DESIGN OF WIRELESS BODY AREA NETWORK

Thesis submitted to Visvesvaraya National Institute of Technology, Nagpur

In partial fulfilment of requirement for the award of degree of

BACHELOR OF TECHNOLOGY

(ELECTRONICS AND COMMUNICATION ENGINEERING)

Submitted By

AMABATI UDAY KAUSHIK	(BT11ECE004)
INDRA KIRAN NANADAPATI	(BT11ECE032)
PVN MANOJ	(BT11ECE060)
RAVULA SAI KIRAN REDDY	(BT11ECE070)
SHAIK ARSHAD	(BT11ECE078)
VEERANKI ABHIJEETH	(BT11ECE089)
GUNDU VENKATESWARLU	(BT11ECE090)
YNV RAGHAVA	(BT11ECE092)

Under the guidance of

Asst. Prof. P.H.GHARE

Department of Electronics and Communication Engineering

Visvesvaraya National Institute of Technology

Nagpur 440 010 (India) April 2015.

Visvesvaraya National Institute of Technology, Nagpur

2014 – 2015

CERTIFICATE

This is to certify that the project work entitled "DESIGN OF WIRELESS BODY AREA NETWORK ", is a bonafide work by Mr. Ambati Uday Kaushik, Mr. Nandipati Indra Kiran Reddy, Mr. Potharlanka Venkata Naga Manoj, Mr. Arshad Shaik ,Mr. Veeranki Abhijeeth ,Mr. Gundu Venkateswarlu and Mr. Yallamanchilli Naga Venkata Sai Raghava in the Department of Electronics and Communication Engineering, Visvesvaraya National Institute of Technology, Nagpur, in partial fulfilment of the requirements of the award of the degree of Bachelor of Technology in Electronics and Communication Engineering.

Prof. P.H.Ghare
Mentor and Coordinator
Assistant Professor
Department of Electronics and
Communication Eng.
VNIT, Nagpur

Prof. Dr. K.D.Kulat
Head of Department
Professor
Department of Electronics and
Communication Eng.
VNIT, Nagpur

Visvesvaraya National Institute of Technology, Nagpur

2014 – 2015

DECLARATION

This is to declare that the project work entitled “DESIGN OF WIRELESS BODY AREA NETWORK”, is a bonafide work performed by us, the below mentioned students. This project work is being submitted and forwarded in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering from Visvesvaraya National Institute of Technology, Nagpur.

To the best of our knowledge this project report has not been submitted to any other institution or university.

Ambati Uday Kaushilk	Nandipati Indra Kiran Reddy	Potharlanka Venkata Naga Manoj

Ravula Sai Kiran Reddy

Shaik Arshad

Veeranki Abhijeeth

Gundu Venkateswarlu

Yellamanchilli Naga Venkata Raghava

Acknowledgement

"The finest task of an Engineer – to survey, to plan, to suggest resources and material, assemble them for task in hand, implement and procedure results"

- **Visvesvaraya**

We take this opportunity to acknowledge with deep sense of gratitude our project guide Prof. P.H.Ghare, Assistant Professor, Department of Electronics and Communication Engineering, VNIT Nagpur for her invaluable guidance, constant motivation, and continuous support which has led to the successful completion of this project.

We also take this opportunity to pay our sincere thanks to Dr. K.D.Kulat, Head of Department, Electronics and Communication Engineering, VNIT Nagpur, for providing the requisite facilities needed to complete the project. We would also like to thank all the teaching and non-teaching staff for supporting us.

We express our thanks to our parents and all our friends for their constant support and encouragement, which helped us, complete this work.

Ambati Uday Kaushik

Nandipati Indra Kiran Reddy

Potharlanka Venkata Naga Manoj

Ravula Sai Kiran Reddy

Arshad Shaik

Veeranki Abhijeeth

Gundu Venkateswarlu

Yallamanchilli Naga Venkata Sai Raghava

Table of Contents

List of figures

List of tables

1. INTRODUCTION

A number of economic and demographic forces are challenging the long-term scalability of existing healthcare systems. In 2014/15 India has allocated a budget of $948 million out of $5 billion, which is very less for a country with 1.2 billion populations. So a health care system has to be designed which is compatible with the needs of poor people in India. Increased life expectancy and retiring baby boomers are compounding the problem – causing a dramatic shift in demographics in the India and worldwide. In this century, it is expected that the elderly will outnumber children for the first time in history.

In a 2011 estimate on the causes of death, death due to cardiovascular disease is the most common death among the people in India. 24.8% of death are caused by cardiovascular diseases, 10.2% of deaths are caused by respiratory diseases. These diseases can be well detected through diagnosis far before it becomes serious, so that life can be saved. It has long been understood the relationship between wellness and physical activity, yet this continues to be a significant health care problem. Early diagnosis is also well understood as a means to reduce overall treatment costs and increase life expectancy and quality of life – especially for cardiovascular disease. Yet, it is estimated that 81% of total health expenditures is spent on treatments, hospital stays, and rehabilitation measures while only 4% is spent on diagnostic measures. These facts underscore the shortcomings of existing healthcare systems. We need a dramatic shift from centralized reactive healthcare systems to distributed and proactive systems focused on managing wellness rather than illness. Wearable systems for continuous health monitoring are a key technology in helping the transition to more proactive and affordable healthcare.

They allow an individual to closely monitor changes in her or his vital signs and provide feedback to help maintain an optimal health status. If integrated into a telemedical system, these systems can even alert medical personnel when life-threatening changes occur. In addition, the wearable systems can be used for health monitoring of patients in ambulatory settings. For example, they can be used as a part of a diagnostic procedure, optimal maintenance of a chronic condition, a supervised recovery from an acute event or surgical procedure, to monitor adherence to treatment guidelines (e.g., regular cardiovascular exercise), or to monitor effects of drug therapy. One of the most promising approaches in building wearable health monitoring systems utilizes emerging wireless body area networks (WBANs). AWBAN consists of multiple sensor nodes, each capable of sampling, processing, and communicating one or more vital signs (heart rate, blood pressure, oxygen saturation, activity) or environmental parameters (location, temperature, humidity, light). Typically, these sensors are placed strategically on the human body as tiny patches or hidden in users' clothes allowing ubiquitous health monitoring in their native environment for extended periods of time. This offers the freedom of mobility and enhances the patient's quality of life.

This thesis presents the prototype WBAN implementation and the specific design objectives. Specifically, we present the architecture and environment for our WBAN development, the power efficient wireless communication system, detail the embedded software, and describe our personal server application. The thesis is organized as follows: Chapter 2 defines the overall system of E-Health care monitoring system. Chapter 3 details the various types of components present in the WBAN. It begins by briefly describing the motivating system architecture and long-term vision, and then details the hardware architecture. Chapter 4 describes the WBAN communication scheme and how its design is well suited for ultra-low power sensors such as the nodes in our WBAN. In Chapter 5 We discuss the details of our implementation.

2. OVERVIEW OF E-HEALTH CARE MONITORING SYSTEM

A sensor node, also known as a mote, is a node in a sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. A mote is a node but a node is not always a mote.

2.1 Components:

The main components of a sensor node are a microcontroller, transceiver, external memory, power source and one or more sensors.

2.2 Sensors:

Sensors are hardware devices that produce a measurable response to a change in a physical condition like temperature or pressure. Sensors measure physical data of the parameter to be monitored. The continual analog signal produced by the sensors is digitized by an analog-to-digital converter and sent to controllers for further processing. A sensor node should be small in size, consume extremely low energy, operate in high volumetric densities, be autonomous and operate unattended, and be adaptive to the environment. As wireless sensor nodes are typically very small electronic devices, they can only be equipped with a limited power source of less than 0.5-2 ampere-hour and 1.2-3.7 volts.

Sensors are classified into three categories: passive, Omni-directional sensors; passive, narrow-beam sensors; and active sensors. Passive sensors sense the data without actually manipulating the environment by active probing. They are self-powered; that is, energy is needed only to amplify their analog signal. Active sensors actively probe the environment, for example, a sonar or radar sensor, and they require continuous energy from a power source. Narrow-beam sensors have a well-defined notion of direction of measurement, similar to a camera. Omni-directional sensors have no notion of direction involved in their measurements.

The overall theoretical work on WSNs works with passive, Omni-directional sensors. Each sensor node has a certain area of coverage for which it can reliably and accurately report the particular quantity that it is observing. Several sources of power consumption in sensors are: signal sampling and conversion of physical signals to electrical ones, signal conditioning, and analog-to-digital conversion. Spatial density of sensor nodes in the field may be as high as 20 nodes per cubic meter.

The sensors used for e-health care are-

2.2.1 ECG:

Electrocardiography is the process of recording the electrical activity of the heart over a period of time using electrodes placed on a patient's body. These electrodes detect the tiny electrical changes on the skin that arise from the heart muscle depolarizing during each heartbeat.

In a conventional 12 lead ECG, ten electrodes are placed on the patient's limbs and on the surface of the chest. The overall magnitude of the heart's electrical potential is then measured from twelve different angles ("leads") and is recorded over a period of time (usually 10 seconds). In this way, the overall magnitude and direction of the heart's electrical depolarization is captured at each moment throughout the cardiac cycle. The graph of voltage versus time produced by this noninvasive medical procedure is referred as Electrocardiography (abbreviated ECG or EKG).

An ECG can be used to measure the rate and rhythm of heartbeats, the size and position of the heart chambers, the presence of any damage to the heart's muscle cells or conduction system, the effects of cardiac drugs, and the function of implanted pacemakers

2.2.2 Pulse sensor:

Pulse sensing is a non-invasive method for monitoring a person's O_2 saturation.

In its most common (transmissive) application mode, a sensor device is placed on a thin part of the patient's body, usually a fingertip or earlobe, or in the case of an infant, across a foot. The device passes two wavelengths of light through the body part to a photo detector. It measures the changing absorbance at each of the wavelengths, allowing it to determine the absorbance's due to the pulsing arterial blood alone, excluding venous blood, skin, bone, muscle, fat, and (in most cases) nail polish.

Reflectance pulse oximetry may be used as an alternative to Transmissive pulse oximetry described above. This method does not require a thin section of the person's body and is therefore well suited to more universal application such as the feet, forehead and chest, but it also has some limitations. Vasodilation and pooling of venous blood in the head due to compromised venous return to the heart, as occurs with congenital cyanotic heart disease patients, or in patients in the Trendelenburg position, can cause a combination of arterial and venous pulsations in the forehead region and lead to spurious SpO_2 (Saturation of peripheral oxygen) results.

2.2.3 Temperature Sensor:

Normal human body temperature, also known as normothermia or euthermia, depends upon the place in the body at which the measurement is made, the time of day, as well as the activity level of the person. Nevertheless, commonly mentioned typical values are:

Temperature classification	
	Core (rectal, esophageal, etc.)
Hypothermia	<35.0 °C (95.0 °F)
Normal	36.5–37.5 °C (97.7–99.5 °F)
Fever	>37.5 or 38.3 °C (99.5 or 100.9 °F)
Hyperthermia	>37.5 or 38.3 °C (99.5 or 100.9 °F)
Hyperpyrexia	>40.0 or 41.5 °C (104.0 or 106.7 °F)
Oral (under the tongue)	= 36.8° ± 0.4 °C (98.2° ± 0.7 °F)
Note: The difference between fever and hyperthermia is the underlying mechanism. Different sources have different cuts offs for fever, hyperthermia and hyperpyrexia.	

Table – 1 Temperature classification of temperature sensor

2.3 Transceivers

Sensor nodes often make use of ISM band, which gives free radio, spectrum allocation and global availability. The possible choices of wireless transmission media are radio frequency (RF), optical communication (laser) and infrared. Lasers require less energy, but need line-of-sight for communication and are sensitive to atmospheric conditions. Infrared, like lasers, needs no antenna but it is limited in its broadcasting capacity. Radio frequency-based communication is the most relevant that fits most of the WSN applications. WSNs tend to use license-free communication frequencies: 173, 433, 868, and 915 MHz; and 2.4 GHz. The functionality of both transmitter and receiver are combined into a single device known as a transceiver. Transceivers often lack unique identifiers. The operational states are transmitting, receive, idle, and sleep. Current generation transceivers have built-in state machines that perform some operations automatically.

Most transceivers operating in idle mode have a power consumption almost equal to the power consumed in receive mode. Thus, it is better to completely shut down the transceiver rather than leave it in the idle mode when it is not transmitting or receiving. A significant amount of power is consumed when switching from sleep mode to transmit mode in order to transmit a packet.

The radio design used by ZigBee has been carefully optimized for low cost in large scale production. It has few analog stages and uses digital circuits wherever possible.

Though the radios themselves are inexpensive, the ZigBee Qualification Process involves a full validation of the requirements of the physical layer. All radios derived from the same validated semiconductor mask set would enjoy the same RF characteristics. An uncertified physical layer that malfunctions could cripple the battery lifespan of other devices on a ZigBee network. ZigBee radios have very tight constraints on power and bandwidth. Thus, radios are tested with guidance given by Clause 6 of the 802.15.4-2006 Standard. Most vendors plan to integrate the radio and microcontroller onto single chip getting smaller devices.

This standard specifies operation in the unlicensed 2.4 GHz (worldwide), 915 MHz (Americas and Australia) and 868 MHz (Europe) ISM bands. Sixteen channels are allocated in the 2.4 GHz band; with each channel spaced 5 MHz apart, though using only 2 MHz of bandwidth. The radios use spectrum coding, which is managed by the digital stream into the modulator. Binary phase-shift keying (BPSK) is used in the 868 and 915 MHz bands, and offset quadrature phase-shift keying (OQPSK) that transmits two bits per symbol is used in the 2.4 GHz band.

The raw, over-the-air data rate is 250 Kbit/s per channel in the 2.4 GHz band, 40 Kbit/s per channel in the 915 MHz band, and 20 Kbit/s in the 868 MHz band. The actual data throughput will be less than the maximum specified bit rate due to the packet overhead and processing delays. For indoor applications at 2.4 GHz transmission distance may be 10–20 m, depending on the construction materials, the number of walls to be penetrated and the output power permitted in that geographical location. Outdoors with line-of-sight, range may be up to 1500 m depending on power output and environmental characteristics. The output power of the radios is generally 0-20 dBm (1-100 mW).

2.4 Controller:

The controller performs tasks, processes data and controls the functionality of other components in the sensor node. While the most common controller is a microcontroller, other alternatives that can be used as a controller are: a general purpose desktop microprocessor, digital signal processors, FPGAs and ASICs. A microcontroller is often used in many embedded systems such as sensor nodes because of its low cost, flexibility to connect to other devices, ease of programming, and low power consumption. A general purpose microprocessor generally has higher power consumption than a microcontroller; therefore it is often not considered a suitable choice for a sensor node. Digital Signal Processors may be chosen for broadband wireless communication applications, but in Wireless Sensor Networks the wireless communication is often modest: i.e., simpler, easier to process modulation and the signal processing tasks of actual sensing of data is less complicated. Therefore the advantages of DSPs are not usually of much importance to wireless sensor nodes. FPGAs can be reprogrammed and reconfigured according to requirements, but this takes more time and energy than desired.

MSP430 is used as microcontroller for the project. The MSP430 is a mixed-signal microcontroller family from Texas Instruments. Built around a 16-bit CPU, the MSP430 is designed for low cost and, specifically, low power consumption embedded applications. The main advantages of this controller over other devices are -

- The MSP430 Value Line controllers are cheaper than the higher-end AT Mega controllers used in the Arduino platform.
- You don’t need an external crystal, because the MSP430 controllers can run at the full 16 MHz from their internal clock source.
- With the MSP430 Launchpad, TI offers a development board that includes everything you need for about a quarter of the cost of an Arduino board:
    - A programmer for the MSP430 controllers
    - comes with two microcontrollers (a MSP430G2553 and a MSP430G2452)
    - In addition to connecting the programmer, the USB connection also allows you to talk to the microcontroller via a serial connection at 9600 baud, so you don’t need a separate “FTDI cable”.

2.5 External memory:

From an energy perspective, the most relevant kinds of memory are the on-chip memory of a microcontroller and Flash memory—off-chip RAM is rarely, if ever, used. Flash memories are used due to their cost and storage capacity. Memory requirements are very much application dependent. Two categories of memory based on the purpose of storage are: user memory used for storing application related or personal data, and program memory used for programming the device. Program memory also contains identification data of the device if present.

This is optional, because there is no need of storing the patient reading for every second .as we are interested in the data where there is aberrant condition in patient health .we are also sending the data to the other devices like mobile or system using Bluetooth where we can store the required data in the cloud.so indirectly our external memory storage is cloud.

2.6 Power source:

A wireless sensor node is a popular solution when it is difficult or impossible to run a mains supply to the sensor node. However, since the wireless sensor node is often placed in a hard-to-reach location, changing the battery regularly can be costly and inconvenient. An important aspect in the development of a wireless sensor node is ensuring that there is always adequate energy available to power the system. The sensor node consumes power for sensing, communicating and data processing. More energy is required for data communication than any other process. The energy cost of transmitting 1 Kb a distance of 100 meters (330 ft.) is approximately the same as that used for the execution of 3 million instructions by a 100 million instructions per second/W processor. Power is stored either in batteries or capacitors. Batteries, both rechargeable and non-rechargeable, are the main source of power supply for sensor nodes. They are also classified according to electrochemical material used for the electrodes such as NiCd (nickel-cadmium), NiZn (nickel-zinc), NiMH (nickel-metal hydride), and lithium-ion. Current sensors are able to renew their energy from solar sources, temperature differences, or vibration. Two power saving policies used are Dynamic Power Management (DPM) and Dynamic Voltage Scaling (DVS). DPM conserves power by shutting down parts of the sensor node which are not currently used or active. A DVS scheme varies the power levels within the sensor node depending on the non-deterministic workload. By varying the voltage along with the frequency, it is possible to obtain quadratic reduction in power consumption.

2.7 Block diagram:

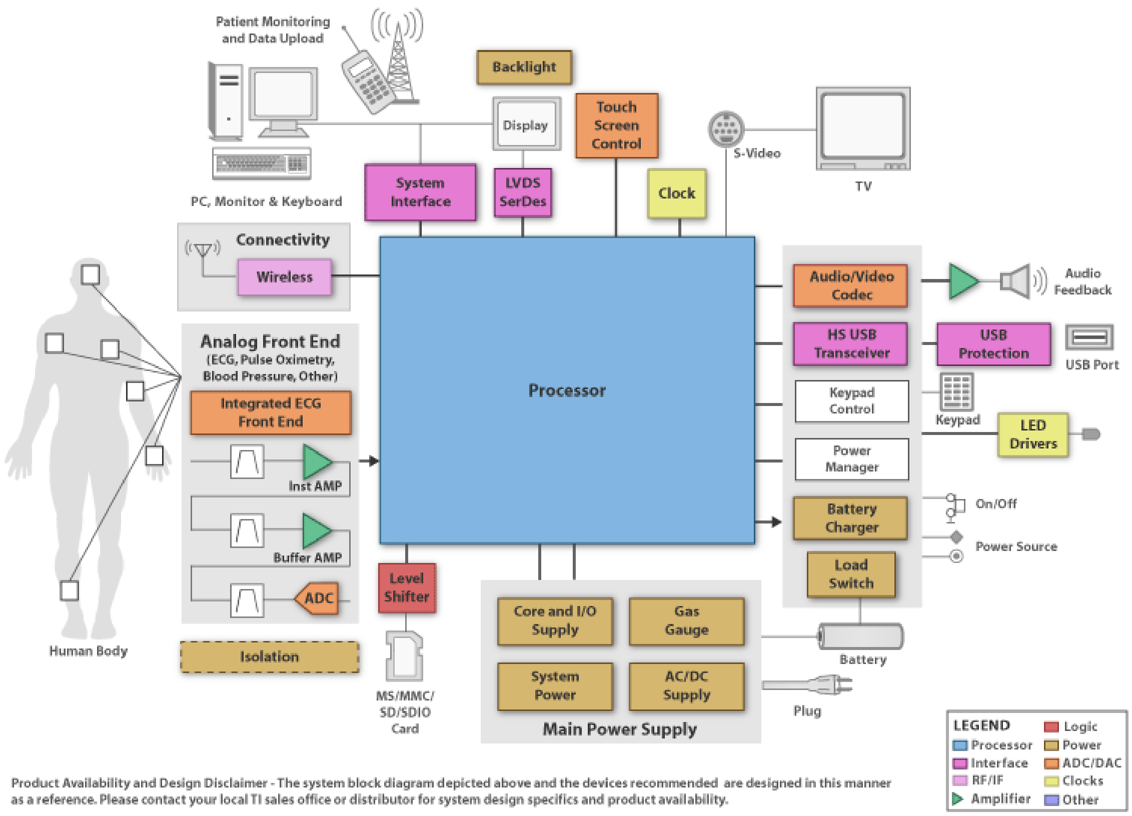

Figure -1 Bloch Diagram of a Typical WBAN System

2.8 Wearable devices for Health monitoring:

1. Lumo Lift:

This little gadget nudges you to stand up straight. The Lumo Lift which comes in a variety of colours detects your body's positioning and when you start to slouch, the device vibrates and reminds you to adjust your stance. The Lift can be clasped to an undershirt, collar or bra strap.

2. Jabra's Heart-Rate Tracking Ear buds:

The Jabra Sports Pulse Wireless Ear buds come with a built-in biometric heart rate monitor that tracks and evaluates your workout. This is a smart gift for anyone who's training for a marathon or other high-intensity activity.

3. June Bracelet:

The June bracelet ($99) -- from Netatmo, known for its personal weather station monitors -- measures the sun's impact. The jewel on the wristband syncs with an iOS device and alerts users when the skin has had too much exposure to the sun. It also monitors UV intensity, tracks daily habits and advises women how to better take care of their skin.

4. Mimo Baby Monitor:

Figure –2 A MIMO Baby Monitor

This smart baby onesie is a good pick for new parents: it monitors the respiration, skin temperature, body position, and sleeping and activity levels of infants. The data is then sent to a smartphone app in real time -- and users can set up alerts if there are any changes. This means parents don't have to get up and check on the baby throughout the night any more than they need to.

5. Babolat Play:

This Babolat Play ($399) tells you why you keep hitting the ball out of bounds. Sensors on the handle collect data about the player's swing, power, endurance, technique and ball impact. That information is then sent to its accompanying app, so users can analyze their skills and ultimately improve how they play. This is a fun gift for both beginners and advanced players.

6. Fit bit Charge:

The Fitbit Charge ($129.95) is the replacement to the Fitbit Force that was recalled six months ago after users reported allergic reactions to the wristband. Just like its predecessor, it comes with a display (tells time, highlights steps taken, calories burned and so on), but it has a neat new perk: it has caller ID too, showcasing who's calling you on your smartphone in real time.

The design is slightly different than the Force too; it's lighter, the clasp works better and it comes with a cool-looking texturized wristband. There are software updates too,specifically gamification features that let you compete against friends, and you no longer have to put it in sleep mode for it to tracking your sleeping patterns.

7. Tory Burch Fit bit Accessories:

For those who already have a Fitbit Flex -- in which the core tracking piece pops out of the wristband -- designer Tory Burch has a beautiful line of high-end accessories for fashionable fitness fans. The collection brings a refresh to the unattractive fitness tracker design market. Items in the line include a silicone printed bracelet ($38), a pendant necklace ($175) and a hinged bracelet ($195). The pieces are stylish to wear alone, so the fact that these disguise a fitness tracker is an added bonus.

8. Microsoft Band:

Figure -3 Microsoft band; a wearable health gadget

If you're willing to splurge a little more for a fitness wristband tracker, the Microsoft Band ($199) is your best investment. In addition to monitoring steps, workouts, calories burned, sleep cycle and heart rate, it displays calendar alerts, call and text notifications (the Fit bit Charge only does the latter) and even your tweets. It acts more like a smart watch than most fitness trackers we've seen so far.

9. Shoe Pouch for Fitness Trackers:

This shoe pouch ($14.99) from tech accessory company Griffin fits small fitness trackers like the Misfit Shine. The shoe pouch straps on to shoelaces and lets you track your runs, without wearing a clunky wristband device.

10. Fit bug KiQPlan:

Fitbug has a whole line of Bluetooth wearable’s, but its new 12-week program called KiQPlan is 12-week program called KiQPlan aims to really get you in shape. Consider it a virtual, modern-day Jenny Craig that works with other trackers and apps like Fitbit, Jawbone, MyFitnessPal and your smartphone. The virtual coaches help you lose baby weight or shed beer belly pounds -- eventually, the company will offer plans for helping people run their first 5k or marathon. Activation cards ($19.99) are sold in Target and online -- downloadable codes are available too, if you don't want to send a tangible card to someone.

11. Muse Headband:

This $300 brain-sensing headband the Muse, which was born out of a popular Indiegogo campaign, works with a series of calming exercises that are supposed to chill you out in a few minutes yet give you the effects of a nearly 30-minute yoga session. The Muse syncs with an app and focuses on training your breathing, so you can unwind from anywhere, from the office to an airplane. And it actually works: we felt the results after just one three-minute session.

