LOW LATENCY MULTICHANNEL LMAC PROTOCOL FOR WIRELESS SENSOR NETWORKS

by

Abul Kalam Azad

MASTER OF SCIENCE

IN

INFORMATION AND COMMUNICATION TECHNOLOGY

INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

2016

The thesis titled "**LOW LATENCY MULTICHANNEL LMAC PROTOCOL FOR WIRELESS SENSOR NETWORKS**" submitted by Abul Kalam Azad, Roll No. 0411312033F, and Session: April, 2011, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Information and Communication Technology on 31 July, 2016.

BOARD OF EXAMINERS

1. Dr. Mohammad Shah Alam Associate Professor IICT, BUET, Dhaka	Chairman (Supervisor)
2. Dr. Md. Liakot Ali Professor & Director IICT, BUET, Dhaka	Member (Ex-Officio)
3. Dr. Md. Saiful Islam Professor IICT, BUET, Dhaka	Member
4. Dr. Lutfa Akter Associate Professor Dept. of EEE, BUET, Dhaka	Member
5. Dr. Md. Shariful Islam Associate Professor Institute of Information Technology University of Dhaka	Member (External)

DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidate

Abul Kalam Azad

DEDICATION

THIS THESIS IS DEDICATED

TO

MY BELOVED PARENTS

Table of Contents

CHAPTER 3

List of Tables

List of Figures

List of Abbreviations

AI-LMAC	Information-Centric Lightweight Medium Access Control
A-MAC	Asynchronous Medium Access Control
BS	Base Station
CCA	Check Channel Assessment
CF	Common Frequency
CM	Control Message
CPU	Central Processing Unit
CR	Communication Request
CRA	Contention Resolution Algorithm
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access/ Collision Avoidance
CTS	Clear To Send
DA	Demand Assignment
DDT	Data Distribution Table
DFI	Data Forwarding Interruption
DLL	Data Link Layer
DM	Data Message
DSSS	Direct Sequence Spread Spectrum
DTG	Distance To Gateway
EMAC	Eyes Medium Access Control
FDMA	Frequency Division Multiple Access
FFD	Full-Function Devices
FHSS	Frequency Hopping Spread Spectrum
ID	Identification Number
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISA	Instrumentation Systems and Automation society
ISM	Industrial, Scientific and Medical
LEACH	Low-Energy Adaptive Clustering Hierarchy

LLC	Logical Link Control
LL-MCLMAC	Low Latency Multichannel Lightweight Medium Access Control
LMAC	Lightweight Medium Access Control
MAC	Medium Access Control
MACA	Multiple Access Collision Avoidance
MANET	Mobile Ad-hoc NETworks
MC-LMAC	Multi-Channel Lightweight Medium Access Control
NAT	Network Address Translator
NIC	Network Interface Card
PDA	Personal Digital Assistance
QoS	Quality of Service
RF	Radio Frequency
RFD	Reduced-Function Devices
RRA	Random Reservation Access
RTS	Ready To Send
SDMA	Space division multiple access
S-MAC	Sensor Medium Access Control
SMP	Sensor Management Protocol
SNSP	Sensor Network Services Platform
TC	Traffic Control
TDMA	Time Division Multiple Aceess
T-MAC	Timeout Medium Access Control
WLAN	Wireless Local Area Network
WPAN	Wireless personal Area Network
WSN	Wireless Sensor Network

Acknowledgments

All praises are for the Almighty Allah for giving me the strength, without which I couldn't afford to attempt this research work.

I would like to express my sincere and heartiest gratitude to my honorable thesis supervisor Dr. Mohammad Shah Alam, Associate Professor, Institute of Information and Communication Technology (IICT), Bangladesh University of Engineering and Technology (BUET), Dhaka for his continuous motivation, guidance and keen encouragement which helped me throughout the time of my research work. Nothing is comparable to his keen advice and the freedom he provided for me in research. I am grateful to him for his cooperation throughout my thesis work.

