;
TXOP1=1;
TXOP2=2;
TXOP3=3;
TXOP4=4;
len=1;
count=0;
check=0;
check1=0;
check2=0;
check3=0;
ms_length=length(sort_priority);
n=0;
frame_size=100;
persistent tp;
persistent de;

persistent gtime;
while (len~=ms_length)

   if sort_priority(len)==3



         sort_priority=sort(sort_priority,'descend')

      t = timer('StartDelay', TXOP4,'TimerFcn','disp(''NEXT TC'')');
          start(t);
          'Traffic category 3'

          wait(t);
          delete(t);
```

```
            n=n+1;


            for j=1:n
            gtime(n)=TXOP4;



            end
            %tcat=sort_priority(len);
            priority=3;
            tcat=10;;
            t=TXOP4;

[sort_priority,ms_length,tcat,check,priority,gtime,n]=interrupt3(sort_priority,ms_lengt
h,tcat,check,t,priority,gtime,n);

           % backoff_time=AIFS4;
           % tcat=sort_priority(len);
           % sort_priority=interruption_cat3(tcat,backoff_time,sort_priority);
           %ms_length=length(sort_priority);
           % sort_priority

   end
     if sort_priority(len)==2
   t = timer('StartDelay', TXOP3,'TimerFcn','disp(''NEXT TC'')');
            start(t);
            'Traffic category 2'

            wait(t);
            delete(t);
            %priority=sort_priority(len);
             tcat=10;
              n=n+1;
            for j=1:n
            gtime(n)=TXOP3;
            end
             priority=2;
             tcat=10;
             t=TXOP3;

[sort_priority,ms_length,tcat,check,priority,gtime,n]=interrupt2(sort_priority,ms_lengt
h,tcat,check,t,priority,gtime,n);
```

```
%[sort_priority,ms_length,tcat,ch]=interrupt3(sort_priority,ms_length,tcat,check1);
   end
     if sort_priority(len)==1
   t = timer('StartDelay', TXOP2,'TimerFcn','disp(''NEXT TC'')');
          start(t);
          'Traffic category 1'

          wait(t);
          delete(t);
           n=n+1;
          for j=1:n
          gtime(n)=TXOP2;
          end
          priority=1;
           tcat=10;
          t=TXOP4;

[sort_priority,ms_length,tcat,check,priority,gtime,n]=interrupt1(sort_priority,ms_lengt
h,tcat,check,t,priority,gtime,n);

     end
     if sort_priority(len)==0
   t = timer('StartDelay', TXOP1,'TimerFcn','disp(''NEXT TC'')');
          start(t);
          'Traffic category 0'

          wait(t);
          delete(t);
           n=n+1;
          for j=1:n
          gtime(n)=TXOP1;
          end
          priority=sort_priority(len);
           tcat=10;
   t=TXOP1;

[sort_priority,ms_length,tcat,check,priority,gtime,n]=interrupt0(sort_priority,ms_lengt
h,tcat,check,t,priority,gtime,n);

     end

   len=len+1;
end
gtime
```

```
for i=1:length(sort_priority)
    turn(i)=i;
end
z=length(sort_priority);
for j=1:length(sort_priority)
    if sort_priority(z)==3
        gtime(z)=4;
    elseif  sort_priority(z)==2
        gtime(z)=3;
    elseif  sort_priority(z)==1
        gtime(z)=2;
    elseif sort_priority(z)==0
        gtime(z)=1;
    end
end
gtime
plot(gtime,sort_priority,'--rs')
figure;
plot3(gtime,sort_priority,turn,'o')
xlabel('Priority');
ylabel('turn');
zlabel('Delay');

figure;
plot3(gtime,sort_priority,turn,'--rs')
```

Appendix E

Function WiMAX_scheduler ( ):

```
function scheduler(SFID,CID,time)
persistent channel;

persistent critical_time;
persistent continuous_time;
persistent periodic_time;
persistent noncritical_time;
global time_array;
persistent gtime;
n=0;
m=0;
assign_time=time;
persistent time_remainder;
remainder=0;
channel=[0,0,0,0,0];
SFID_array=[2222];
%wait_time=10000;
slot_time=[2,3];
persistent sfid_arr;
persistent slot_number;
for i=1:length(SFID_array)
        if SFID_array[i]==1111
            'Service Available';
        break;

        end
end
%active set
active_set(1,1)=100; % base station id
active_set(1,2)=1;   % signal strength
active_set(2,1)=200;
active_set(2,2)=3;
active_set(3,1)=300;
active_set(3,2)=1;
%active set
```