12. Oral-B Toothbrush:

This web-connected toothbrush from Oral-B syncs with an app and gives you real-time data about how well you're brushing and if you're missing some key spots. It's also a good tool for kids too -- parents can monitor just how well they're doing and how they might need to improve.

13. Withings Aura:

This sleep-tracking device is undoubtedly pricy ($299.95), but for the right person, it could change their life. The device tracks REM light and deep sleep, makes suggestions on how to improve the quality of your snoozing and it emits a light that helps you both fall asleep and wake up. What's nice is that it's not a wearable you have to sleep with -- those are often uncomfortable at night -- and it also serves as an alarm clock.

2.9 Challenges:

Problems with the use of this technology could include:

- Security: Considerable effort would be required to make BAN transmission secure and accurate. It would have to be made sure that the patient ‘’secure‘’ data is only derived from each patient's dedicated BAN system and is not mixed up with other patient's data. Further, the data generated from WBAN should have secure and limited access. Although security is a high priority in most networks, little study has been done in this area for WBANs. As WBANs are resource-constrained in terms of power, memory, communication rate and computational capability, security solutions proposed for other networks may not be applicable to WBANs. Confidentiality, authentication, integrity, and freshness of data together with availability and secure management are the security requirements in WBAN.

- Interoperability: WBAN systems would have to ensure seamless data transfer across standards such as Bluetooth, Zigbee etc. to promote information exchange, plug and play device interaction. Further, the systems would have to be scalable, ensure efficient migration across networks and offer uninterrupted connectivity.
- System devices: The sensors used in WBAN would have to be low on complexity, small in form factor, light in weight, power efficient, easy to use and reconfigurable. Further, the storage devices need to facilitate remote storage and viewing of patient data as well as access to external processing and analysis tools via the Internet.
- Invasion of privacy: People might consider the WBAN technology as a potential threat to freedom, if the applications go beyond "secure" medical usage. Social acceptance would be key to this technology finding a wider application.
- Sensor validation: Pervasive sensing devices are subject to inherent communication and hardware constraints including unreliable wired/wireless network links, interference and limited power reserves. This may result in erroneous datasets being transmitted back to the end user. It is of the utmost importance especially within a healthcare domain that all sensor readings are validated. This helps to reduce false alarm generation and to identify possible weaknesses within the hardware and software design.
- Data consistency: Data residing on multiple mobile devices and wireless patient notes need to be collected and analysed in a seamless fashion. Within body area networks, vital patient datasets may be fragmented over a number of nodes and across a number of networked PCs or Laptops. If a medical practitioner′s mobile device does not contain all known information then the quality of patient care may degrade.
- Interference: The wireless link used for body sensors should reduce the interference and increase the coexistence of sensor node devices with other network devices available in the environment. This is especially important for large scale implementation of WBAN systems.
- Data Management: As BANs generate large volumes of data, the need to manage and maintain these datasets is of utmost importance.

Besides hardware-centric challenges, the following human-centric challenges should be addressed for practical BAN development. These include

- Cost: Today's consumers expect low cost health monitoring solutions which provide high functionality. WBAN implementations will need to be cost optimized to be appealing alternatives to health conscious consumers.
- Constant monitoring: Users may require different levels of monitoring, for example those at risk of cardiac ischemia may want their WBANs to function constantly, while others at risk of falls may only need WBANs to monitor them while they are walking or moving.

3. WBAN WIRELESS COMMUNICATIONS

Long-life, persistent sensor nodes require efficient power management. With highly integrated electronics, the sensor size and weight becomes dominated by battery selection. An implementation must address conflicting requirements for small size and infrequent battery maintenance, striving for a balance that will maximize user compliance. It is our challenge as designers to minimize sensor power consumption and thus maximize battery life for a given size. In designing our prototype we have held low power consumption as a primary design goal in every component of the system – in processor and technology selection, in managing sensor data, in network organization, and in efficient communications.

Power consumption of the sensor node is dominated by the wireless radio. Nearly 85% can be attributed to CC2420 controller – even when not actively transmitting. The CC2420, although the lowest power of its kind, still draws 17.4mA when transmitting and 19.7mA when receiving. In contrast, the MSP430 utilizes 250µA/MIPS – typically just over 1mA when active. As an example, the MSP430 can execute 100,000 instructions for the same cost of transmitting a single 40 byte message. With that in mind, power savings can be realized by disabling the radio when not in use as well as reducing the total quantity of transmission – even if extensive computation is required. Besides power efficiency, we were motivated to implement simple and scalable communications, to use standards-based protocols, and to support multiple simultaneous WBANs within close proximity of one another. The resulting solution spans multiple layers, is IEEE 802.15.4 compliant and upholds the Zigbee star network topology. It leverages existing communication framework within TinyOS and addresses practical WBAN implementation issues.

3.1 IEEE 802.15.4 and ZigBee

Our prototype WBAN utilizes the IEEE 802.15.4 compliant XBEE series2 radio for wireless communications. The IEEE 802.15.4 standard defines communications for nodes in a low-rate wireless personal area network (LR-WPAN) and is well suited for our prototype WBAN. The standard specifies the physical (PHY) layer and data link / media access control (MAC) layer. At the physical layer, IEEE 802.15.4 defines three frequency bands, spread spectrum chip rate, and data encoding [IEEE802.15.4]. XBEE series2 radio works at a maximum frequency of 2.4 GHz. The standard specifies 16 channels in the 2.4 GHz ISM band. Series 2 modules allows create complex mesh networks based on the XBee ZB ZigBee mesh firmware Channel selection is based on the programming tool

XCTU provided by the XBEE manufacturers. By exploiting different 802.15.4 channels, we have been able to operate multiple simultaneous WBANs in close proximity without interference. At present this feature is statically assigned, but satisfies proof of concept. IEEE 802.15.4 employs a carrier sense multiple access with collision avoidance (CSMA-CA) scheme for peer-to-peer communications. In the simplest form, communications are asynchronous and random access can occur. IEEE 802.1.5.4 includes specification for an optional super frame structure utilizing device timeslots which we exploit in the next section. ZigBee and IEEE 802.15.4 are cooperating protocol stacks. ZigBee is tightly coupled to 802.15.4 in that the PHY and MAC layers are specified to be IEEE 802.15.4; however, the ZigBee specification details the upper protocol layers – network, application and application sub layer, and security. It specifies network topologies, routing mechanisms and dynamic discovery and registration of nodes as they enter and exit the network. ZigBee defines three network topologies: star, tree, and mesh. In the star topology a single node serves as the network coordinator; nodes communicate directly to the network coordinator, but not peer-to-peer. In a tree topology, nodes are arranged hierarchically so that each node communicates to a designated router node. Traffic propagates through the network by visiting router nodes. A mesh network topology allows full peer-to-peer communications [ZigBee]. Based on the human-centric WBAN model, we are able to exploit the simplified star topology. Whenever possible we upheld the spirit of the ZigBee specification, but did not restrict ourselves to conform to the ZigBee specification. Instead of developing this functionality, we chose to maintain the spirit of ZigBee, but allow for a simpler implementation. By doing so, our efforts could be focused on exploring more general challenges in WBAN implementation and maintaining flexibility for future implementations that may not use ZigBee – a Bluetooth WBAN for example. Compared to Bluetooth which is primarily designed for wireless cable replacement for electronic devices, 802.15.4 / ZigBee offers lower data rates and lower power consumption. Bluetooth is limited to a relatively small number of network participants while 802.15.4 scales upward to 65,536 nodes. In addition, 802.15.4 implementations have smaller memory footprints.

3.2 IEEE 802.15.4 SUPER FRAME STRUCTURE:

In the beacon-enabled mode of IEEE 802.15.4, each node employs two system parameters: BO and SO, which define beacon interval and SD, respectively, i.e.

$$\mathrm{BI} = \mathrm{BaseSuperframeDuration} \times 2^{\mathrm{BO}} \text{ and } \mathrm{SD} = \mathrm{BaseSuperframeDuration} \times 2^{\mathrm{SO}}, \text{ for } 0 \leq \mathrm{SO} \leq \mathrm{BO} \leq 14.$$

BaseSuperframeDuration denotes the minimum number of symbols in an active period, which is fixed to 960 symbols. The active period of each superframe consists of three parts: beacon, contention access period (CAP) and contention free period (CFP), while the active period is further equally divided into 16 time slots called NumSuperframeSlots. The length of one slot is equal to BaseSlotDuration × 2SO symbols, where BaseSlotDuration is the minimum number of symbols in a slot and equal to 60 symbols.

In IEEE 802.15.4 standard, BO and SO shall be equal for all superframes on a PAN. All devices shall interact with the PAN only during the active portion of a superframe. In CAP, each node performs the CSMA/CA algorithm before transmitting data frame or MAC command frame. Each device maintains three parameters: the number of backoff (NB), contention window (CW), and backoff exponent (BE). NB denotes the required NB while attempting to transmit data; CW denotes the number of backoff periods that need to be clear before committing transmission; and BE denotes how many backoff periods a device need to wait before trying to access the channel. The initial value of NB, CW, and BE are equal to 0, 2, and macMinBE, respectively, where macMinBE is equal to 3. In the located boundary of the next backoff period, a device takes delay for random backoff between 0 and 2BE – 1 unit backoff period (UBP), where UBP is equal to 20 symbols (or 80 bits). A device performs clear channel assessment (CCA) to make sure whether the channel is idle or busy, when the number of random backoff periods is decreased to 0. The value of CW will be 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15.

$$\text{Inactive Period} = \text{BI} - \text{BaseSuperframeDuration} * 2^{SO} \text{ symbols}$$

$$\text{BI} = \text{BaseSuperframeDuration} * 2^{BO} \text{ symbols}$$

It is decreased by one if the channel is idle; and the second CCA will be performed if the value of CW is not equal to 0. If the value of CW is equal to 0, it means that the channel is idle after twice CCA; then a device is committed the data transmission. However, if the CCA is busy, the value of CW will reset to 2; the value of NB is increased by 1; and the value of BE is increased by 1 up to the maximum BE (macMaxBE), where the value macMaxBE is equal to 5. The device will repeatedly take random delay if the value of NB is less than the value of macMaxCSMABackoff, where the value of macMaxCSMABackoff is equal to 4; and the transmission attempt is decided to be failure if the value of NB is greater than the value of macMaxCSMABackoff.

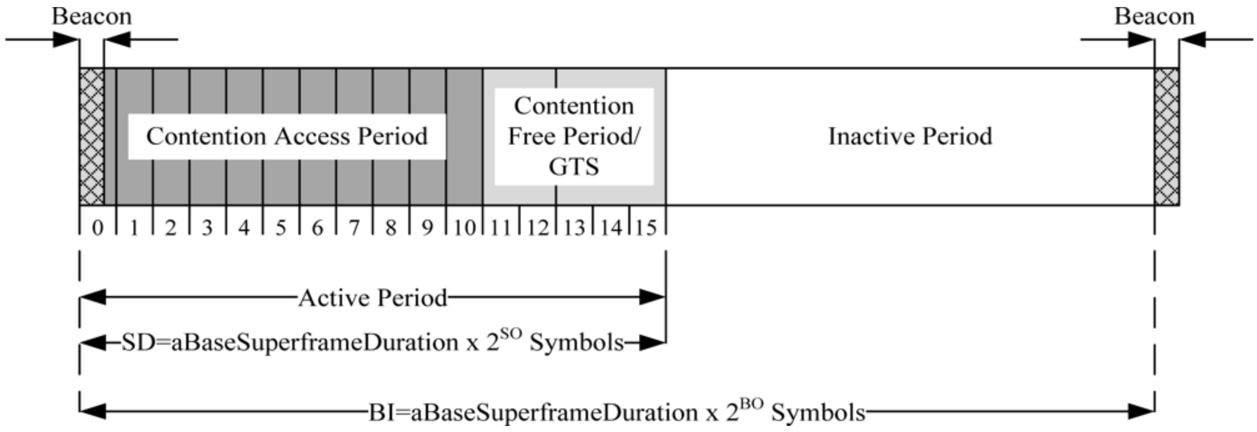

Figure- 4 IEEE 802.15.4 SUPER FRAME STRUCTURE

3.3 IEEE 802.15.6

As a branch of WSN and an important part of the internet of things (IOT), the wireless body area network (WBAN) was originally created to improve the level of personal health care. And with the development of wireless technologies, the application of the WBAN has expanded to entertainment, leisure, military and other fields. So far, IEEE 802 Task Group has developed a new IEEE 802.15.6 standard.

Various IEEE standards are shown below,

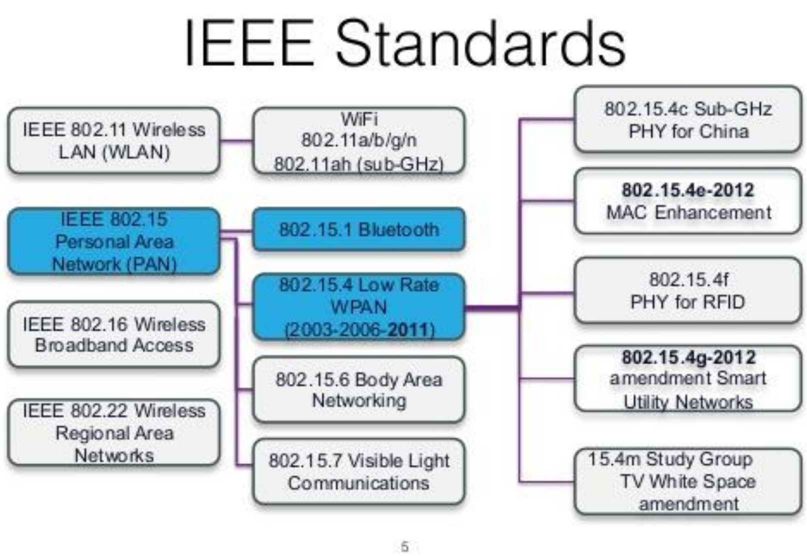

Figure -5 Various IEEE Standards

Comparison between data rate and battery life of different various IEEE standards are given below

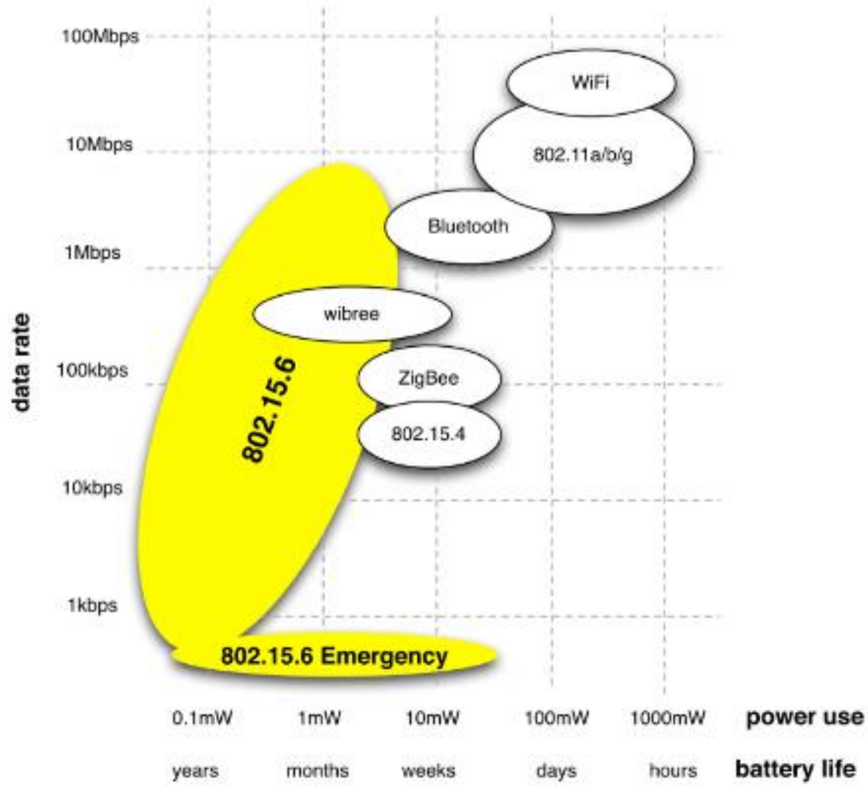

Figure -6 Plots between data rate and power use for different IEEE standards

Introduction-

Wireless Body Area Networks (WBAN) has emerged as a key technology to provide real-time health monitoring of a patient and diagnose many life threatening diseases. WBAN operates in close vicinity to, on, or inside a human body and supports a variety of medical and non-medical applications. IEEE 802 has established a Task Group called IEEE 802.15.6 for the standardization of WBAN. The purpose of the group is to establish a communication standard optimized for low-power in-body/on-body nodes to serve a variety of medical and non-medical applications. This paper explains the most important features of the new IEEE 802.15.6 standard. The standard defines a Medium Access Control (MAC) layer supporting several Physical (PHY) layers. We briefly overview the PHY and MAC layers specifications together with the bandwidth efficiency of IEEE 802.15.6 standard.

In December 2011, the IEEE 802.15.6 task group approved a draft of a standard for Body Area Network (BAN) technologies. The draft was approved on 22 July 2011 by Letter Ballot to start the Sponsor Ballot process. Task Group 6 was formed in November 2007 to focus on a low-power and short-range wireless standard to be optimized for devices and operation on, in, or around the human body (but not limited to humans) to serve a variety of applications including medical, consumer electronics, and personal entertainment.

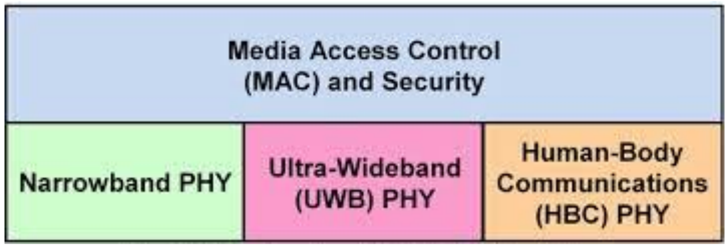

Figure -7 High level overview of the IEEE 802.15.6 Architecture

The WBAN applications targeted by the IEEE 802.15.6 standard are divided into medical and non-medical applications as given in Fig. 8. Medical applications include collecting vital. Information of a patient continuously and forward it to a remote monitoring station for further analysis. This huge amount of data can be used to prevent the occurrence of myocardial infarction and treat various diseases such as gastrointestinal tract, cancer, asthma, and neurological disorder. WBAN can also be used to help people with disabilities. For example, retina prosthesis chips can be implanted in the human eye to see at an adequate level. Non-medical applications include monitoring forgotten things, data file transfer, gaming, and social networking applications. In gaming, sensors in WBAN can collect coordinates movements of different parts of the body and subsequently make the movement of a character in the game, e.g., moving soccer player or capturing the intensity of a ball in table tennis. The use of WBAN in social networking allows people to exchange digital profile or business card only by shaking hands.

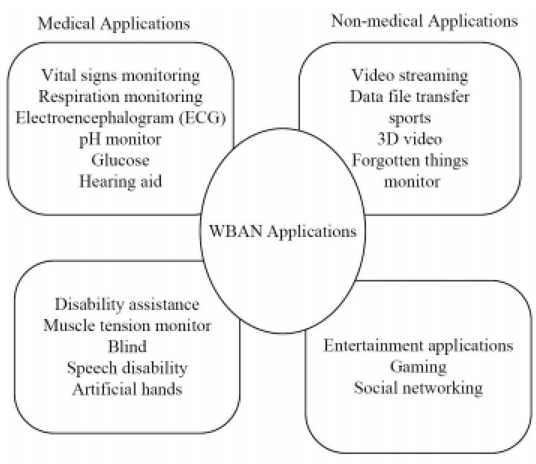

Figure -8 WBAN Applications

The current IEEE 802.15.6 standard defines three PHY layers, i.e., Narrowband (NB), Ultra wideband (UWB), and Human Body Communications (HBC) layers. The selection of each PHY depends on the application requirements. On the top of it, the standard defines a sophisticated MAC protocol that controls access to the channel. For time referenced resource allocations, the hub (or the coordinator) divides the time axis (or the channel) into a series of superframes. The superframes are bounded by beacon periods of equal length. To ensure high level security, the standard defines three levels: 1) level 0 - unsecured communication, 2) level 1 – authentication only, 3) level 2 - both authentication and encryption. We briefly overview the PHY and MAC layers specifications together with the bandwidth efficiency of IEEE 802.15.6 standard for different frequency bands and data rates.

According to the IEEE 802.15.6 standard, the nodes are organized into one- or two-hop star WBANs. A single coordinator or hub controls the entire operation of each WBAN. The WBAN must have one hub and a number of nodes, ranging from zero to mMaxBANSize. In a two-hop start WBAN, a relay-capable node may be used to exchange data frames between a node and the hub. The standard divides the time axis or channel into beacon periods or super frames of equal length. Each super frame contains a number of allocation slots that are used for data transmission. These slots have equal duration and are numbered from 0 to, where. The hub transmits beacons to define the super frame boundaries and allocate the slots. For nonbeacon modes, the super frame boundaries where beacons are not used are defined by polling frames. Generally, the hub transmits beacons in each super frame except those that are inactive. The hub may shift or rotate the offsets of the beacon periods, thus shifting the schedule allocation slots. The following sections present the MAC frame format, communication modes, and access mechanisms defined in the IEEE 802.15.6 standard.

Frequency Band	Packet Component	Modulation	Symbol Rate (Kbps)	Code Rate BCH (n,k)	Information Data Rate (Kbps)
402 - 405 MHz	PLCP Header	π/2-DBPSK	187.5	(31,19)	57.5
	PSDU	π/2-DBPSK	187.5	(63,51)	75.9
	PSDU	π/4-DQPSK	187.5	(63,51)	303.6
420 - 450 MHz	PLCP Header	GMSK	187.5	(31,19)	57.5
	PSDU	GMSK	187.5	(63,51)	75.9
	PSDU	GMSK	187.5	(63,51)	151.8
863 - 870 MHz	PLCP Header	π/2-DBPSK	250	(31,19)	76.6
	PSDU	π/2-DBPSK	250	(63,51)	101.2
	PSDU	π/4-DQPSK	250	(63,51)	404.8
902 - 928 MHz	PLCP Header	π/2-DBPSK	300	(31,19)	91.9
	PSDU	π/2-DBPSK	300	(63,51)	121.4
	PSDU	π/4-DQPSK	300	(63,51)	485.7
950 - 956 MHz	PLCP Header	π/2-DBPSK	250	(31,19)	76.6
	PSDU	π/2-DBPSK	250	(63,51)	101.2
	PSDU	π/4-DQPSK	250	(63,51)	404.8
2360-2400 MHz 2400-2483.5 MHz	PLCP Header	π/2-DBPSK	600	(31,19)	91.9
	PSDU	π/2-DBPSK	600	(63,51)	121.4
	PSDU	π/2-DBPSK	600	(63,51)	485.7

Table- 2 Modulation Parameters for PLCP header and PSDU

3.2.1 PHY Layer Specification

As mentioned earlier, the IEEE 802.15.6 supports three different PHYs, i.e., NB, UWB, and HBC.

1) Narrowband PHY (NB): The NB PHY is responsible for activation/deactivation of the radio transceiver, Clear Channel Assessment (CCA) within the current channel and data transmission/reception. The Physical Protocol Data Unit (PPDU) frame of NB PHY contains a Physical Layer Convergence Procedure (PLCP) preamble, a PLCP header, and a PHY Service Data Unit (PSDU) as given below. The PLCP preamble helps the receiver in the timing synchronization and carrier-offset recovery. It is the first component being transmitted. The PLCP header conveys information necessary for a successful decoding of a packet to the receiver. The PLCP header is transmitted after PLCP preamble using the given header data rate in the operating frequency band. The last component of PPDU is PSDU which consists of a MAC header, MAC frame body, Frame Check Sequence (FCS) and is transmitted after PLCP header using any of the available data rates in the operating frequency band. A WBAN device should be able to support transmission and reception in one of the frequency bands summarized in Table I. The table further shows the data-rate dependent modulations parameters for PLCP header and PSDU. In NB PHY, the standard uses Differential Binary

Phase-shift Keying (DBPSK), Differential Quadrature Phase-shift Keying (DQPSK), and Differential 8-Phase-shift Keying (D8PSK) modulation techniques except 420-450 MHz which uses a Gaussian minimum shift keying (GMSK) technique.