I would like to thank all the members of the board of examiner for their precious time in understanding my work and their insightful comments. I would like to thank to all of my friends and colleagues for their cooperation. Last but not least I am grateful to my parents for their continuous supports and cooperation.

Abstract

Wireless sensor networks consist of a large number of resource constraint sensor nodes. Due to it's wide range of applications, it has drawn much attention in the research community. As wireless sensor networks use shared medium, designing an energy efficient MAC protocol is highly desirable for maintaining other application specific performance requirements. Between contention-based and schedule-based MAC protocols; contention-based MAC protocol consumes much energy and cost higher for collision, idle listening and overhearing problem. In the contrary, schedule-based MAC protocol ensures energy efficient operation at the cost of high end-to-end delay. Although existing lightweight medium access control (LMAC) protocol is an energy efficient protocol, it introduces high end-to-end delay and low network throughput due to the long waiting time. The low throughput problem of LMAC protocol has already been mitigated in multichannel LMAC (MC-LMAC) protocol by ensuring energy efficient operation. However, the delay problem still exists for the same reason of the long waiting time. Therefore, it is highly expected to design an energy efficient MAC protocol for delay sensitive and high throughput applications.

In this research work, a low latency multichannel LMAC (LL-MCLMAC) protocol is presented for WSNs which significantly reduces the end-to-end delay while improving the system throughput further. In the proposed protocol, the delay problem of existing LMAC and MC-LMAC protocol has been significantly improved by allowing a node to send its packet using two timeslots of a channel from multiple channel. The timeslot pair are kept half of the frame time separated which is decided by an algorithm. Moreover, the schedule-based operations ensure an energy efficient operation and use of multichannel results high throughput. A simulation model using OMNeT++ and MiXiM is designed to show the effectiveness of the proposed protocol. Finally, performance of proposed protocol is compared with the existing LMAC and MC-LMAC protocol, in terms of number of received packets, end-to-end delay, energy consumption and network lifetime. Simulation results clearly show that LL-MCLMAC achieves significant improvement in end-to-end delay and network throughput over existing LMAC and MC-LMAC protocols while ensuring an energy efficient operation.

CHAPTER 1

INTRODUCTION

For the last few years wireless sensor networks (WSNs) have been the subject of research as well as commercial interest due to its wide range of applications, such as target tracking, habitant sensing and fire detection in hazardous, hostile and remote areas. Such kind of networks are often deployed in inaccessible application areas and consist of a huge number of resource constrained sensor nodes. In wireless sensor networks, these nodes are organized in a multi hop fashion and generally perform a common task by coordination among each other. The main components of each node in wireless sensor networks are: controller, memory, sensors and actuators, communication devices and normally limited battery operated power supply and for most of the applications it is not feasible to provide the power supply by recharging or replacing the batteries for sensor nodes. So to sustain the network for long time, energy efficient operations becomes the most important issue of attention. So for WSN the issue of energy efficiency becomes more and more important whereas depending on the applications other performance criteria like scalability (when network topology, node density or network size changes, because some nodes may join in the network; some nodes may die or some nodes may also move to a different locations), delay (time difference between a packets originating time and arriving time in the sink node), fairness (giving each node almost equal facilities to share the medium/ network resources), throughput and bandwidth utilization are also important.

Research studies reveals that, for shared-medium wireless networks energy wastage occurs mainly from collision, overhearing, control packet overhead and idle listening. All of the above problems are handled by the medium access control (MAC) techniques. The main goal of a WSN MAC layer protocol is to guarantee that two or more nodes will not transmit in their collision domain by allocating a shared wireless medium among sensor nodes as fairly as possible.

Since for wireless sensor networks, energy efficiency is the primary concern and most of the reasons of energy wastage are handled by MAC protocols. So, to design a good MAC protocol for WSN is important.