```
 time_count=0;
if SFID==1111
    'Critical Service Needed'
    critical_time=time;
assign_time=critical_time;

    for j=1:length(channel)

        if channel(j)==0
            channel(j)=1;
            str2=int2str(j);
           str='  Channel available= ';
           str3=strcat(str,str2)


          critical_time=critical_sensor_time();
          assign_time=critical_time;


          i=1;

          while(assign_time(i)~=0)

              if(i>=2)
                  'Time increased';
              end


              assign_time(i);

              %check_frame_time(assign_time(i));

              if assign_time(i)<=5
              time_count=assign_time(i)+time_count
              if time_count>5
                  'New Frame is Started'
                  %assign_time(i)=5-assign_time(i);
                  %remainder=time_count-5;
                   time_count=0;
              end
              t = timer('StartDelay', assign_time(i),'TimerFcn','disp(''Next Slot'')');
              start(t);
              assign_time(i)
              string1='Waiting for the slot to get free'
              wait(t);
```

```
delete(t);
g_time=assign_time(i);
n=n+1;
for mn=1:n
gtime(n)=g_time;

end
m=m+1
  sfid=SFID;
for no=1:m
  sfid_arr(m)=sfid;
end


elseif assign_time(i)>5


[remainder,assign_time(i)]=check_remainder(assign_time(i));
while (remainder~=0)

'time_count is '
time_count=assign_time(i)+time_count
if time_count>5
   'New Frame is Started'
   time_count=0;
end

t = timer('StartDelay', assign_time(i),'TimerFcn','disp(''Next Slot'')');
start(t);
assign_time(i)
string1='Waiting for the slot to get free'
wait(t);
delete(t);
g_time=assign_time(i);
 n=n+1;
for mn=1:n
gtime(n)=g_time;



end
  sfid=SFID;
m=m+1
for no=1:m
  sfid_arr(m)=sfid;
```

```
end



if remainder>0

'time_count is '
time_count=remainder+time_count

if time_count>5
   'New Frame is Started'
   time_count=0;

end
t = timer('StartDelay', remainder,'TimerFcn','disp(''Next Slot'')');
start(t);
'waiting for remainder'
remainder
string1='Waiting for the slot to get free'
wait(t);
delete(t);
n=n+1;
g_time=assign_time(i);
for mn=1:n
gtime(n)=g_time;



end
  sfid=SFID;
m=m+1
for no=1:m
   sfid_arr(m)=sfid;
end
%i=i+1;


end
[remainder,assign_time(i)]=check_remainder(assign_time(i));



end
```

```
            end
          i=i+1;

          end

         break;
         elseif channel(j)==1
             slot_number(j)=j;
            str2=int2str(j);
            str='  Channel not available= ';
            str3=strcat(str,str2)
            'slot'
            slot_number(j)
            'assigned to Service'
            SFID_array(j)
            'and slot time is'
            slot_time(j)
            if channel(j)==channel(end)


              %assign_time
                 %assign_channel(assign_time);
             end


         end


     end
 elseif SFID==2222
     'Continuous Service Needed'
     continuous_time=time;

     for j=1:length(channel)

             if channel(j)==0
             channel(j)=1;
             str2=int2str(j);
            str='  Channel available= ';
            str3=strcat(str,str2)


          critical_time=continuous_sensor_time();
          assign_time=critical_time;
```

```
    for mn=1:n
    gtime(n)=g_time+5;



    end
i=1;

while(assign_time(i)~=0)

    if(i>=2)
        'Time increased';
    end


    assign_time(i);

    %check_frame_time(assign_time(i));

    if assign_time(i)<=5
    time_count=assign_time(i)+time_count
    if time_count>5



        'New Frame is Started'
        'remaindererererererer'
        remainder=time_count-5;
        sfid=check_sfid(SFID);
        remainder
        t = timer('StartDelay', remainder,'TimerFcn','disp(''Next Slot'')');
    start(t);
    assign_time(i)
    string1='Waiting for the slot to get free'
    wait(t);
    delete(t);
    g_time=assign_time(i);
      n=n+1;
    for mn=1:n
    gtime(n)=g_time+5;



    end
      sfid=SFID;
```

```
m=m+1
for no=1:m
    sfid_arr(m)=sfid;
end
    %assign_time(i)=5-assign_time(i);
    %remainder=time_count-5;
     time_count=0;
end
t = timer('StartDelay', assign_time(i),'TimerFcn','disp(''Next Slot'')');
start(t);
assign_time(i)
string1='Waiting for the slot to get free'
wait(t);
delete(t);
g_time=assign_time(i);
  n=n+1;
for mn=1:n
gtime(n)=g_time+5;



end
   sfid=SFID;
m=m+1
for no=1:m
sfid_arr(m)=sfid;
end

elseif assign_time(i)>5


[remainder,assign_time(i)]=check_remainder(assign_time(i));
while (remainder~=0)