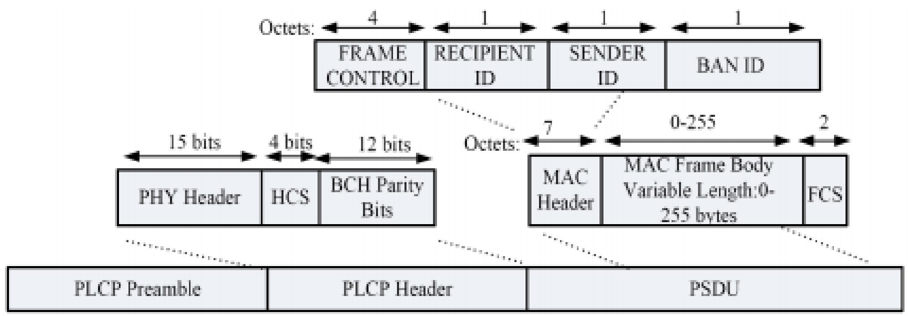

Figure -9 IEEE 802.15.6 NB PPDU structure

2) Ultra Wideband PHY (UWB): UWB PHY operates in two frequency bands: low band and high band. Each band is divided into channels, all of them characterized by a bandwidth of 499.2 MHz .The low band consists of 3 channels (1-3) only the channel 2 has a central frequency of 3993.6 MHz and is considered a mandatory channel. The high band consists of eight channels (4-11) where channel 7 with a central frequency 7987.2 MHz is considered a mandatory channel, while all other channels are optional. A typical UWB device should support at least one of the mandatory channels. The UWB PHY transceivers allow low implementation complexity and generate signal power levels in the order of those used in theMICS band. Fig. 4 shows the UWB PPDU that contains a Synchronization Header (SHR), a PHY Header (PHR), and PSDU. The SHR is composed of a preamble and a Start Frame Delimiter (SFD). The PHR conveys information about the data rate of the PSDU, length of the payload and scrambler seed. The information in the PHR is used by the receiver in order to decode the PSDU. Typical data rates range from 0.5 Mbps up to 10 Mbps .

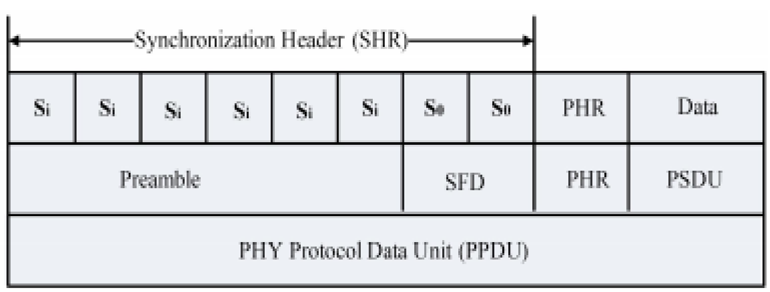

Figure -10 IEEE 802.15.6 UWB PPDU structure

3) Human Body Communications PHY (HBC): HBC PHY operates in two frequency bands centered at 16 MHz and 27 MHz with the bandwidth of 4 MHz. Both operating bands are valid for the United States, Japan, and Korea, and the operating band at 27MHz is valid for Europe. HBC is the Electrostatic Field Communication (EFC) specification of PHY, which covers the entire protocol for WBAN such as packet structure, modulation, preamble/SFD, etc. Fig. 5 describes the PPDU structure of EFC that is composed of a preamble, SFD, PHY header and PSDU. The preamble and SFD are fixed data patterns. They are pre-generated and sent ahead of the packet header and payload. The preamble sequence is transmitted four times in order to ensure packet synchronization while the SFD is transmitted only once. When the packet is received by the receiver, it finds the start of the packet by detecting the preamble sequence, and then it finds the start of the frame by detecting the SFD.

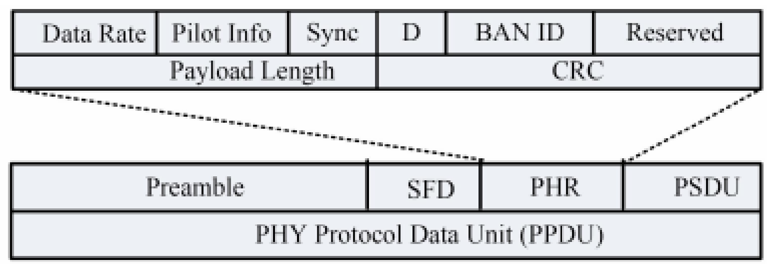

Figure -11 IEEE 802.15.6 EFC PPDU structure

3.2.2 IEEE 802.15.6 MAC Frame Format

Figure 13 shows the general MAC frame format consisting of a 56-bit header, variable length frame body, and 18-bit Frame Check Sequence (FCS). The maximum length of the frame body is 255 octets. The MAC header further consists of 32-bit frame control, 8-bit recipient Identification (ID), 8-bit sender ID, and 8-bit WBAN ID fields. The frame control field carries control information including the type of frame, that is, beacon, acknowledgement, or other control frames. The recipient and sender ID fields contain the address information of the recipient and the sender of the data frame, respectively. The WBAN ID contains information on the WBAN in which the transmission is active. The first 8-bit field in the MAC frame body carries message freshness information required for nonce construction and replay detection. The frame payload field carries data frames, and the last 32-bit Message Integrity Code (MIC) carries information about the authenticity and integrity of the frame.

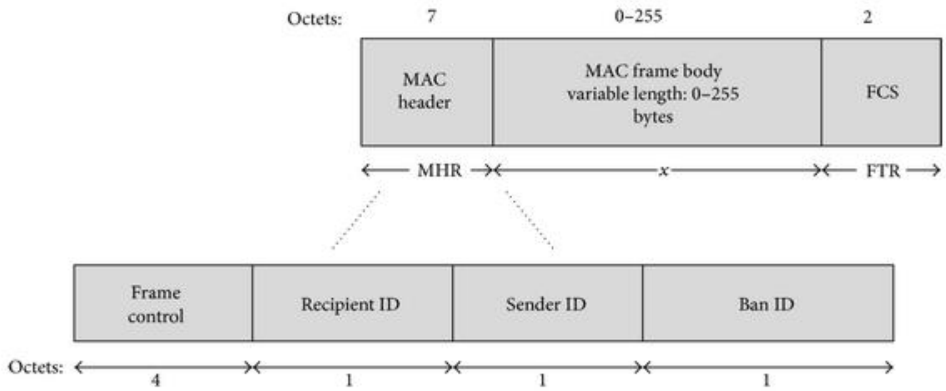

Figure -12 IEEE 802.15.6 MAC Frame Format

The IEEE 802.15.6 supports the following communication modes.

- Beacon Mode with Super frame Boundaries

In this mode, the hub transmits beacons in active super frames. The active super1 frames may be followed by several inactive super frames whenever there is no scheduled transmission. As illustrated in Figure 13.1, the super frame structure is divided into Exclusive Access Phases (EAP1 and EAP2), Random Access Phases (RAP1 and RAP2), a Managed Access Phase (MAP), and a Contention Access Phase (CAP). The EAPs are used to transfer high-priority or emergency traffic. The RAPs and CAP are used for nonrecurring traffic. The MAP period is used for scheduled and unscheduled blink allocations, scheduled uplink and downlink allocations, and Type I (not Type II) polled and posted allocations. The length of Type I and Type II allocations is represented in terms of the transmission time and number of frames, respectively.

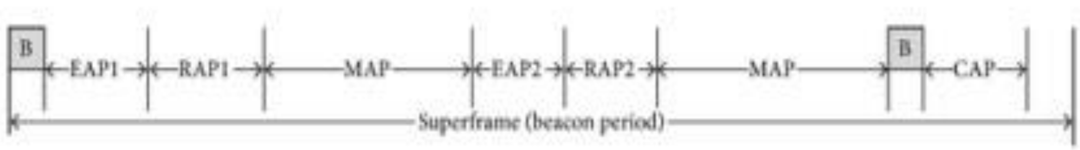

Figure -13.1 Beacon mode with super frame boundaries

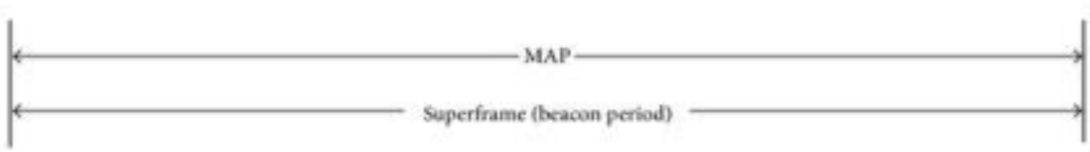

Figure -13.2 Nonbeacon mode with super frame boundaries

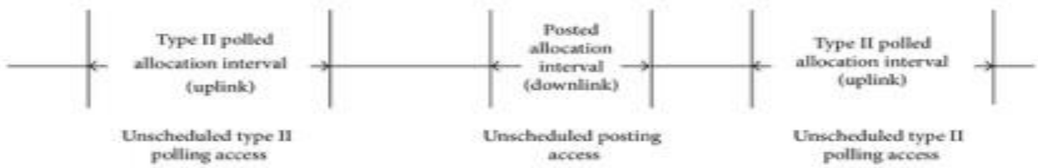

Figure -13.3 Nonbeacon mode without super frame boundaries

Figure 13: IEEE 802.15.6 communication modes.

- Nonbeacon Mode with Super frame Boundaries

In this mode, the hub operates during the MAP period only, as illustrated in Figure 13.2.

- Nonbeacon Mode without Super frame Boundaries

In this mode, the hub provides unscheduled Type II polled or posted allocations or a combination of both, as depicted in Figure 13.3.

- Random Access Mechanism

In EAP, RAP, and CAP periods, the hub may employ either a slotted ALOHA or Carrier Sensor Multiple Access/Collision Avoidance (CSMA/CA) protocol, depending on the PHY. The hub considers slotted ALOHA and CSMA/CA protocols for UWB and NB PHYs, respectively. To send high-priority data frames using CSMA/CA, the hub may combine EAP1 and RAP1 into a single EAP1 period and EAP2 and RAP2 into a signal EAP2 period. When using slotted ALOHA for high-priority traffic, EAP1 and EAP2 are not extended, but RAP1 and RAP2 are simply replaced by another EAP1 and EAP2 period. The following sections briefly describe the slotted ALOHA and CSMA/CA protocols.

- Slotted ALOHA Protocol.

In the slotted ALOHA protocol, the nodes access the channel using predefined User Priorities (UPs), as given in Table -3. These priorities are used to classify the high- and low-priority traffic. Initially, the Collision Probability (CP) is selected according to the UPs. The nodes obtain contended allocation if, where is randomly selected from the interval. If the node fails to transmit, the CP remains unchanged into an odd number of failures and the node divides it equally for an even number of failures.

User Priorities	Slotted-ALOHA		CSMA/CA	
	CP_{max}	CP_{min}	CW_{min}	CW_{max}
0	0.125	0.0625	16	64
1	0.125	0.0937	16	32
2	0.25	0.0937	8	32
3	0.25	0.125	8	16
4	0.375	0.125	4	16
5	0.375	0.1875	4	8
6	0.5	0.1875	2	8
7	1	0.25	1	4

Table -3 Bounds for slotted-ALOHA and CSMA/CA protocols

- CSMA/CA Protocol.

In the CSMA/CA protocol, the node initially sets its back off counter to a random integer that is uniformly distributed over the interval, where. As given in Table -3, the values of and are selected according to the UPs. The high-priority traffic will have a small contention window compared to that of low-priority traffic, which increases the probability of accessing the channel to report emergency events. The node starts decrementing the backoff counter by one for each idle CSMA slot with a length equal to CSMA SlotLength. Particularly, the node considers a CSMA slot to be idle if it determines that the channel has been idle between the start of the CSMA slot and CCATime. The node decreases the backoff counter CCATime after the start of the CSMA slot. Once the back-off counter reaches zero, the node transmits the frame. If the channel is busy due to frame transmission, the node locks its backoff counter until the channel is idle. It is doubled for an even number of failures until it reaches. Figure 16 shows an example of the CSMA/CA protocol. As shown in the figure, the node unlocks the backoff counter in RAP1. However, the contention fails and the value of remains unchanged because does not change for an odd number of failures. In the following CAP period, the backoff counter is set to five; however, it is locked at two because the time between the end of the slot and the end of the CAP is not sufficient to accommodate the data frame transmission and Nominal Guard Time. The backoff counter is then unlocked in the RAP2 period. This time, the value of is doubled because there is an even number of contention failures. The backoff counter is set to eight and is unlocked. Once the backoff counter reaches zero, the data are transmitted and the value of is set to.

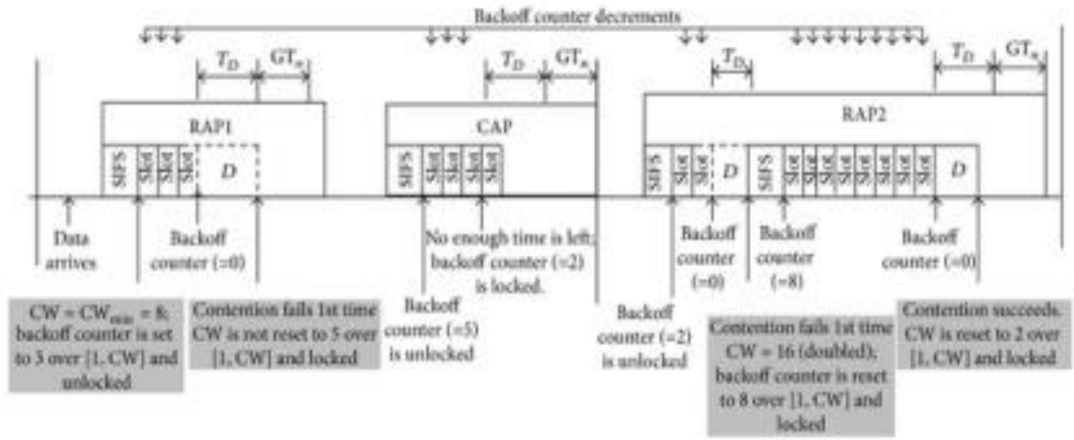

Figure-14: IEEE 802.15.6 CSMA/CA protocol: slot = CSMA slot SIFS = Psifs, frame transaction initiated by node 1 in a contended allocation (e.g., a data type frame and an I-Ack frame with pSIFS in between), = time required to complete, nominal guard time.

- Improvised and Unscheduled Access Mechanism

As discussed above, the hub may use improvised access to send poll or post commands without preservation or advance notice in beacon or Nonbeacon modes with super frame boundaries. These commands are used to initiate the transactions of one or more data frames by the nodes or hub outside the scheduled allocation interval. The polls are used to grant Type I or Type II polled allocation to the nodes, while the posts are used to send management frames. The Type I polled allocation starts after the duration of pSIFS and stops at the end of the allocated slot in the current super frame. Figure15 illustrates an example of immediate polled allocations.

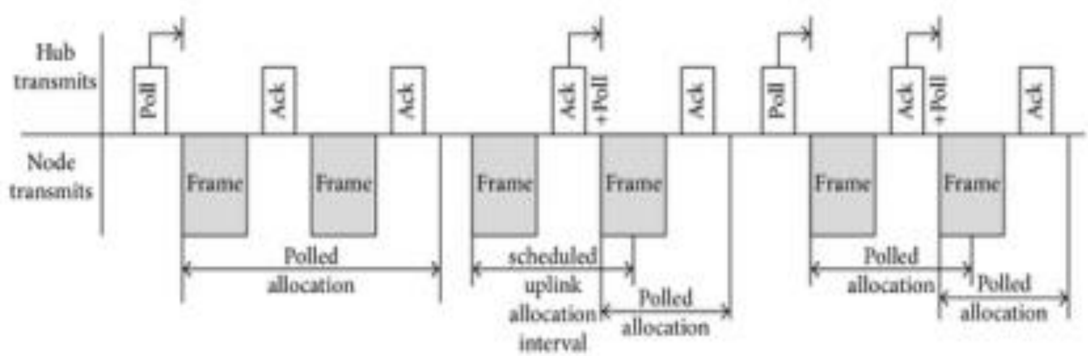

Figure 15: Immediate polled allocations.

The hub may also use an unscheduled access mechanism to obtain an unscheduled bilink allocation. The unscheduled bilink allocation may be (1) one-periodic, where frames are exchanged between the nodes and hub every super frame, or (2) multiple-periodic (m-periodic), where frames are exchanged every m super frames thus allowing the devices to sleep between m super frames. An m-periodic bilink allocation is suitable for low-duty cycle nodes because nodes in m-periodic allocation sleep between m super frames.

- Scheduled and Scheduled-Polling Access Mechanisms

Unlike unscheduled allocation, the scheduled access mechanism is used to obtain scheduled uplink, downlink, and bilink allocations. In addition, the scheduled polling is used for polled and posted allocations. These allocations may be one-periodic or m-periodic; however, neither of these allocations is allowed in a single WBAN at the same time. The nodes consider the super frame periods (with allocated slots) as the wakeup periods. The uplink and downlink allocations are used to send management and data frames to and from the hub, respectively. Figure 16 illustrates an example of scheduled one-periodic allocations.

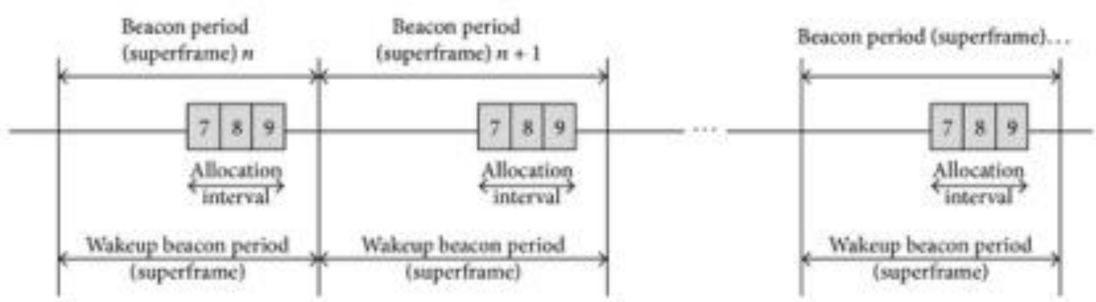

Figure- 16 Scheduled one-periodic allocation

4. WORKING MODEL

4.1Block Diagram:

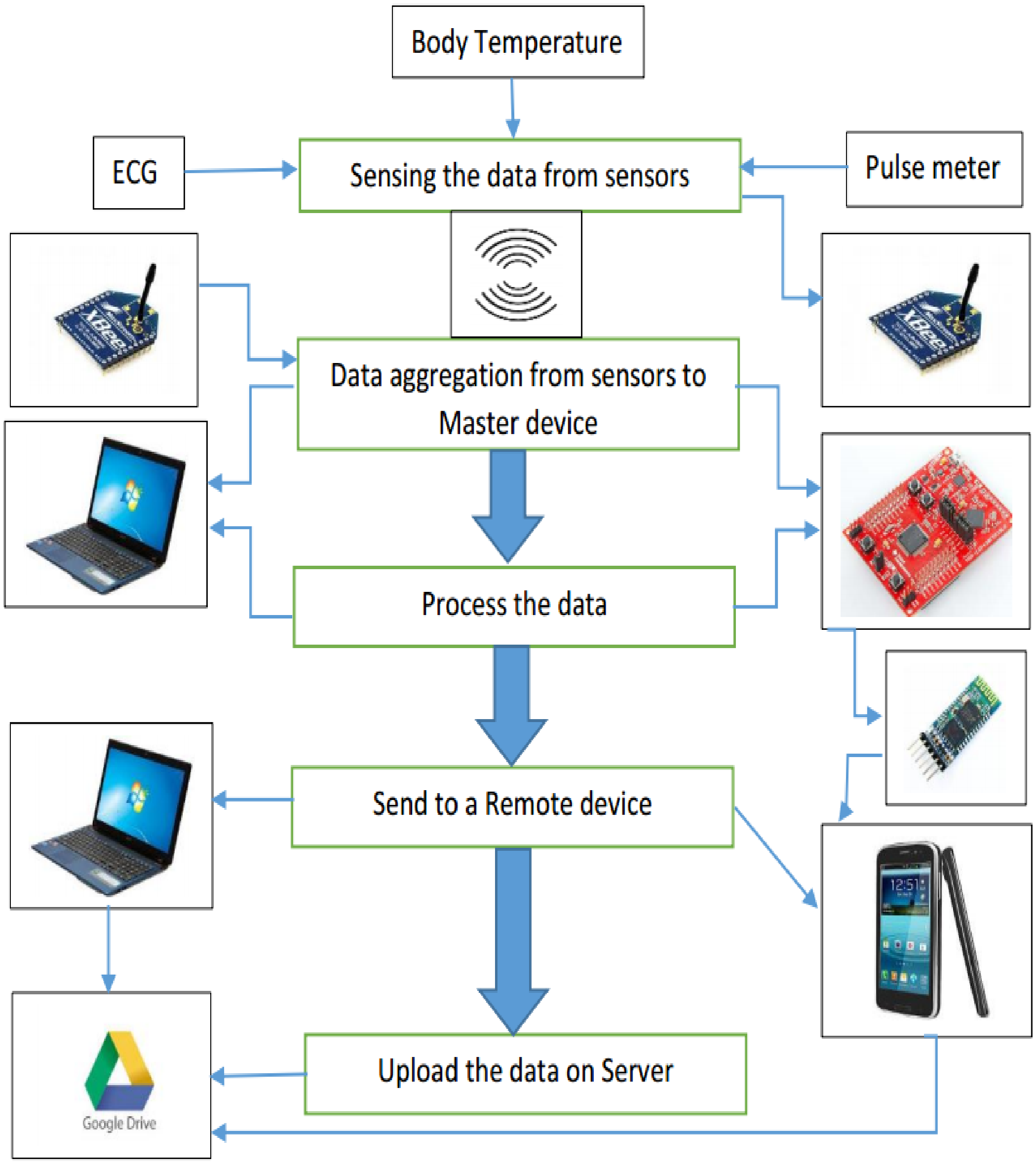

Figure -17 Block Diagram of our Working Model

The whole system architecture is shown in figure. It is composed of medical sensor nodes, a hand-held personal server, a ZigBee module, microcontroller and related services like server, cloud and mobile app. In this system, medical sensor nodes are used to collect physiological signals including bio-signals, medical images, and voice signals. These obtained signals are fed into the personal server through wireless personal area network (WPAN) using ZigBee. The wireless communication between the sensor nodes on patient body and the hand-held personal controller which gets data from the sensors uses IEEE 820.15.4/ZigBee standard. All this communication was established and maintained by the head or the Xbee which is at the server side. After arriving at the personal server, the data are either stored in the clinical data base, or available to a concerned doctor or pc which monitors all the data through a local area network (LAN) or Bluetooth. Then clinicians can analyze the physiological data and give diagnosis advices accordingly. There are many other applications that are done using the same architecture, as we need not consider the data when the patient status is normal but we need attention when reading are out of the threshold. So to reduce the work we can program the microcontroller that when the received data is out of the predefined set then immediately send the information to the doctor otherwise it can just store the data for any references and delete if it runs out of space or when data was of no use. There can also another application like a GPS module is placed on patient body and it continuously send the location to the controller ,when the patient condition is too Sevier as the patient reading are very aberrant from safe condition like if he goes to comma or he can't make a call. In such condition the controller will immediately send the data to the nearby hospital, so that immediately we can have ambulance facility and arrangements can be made in hospital for treatment.

Many more applications are possible with WBAN, where patient can do his day to day activities without any problem while WBAN will take care of the health condition with the continuous monitoring.

The main tasks of the medical sensors are to collect physiological signals and send them to the personal server. Typical medical sensors and characteristics of the signals are given in the next sub section.

4.1 Sensor Nodes:

4.1.1 ECG:

The aim is to design a sensitive amplifier circuit that can detect ECG (Electrocardiogram) signals found from electrodes applied at the left arm (LA), Right arm (RA) and right leg (RL). The difference between the right-arm (RA) lead and the left-arm (LA) lead is amplified by circuit with the right-leg (RL) lead as the ground or reference node. The circuit needs a two stage instrumentation amplifier which has buffering stage and amplification stage. After that filtering stage is required for signal filtration.

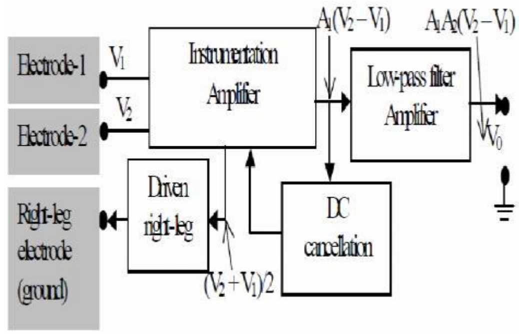

Figure-18: General Block Diagram of ECG Amplifier

A general block diagram of a modern ECG amplifier is shown in Fig. 18. It is designed around an instrumentation amplifier. In addition, the instrumentation amplifier has feature of an automatic offset (DC) cancellation circuit to keep the output always zero averaged. The electrical safety is provided by driven-right leg circuit and the interference is reduced under normal operational conditions. Low-pass filter provides further amplification and limit the bandwidth of the circuit.