1.1 Related Research Works

From the invention of wireless sensor networks, researchers continually studying to offer a new method for the controlling of medium access shared among many nodes. The performance of WSN mostly rely on the use of a suitable MAC protocol. There are two broad categories of WSN MAC protocols: contention-based protocols and schedule-based protocols. Contention-based MAC protocols are based on carrier sense multiple access (CSMA) technique in which for a given transmit opportunity every transmitting nodes can try their luck to transmit at the risk of collision and costs higher for idle listening and overhearing. Whereas, schedule-based MAC protocols are based on time division multiple access (TDMA) technique. A lot of research studies have been done on the above mentioned two categories for wireless sensor networks. Here the focus is on schedule-based MAC protocols for WSN. In the context of WSNs, there exist recent proposals [1-14] that use the concept of schedule-based channel access to improve energy consumption by overcoming collision, overhearing and idle listening problems. However, most of the schedule-based MAC protocols have been suffering from low throughput and high end-to-end delay. W. R. Heinzelman et al. [1] has proposed a TDMA based protocol for dense sensor network by partitioning the sensor nodes into clusters and each cluster is coordinated by a dedicated clusterhead which creates a TDMA schedule for its cluster member and aggregates data of its members then send directly to the sink. For large geographic areas this protocol may encounter problem, because a clusterhead may not have enough energy to reach the sink. T. Nieberg et al. [2] has proposed a connection oriented TDMA based MAC protocol, where time is divided into timeslots and each node allows to take control over a timeslot which it can use to send data with high latency without having contended for the medium. In [4], Lu et al. has designed an adaptive energy efficient MAC protocol for fixed WSN, which addresses data forwarding interruption (DFI) problem by sleep delay and allows continuous packet forwarding. However, this protocol is not suitable for data transmission and flow between arbitrary source and destination and does not provide reliability for end to end data delivery. Furthermore, this protocol does not take fairness into account. R. Kalidindi et al. [5] has introduced a TDMA based distributed energy-aware MAC protocol with the objective to prolong network lifetime by allowing critical (low power) nodes to sleep more time than the other nodes. However, the performance of this protocol may be seriously affected by the overhearing problem. Lightweight medium access control (LMAC) [3] is another

TDMA based self-organizing (since timeslot assignment and synchronization was done in a distributed manner) MAC protocol proposed by L.F .W. Van Hoesel et al. in which a node can only send its packet using a timeslot in a frame like Eyes MAC (EMAC) [2], therefore if a node has more packets to send, it has to wait consecutive frame durations for sending its packets which results in high delay and low throughput as well. LMAC allows a node to own only one timeslot, whatever the data traffic will flow through that node. An improved version of LMAC protocol named an adaptive, information-centric and lightweight MAC (AI-LMAC) [6] has been proposed by S. Chatterjea et al. In AI-LMAC protocol, if a node is expected to flow more traffic, then it is allowed for this node to use more than one timeslots. Despite benefits provided by AI-LMAC, in high dense WSNs, it becomes difficult for a node to get more than one free slot between two hop neighbors to avoid hidden terminal problem. So timeslot shortage problems may happen and also there have a chance of timeslot conflict problem here. So, the delay problem has not been reduced here as expected. Multichannel LMAC (MC-LMAC) [7] is another schedule based energy efficient MAC protocol designed by O. D. Incel et al. with the objective to increase throughput. Here, although a node is allowed to take control of one timeslot from multiple channels, a node is allowed to send packet using only one timeslot per frame like LMAC. So the delay problem has not been resolved in MC-LMAC efficiently as well. O. D. Incel et al. has proposed another multi-channel scheduling MAC protocol [8] using a different algorithm for channel and timeslot assignment. M. A. Hamid et al. has introduced a new schedule-based multi-channel MAC protocol [9] for WSN, where receiving nodes can select timeslots over which it may receive data from the intending sender. In this protocol, neighboring nodes are spilitted into different groups, where nodes of a group may select the slots from the allocated slots of that group only. G. Wang et al. has proposed a low latency MAC protocol [11], where asynchronous duty cycling concepts are used and allows a node to adjust their sleep time adaptively according to the amount of the received data packets and hence to reduce end-to-end delay. S. Gajjar et al. has proposed a TDMA based low latency cross layer MAC protocol [12], where a node allows to use a slot of another node if slot holder node does not have data to send. A low-latency communication protocol for target tracking in WSNs[13] has been introduced by T. N. Quynh et al, which focus on the joint optimization of routing and a MAC strategy by incorporating a new cluster working-cycle synchronization procedure based routing algorithm and a new low-duty-cycle MAC protocol. Survey paper [14], summarizes the characteristics of some of the schedule-based MAC protocols.