'time_count is '
time_count=assign_time(i)+time_count
if time_count>5
    'New Frame is Started'
    time_count=0;
end

t = timer('StartDelay', assign_time(i),'TimerFcn','disp(''Next Slot'')');
start(t);
assign_time(i)
string1='Waiting for the slot to get free'
```

```
wait(t);
delete(t);
g_time=assign_time(i);
  n=n+1;
for mn=1:n
gtime(n)=g_time+5;



end
   sfid=SFID;
m=m+1
for no=1:m
  sfid_arr(m)=sfid;
end



if remainder>0

'time_count is '
time_count=remainder+time_count

if time_count>5
    'New Frame is Started'
    time_count=0;

end
t = timer('StartDelay', remainder,'TimerFcn','disp(''Next Slot'')');
start(t);
'waiting for remainder'
remainder
string1='Waiting for the slot to get free'
wait(t);
delete(t);
g_time=assign_time(i);
  n=n+1;
for mn=1:n
gtime(n)=g_time+5;



end
   sfid=SFID;
m=m+1
```

```
          for no=1:m
            sfid_arr(m)=sfid;
          end
          %i=i+1;


          end
          [remainder,assign_time(i)]=check_remainder(assign_time(i));



          end


          end
        i=i+1;
        if SFID==1111
              %critical_sensor();

          end
        end

       break;
       elseif channel(j)==1
         str2=int2str(j);
         str='  Channel not available= ';
         str3=strcat(str,str2)


       end


   end
  elseif SFID==3333
 'Periodic Service Needed'
   periodic_time=time;

   for j=1:length(channel)

          if channel(j)==0
          channel(j)=1;
          str2=int2str(j);
         str='  Channel available= ';
         str3=strcat(str,str2)
```

```
critical_time=periodic_sensor_time();
assign_time=critical_time;


i=1;

while(assign_time(i)~=0)

    if(i>=2)
        'Time increased';
    end


    assign_time(i);

    %check_frame_time(assign_time(i));

    if assign_time(i)<=5
    time_count=assign_time(i)+time_count
    if time_count>5
        'New Frame is Started'
        'remaindererererererererer'
        remainder=time_count-5;
        sfid=check_sfid(SFID);
        remainder
        t = timer('StartDelay', remainder,'TimerFcn','disp(''Next Slot'')');
    start(t);
    assign_time(i)
    string1='Waiting for the slot to get free'
    wait(t);
    delete(t);
    g_time=assign_time(i);
      n=n+1;
    for mn=1:n
    gtime(n)=g_time;



    end
       sfid=SFID;
    m=m+1
    for no=1:m
        sfid_arr(m)=sfid;
    end
```

```
    %assign_time(i)=5-assign_time(i);
    %remainder=time_count-5;
     time_count=0;
  end
  t = timer('StartDelay', assign_time(i),'TimerFcn','disp(''Next Slot'')');
  start(t);
  assign_time(i)
  string1='Waiting for the slot to get free'
  wait(t);
  delete(t);
  g_time=assign_time(i);
   n=n+1;
  for mn=1:n
  gtime(n)=g_time;



  end
    sfid=SFID;
  m=m+1
  for no=1:m
  sfid_arr(m)=sfid;
  end

  elseif assign_time(i)>5


[remainder,assign_time(i)]=check_remainder(assign_time(i));
  while (remainder~=0)

  'time_count is '
  time_count=assign_time(i)+time_count
  if time_count>5
    'New Frame is Started'
    time_count=0;
  end

  t = timer('StartDelay', assign_time(i),'TimerFcn','disp(''Next Slot'')');
  start(t);
  assign_time(i)
  string1='Waiting for the slot to get free'
  wait(t);
  delete(t);
  g_time=assign_time(i);
   n=n+1;
```

```
for mn=1:n
gtime(n)=g_time;



end
  sfid=SFID;
m=m+1
for no=1:m
 sfid_arr(m)=sfid;
end



if remainder>0

'time_count is '
time_count=remainder+time_count

if time_count>5
   'New Frame is Started'
   time_count=0;

end
t = timer('StartDelay', remainder,'TimerFcn','disp(''Next Slot'')');
start(t);
'waiting for remainder'
remainder
string1='Waiting for the slot to get free'
wait(t);
delete(t);
g_time=assign_time(i);
 n=n+1;
for mn=1:n
gtime(n)=g_time;



end
  sfid=SFID;
m=m+1
for no=1:m
 sfid_arr(m)=sfid;
end
%i=i+1;
```

```
        end
        [remainder,assign_time(i)]=check_remainder(assign_time(i));



        end


        end
      i=i+1;
      if SFID==1111
           critical_sensor();

        end
      end

    break;
    elseif channel(j)==1
       str2=int2str(j);
       str='  Channel not available= ';
       str3=strcat(str,str2)


    end


end



   elseif SFID==4444
'Non Critial Service Needed'
noncritical_time=time;

for j=1:length(channel)

        if channel(j)==0
        channel(j)=1;
        str2=int2str(j);
       str='  Channel available= ';
       str3=strcat(str,str2)
```