ECG is recording most commonly between the Right Arm (RA) and the Left Arm (LA). Sometimes another two combinations using the Left Leg (LL) are also used clinically (RA−LL and LA−LL). For common ground of the Instrumentation amplifier, there are one another electrode connects to the patient. This is attached to the right leg.

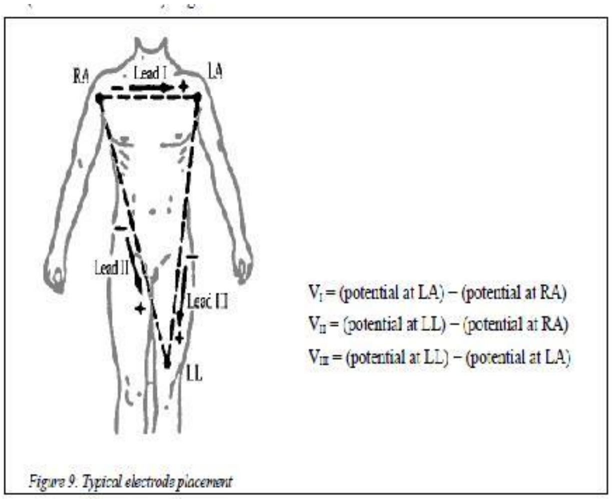

Figure-19: Typical Electrode Placement

The movement of electrode with respect to the electrolyte is mechanically disturbed. So, the distribution of charge at the interface and results in a momentaneous change of the half–cell potential until equilibrium can be restored. If one electrode is moved while the other remains stable, a potential difference appears between the two electrodes during this movement. Due to this kind of movement the potential is referred to as movement artifacts and this can be a grievous cause of interference in the measurement of ECG.

To fulfil the requirements of our ECG amplifier, we need to design a cascade circuit, which is a combination of an Instrumentation Amplifier, a Low Pass Filter, a High Pass Filter and a gain stage. For reducing noises, the order of cascade stages is considered. For example, in the following cascade Figure, the output noise is

$$((n_1 * A_1 + n_2) * A_2 + n_3) * A_3 = A_1 A_2 A_3 * n_1 + A_2 A_3 * n_2 + A_3 * n_3, \; A_1 > A_2 > A_3.$$

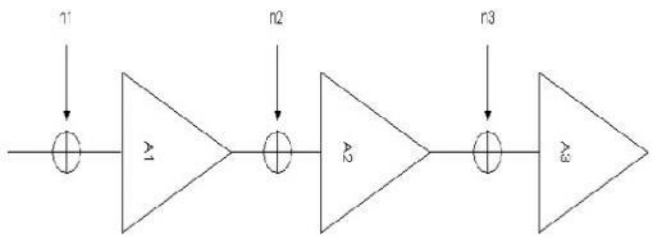

Figure - 20: Cascade Design of Amplifier

Cascade design is on the basis of placing high gain stages in the signal path. All the same, the High Pass Filter stage should be placed immediately after the differential amplifier to remove DC offset.

The AD620 is a low cost, high accuracy of 40 ppm with maximum nonlinearity instrumentation amplifier that requires only one external resistor to set gains of 1 to 1000. The AD620 has low offset voltage of 50 mV max and offset drift of 0.6 mV/°C max.

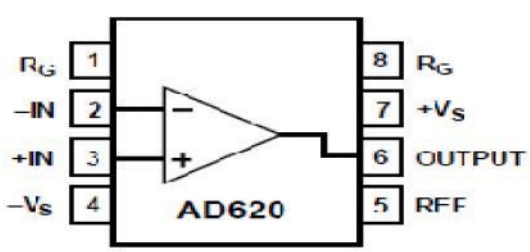

Figure - 21: Pin Diagram of AD620

Furthermore, it has low noise, low input bias current and low power which makes it well suited for medical applications such as ECG and non-invasive blood pressure monitors. Gain of the AD620 is set by connecting a single external resistor RG, as shown commonly used gains and RG resistor values. The internal gain resistors R1 and R2. Trimmed to an absolute value of 24.7KΩ, allowing the gain to be programmed accurately with a single external resistor. These resistors are affected by the accuracy and temperature coefficient included in the gain accuracy.

1% Std Table Value of R_G, Ω	Calculated Gain	0.1% Std Table Value of R_G, Ω	Calculated Gain
49.9 k	1.990	49.3 k	2.002
12.4 k	4.984	12.4 k	4.984
5.49 k	9.998	5.49 k	9.998
2.61 k	19.93	2.61 k	19.93
1.00 k	50.40	1.01 k	49.91
499	100.0	499	100.0
249	199.4	249	199.4
100	495.0	98.8	501.0
49.9	991.0	49.3	1,003

Table -4 Values of Rg for settling Gain of AD620

The stability and temperature drift of the external gain setting resistor, RG, also affects gain. RG's contribution to gain accuracy and drift can be directly inferred from the gain equation. There are many important features of AD620 such as gain set with one resistor, wide range power supply, 100dB min CMRR (common-mode rejection ratio), low noise and excellent DC performance. AD620 is used in many applications like portable battery operated system, physiological amplifier: EEG, ECG, EMG, multi-channel data acquisition, ECG and medical instrumentation etc.

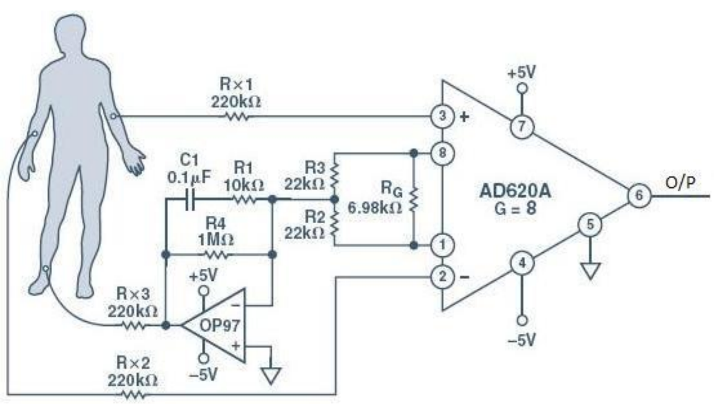

Figure-22: ECG Circuit with Right Leg Driven Circuit

The average heart rate of a person is around 1.1Hz and the signal level is very weak thus to avoid interference from other signals, a band pass filter is needed. The desired frequencies (between 0.05 Hz and 150Hz) can be achieved via operational amplifier, capacitors and Resistors where the values are found for the high pass and low pass respectively.

Low pass 0.05 Hz= $(2*pi * R* C)^{-1}$

High Pass 150 Hz = $(2*pi * R* C)^{-1}$

For High Pass filter R = 3.18 MΩ, C= 1μF

For Low Pass filter R = 106 KΩ, C= 0.01μF

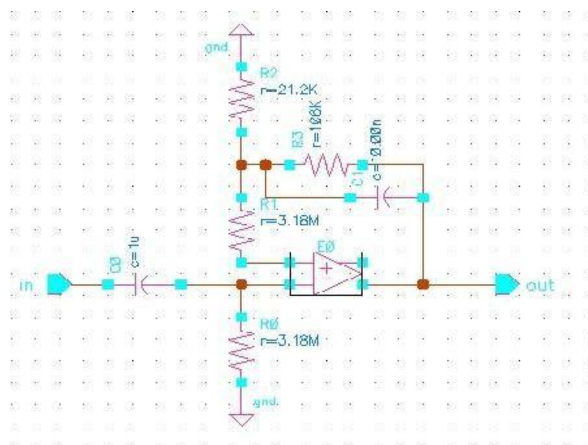

Figure -23: Filter Circuit for ECG

Thus the filter supplies a gain of 5. The Resistor 3.18MΩ connected to non-inverting input is for keeping the balance and symmetry.

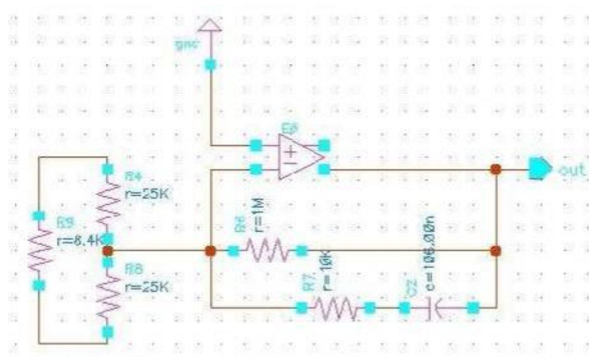

Figure - 24: Right Leg Driven Circuit

The aim of right leg driven circuit is reducing the effect of noise. The common mode signal taken from the both ends of the Rg (gain resistance of AD620) is given back to body as a reference.

The values of the resistances are 25KΩ for the input resistance. Also a low pass filter with 150Hz cut-off frequency is existent with R = 10KΩ and c =106nf. The gain of the circuitry is defined by 1MΩ resistance.

The last stage of the circuit is both designed to provide the necessary gain and the necessary voltage shifting to make the signal appear on the limits is 0 to 5 volts. This stage is expected to have a gain of 25. Since the amount of voltage shifting is unknown the voltage shifting is applied after the first ECG signal is observed. Initially an inverting amplifier with R1=5KΩ and R2=125KΩ used.

The circuit is first realized on breadboard. Some of the values have changed due to the availability of the resistors and capacitors. For the operational amplifier op07 is used in all stages. While the circuitry is being built each stage is checked by a given signal which would not cause clipping.

Changes with Realization:

1) AD620: Inserting a 27KΩ resulted with a gain of 10, even though it was unexpected, in order to not using a larger resistance, the 27KΩ is accepted and the extra gain is corrected by changing the last Stage.

2) Band-pass stage: For the high-pass filter a resistance of 3.18MΩ is used. The resistance is in fact labelled as 3MΩ, however its exact value was 3.18MΩ. For the low-pass filter the 106 is realize as an 112KΩ and the 21.2KΩ is realize as a 21.8KΩ.

3) Output DC Level Shifting: The amount of shifting is done by supplying is done by supplying the inverting input and supplying the shifting dc voltage from the non-inverting input, thus the amount of dc input can be found by dividing the desired DC voltage level by non-inverting gain of the stage

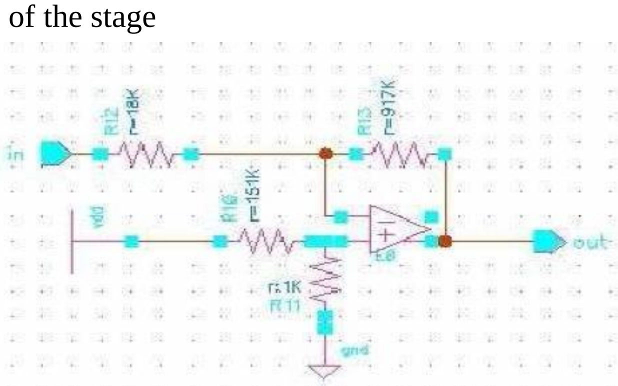

Figure - 25: Gain Stage and DC level Shift

However due to non-ideal of op07 the non-inverting input lowered the gain. With an interactive method the correct values are found. The input dc voltage is supplied from Vdc by a voltage divider.

To prevent the signal being lost in the 50Hz interface, a 50Hz notch filter is designed and implemented.

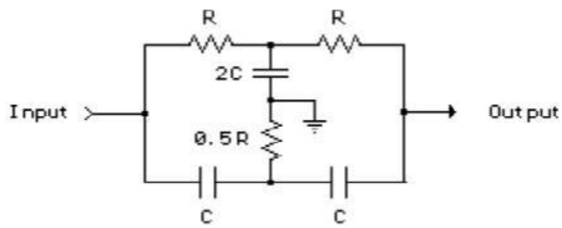

Figure - 26: Schematic of Notch Filter

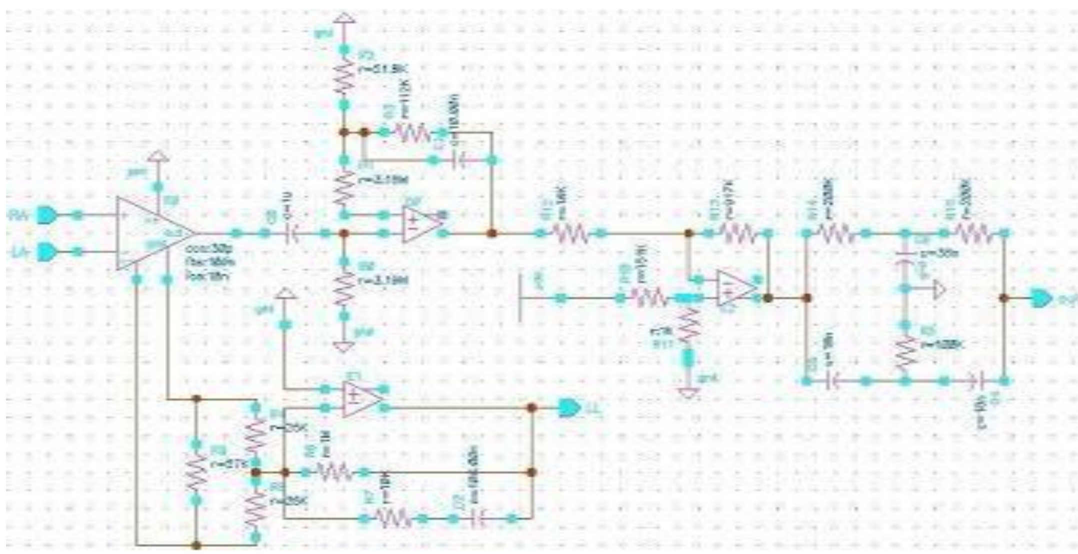

Figure - 27: Complete ECG Circuit

4.1.2 Pulse meter:

Heart rate measurement indicates the soundness of the human cardiovascular system. This project demonstrates a technique to measure the heart rate by sensing the change in blood volume in a finger artery while the heart is pumping the blood. It consists of an infrared LED that transmits an IR signal through the fingertip of the subject, a part of which is reflected by the blood cells. The reflected signal is detected by a photo diode sensor. The changing blood volume with heartbeat results in a train of pulses at the output of the photo diode, the magnitude of which is too small to be detected directly. Therefore, a two-stage high gain, active low pass filter is designed using two Operational Amplifiers (OpAmps) to filter and amplify the signal to appropriate voltage level so that the pulses can be counted

Heart rate is the number of heartbeats per unit of time and is usually expressed in beats per minute (bpm). In adults, a normal heart beats about 60 to 100 times a minute during resting condition. The resting heart rate is directly related to the health and fitness of a person and hence is important to know. You can measure heart rate at any spot on the body where you can feel a pulse with your fingers. The most common places are wrist and neck. You can count the number of pulses within a certain interval (say 15 sec) and easily determine the heart rate in bpm.

This project describes heart rate measurement system that uses optical sensors to measure the alteration in blood volume at fingertip with each heartbeat. The sensor unit consists of an infrared light-emitting-diode (IR LED) and a photodiode, placed side by side as shown below. The IR diode transmits an infrared light into the fingertip (placed over the sensor unit) and the photodiode senses the portion of the light that is reflected back. The intensity of reflected light depends upon the blood volume inside the fingertip. So, each heart beat slightly alters the amount of reflected infrared light that can be detected by the photodiode. With a proper signal conditioning, this little change in the amplitude of the reflected light can be converted into a pulse. The pulses can be later counted by the microcontroller to determine the heart rate.

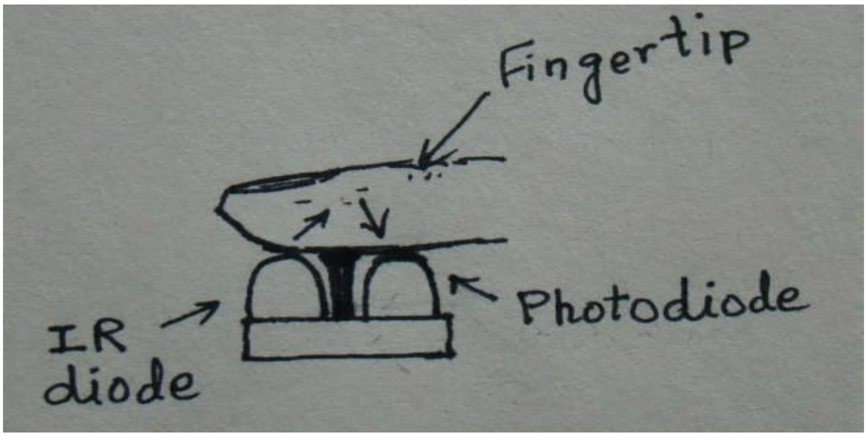

Figure – 28 Overview Picture of Pulse meter

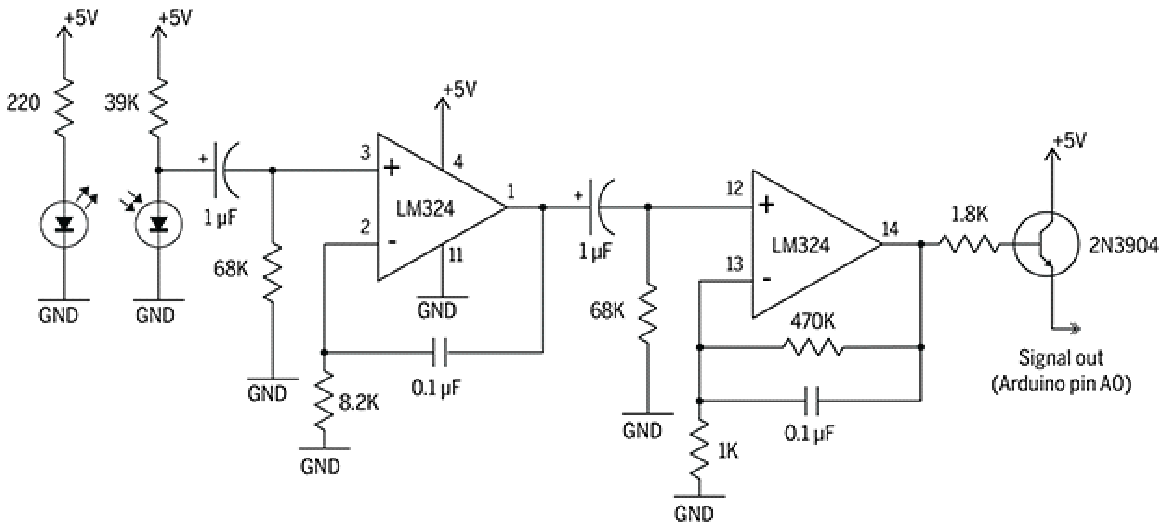

Figure- 29 Circuit Diagram of Pulse meter

- WORKING:

The sensor itself consists of an infrared emitter and detector mounted side-by-side and pressed closely against the skin. When the heart pumps, blood pressure rises sharply, and so does the amount of infrared light from the emitter that gets reflected back to the detector. The detector passes more current when it receives more light, which in turn causes a voltage drop to enter the amplifier circuitry. This design uses two consecutive operational amplifiers to establish a steady baseline for the signal, emphasize the peaks, and filter out noise. Both op-amps are contained in a single integrated circuit and hooking them up is really just a matter of interconnecting the pins correctly. The two op-amps output a clean but weak signal which is amplified by the transistor before output.

4.1.3 Temperature Sensor:

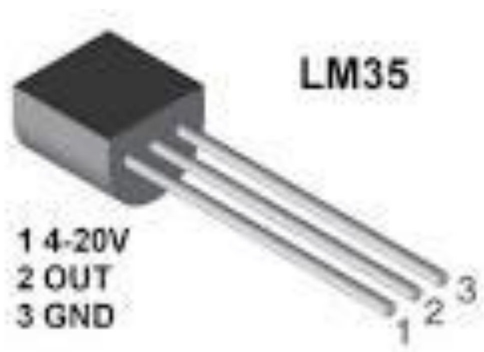

Figure – 30 Temperature Sensor LM35

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in˚ Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of±1⁄4˚C at room temperature and±3⁄4˚C over a full −55 to +150˚Ctemperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1˚C in still air. The LM35 is rated to operate over a −55˚ to +150˚C temperature range, while the LM35C is rated for a −40˚ to +110˚C range (−10˚with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while theLM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package.

- Features:
  - Calibrated directly in ˚ Celsius (Centigrade)
  - Linear + 10.0 mV/˚C scale factor
  - 0.5˚C accuracy guaranteeable (at +25˚C)
  - Rated for full −55˚ to +150˚C range
  - Suitable for remote applications
  - Low cost due to wafer-level trimming
  - Operates from 4 to 30 volts
  - Less than 60 µA current drain
  - Low self-heating, 0.08˚C in still air
  - Nonlinearity only±1
  - ⁄4˚C typical
  - Low impedance output, 0.1Ωfor 1 mA

4.3 Communication Modules:

4.3.1 Xbee Modules:

Figure – 31 Series 2 XBee Module

Series 2 modules allow creating complex mesh networks based on the XBee ZB ZigBee mesh firmware. These modules allow a very reliable and simple communication between microcontrollers, computers, systems, really anything with a serial port! Point to point and multi-point networks are supported.

These are essentially the same hardware as the older Series 2.5, but have updated firmware. They will work with Series 2.5 modules if you update the firmware through X-CTU.

- Features:
    - 3.3V @ 40mA
    - 250kbps Max data rate
    - 2mW output (+3dBm)
    - 400ft (120m) range
    - Built-in antenna
    - Fully FCC certified
    - 6 10-bit ADC input pins
    - 8 digital IO pins
    - 128-bit encryption
    - Local or over-air configuration
    - AT or API command set

4.3.2 XCTU Software:

X-CTU is a Windows-based application provided by Digi. This program was designed to interact with the firmware files found on Digi's RF products and to provide a simple-to-use graphical user interface to them.

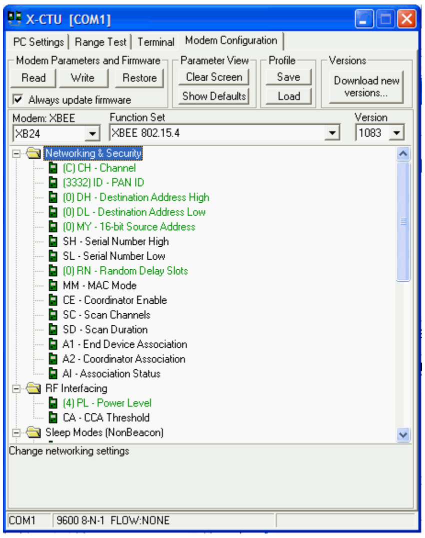

Figure – 32 XCTU Software

4.3.3 XBee Configuration:

For wireless transmission of data xbee is used.

One set of xbee is connected to the sensor nodes while other co-coordinator xbee is connected to the CPU or laptop. For establishment of path for the data transfer bot routers and coordinator should be configured such as id, channel, and baud rate.

Figure – 33 Pin Diagram of ZigBee Module

With using one zigbee module we can transmit 8 digital bits and two among signals.

Max.voltage that can be given to a zigbee module is 3.3 V, and have a ground.

This configures two XBee modules to connect to each other automatically upon power-up and continuously Trans-receive data. This setup can be used as a Wireless UART to connect any two PCs or two MCUs with serial ports.

Use two UartSBee or two Grove - XBee Carrier with two XBee ZB modules.

Connect these modules to PCs Serial port and make sure the drivers are installed.

Open X-CTU, select the USB-Serial Ports of the device as shown below:

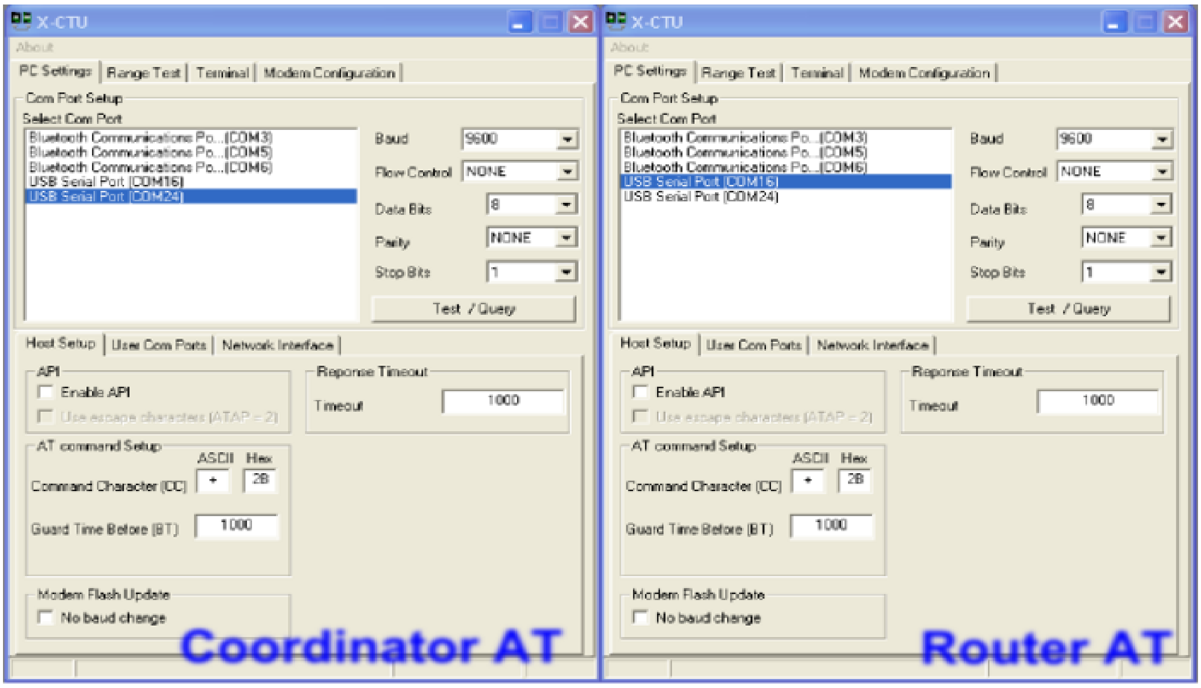

Figure- 34 Selecting the USB Serial Ports of Coordinator AT and Router AT

Program one module with **Coordinator AT.** Function-set firmware and another module with **Router AT** function-set firmware are done. Always use the latest version of firmware.