1.2 Motivation

Since wireless sensor network may be used for a variety of applications. For some applications, where low end-to-end packet delay and high data throughput are expected while ensuring the energy efficient operations. This type of applications may include:surveillance applications and intrusion detection. So to design an energy efficient MAC protocol with low latency and high throughput is highly required for delay sensitive applications. Between contention-based and schedule-based MAC protocols, schedule-based MAC protocols are more energy efficient but suffers from high end-to-end delay. Thus it is desirable to devise a new MAC protocol which will be energy efficient and ensures low end-to-end delay and high throughput.

As mentioned in the related research work section, LMAC and MC-LMAC protocols suffers from high end-to-end delay. The ability of newer generations commercially available radios to tune their operating frequencies over different channels provides an opportunity to relieve the effects of interference and hence to improve the network performance. So like MC-LMAC protocol, multichannel concept can be used to achieve high throughput. Moreover, a slot selection algorithm depending on the free timeslots of a channel can be used to assign two timeslots of a channel to a node by ensuring the two hop neighbor timeslot conflict avoidance over the same channel. Hence, delay problem should be minimized. Which is the main motivation of this research work.

1.3 Objectives

Basic consideration of all of the WSN MAC protocol is energy efficient operation by controlling the factors that results energy wastage. Besides this other considerations like: scalability, traffic adaptivity, latency, fairness and throughput are also important. The main objectives of this thesis work is to design a new low latency multichannel medium access control protocol for wireless sensor networks. Throughout the process of designing this proposed protocol the following basic milestones are the objectives of this thesis.

1. To propose a new schedule based MAC protocol, low latency multichannel LMAC protocol (LL-MCLMAC), for energy efficient delay sensitive WSNs applications.
2. To carry out performance analysis of the proposed LL-MCLMAC protocol in terms of performance parameters such as number of received packet for throughput, end-to-end delay, energy consumption and network lifetime.

3. Finally, to show the efficacy of the proposed MAC protocol, performance comparison between the proposed protocol and other related existing protocols are investigated.

1.4 Organization of Thesis

This thesis work consists of six chapters. Following is the brief discussion of each chapter of this thesis work:

Chapter one briefly introduces wireless sensor network and role of MAC layer in wireless sensor network. Related research regarding schedule-based MAC protocols as well as motivation and objectives are also presented in this chapter.

Chapter two of this thesis work covers a detail overview of wireless sensor network including different types of network nodes, network types, types of applications and challenges of WSN etc.

Chapter three describes the role of MAC layer in network architecture, MAC layer properties for WSN, different issues of MAC design, multiple access technique, MAC performance metrics and classification of wireless MAC protocol etc.

The details of the proposed LL-MCLMAC protocol and proposed slot selection algorithm is explained in chapter four. Some related schedule-based MAC protocols like EMAC, LMAC, AI-LMAC and MC-LAC are also described.

In chapter five, for different node densities the performance of the proposed approach is investigated in terms of performance parameters such as number of received packets for throughput, end-to-end delay for latency, energy consumption for each packet received by sink node and network lifetime. Moreover, in order to show the efficacy of the proposed scheme, a performance comparison between proposed protocol and existing LMAC and MC-LMAC protocols is carried out in this chapter as well.