```
critical_time=noncritical_sensor_time();
  assign_time=critical_time;


  i=1;

  while(assign_time(i)~=0)

      if(i>=2)
          'Time increased';
      end


      assign_time(i);

      %check_frame_time(assign_time(i));

      if assign_time(i)<=5
      time_count=assign_time(i)+time_count
      if time_count>5
          'New Frame is Started'
          'remaindererererererer'
          remainder=time_count-5;
          sfid=check_sfid(SFID);
          remainder
          t = timer('StartDelay', remainder,'TimerFcn','disp(''Next Slot'')');
      start(t);
      assign_time(i)
      string1='Waiting for the slot to get free'
      wait(t);
      delete(t);
      g_time=assign_time(i);
        n=n+1;
      for mn=1:n
      gtime(n)=g_time*n;



      end
         sfid=SFID;
      m=m+1
      for no=1:m
          sfid_arr(m)=sfid;
      end
          %assign_time(i)=5-assign_time(i);
```

```
    %remainder=time_count-5;
     time_count=0;
  end
  t = timer('StartDelay', assign_time(i),'TimerFcn','disp(''Next Slot'')');
  start(t);
  assign_time(i)
  string1='Waiting for the slot to get free'
  wait(t);
  delete(t);
  g_time=assign_time(i);
   n=n+1;
  for mn=1:n
  gtime(n)=g_time*n;



  end
    sfid=SFID;
  m=m+1
  for no=1:m
  sfid_arr(m)=sfid;
  end

  elseif assign_time(i)>5


[remainder,assign_time(i)]=check_remainder(assign_time(i));
  while (remainder~=0)

  'time_count is '
  time_count=assign_time(i)+time_count
  if time_count>5
    'New Frame is Started'
    time_count=0;
  end

  t = timer('StartDelay', assign_time(i),'TimerFcn','disp(''Next Slot'')');
  start(t);
  assign_time(i)
  string1='Waiting for the slot to get free'
  wait(t);
  delete(t);
  g_time=assign_time(i);
   n=n+1;
  for mn=1:n
```

```
gtime(n)=g_time*n;



end
  sfid=SFID;
m=m+1
for no=1:m
 sfid_arr(m)=sfid;
end



if remainder>0

'time_count is '
time_count=remainder+time_count

if time_count>5
   'New Frame is Started'
   time_count=0;

end
t = timer('StartDelay', remainder,'TimerFcn','disp(''Next Slot'')');
start(t);
'waiting for remainder'
remainder
string1='Waiting for the slot to get free'
wait(t);
delete(t);
g_time=assign_time(i);
 n=n+1;
for mn=1:n
gtime(n)=g_time*n;



end
  sfid=SFID;
m=m+1
for no=1:m
 sfid_arr(m)=sfid;
end
%i=i+1;
```

```
            end
            [remainder,assign_time(i)]=check_remainder(assign_time(i));



            end


            end
          i=i+1;



              if SFID==1111
              critical_sensor();
              periodic_sensor();


                end

          end

        break;
        elseif channel(j)==1
           str2=int2str(j);
           str='  Channel not available= ';
           str3=strcat(str,str2)


        end


    end

end
for i=1:length(gtime)
    turn(i)=i;
end

assign_time
gtime
sfid_arr
plot(sfid_arr,gtime,'o')
figure;
```

```
plot(sfid_arr,gtime)
figure;

for i=1:length(sfid_arr)
   if sfid_arr(i)==1111
      sfid_arr(i)=1
   elseif sfid_arr(i)==2222
      sfid_arr(i)=2

   elseif sfid_arr(i)==3333

      sfid_arr(i)=3
   elseif sfid_arr(i)==4444
      sfid_arr(i)=4
   end
end


plot3(sfid_arr,turn,gtime,'o')
xlabel('Priority');
ylabel('Turn');
zlabel('Delay');
figure;
plot3(sfid_arr,turn,gtime,'--rs')

%channel
%wait(t);
 %          delete(t);
%assign_channel(assign_time,channel,SFID);
```