Now, Open Modem Configuration Tab and configure the destination addresses of both the modules as follows:

Set the destination address high of COORDINATOR to serial number high of ROUTER.

Set the destination address low of COORDINATOR to serial number low of ROUTER.

Set the destination address high of ROUTER to serial number high of COORDINATOR.

Set the destination address low of ROUTER to serial number low of COORDINATOR.

Write these parameters to the modules.

PAN ID was not modified during parameters configuration. It was left to 0, as there is only one Zigbee network in the vicinity. You might want to change these PAN IDs (of both the modules) to a 16bit number.

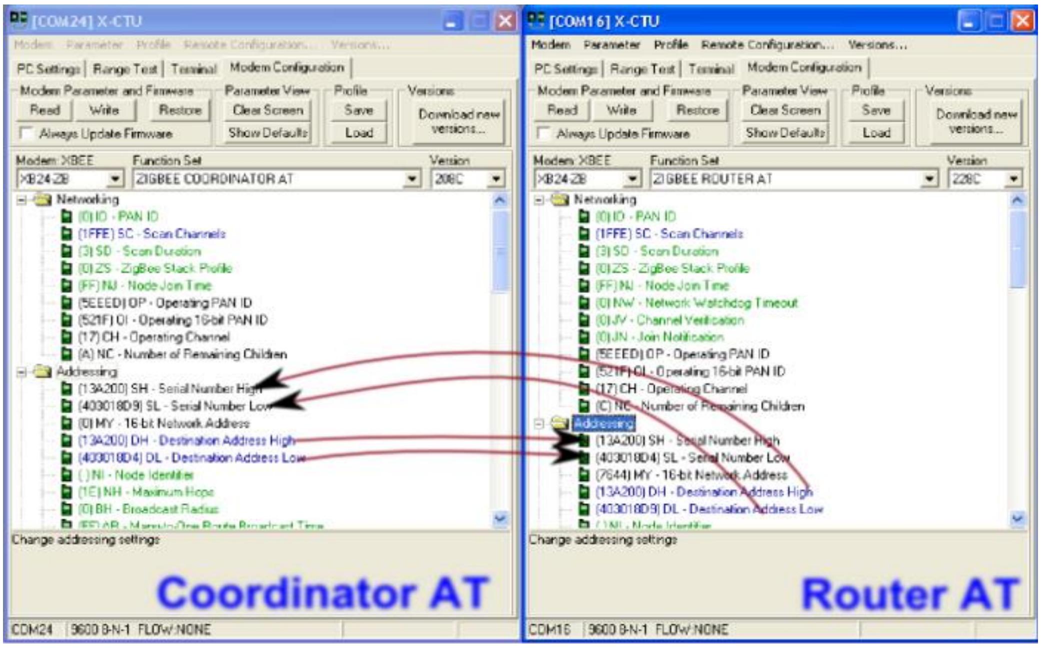

Figure – 35 Coordinator AT and Router AT

Open the Terminal Tabs for both the modules and click the Show Hex buttons.

Type a message in COORDINATOR terminal and it will be sent to ROUTER via wireless. Similarly, any text typed in ROUTER terminal is sent to COORDINATOR terminal.

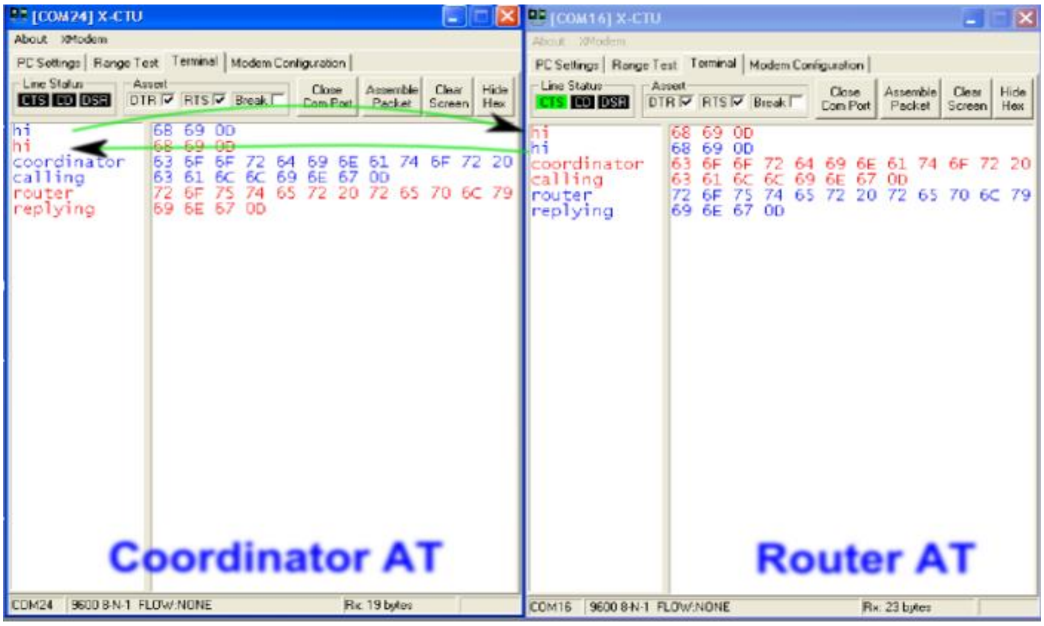

Figure – 36 Transfer of the message from Coordinator AT to Router AT and Vice versa

Security

ZigBee supports various levels of security that can be configured depending on the needs of the application. Security provisions include:

•128-bit AES encryption

•Two security keys that can be preconfigured or obtained during joining

•Support for a trust center

•Provisions to ensure message integrity, confidentiality, and authentication.

The first half of this chapter describes various security features defined in the ZigBee-PRO specification, while the last half illustrates how the XBee and XBee-PRO modules can be configured to support these features Security Modes The ZigBee standard supports three security modes – residential, standard, and high security. Residential security was first supported in the ZigBee 2006 standard. This level of security requires a network key be shared among devices. Standard security adds a number of optional security enhancements over residential security, including an APS layer link key. High security adds entity authentication, and a number of other features not widely supported. XBee ZB modules primarily support standard security, although end devices that support residential security can join and interoperate with standard security devices. The remainder of this chapter focuses on material that is relevant to standard security.

Troubleshooting

If everything works perfectly the first time around, that's GREAT! However, experience shows that it sometimes takes a few tries to get everything right. You've just set up a pretty complex system. Don't despair if your chat doesn't work right away. In almost every case, there's nothing wrong with any of your hardware or even with most of your setup. It takes only one wrong parameter to throw a wrench in the works. Learning how to find that wrench and fix it is an essential skill, so here are some tips on what to try if things don't work at first.

- Start with the simple stuff. Make sure your radios are seated properly in the adapter boards and that all the USB cables are plugged in the way they should be.
- Check that each radio is responding properly in the terminal window by trying to use +++ to put it into command mode. If you don't get an OK back, check your port selection, baud rate, and the other settings until you find the reason the radio is not communicating properly.
- If both radios are responding, use AT commands to check the settings. The most common problems are: not using the same PAN ID on both radios, not setting the destination address on each radio to the address number of the other radio, and not saving the settings properly.
- If the settings all seem to be correct, check to make sure that you have the coordinator firmware on one radio and the router firmware on the other radio. You can use the ATVR command to show which firmware is in use. If you see old version, go back to X-CTU and load the proper firmware.
- Sometimes the radios will be perfectly configured and connected but one of your radios will accidentally be talking to somebody else's. This often happens in classroom situations, where many people are using the same PAN ID in the same room. Try using different PAN IDs for each pair.
- Sometimes setting both radios back to factory defaults and reconfiguring them will flush out a bad setting that was left over from a previous setup, or an unrecognized typo. The ATRE command will wipe out your radio's custom configuration and leave the firmware set cleanly to factory defaults. Follow it with the ATWR command to write those defaults to the firmware, then go back to the configuration steps and try putting in your settings again.
- Don't forget that Digi's technical support is a great resource. While they needn't be your first step, if you're really stuck they can help you confirm that your radios are working properly at the hardware level. There are also a number of great online resources and forums you can read for ideas and where you can ask for more help.

4.3.4 HC-05:

HC serial Bluetooth products consist of Bluetooth serial interface module and Bluetooth adapter, such as: (a) Bluetooth serial interface module:

- Industrial level: HC-03, HC-04(HC-04-M, HC-04-S)
- Civil level: HC-05, HC-06(HC-06-M, HC-06-S) HC-05-D, HC-06-D (with baseboard, for test and evaluation)

(b) Bluetooth adapter: HC-M4 HC-M6

Bluetooth serial module is used for converting serial port to Bluetooth. These modules have two modes: master and slaver device. The device named after even number is defined to be master or slaver when out of factory and can’t be changed to the other mode. But for the device named after odd number, users can set the work mode (master or slaver) of the device by AT commands. HC-04 specifically includes: Master device: HC-04-M, M=Master Slave device: HC-04-S, S=slaver the default situation of HC-04 is slave mode.

The main function of Bluetooth serial module is replacing the serial port line, such as:

a) There are two MCUs want to communicate with each other. One connects to Bluetooth master device while the other one connects to slave device. Their connection can be built once the pair is made. This Bluetooth connection is equivalently liked to a serial port line connection including RXD, TXD signals. And they can use the Bluetooth serial module to communicate with each other.
b) When MCU has Bluetooth salve module, it can communicate with Bluetooth adapter of computers and smart phones. Then there is a virtual communicable serial port line between MCU and computer or smart phone.
c) The Bluetooth devices in the market mostly are salve devices, such as Bluetooth printer, Bluetooth GPS. So, we can use master module to make pair and communicate with them. Bluetooth Serial module’s operation doesn’t need drive, and can communicate with the other Bluetooth device that has the serial. But communication between two Bluetooth modules requires at least two conditions:
    a. The communication must be between master and slave.
    b. The password must be correct. However, the two conditions are not sufficient conditions. There are also some other conditions basing on different device model.

The Bluetooth serial module named even number is compatible with each other; the salve module is also compatible with each other. In other word, the function of HC-04 and HC-06, HC-03 and HC-05 are mutually compatible with each other. HC-04 and HC-06 are former version that user can’t reset the work mode (master or slave). And only a few AT commands and functions can be used, like reset the name of Bluetooth (only the slaver), reset the password, reset the baud rate and check the version number. The command set of HC-03 and HC-05 are more flexible than HC-04 and HC-06’s. Generally, the Bluetooth of HC-03/HC-05 is recommended for the user.

The PIN definitions of HC-03, HC-04, HC-05 and HC-06 are kind of different, but the package size is the same: 28mm * 15mm * 2.35mm. The following figure 1 is a picture of HC-06 and its main PINs. Figure 2 is a picture of HC-05 and its main PINs.

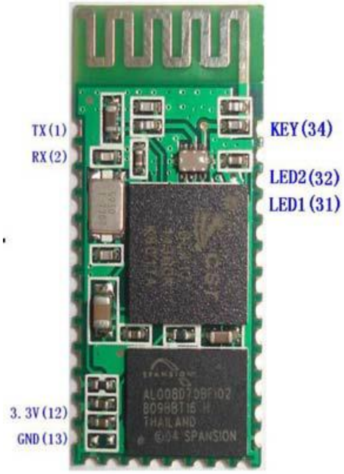

Figure -37 Pin Diagram of HC-05

a) PIN1 UART_TXD, Bluetooth serial signal sending PIN, can connect with MCU's RXD PIN
b) PIN2 UART_RXD, Bluetooth serial signal receiving PIN, can connect with the MCU's TXD PIN; there is no pull-up resistor in this PIN. But it needs to be added an eternal pull-up resistor.
c) PIN11 RESET, the reset PIN of module, inputting low level can reset the module, when the module is in using, this PIN can connect to air.
d) PIN12 VCC, voltage supply for logic, the standard voltage is 3.3V, and can work at 3.0-4.2V
e) PIN13 GND
f) PIN31 LED1, indicator of work mode. Have 3 modes: When the module is supplied power and PIN34 is input high level, PIN31 output 1Hz square wave to make the LED flicker slowly.
g) PIN32 Output terminal. Before paired, it output low level. Once the pair is finished, it output high level.
h) PIN34 Mode switch input. If it is input low level, the module is at paired or communication mode. If it's input high level, the module will enter to AT mode.

HC-05 has many functions and covers all functions of HC-06. Besides this, HC-05 leaves lots of space for user. So HC-05 is better than HC-06 and recommended. HC-03 is similar with HC-05. The function set also suits HC-03.

The pins TX and RX pin of the HC 05 form the path for data transmission and reception. These TX pin of HC05 must be connected to the RX pin of microcontroller and vice versa. Whereas the key pin of the module is used to set the password for pairing the module with our devices.

Figure below shows how to interface the Bluetooth with microcontroller. Bluetooth technology handles the wireless part of the communication channel; it transmits and receives data wirelessly between these devices. It delivers the received data and receives the data to be transmitted to and from a host system through a host controller interface (HCI). The most popular host controller interface today is either a UART or a USB. Here, I will only focus on the UART interface; it can be easily show how a Bluetooth module can be integrated on to a host system through a UART connection.

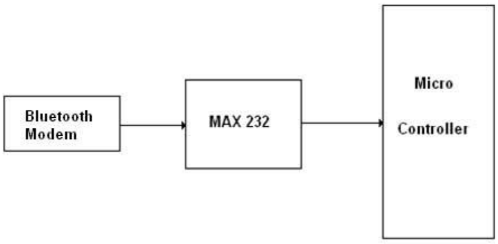

Figure – 38 Interface of Bluetooth module and Micro Controller

APPLICATION:

In BAN number of intelligent physiological sensors can be integrated into a wearable wireless body area network, which can be used for computer-assisted rehabilitation or early detection of medical conditions. This area relies on the feasibility of implanting very small biosensors inside the human body that are comfortable and that don't impair normal activities. The implanted sensors in the human body will collect various physiological changes in order to monitor the patient's health status no matter their location. The information will be transmitted wirelessly to an external processing unit. This device will instantly transmit all information in real time to the doctors throughout the world. Forthis purpose in our project we are using Bluetooth for wireless transmission.

The information that is obtained through Xbee module is to be transmitted via Bluetooth module like HC 05.So we need microcontroller for interfacing Xbee module as well as HC05 module. The information collected in Xbee is taken into microcontroller and transmitted through Bluetooth via proper interface.

SCHEMATIC DESIGN:

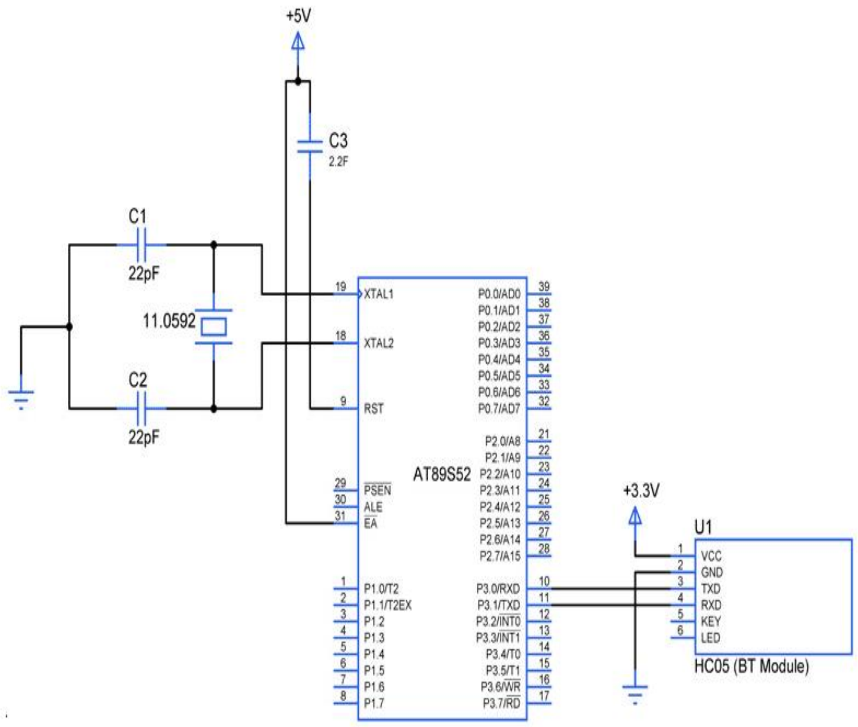

Figure – 39 Schematic Design of connection of HC-05 and AT89S52

STEPS TO PROGRAM:

- This uses Serial Communication or UART protocol in the microcontroller.
- Initialize the Serial communication in 8051 using Timer and serial registers.
- Generate the required baud rate for the communication to take place. The default baud rate of the HC05 is 9600.
- Initialize serial interrupts in case you need to control the tasks performed by your microcontroller or receive data when requested.

	UART CONNETOR UART DB-9 Connector	MICROCONTROLLER LINES[MSP430]
UART0(P1)	TXD-0	P1_1
ISP PGM	RXD-0	P1_2

Table -5 Interface of Bluetooth module and MSP430

4.4 Processing Unit:

4.4.1 MSP430:

The MSP430 is a mixed-signal microcontroller family from Texas Instruments. Built around a 16-bit CPU, the MSP430 is designed for low cost and, specifically, low power consumption embedded applications.

The MSP430 can be used for low powered embedded devices. The current drawn in idle mode can be less than 1 µA. The top CPU speed is 25 MHz. It can be throttled back for lower power consumption. The MSP430 also uses six different low-power modes, which can disable unneeded clocks and CPU. Additionally, the MSP430 is capable of wake-up times below 1 microsecond, allowing the microcontroller to stay in sleep mode longer, minimizing its average current consumption. The device comes in a variety of configurations featuring the usual peripherals: internal oscillator, timer including PWM, watchdog, USART, SPI, 10/12/14/16/24-bit ADCs. Some less usual peripheral options include comparators, on-chip op-amps for signal conditioning, 12-bit DAC, LCD driver, hardware multiplier, USB, and DMA for ADC results. There are, however, limitations that preclude its use in more complex embedded systems. The MSP430 does not have an external memory bus, so it is limited to on-chip memory (up to 512 KB flash memory and 66 KB RAM) which may be too small for applications that require large buffers or data tables.

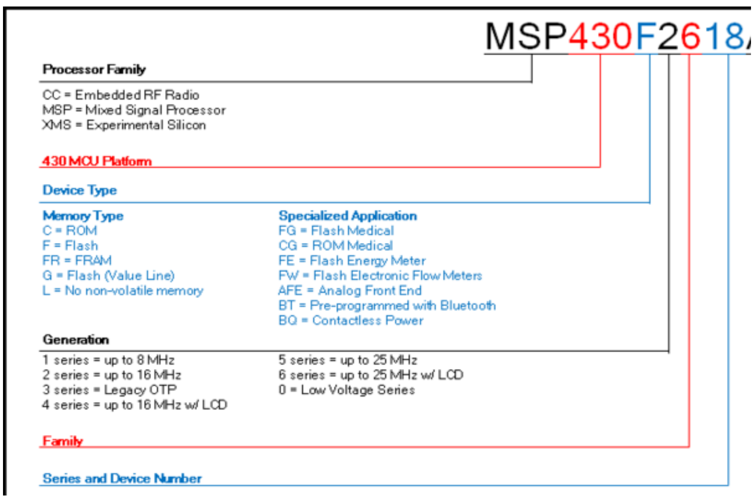

Figure – 40 MSP430 Nomenclatures

<u>MSP430G2XX SERIES:</u>

The MSP430G2xx Value Series features flash-based Ultra-Low Power MCUs up to 16 MIPS with 1.8–3.6 V operation. It includes the Very-Low power Oscillator (VLO), internal pull-up/pull-down resistors, and low-pin count options, at lower prices than the MSP430F2xx series.

- Ultra-Low Power, as low as 2.2 V
- µA RAM retention
- 0.4 µA Standby mode (VLO)
- 0.7 µA real-time clock mode
- Ultra-Fast Wake-Up From Standby Mode in <1 µs

➢ Device parameters

- Flash options: 0.5–56 KB
- RAM options: 128 B-4 KB
- GPIO options: 10, 16, 24, 32 pins
- ADC options: Slope, 10-bit SAR
- Other integrated peripherals: Capacitive Touch I/O, up to 3 16-bit timers, watchdog timer, brown-out reset, USI module (I^2C, SPI), USCI module, Comparator A+, Temp sensor
- High performance for cost-sensitive applications The MSP430G2xx and MSP430i2x 16-bit microcontrollers feature Flash-based ultra-low-power MCUs up to 16 MIPS with 1.8V – 3.6V operation. Includes the very-low power oscillator (VLO), internal pull-up/pull-down resistors and low-pin count options.

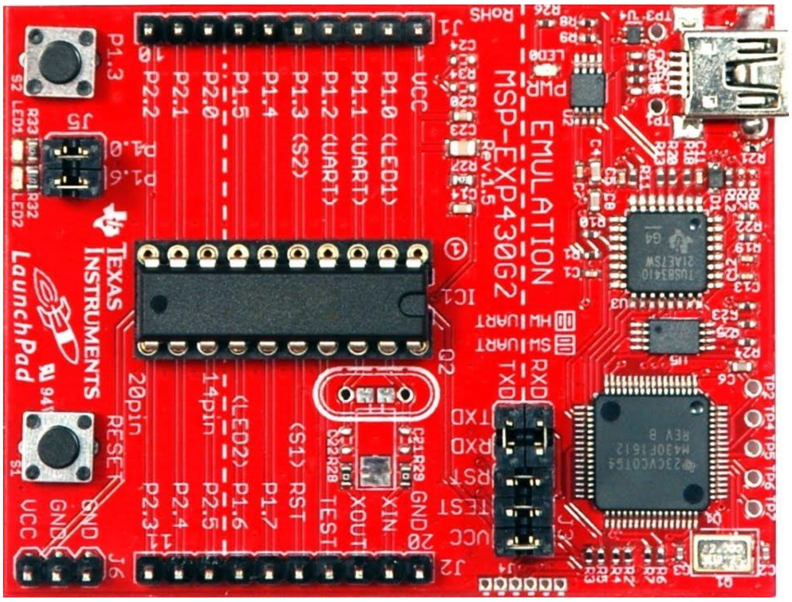

Figure – 41 MSP430

PERIPHERALS:

a) Analog-to-Digital Converter

The MSP430 line offers two types of Analog-to-Digital Conversion (ADC) 10- and 12-bit Successive Approximation converters, as well as a 16-bit Sigma-Delta converter.

b) Op Amps
   Feature single supply, low current operation with rail-to-rail outputs and programmable settling times. Software selectable configuration options: unity gain mode, comparator mode, inverting PGA, non-inverting PGA, differential and instrumentation amplifier.
c) Basic timer (BT)
   The BT has two independent 8-bit timers that can be cascaded to form a 16-bit timer/counter. Both timers can be read and written by software. The BT is extended to provide an integrated RTC. An internal calendar compensates for months with less than 31 days and includes leap-year correction.
d) Real-Time Clock
   RTC_A/B is 32-bit hardware counter modules that provide clock counters with a calendar, a flexible programmable alarm, and calibration. The RTC_B includes a switchable battery backup system that provides the ability for the RTC to operate when the primary supply fails.
e) 16-bit timers
   Timer_A, Timer_B and Timer_D are asynchronous 16-bit timers/counters with up to seven capture/compare registers and various operating modes.
f) Capacitive Touch Sense I/Os
   The integrated capacitive touch sense I/O module offers several benefits to touch button and touch slider applications.
g) General Purpose I/Os
   MSP430 devices have up to 12 digital I/O ports implemented. Each port has eight I/O pins. Every I/O pin can be configured as either input or output, and can be individually read or written to.
h) USART (UART, SPI, I²C)
   The universal synchronous/asynchronous receive/transmit (USART) peripheral interface supports asynchronous RS-232 and synchronous SPI communication with one hardware module.
i) USB
   The USB module is fully compliant with the USB 2.0 specification and supports control, interrupt and bulk transfers at a data rate of 12 Mbps (full speed). The module supports USB suspend resume and remote wake-up operations and can be configured for up to eight input and eight output endpoints.
j) Direct memory access
   The MSP MCUs also feature a direct memory access controller, enabling memory transfer with no CPU intervention. This means higher throughput of peripheral modules and lower system power.