Chapter six concludes this thesis along with some limitations and future research scopes.

1.5 Summary

This chapter introduces a very brief introduction of wireless sensor networks with its MAC layer responsibilities. A detail analysis of some related schedule-based MAC protocols for WSN are included here. Motivations that enticed us to conduct this thesis are also keep in focus here. Finally, the objectives to propose a new MAC protocol for WSN are mentioned in this chapter.

CHAPTER 2

WSN: A DETAIL OVERVIEW

2.1 Introduction

This chapter introduces a overview of wireless sensor networks, regarding the different WSN standards, WSN protocol stack, different types of network nodes, mobility types, single node architecture and the basic principles of turning individual sensor nodes into a wireless sensor network. This chapter also covers different types of WSN application examples and challenges for WSNs. Finally a comparison between wireless sensor networks and mobile ad-hoc networks are also included in this chapter.

2.2 WSN Standards

This section presents some of the currently used wireless sensor networks (WSNs) standards [15] like: Zigbee, WirelessHART and ISA.100, are currently available for industrial applications.

1. **Zigbee**

   Zigbee is a specification which is based on IEEE 802.15.4 and uses low power digital radios for high level communication protocols. The technology defined by the Zigbee specification is simple and less expensive than most of the other consumer WPANs, such as Bluetooth. Zigbee specification is targeted at low data rate, long battery life radio-frequency (RF) applications and for secure networking. The low cost of Zigbee allows the technology to be widely used in monitoring and wireless control applications.

   In Zigbee standards, there are two types of network devices, such as: Full-Function Devices (FFD) and Reduced-Function Devices (RFD). Using FFD allows the networks to form of any desired type: such as star, mesh and hybrid; whereas RFD can only connect to a full function node [16].

2. **WirelessHART**

   WirelessHART is a mesh networking technology operating in the 2.4 ISM radio band. It uses IEEE 802.15.4 compatible direct sequence spread spectrum (DSSS) radios with channel hopping functionality on a packet by packet basis. Frequency hopping spread spectrum (FHSS) are also defined in the IEEE 802.15.4 standard which allows wirelessHART to hop across the 16 channels. Communication is performed using TDMA technology to coordinate the communications between network devices. WirelessHART is backward compatible with core HART technology.

3. **ISA.100**

ISA-100 technology also uses IEEE 802.15.4 radio technology and stands for international society of automation (ISA). ISA-100 is an emerging technology which promises many new features. This standard aims to enable a single, integrated wireless infrastructure platform for plants and delivers a number of related standards defining wireless systems for industrial automation and also for control applications [17]. The architecture supports wireless systems that span the physical range from a single, small and isolated network. The network may include many thousands of devices and multiple networks that can cover a multi-square-km plant [17].

ISA.100 standard supports channel hopping to avoid any interference from other RF devices operating in the same band and provides the robustness to mitigate multipath interference.

2.3 WSN Protocol Stack

The protocol stack for WSNs consists of five protocol layers [18-19] shown in Figure 2.1.

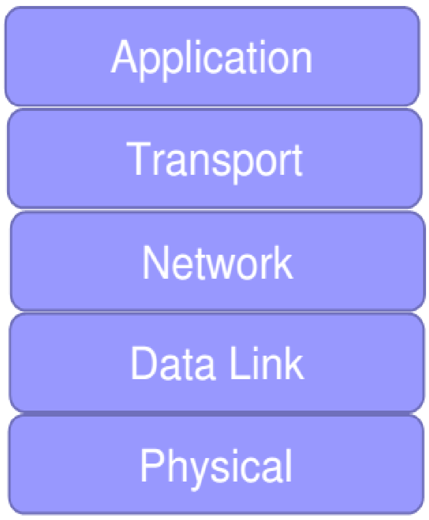


Figure 2.1 Protocol stack for sensor networks.