GENERAL-PURPOSE I/O PORTS:

As is standard on microcontrollers, most pins connect to a more specialized peripheral, but if that peripheral is not needed, the pin may be used for general-purpose I/O. The pins are divided into 8-bit groups called "ports", each of which is controlled by a number of 8-bit registers. In some cases, the ports are arranged in pairs which can be accessed as 16-bit registers. The MSP430 family defines 11 I/O ports, P0 through P10, although no chip implements more than 10 of them. P0 is only implemented on the '3xx family. P7 through P10 are only implemented on the largest members (and highest pin count versions) of the '4xx and '2xx families. The newest '5xx and '6xx families has P1 through P11, and the control registers are reassigned to provide more port pairs. Each port is controlled by the following registers. Ports which do not implement particular features (such as interrupt on state change) do not implement the corresponding registers.

4.5 Data Acquisition:

Data Acquisition is done by using pc using RF modules and to mobile phone using bluetooth modules

4.5.1 Personal Computer:

The RF module is connected to one of the serial COM ports of the pc. From the serial COM port, the data is collected and plotted using Matlab.

4.5.2 Mobile Phone:

The data is received using the bluetooth module.The serial data is shown using a bluetooth serial application available on the internet.

4.6 Data Storage:

The Data acquired from the data acquisition devices is outputted into an excel file or a text file so that a patient record is maintained.

4.6.1 Memory:

One way of storing the data is to store it an external memory

4.6.2 Google Drive:

Google Drive is a file storage and synchronization service created by Google. It allows users to cloud, share. Google Drive can be synched directly from a computer so that the stored file can be accessed from a remote location. The file accessed from a remote location can be plotted again using Matlab to show live patient record.

5. NETSIM

5.1 What Is NetSim?

NetSim is popular network simulation software for protocol modelling and simulation, allowing you to analyse computer networks with unmatched depth, power and flexibility. It is a network simulation tool that allows you to create network scenarios, model traffic, and study performance metrics. Network simulation is a technique where a program models the behaviour of a network either by calculating the interaction between the different network entities. It is also used for network design, planning, network R & D and defence applications. Various technologies such as Wireless Sensor Networks, Wireless LAN, Wi Max, TCP, IP, etc. are covered in NetSim. It is being used by the world's most prestigious institutions for network lab research and experimentation. It is available as Standard or Academic version and is built on a common design framework of high level architecture and code.

5.2 Why Need NetSim?

Communication Networks have become too complex for traditional analytical methods to provide an accurate understanding of system behaviour and possible problems and solutions. NetSim provides network performance metrics at various abstraction levels such as Network, sub-network, link, node along with a detailed packet trace and event trace. Using NetSim modelling& simulation services are provided in a variety of networking technologies and protocols including MANET, Wi-Fi, WI-Max, IP, MPLS, WSN, QOS and VoIP etc. This can help avoid the time consuming process of programming, customization and configuration commercial simulators to meet customer specific needs.

5.3 Use of NetSim

Over 300 customers across 15 countries use NetSim. Several defence agencies and DRDO labs use NetSim for modelling the unique requirements of defence R & D crucial to Network-centric warfare. NetSim comes with an in-built development environment, which serves as the interface between User's code and NetSim's protocol libraries and simulation kernel. Protocol libraries are available as open C code for user modification. De-bugging custom code during simulation is an advanced feature: i.e. a simulation can be started and then at user determined breakpoints in the code, users can perform single-step, step-in, step over etc. This can be carried out at various levels (depending on where the user code links) including at a per-packet interval.

5.4 INTERFACE AND GUI OF NETSIM

5.4.1 Modelling and Simulation - Create scenario

- Create network scenarios using NetSim's GUI or using XML config files
- Click and Drop devices, links, application etc. into the environment using NetSim's GUI
- Set properties of the channels. Layer-wise parameters can be edited
- Model large and complicated networks using the XML config file which comes with automatic verification

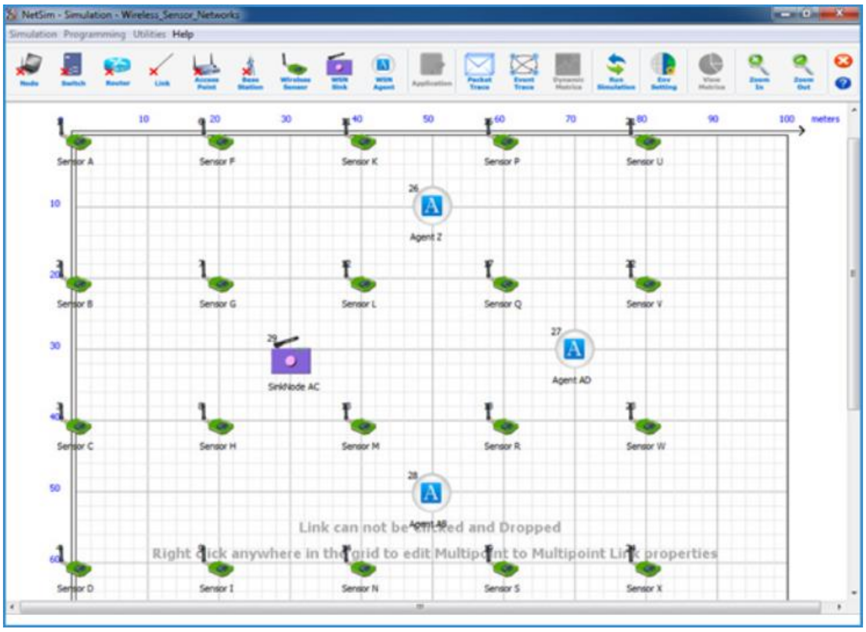

Figure – 42 Typical GUI of NetSim

5.4.2 Running the simulation

- Run the discrete event simulation (DES) through the GUI or via Command line interface (CLI)
- Log packet traces that reports parameters such as arrival time, queuing time, payload, overhead, error etc. for every packet as it flows through the network
- Record event traces which logs every single event in the protocol finite state machine (FSM) transitions along with associated information like time-stamp, Event ID, Event Type etc.

5.4.3 Visualizing the simulation using the packet animator

- Animate packet flow over wired and wireless links
- Colour variation for control packets, data packets and error packets
- Control animation with play, pause and simulation timeline

5.4.4 Analysing the results

- Examine output performance metrics at multiple levels - network, sub network, link, queue, application etc.
- Study a variety of metrics such as throughput, delay, loss, packet error, link utilization.
- Interpret metrics using in-built plots and graphs
- Export packet and event trace files easily to tools like Excel, Notepad etc. for post-processing and statistical analysis

5.4.5 Developing your own protocol / algorithm

- Extend existing algorithms by modifying NetSim's source C code
- Create custom protocols using NetSim's simulation API's
- Interface with other software products
- Debug your code (step-in, step-out, step-over, continue) and watch your variables in sync with simulation

5.5 LEACH PROTOCOL

In this project we followed the leach protocol. We modified the existing code to create a new code which would be helpful in making it possible to achieve this. Basically the leach protocol follows certain predefined criteria which are to be satisfied.

5.5.1 Working of leach protocol

- The LEACH protocol aims to minimize energy consumption in WSNs through a cluster-based operation.
- The goal of LEACH is to dynamically select sensor nodes as cluster heads and form clusters in the network.
- The communications inside the clusters are directed to the cluster head, which performs aggregation.
- Cluster heads then directly communicate with the sink to relay the collected information from each cluster.
- LEACH also changes the cluster head role dynamically such that the high-energy consumption in communicating with the sink is spread to all sensor nodes in the network.
- Firstly, a sensor node chooses a random number between 0 and 1.
- If this random number is less than a threshold T (n), the sensor node becomes a cluster head.

- T (n) is calculated as

$$T(n) = \frac{P}{1 - P \times \left(r \bmod \frac{1}{P} \right)}$$

$$T(n) = 0$$

Where P is the desired percentage to become a cluster head, r is the current round, and G is the set of nodes that have not been selected as a cluster head in the last 1/P rounds.

But the main problem in using the leach protocol that it assigns the cluster head of the particular cluster at that particular time based on the probability obtained from the above equation. In our current scenario we do not want the cluster head to be changing as the tracking of the received packets and the transmitted packets will be difficult. For accurately tracking the packets and obtaining the ratio of the packets dropped to packets received which gives us the packet error ratio, we need the cluster head to be the same node. So some changes are done in the code so that the cluster head remains same for respective cluster for the whole simulation time.

5.6 NETSIM PARAMETERS

5.6.1 Channel Parameters

The channel follows multi point to multi point communication. This involves more than two members of the same group to that wish to exchange the information; in contrast point to pint applications involves only two users. As in the wireless sensor network the sensor nodes are situated far away from each other they communicate with one another which are close by. So it follows multipoint to multipoint communication.

The channel link medium is wireless. Wireless communication is the transfer of information form one point to another point or other points without any presence of wired electrical cable.

The channel link mode is half duplex. System composed of two connected devices which can communicate with one another in both directions is known as duplex communication. Half duplex system provides for the communication is both the directions but only one direction at a time (not simultaneously). Typically once a device starts receiving a signal it must wait for the transmitter to stop transmitting, before replying.

The WBAN uplink speed is the rate at which he files/ data can be uploaded from a local machine to a remote source. It may be in kbps or in mbps. But here in our case it is in mbps. The value is 0.25 mbps i.e. 250kbps

The WBAN downlink speed is the rate at which he files/ data can be downloaded from a remote source to a local machine. It may be in kbps or in mbps. But here in our case it is in mbps. The value is 0.25 mbps i.e. 250kbps

5.6.2 ISM Band

There are 16 channel numbers in NetSim. These are sixteen non overlapping channel numbers from 11 to 26 for 2450 MHz band which is ISM band in 802.15.4 standard. For channel number 11 the Frequency is 2405 MHz Each channel is separated by 5 MHz so as to avoid the interference between the channels. The industrial, scientific and medical (ISM) radio bands are radio bands reserved internationally for the use of radio frequency energy for industrial, scientific and medical purposes other than telecommunications. Individual countries' use of the bands designated in these sections may differ due to variations in national radio regulations. Because communication devices using the ISM bands must tolerate any interference from ISM equipment, unlicensed operations are typically permitted to use these bands, since unlicensed operation typically needs to be tolerant of interference from other devices anyway. The ISM bands share allocations with unlicensed and licensed operations; however, due to the high likelihood of harmful interference, licensed use of the bands is typically low.IEEE 802.15.4, Zigbee and other personal area networks may use the 915 MHz and 2450 MHz ISM bands because of frequency sharing between different allocations.

5.6.3 Placing of the sensor nodes

Basically in NetSim simulation virtual environment is created in which these sensor nodes are placed according to the settings given by the user. The nodes can be placed in uniform arrangement or in the random arrangement where the user decides the position of the nodes by ‘click and drop’ method. Usually uniform arrangement of the nodes is not preferred as in the real time simulation they are not placed equidistantly. The nodes are placed together to form different clusters. Each cluster has some nodes which send the data to the cluster head which is decided by the user.

5.6.4 Other blocks

Basically the data is acquiesced rom the active agent and the data is sent to the cluster head through some packets. This whole data is again sent to the sink. The sink can be considered as the receiver which analyses the data. While we are accessing wireless sensor network mode in netsim nodes ,switches routers links access points and base stations are not available or cannot be accessed because in wireless sensors networks nodes transmit wirelessly directly to one another or cluster head . So there is no need of these other blocks.

5.7 CHANNEL CHARACTERISTICS

In environmental properties we can set the channel characteristics. These are important because the packets dropped and the error ratio depends upon the loss of the medium which is selected to transmit the data. So the simulations are done for different types of channel characteristics and the optimum and reliable channel is selected. In this section the channel characteristics are discussed and their effects on the data transmission and packet drop rate are also discussed.

5.7.1 No path loss

In this scenario the channel through which the data is transmitted is considered to be ideal channel which is considered as lossless channel. So basically no path loss will be introduced and that means the data which is transmitted is assumed to be received at the cluster head. When the signal travels through the air the signal is not degraded due to the environmental conditions. This is called a no path loss channel. No loss is presumed. But the data is lost due to the overlap of the data coming from the different nodes at the same time to the same cluster head. So collision occurs at the cluster head.

5.7.2 Line of Sight LOS

In this scenario the sensor node which is transmitting the data and the cluster head which is receiving the data are assumed to be in line of sight. Sender and transmitter have a clear and unobstructable path between them. In this channel has a path loss exponent which determines the path loss in a particular manner. Path loss exponent indicates the rate at which the path loss increases with distances. The path loss exponent is associated with the number of hops the data makes before it reaches the destination i.e. the cluster head. The more the path loss exponent the more the hops it makes the greater the probability of data loss or packet loss increases. The value depends upon on the specific propagation environment. The path loss exponent value can be set between 2 to 5. This is designated by the symbol 'n'.

5.7.3 Fading channel

In this scenario the transmitting channel is assumed to be a fading channel. Signal can take many different paths between sender and receiver due to reflection, scattering and diffraction which leads to the fading of signal. The fading parameter decides the fading of the signal for the maximum time. By changing the fading parameter, the data i.e. the packet loss or he packet drop rate are not changed. Fading only has path loss exponents which are not mentioned here. The parameter m, the 'fading figure', is defined as the ratio of moments. The value of m=1 corresponds to Rayleigh fading and m>1 indicates better fading condition than Rayleigh and usually indicates Line of Sight available between station and access point. The fading figure should be set any value between 0.5 to 5. In our scenarios we have taken the fading figure as 0.5,1 and 5.

5.7.4 Shadowing

In this scenario the transmitting channel is assumed to have shadowing effect too. This is most realistic channel and the channel in which the interference will be maximum. The buildings and trees of environment create some shadowing effect which in turn increases the packet loss probability. Received signal is shadowed by obstruction such as hills and buildings. Fading and shadowing also has path loss exponent and fading figure. Standard Deviation (dB) is used here. The standard deviation which is used here is Log Normal Shadowing -Standard deviation. Shadowing is caused mainly by terrain features of the radio propagation environment .The mathematical model for shadowing is long normal distribution with standard deviation of 5 to 12 dB The shadowing parameter i.e. standard deviation can be set any value between 5 to 12 dB

5.7.5 Effects of the channel characteristics

In no path loss the collision of the packets is observed to be more. In the LOS (line of sight) channel by changing the path loss exponent the values were changing if path loss exponent is set to 2 then the error is less and for value set to 5, data error is more. By changing the fading figure the data received has no change i.e. the packet error rate and packet drop rate remains the same. By changing shadowing parameters, it is observed that the collision of data is more when the shadowing is more.

5.8 CUSTOM PROTOCOL IMPLEMENTATION

- NetSim's project framework UI serves as simplifying layer over the underlying protocol source code. It raises the level of abstraction enabling users to focus on the protocol source code and eliminating the need to manage the rest of the code infrastructure. Code linking, interfaces, configuration files, statistics etc. will be automatically handled by NetSim per predefined settings.
- NetSim development kit provides sophisticated users with source C code, libraries, resource files, config xml files and features MSVC based debugging and documentation. With this users can create custom protocols, extend existing algorithms, interface with other software products, modify output statistics, log debug information and much more.

5.9 PACKET TRACE AND EVENT TRACE

- Learn protocol internals: Window size variation in TCP, Route Table Formation in OSPF, Medium Access in Wi-Fi, etc., are examples of protocol functionalities that can be easily understood from the trace.

- Log chosen parameters: NetSim's trace option features a checkbox based filter where users can select only those packet / event level parameters / information which they want to log.

- Import easily into spread sheets: Tab ordering enables the trace data to be imported into spread sheets programs such as excel with ease for statistical analysis. Further, the events / packets are printed per line and have a unique ID and time stamp. This enables users to quickly write custom scripts to parse, inspect, plot and analyse protocol parameters

- Debug custom protocol implementations: Users need diagnostics that can catch logical errors without having to set a lot of breakpoints or restarting the program often. A host of information is available in the trace files

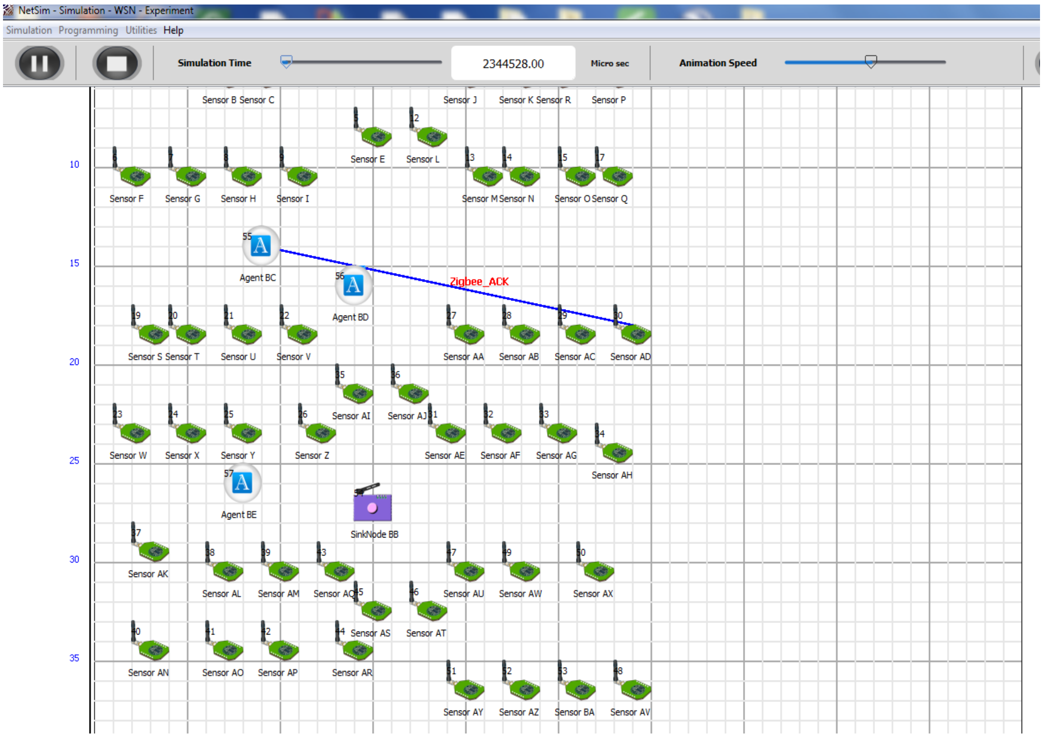

Figure – 43 Packet trace Animation

5.10 MOBILITY MODEL

- No mobility means the mobile node is static at one location i.e. it is not moving in any direction.
- Random Walk Model is that for a random walk on a regular line, where at the each step the walk jumps to another site according to some probability distribution.
- Random Way Point (RWP) model is commonly used synthetic model for the mobility in ad hoc network. It is an elementary model which describes the movement pattern of the independent nodes by some simple terms.

- Group Mobility Model is commonly use synthetic model for the mobility in ad hoc network. It is an elementary model which describes the behaviour of the mobile nodes as they move together as on unit. Basically this type of mobility pattern is used in the cluster mobility. The cluster should move together as one unit so the group mobility is given as some predefined value. The nodes move with some particular velocity. Velocity of a particular node can be given as the rate of change of its position with respect to the background. Its range may vary from 0-100m/sec. basically in our scenarios we have selected the group mobility of the cluster to be 10, 20...i.e. multiples of 10. (m/s)

5.11 LIST OF TERMS USED IN NETSIM

- **Simulation Time (ms)**
  Amount of time for which the simulation is run.

- **Packets Transmitted**
  It is the total number of packets transmitted in the entire network. It is equal to sum of the 'Packets_Transmitted' in all links.

- **Packets Errored**
  It is the total number of packets error in the entire network. It is equal to sum of the 'Error_Packets' in all links.

- **Packets Collided**
  It is the total number of packets collided in the entire network. It is equal to sum of the 'Collided Packets' in all links. Usually packet collisions occur in the wireless links.
  .
- **Bytes Transmitted**
  It is the total number of bytes transmitted in the entire network. It is equal to the sum of the 'Payload Transmitted' (actual payload generated in the application layer) and 'Overhead Transmitted' (layer wise overhead and protocol control packets) in all links.

- **Payload Transmitted**
  It is the total payload transmitted in the entire network in bytes. It is equal to sum of the 'Payload Transmitted' in all the links.

- **Overhead Transmitted**
  It is the total overhead transmitted in the entire network in bytes. It is equal to sum of the 'Overhead Transmitted' in all the links

5.12 GRAPH PLOT OF NETSIM SIMULATIONS

In this section the graph plots of the netsim simulation are shown. The plot is given between the number of nodes and packet error loss percentage. The packet error loss can be given as the ratio of the packets dropped during the transmission to the total packets transmitted by a particular sensor node to the cluster head. So the average of al packet errors is taken to obtain a mean packet error loss for that particular cluster. So the plot can be given as shown in the below figure. The Power level of one cluster is made higher than the other. The clusters which are stationary are maintained at a lower power level and the mobile clusters are at higher power level. Fig-44 gives us the packet error rate for different values of power levels.

Clusters Head	**Packets Transmitted.**	**Packets Received.**	**Percentage Packet Drop.**	**Packet Drop. %**
5 (1-9)	442	400	42	9.5
12 (10-18)	454	428	26	5.72
21(19-27)	452	401	52	11.51
30(28-36)	456	392	65	14.2
40(37-45)	424	379	46	10.84
50(46-54)	481	403	74	15.38

Table -6 Network Analysis of Cluster Configuration with power 10 dbm and 2 clusters mobile

Clusters Head	**Packets Transmitted.**	**Packets Received.**	**Percentage Packet Drop.**	**Packet Drop. %**
5 (1-9)	442	400	43	9.5
12 (10-18)	450	423	27	5.92
21(19-27)	460	398	62	13.51
30(28-36)	456	392	63	13.84
40(37-45)	420	379	53	12.61
50(46-54)	480	403	76	15.83

Table -7 Network Analysis of Cluster Configuration with power 20 dbm and 1 clusters mobile

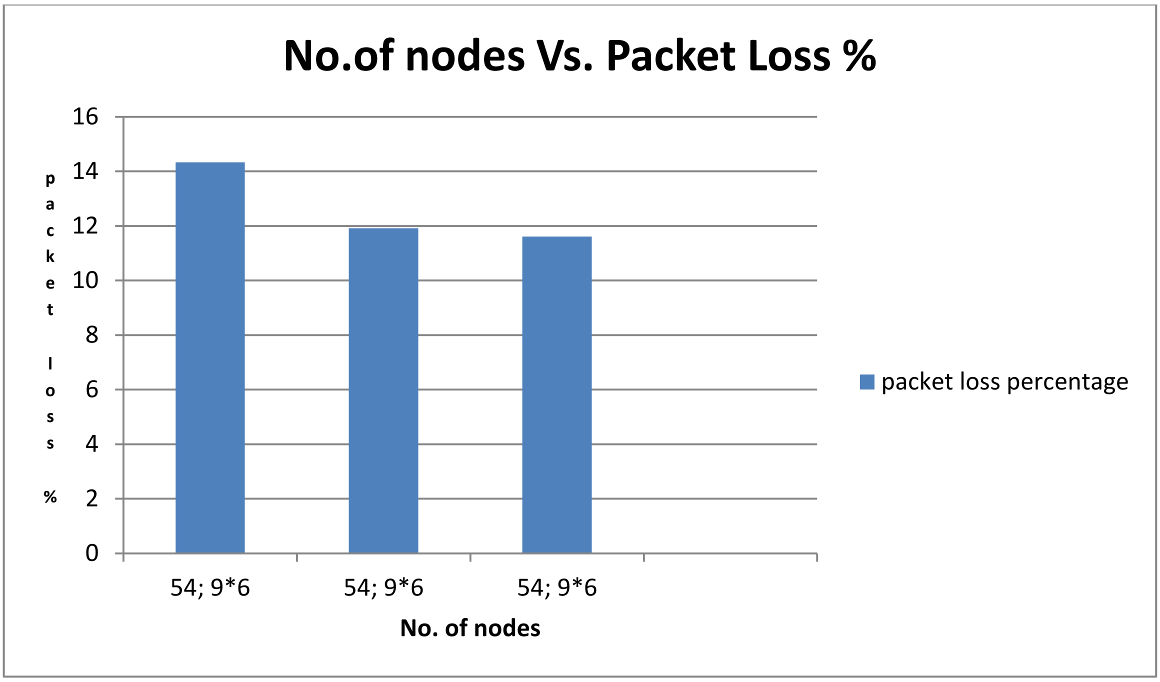

Figure – 44 Graph plot of number of nodes and packet loss

Here the table represents the Network Analysis of Cluster Configuration with power 20 dbm and 1 cluster mobile and other one represents the Network Analysis of Cluster Configuration with power 10 dbm and 2 clusters mobile. Below the graph represents the channel type and its interference in which the packet collision and packet drop are considered as metrics

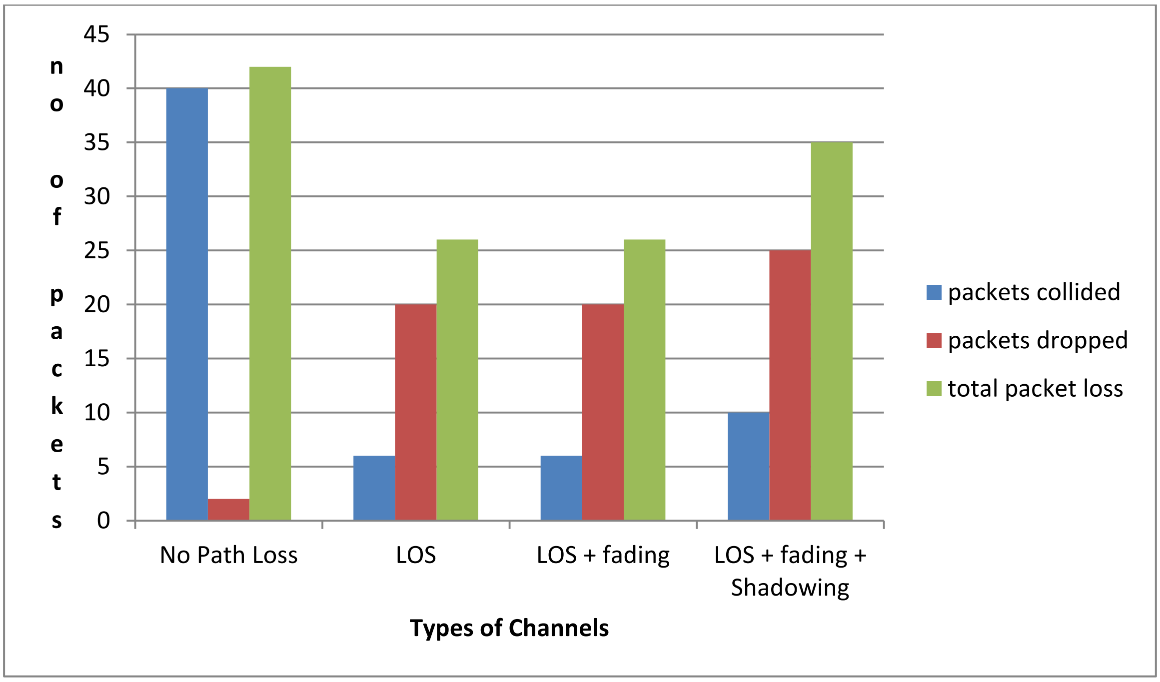

Figure – 45 Graph plot of number of packets and types of channels

6. EXPERIMENT RESULT

Connections are made as mentioned in working model. Sensor nodes such as pulse oximeter and ECG are mounted on the patient body. The data sent over air interface can be seen in Matlab in graphical view.

- **ECG**

The electrodes are connected to patient body one foot while other two on either hands of patient. Dc voltage of +5 V and -5 V are also give at the input of circuit. Output can be connected to both CRO for testing the signal and to Xbee for to data transmission. This o/p is an analog signal so it is connected to analog pin of Xbee. At receiver side we can see the same signal on graph.

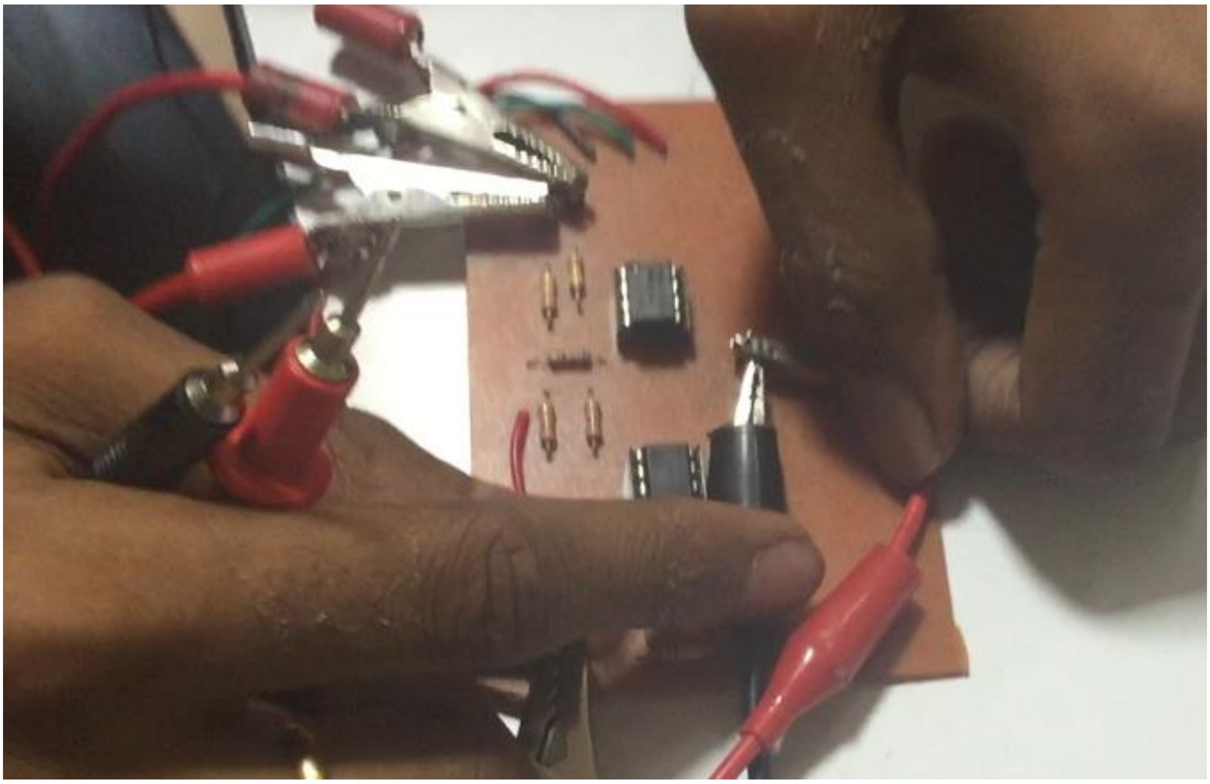

Figure – 46 Connections of ECG module

Output waveform seen on CRO-

As we all know the ECG signal .it will have spike for certain amount of time and the rest of time it will have a DC sweep. When heart beats at a higher rate tae gap between spikes will become less and amplitude will vary abruptly. If the heart beat stops then there will be no signal or dc sweep.

This signal is sent to Xbee and it sampled there at very high rate for digital signal

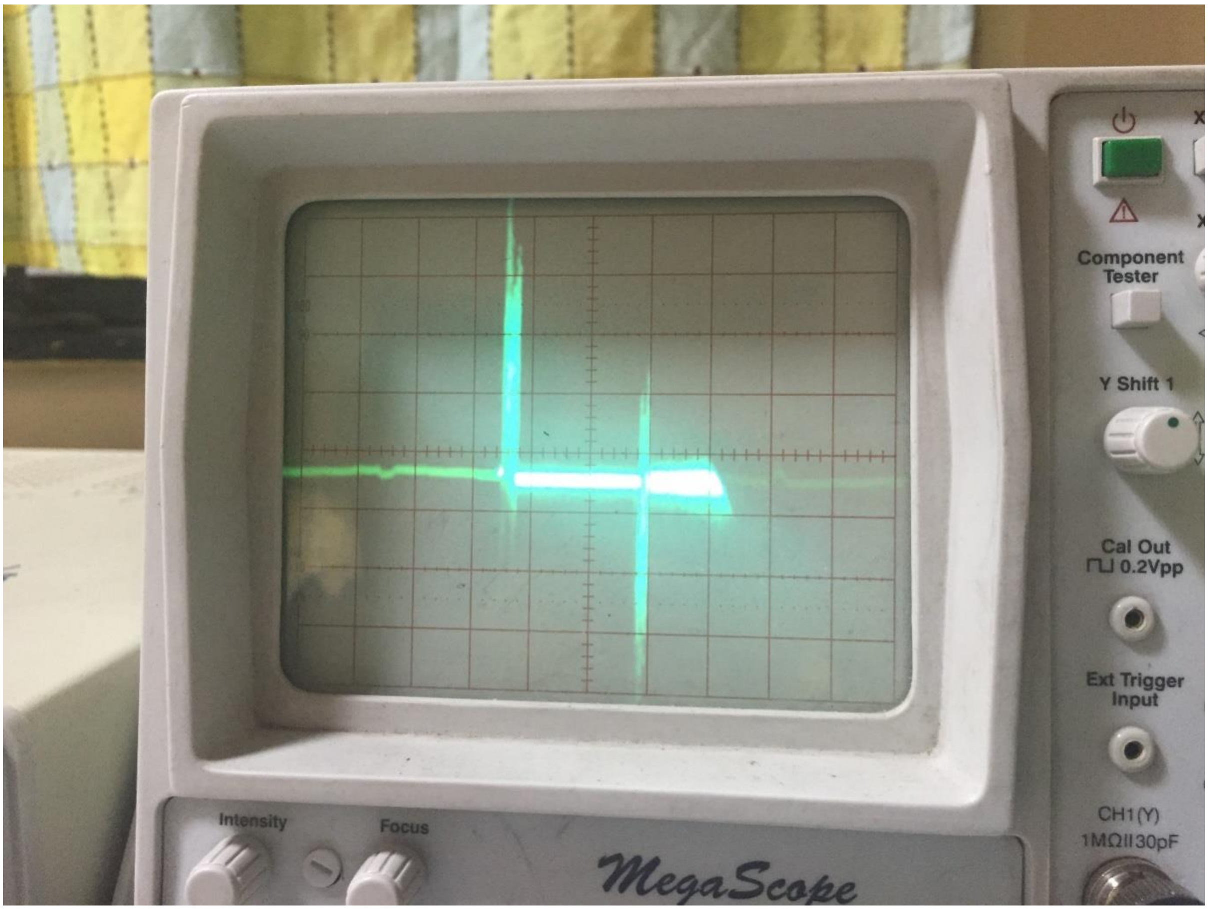

Figure – 46 Output of ECG on CRO

Circuit Diagrams of Sensors:

Pulse oximeter, ECG sensor, Xbee modules are given below:

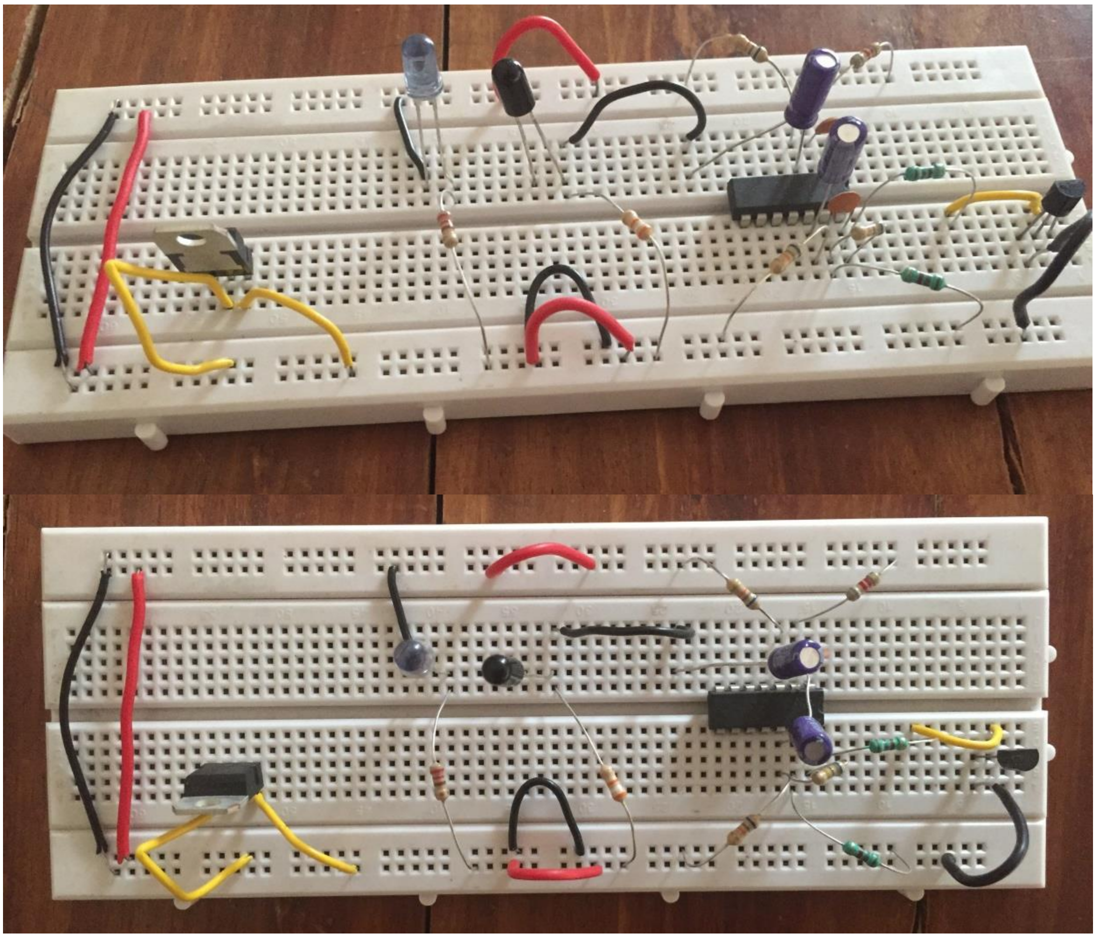

Figure – 47 Pulse Oximeter Circuit

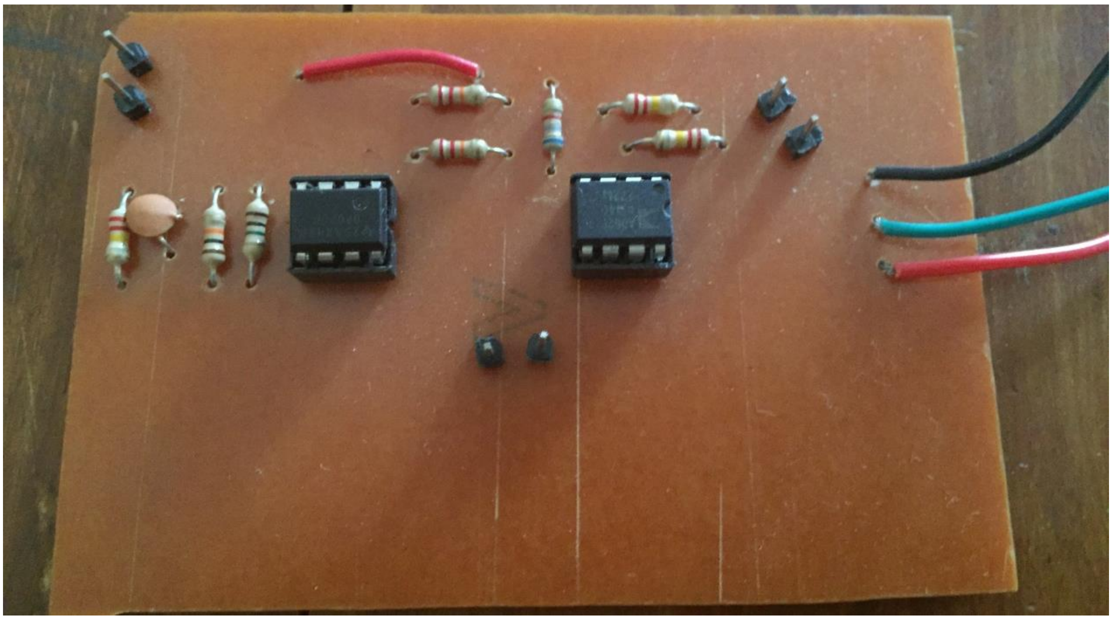

Figure – 48 ECG Circuit

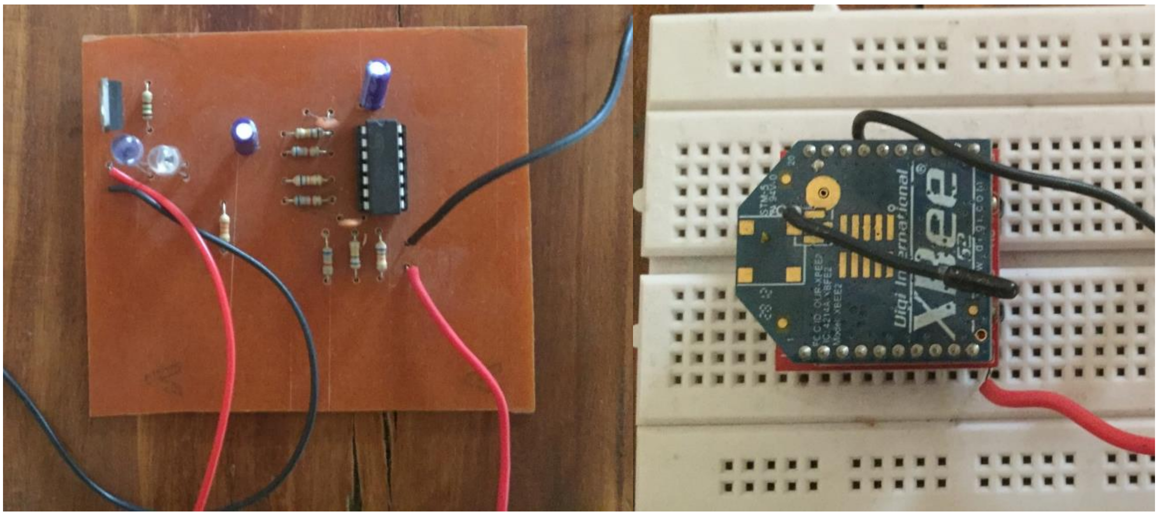

Figure – 49 Pulse oximeter on circuit Board and XBee Module used in our Project

- **Graph on Matlab**

These are the signals send by Xbee to local coordinator.it is then signal processed and displayed with respect to time. There will a small time gap or delay for displaying the signals because of some packet loss and interference. Perfect data can't be seen because of sampling rate.

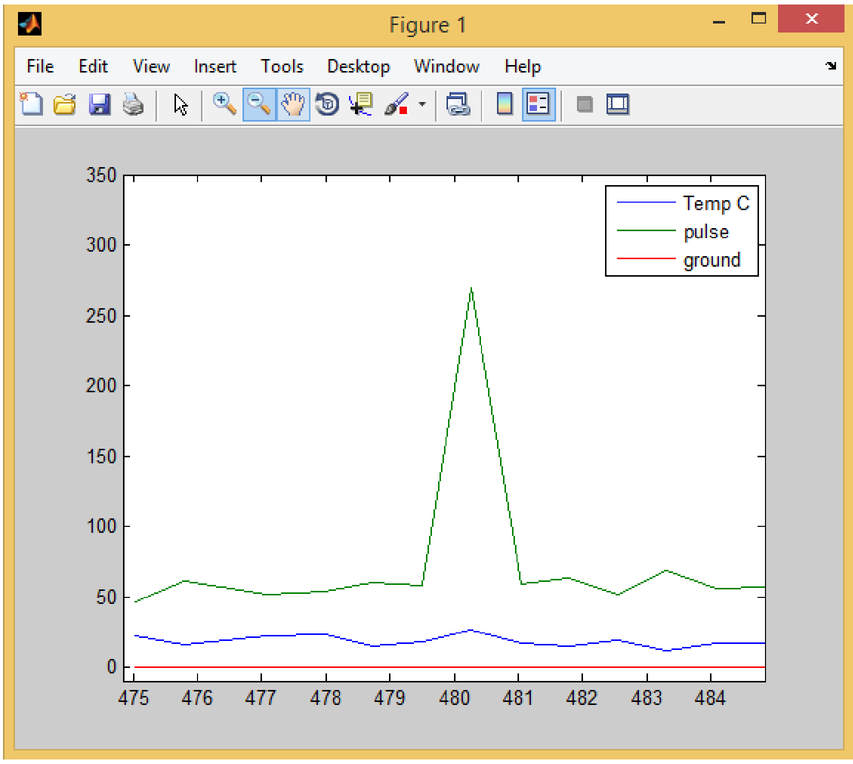

Figure – 50 Matlab Output

7. CONCLUSION

E-Health care system has been developed in this project, which monitors the ECG and pulse of patient in real time and sends the vital data to the nearby server or computer using a radio wave transceiver. For air interface communication Zigbee modules are used. Xbee modules are configured using XCTU software. These Xbee modules are used, one acts as coordinator while the other is router. Router will send data collected from sensors to coordinator.

From now there can be two ways of processing-

- The coordinator is directly connected to the computer using the com port. The data will be processed in system using Matlab software. The required data is taking from series of bits from com port and required processing is done for signals. So that we will get the patient present condition. But there are few constraints like delay time for processing and low sampling rate of analog signal to digital signal. As ECG and pulse oximeter changes their values abruptly.

- By using msp430 in the place of computer, as microcontroller uses very low power and have a high speed operation, it will process the data received by coordinator. The processed data is now sent to any nearby smart devices by using a Bluetooth module HC-05.as now data is present in smart mobile using apps it can be send to various devices. there is an advantage of using controller as it of small size and we can easily program it, like we don’t require data for every second there we can establish some threshold (abnormal conditions) and end-user(doctor) will get message when the patient reading are out of threshold.

NetSim is the simulation tool which is used to get the parameters of wireless networks. In this project the channel characteristics and different variants of wireless networks are observed. The data from all the different cases are taken and observed that no path loss has maximum collision of packets and usually LOS has got the better packet loss error. The mobility model is also observed. The effect of high power level cluster on the low level power level cluster is observed. The packet loss increases as the power level difference increases between the clusters, though the packet loss varies not so much.

8. REFERENCES

1. http://indiatoday.intoday.in/story/India's+no.1+killer:+Heart+disease/1/92422.html
2. http://www.informationweek.com/mobile/10-wearable-health-tech-devices-to-watch/d/d-id/1107148?
3. http://mashable.com/2014/12/02/health-fitness-wearables/
4. Lee et al. EURASIP Journal on Wireless Communications and Networking 2012, 2012:219 Page 2 of 12 http://jis.eurasipjournals.com/content/2012/1/219
5. International Journal of Instrumentation and Control Systems (IJICS) Vol.4, No.1, January 2014 Evaluating ECG Capturing Using Sound-Card of PC/Laptop by B. N. Patel , D. N. Shah
6. http://www.tetcos.com/netsim_gen.html
7. Comparative study of DSR, AODV Routing Protocol for WPAN Using NetSim Simulator by Padmalaya Nayak, Saraswati Bhakare and Nazia Tabassum
8. http://www.freescale.com/files/32bit/doc/app_note/AN4327.pdf?tid=AMdlDR
9. http://www.analog.com/library/analogDialogue/archives/41-01/pulse_oximeter.html
10. Stimulating Student Learning with a Novel "In-House" Pulse Oximeter Design by Jianchu Yao, M.S. and Steve Warren, Ph.D.
11. Introduction to Biomedical Equipment Technology by Joseph J. Carr and John M. Brown.
12. http://www.ti.com/lit/ug/tidu542/tidu542.pdf
13. http://e2e.ti.com/support/amplifiers/precision_amplifiers/f/14/t/148599
14. Performance Analysis of XBee ZB Module Based Wireless Sensor Networks by Rajeev Piyare, Seong-ro Lee
15. http://healthcare.analog.com/en/patientmonitoring/ecg-diagnostic-line-powered/segment/health.html

9. ANNEXTURE

List of codes and NetSim Simulation Screenshots

Now the message is received to the zigbee at laptop side. Themodes of data are serial, and have a specific baudrate.now we can process the data and can process.