- **Application Layer:** It defines a standard set of service and interface primitives available to a programmer independently on their implementation on every kind of platform. An example is the so called sensor network services platform (SNSP). The application layer also provides a variety of application layer protocols that perform various sensor network applications, such as node localization, query dissemination, network security, and time synchronization. For example, the application layer management protocol, sensor management protocol (SMP) is a protocol that provides software operations to perform a variety of tasks, for example, synchronizing sensor nodes, exchanging locations related data, moving sensor nodes, querying the status of sensor nodes and scheduling the sensor nodes as well.

- **Transport Layer:** It helps to maintain the reliable end-to-end flow of data between sensor nodes and sink(s), if the sensor networks application requires it. This layer is especially needed when the sensor system is planned to be accessed through external networks or through the Internet. Due to the limited storage capacity and energy constraints of sensor nodes, without modification traditional transport layer protocols cannot be applied directly to sensor networks.

- **Network Layer:** It takes care of routing the data sensed by source sensor nodes to the sink, directing the process of selecting paths along which to send data in the network.

- **Data Link Layer:** It provides the multiplexing of data streams, data frame creation and detection, error control and MAC. The primary objective of MAC is to fair and efficient use of the shared communication resources or medium among multiple sensor nodes to achieve good network performance in terms of network throughput, delivery latency and most importantly energy consumption.

- **Physical Layer:** It is responsible for converting bit streams from the data link layer to a convenient form of signals that are suitable for transmission over the communication medium. It also deals with carrier frequency generation, signal modulation and detection, frequency selection and data encryption. In addition, it must also be concern with the design of the underlying hardware with its electrical and mechanical interfaces.

2.4 Sensor Node Hardware Overview

A sensor node in a wireless sensor network is able to gather sensory information. The functions of a sensor node may include- to gather information from application area, to process that information and to communicate with other nodes that are within its transmission range. A node can receive signal from another node if the received signal power value is greater than a certain SNR threshold value. A basic sensor node comprises five main components [20] shown in Figure 2.2 – Controller, Memory, Sensors and Actuators, Communication and Power Supply.

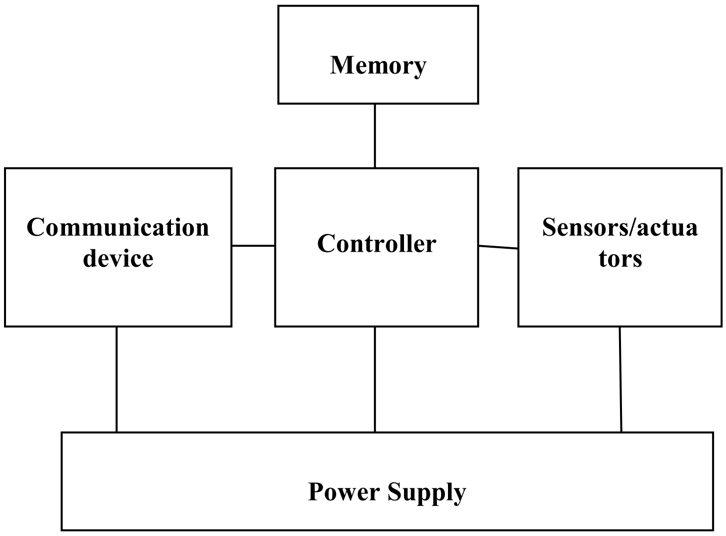


Figure 2.2 Overview of main sensor node hardware.

Controller

The controller is the core of a wireless sensor node. It collects data from the sensors, processes data, and decides when and where to send this data, receives data from other sensor nodes, and decides on the actuators behavior. It has to execute different programs, ranging from time-critical signal processing to application programs; it is called the central processing unit (CPU) of the node.

Memory

Some memory to store programs and intermediate sensor readings, packets from other nodes, and so on; for programs and data different types of memory may be used.

Sensors and Actuators

The actual interface to the physical world; Sensors are used to observe the environment by sensing and actuators are used to control physical objects like: motor, light bulb etc.

Communication

Nodes into a network requires a device for sending information to wireless channel and receiving information from wireless channel. The communication device is used to exchange data between different nodes.

Power Supply

The power source of a sensor node is either non rechargeable also called primary batteries or rechargeable also called secondary batteries by obtaining energy from the environment (e.g. solar cells).

2.5 WSN Topologies

The four wireless sensor network topologies discussed here are [21]: Peer-to-peer (also called point to point), Star, Tree and Mesh.

2.5.1 Peer-to-Peer Networks: In Peer-to-peer network, without going through the centralized communications hub, each node is allowed to communicate directly with another node. In this case each device is able to function as both a "client" and a "server" to the other nodes on the network. An example of a peer-to-peer network is shown in Figure 2.3.

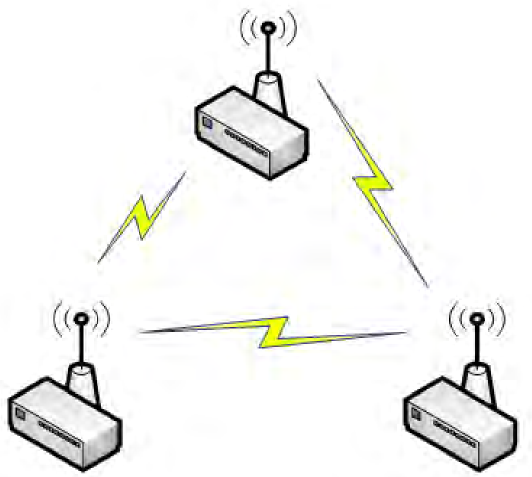


Figure 2.3 Peer-to-Peer Network.

2.5.2 Star Networks: In star network, nodes are connected to a centralized communications hub and all of the communications must be routed through the centralized hub. Nodes cannot communicate directly with one another. Each node is then a "client" while the centralized hub is the "server". A sample example of a star network is shown in Figure 2.4.

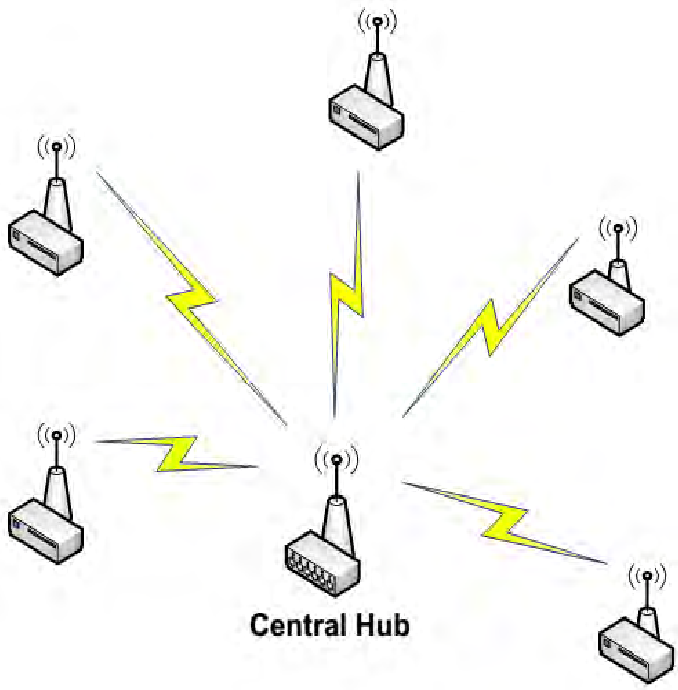


Figure 2.4 Star Network.