This raw data received is converted to message by the Matlab code.

```
Clear;

%User Defined Properties

time = 0;

data = 0; data2 = 0; data3 = 0;

data4 = 0; data5 = 0; data6 = 0;

count = 0;

plotTitle = 'Serial Data Log';  % plot title

xLabel = 'Elapsed Time (s)';    % x-axis label

yLabel = 'Data';                % y-axis label

plotGrid = 'on';                % 'off' to turn off grid

min = -10;                      % set y-min

max = 1023;                     % set y-max

scrollWidth = 10;               % display period in plot, plot entire data log if <= 0

delay = .01;                    % make sure sample faster than resolution

%Open Serial COM Port

while(1) %Loop when Plot is Active   waitforbuttonpress==1

%Define Function Variables

%Set up Plot

plotGraph = plot(time,data3,'ro',time,data4,'go',time,data5,'bo',time,data6,'yo');

file_format = '%5d %5d %5d %5d %5d\n';

title(plotTitle,'FontSize',25);

xlabel(xLabel,'FontSize',15);

ylabel(yLabel,'FontSize',15);

axis([0 10 min max]);
```

```
grid(plotGrid);
serialPort = 'COM7';          % define COM port #
baudeRate = 9600;
s = serial(serialPort, 'BaudRate',baudeRate)
disp('Close Plot to End Session');
fopen(s);
dat = fscanf(s,'%c'); %Read Data from Serial as Float
dat=fread(s,34);
disp(dat);
if(dat(1)==126)
disp('data received successfully');
  %  if(~isempty(dat) && isfloat(dat)) %Make sure Data Type is Correct
       count = count + 1;
       time(count) = toc;    %Extract Elapsed Time
       data(count) = dat(1); %Extract 1st Data Element
       data2(count) = dat(2);
       data31(count) = dat(32); data32(count) = dat(33);
       data41(count) = dat(30); data42(count) = dat(31);
       data51(count) = dat(28); data52(count) = dat(29);
       data61(count) = dat(26); data62(count) = dat(27);
       data3(count)=data31(count)*16 + data32(count);
       data4(count)=data41(count)*16 + data42(count);
       data5(count)=data51(count)*16 + data52(count);
       data6(count)=data61(count)*16 + data62(count);
       disp('port4 data'); disp(data4(count));
       disp(data3(count)); disp('port3 data');
       disp('port2 data'); disp(data5(count));
       disp('port1 data');disp(data6(count));


       data(count); data2(count); data3(count);
```

```
        rec_data=[count data3(count) data4(count) data5(count) data6(count)];

        file_open= fopen('patientrecord.txt','at');

        fprintf(file_open,'%d\t',rec_data(1) );

        fprintf(file_open,'%d\t',rec_data(2) );

        fprintf(file_open,'%d\t',rec_data(3) );

        fprintf(file_open,'%d\t',rec_data(4) );

        fprintf(file_open,'%d\n',rec_data(5) );

        fclose(file_open);


        %Set Axis according to Scroll Width

        if(scrollWidth > 0)

          disp('displaying data');

         plot(time(time > time(count)-scrollWidth),data3(time > time(count)-scrollWidth),time(time >time(count)-
         scrollWidth),data4(time > time(count)-scrollWidth),time(time > time(count)-scrollWidth),data5(time >
         time(count)-scrollWidth),time(time > time(count)-scrollWidth),data6(time > time(count)-scrollWidth));

        legend('Temp','pulse','ECG','ground');

        axis([time(count)-scrollWidth time(count) min max]);

        else

        set(plotGraph,'XData',time,'YData',data3);

        axis([0 time(count) min max]);

        end

              %Allow MATLAB to Update Plot

        pause(delay);

end

%Close Serial COM Port and Delete useless Variables

fclose(s);

disp('Session Terminated...');

end;
```

```
#include "main.h"

#include "DSR.h"

#include "List.h"

#include "802_15_4.h"

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#define SIZEOFCLUSTERS 4                                  //SIZEOFCLUSTERS can be 1,4,9,16,25

#define NUMBEROFCLUSTERS 25

//int NUMBEROFCLUSTERS=claro();

static int CHcount[NUMBEROFCLUSTERS];

static int prevCH[NUMBEROFCLUSTERS];

int number;

/* initialize random seed: */

srand ( time(NULL) );

/* Generate a random number: */

number = rand() % 10 + 1;

return number;


// Custom code for 6 clusters and size of clusters being 9

//int ClusterElements[NUMBEROFCLUSTERS][SIZEOFCLUSTERS] = {{1,2,3,4,5,6,7,8,9},\
                                                       {10,11,12,13,14,15,16,17,18},\
                                                                               {19,20,21,22,23,24,25,26,27},\
                                                                       {28,29,30,31,32,33,34,35,36},\
                                                                       {37,38,39,40,41,42,43,44,45},\
      {46,47,48,49,50,51,52,53,54}};

for(i=0;i<NUMBERFCLUSTERS;i++)

{

  for(j=0;j<SIZEOFCLUSTERS;j++)

  {

   a[i][j] =k;

   k=k+1;

  }
```

```
}

//For 36 sensors and SIZEOFCLUSTERS = 9, uncomment this
//int ClusterElements[NUMBEROFCLUSTERS][SIZEOFCLUSTERS]=
{{1,2,3,7,8,9,13,14,15},{4,5,6,10,11,12,16,17,18},{19,20,21,25,26,27,31,32,33},{22,23,24,28,29,30,34,35,36}};
int fn_NetSim_LEACH_CheckDestination(NETSIM_ID nDeviceId, NETSIM_ID nDestinationId)
//Function to check whether the Device ID is same as the Destination ID
{
	if(nDeviceId == nDestinationId)
		return 1;
	else
		return 0;
}

int fn_NetSim_LEACH_GetNextHop(NetSim_EVENTDETAILS* pstruEventDetails)
{
	int nextHop;
	NETSIM_ID nInterface;

	//int CH[NUMBEROFCLUSTERS] = {23,28,73,78};
	int CH[NUMBEROFCLUSTERS] = {5,12,21,30,40,50};
	//int CH[NUMBEROFCLUSTERS] = {8,11,26,29};
	//int CH[NUMBEROFCLUSTERS] = {6,7,10,11};
	//int CH[NUMBEROFCLUSTERS] = {1,2,3,4};

	int i;
	int ClusterId;

	//This for loop dynamically assigns the Cluster Heads based on their energy.
	//Comment this for loop to enable fixed cluster heads.
	/*for (i=0; i<NUMBEROFCLUSTERS ; i++)
```

```
{
    if(CHcount[i] == 4)
    {
        CH [i]= fn_NetSim_LEACH_AssignClusterHead(i,pstruDevicePower);
        prevCH[i] = CH[i];
        CHcount[i] = 0;
    }
    else
    {
        CHcount[i]++;
        if(prevCH[i] != 0)
            CH[i] = prevCH[i];
    }
}*/



//Static Routes defined for 4 Clusters.
//If the sensor is the Cluster Head, it forwards it to the next Cluster Head or
//to the Sink.Otherwise, it forwards the packet to the Cluster Head of its cluster.
if(pstruEventDetails->pPacket->nSourceId == pstruEventDetails->nDeviceId)
//For the first hop
{
/*if(pstruEventDetails->nDeviceId == CH[0])
        nextHop = CH[2];
    else if(pstruEventDetails->nDeviceId == CH[1])
        nextHop = CH[3];
    else */
    if(pstruEventDetails->nDeviceId == CH[0] || pstruEventDetails->nDeviceId ==
CH[1]||pstruEventDetails->nDeviceId == CH[2] ||pstruEventDetails->nDeviceId == CH[3] ||pstruEventDetails->nDeviceId
== CH[4] ||pstruEventDetails->nDeviceId == CH[5] ||pstruEventDetails->nDeviceId == CH[6] ||pstruEventDetails-
>nDeviceId == CH[7] ||
        pstruEventDetails->nDeviceId == CH[8] ||pstruEventDetails->nDeviceId == CH[9])
        nextHop = pstruEventDetails->pPacket->nDestinationId;
```

```
            else
            {
                ClusterId = fn_NetSim_LEACH_IdentifyCluster(pstruEventDetails->nDeviceId);
                nextHop = CH[ClusterId];
            }
        }
        else
  {
            ClusterId = fn_NetSim_LEACH_IdentifyCluster(pstruEventDetails->nDeviceId);
            /*if(ClusterId < 2)
                nextHop = CH[ClusterId + 2];
            else */
                nextHop = pstruEventDetails->pPacket->nDestinationId;
        }


        //Updating the Transmitter ID, Receiver ID and NextHopIP in the pstruEventDetails
        free(pstruEventDetails->pPacket->pstruNetworkData->szNextHopIp);
        pstruEventDetails->pPacket->pstruNetworkData->szNextHopIp = DSR_DEV_IP_COPY(nextHop);
        pstruEventDetails->pPacket->nTransmitterId = pstruEventDetails->nDeviceId;
        pstruEventDetails->pPacket->nReceiverId = nextHop;


        return 1;
}

int fn_NetSim_LEACH_AssignClusterHead (int ClusterNumber,POWER **pstruDevicePower)
//Function to dynamically assign Cluster Heads based on the sensors' remaining energy.
{
        int  ClusterHeadID,i,DeviceId;
        double ClusterHeadPower,MaximumPower;
```

```
	if(SIZEOFCLUSTERS % 2 ==0)
		DeviceId = ClusterElements[ClusterNumber][SIZEOFCLUSTERS/2];
	else
		DeviceId = ClusterElements[ClusterNumber][(SIZEOFCLUSTERS-1)/2];

	MaximumPower = pstruDevicePower[DeviceId-1]->dRemainingPower;
	ClusterHeadID = DeviceId;

	for(i=0; i<SIZEOFCLUSTERS; i++)
	{
		DeviceId = ClusterElements[ClusterNumber][i];
		ClusterHeadPower = pstruDevicePower[DeviceId-1]->dRemainingPower;
		if(MaximumPower < ClusterHeadPower)
		{
		MaximumPower = ClusterHeadPower;
		ClusterHeadID = DeviceId;
		}
	}
	return ClusterHeadID;
}
int fn_NetSim_LEACH_IdentifyCluster(DeviceId)
//Function to identify the cluster of the sensor.
{
	int i,j;
	for(i=0; i<NUMBEROFCLUSTERS; i++)
	{
		for(j=0; j<SIZEOFCLUSTERS; j++)
		{
			if(DeviceId == ClusterElements[i][j])
				return i;
		}
	}
```

}

NetSim Simulation ScreenShots:

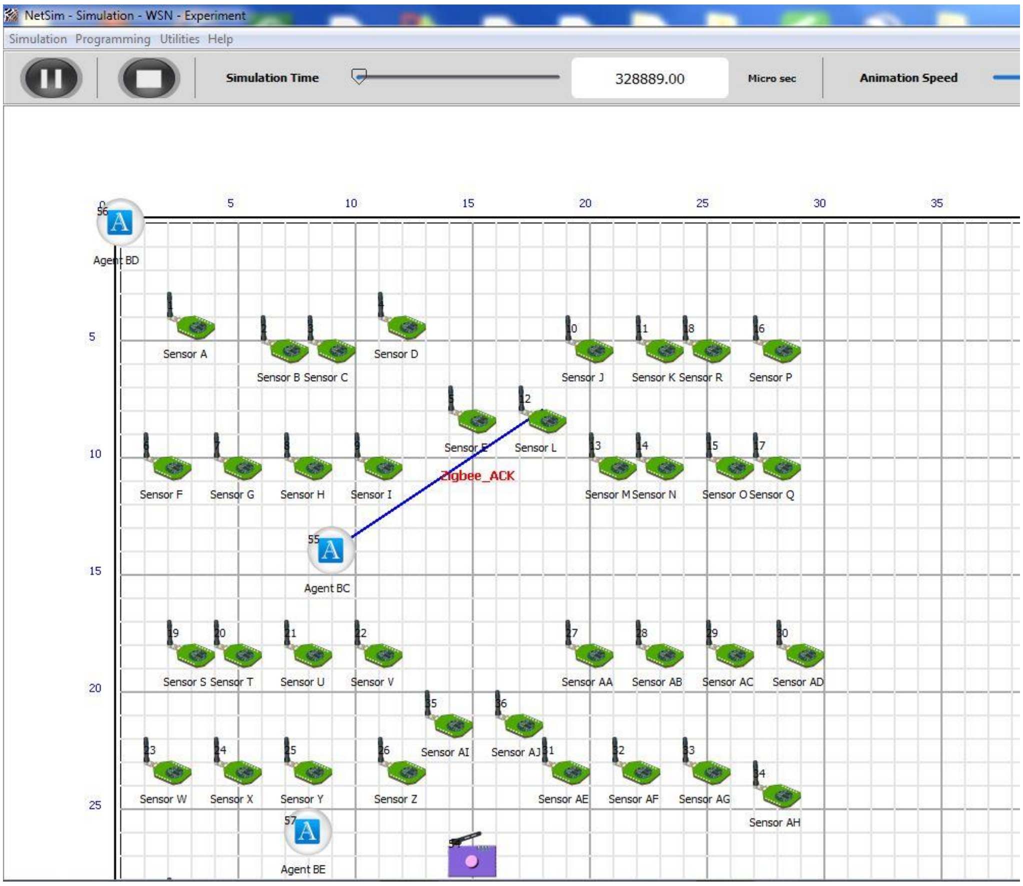

Figure – 51 Packet Trace Animation

Figure – 52 Packet trace animation at different time (change of position of Agent)

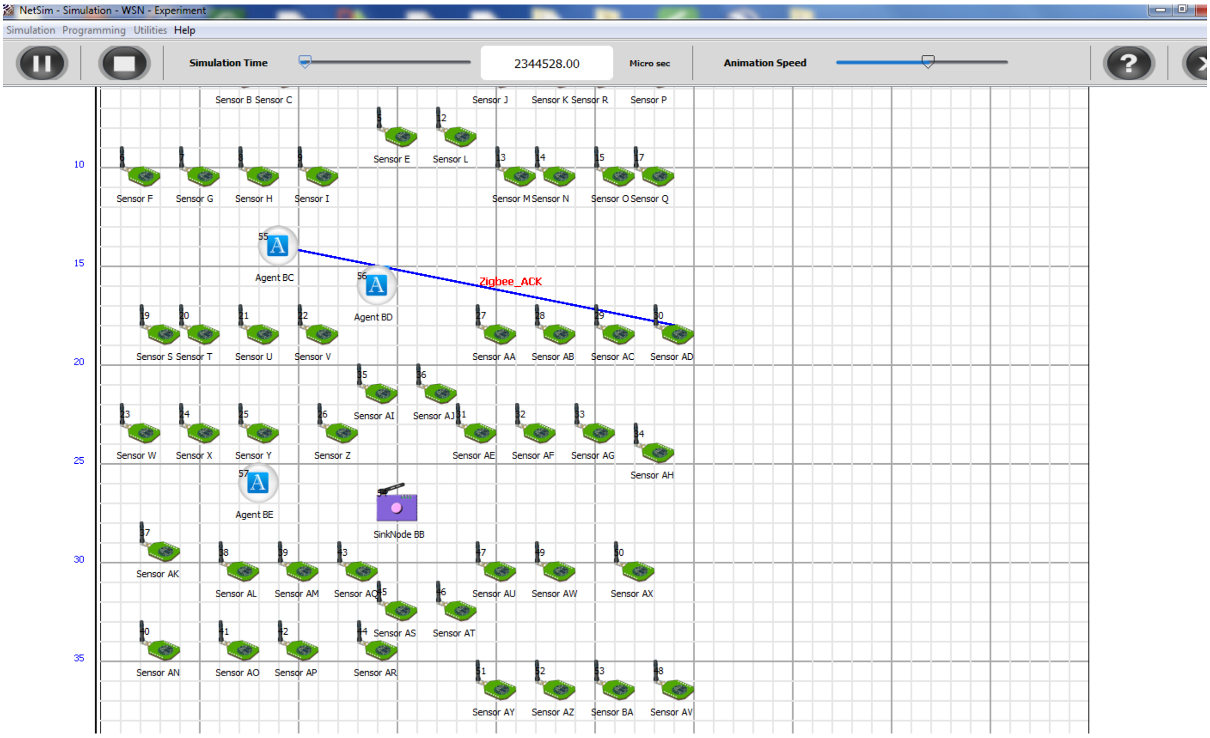

Figure – 53 Packet trace animation with different cluster arrrangement

LEACH Metrics

DeviceID	Data Sent(Network Layer)	Data Sent(Physical Layer)	Data Received(Network Layer)	Data Received(Physical Layer)	Lifetime(sec)
1	89	62	0	0	0.000000
2	83	60	0	0	0.000000
3	78	1	0	0	0.000000
4	62	46	0	0	0.000000
5	59	47	0	0	0.000000
6	56	45	0	0	0.000000
7	92	84	0	0	79.008128
8	85	67	0	0	0.000000
9	79	59	0	0	0.000000
10	63	55	0	0	0.000000
11	60	44	0	0	0.000000
12	57	46	0	0	0.000000
13	94	70	0	0	0.000000
14	87	65	0	0	0.000000
15	81	60	0	0	0.000000
16	76	60	0	0	0.000000
17	61	47	0	0	0.000000
18	58	44	0	0	0.000000
19	55	45	0	0	0.000000
20	89	68	0	0	0.000000

Figure – 54 Leach metrics of the simulated WSN network

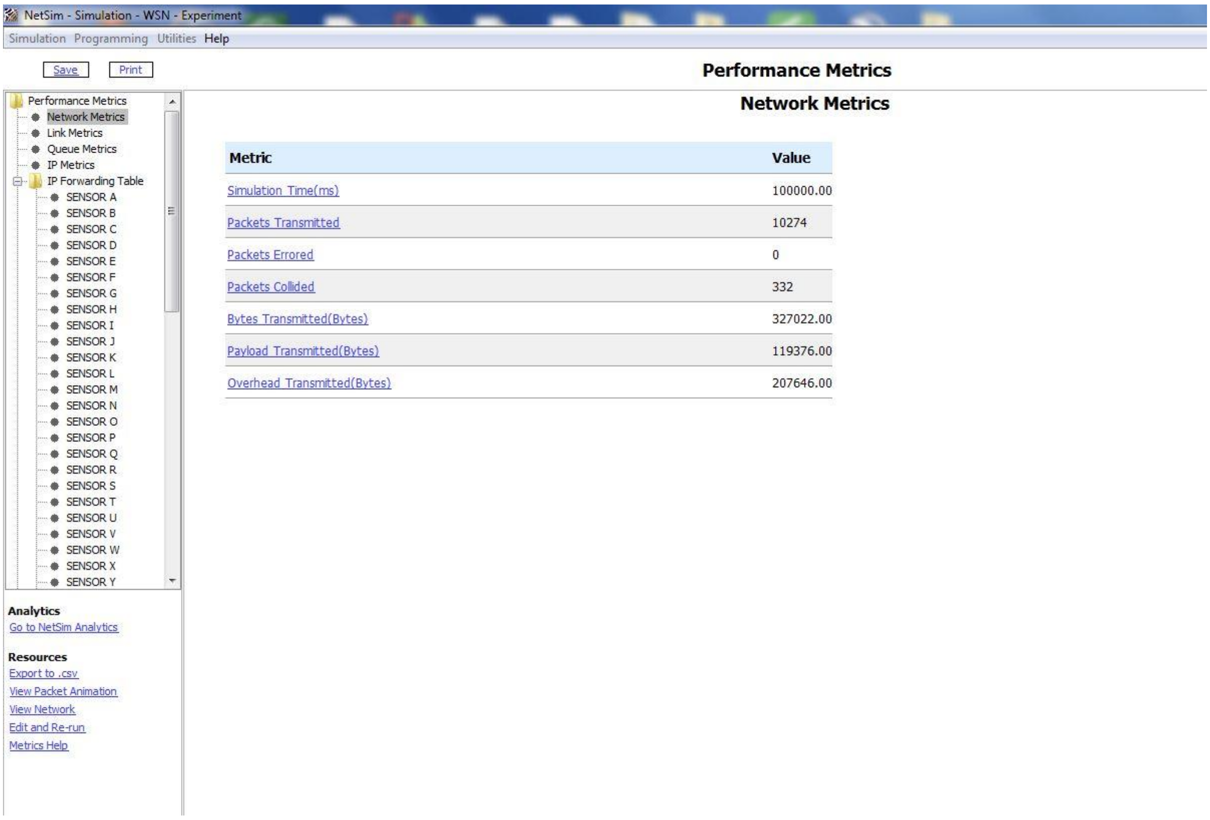

Figure -55 The dialog Box showing Performance metrics after the simulation

Network Metrics

Metric	Value
Simulation Time(ms)	50000.00
Packets Transmitted	66
Packets Errored	0
Packets Collided	6
Bytes Transmitted(Bytes)	2761.00
Payload Transmitted(Bytes)	823.00
Overhead Transmitted(Bytes)	1938.00

Figure – 56 The detailed view of Network metrics of WSN network

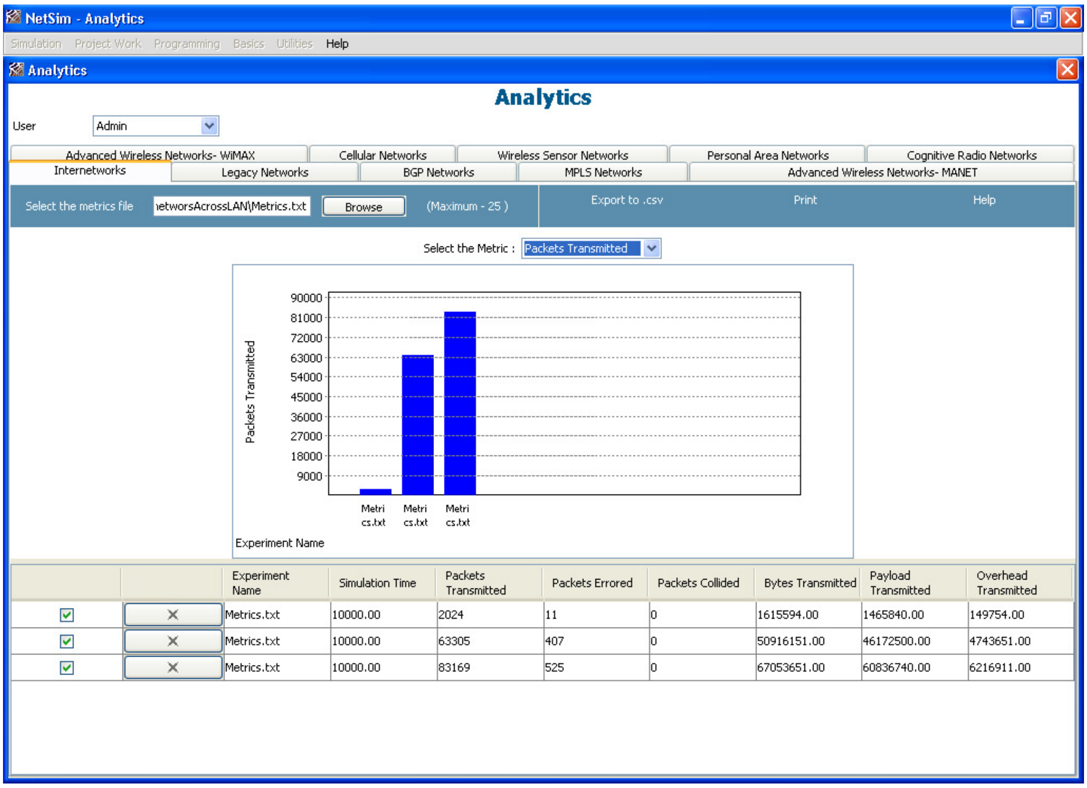

Figure – 57 Network Analysis which shows Metrics that compares the metrics of system

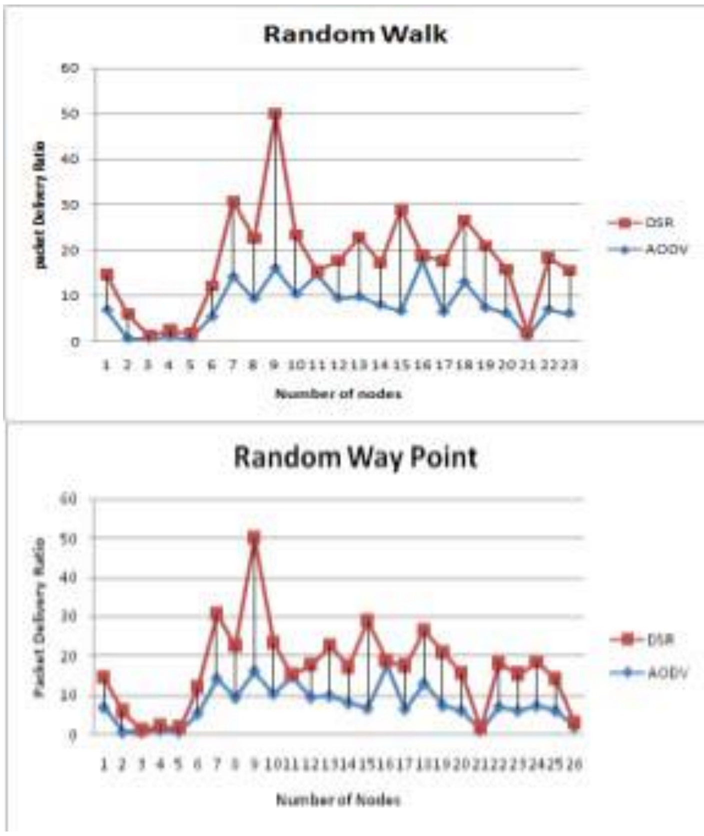

Figure – 58 Analysis of Packet delivery rate V/s Mobility model types