2.5.3 Tree Networks: Tree network use a central hub called a root node and it's the main communications router. One level down from the root node in the hierarchy is called the central hub and this lower level then forms a star network. The tree network consists of both star and peer-to-peer networking topologies and generally called a hybrid network topology. An example of a tree network is shown in Figure 2.5.

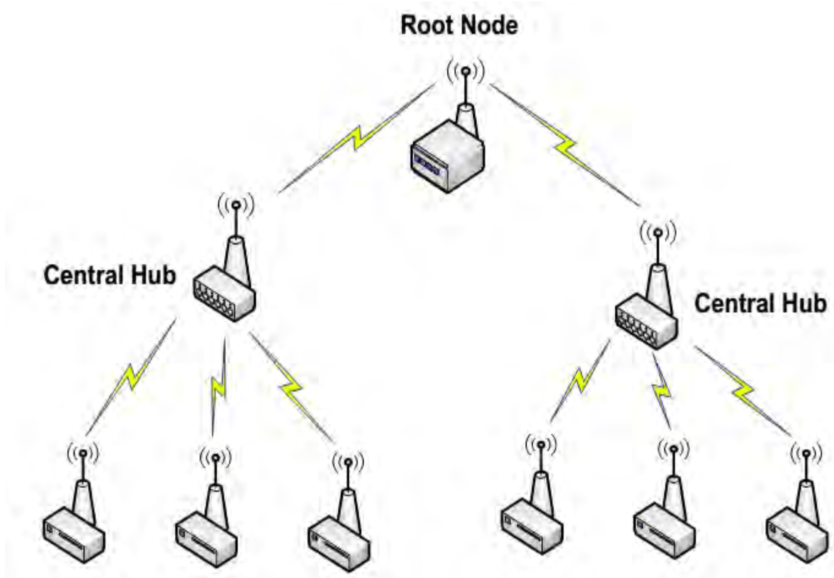


Figure 2.5 Tree Network.

2.5.4 Mesh Networks: Mesh network allow data to hop from node to node. So here each node is then able to communicate with each other until the routed data reaches the desired location. A sample example of a Mesh network is shown in Figure 2.6.

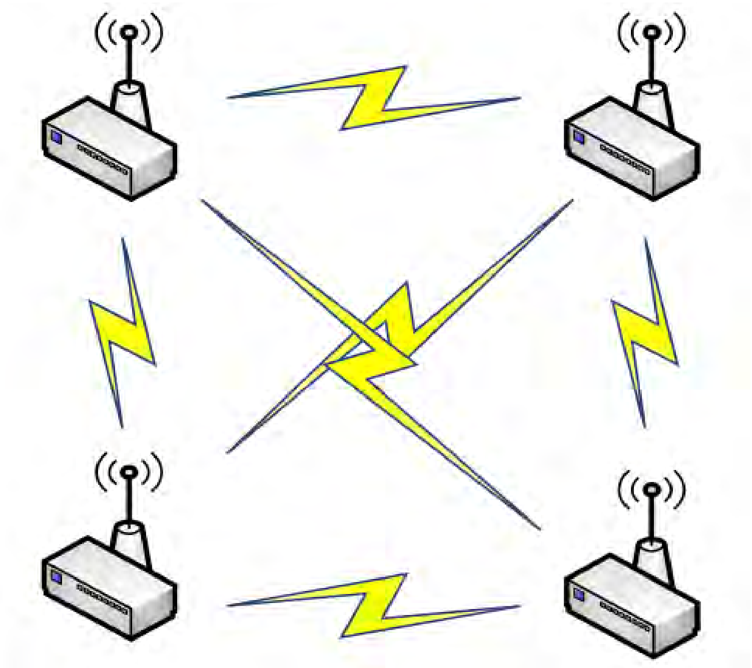


Figure 2.6 Mesh Network.

2.6 Types of Network Nodes

In general there are two types of network nodes [2], sensor nodes that mainly transmit their own sensor readings are also called the source nodes and relay nodes that mainly relay messages from other nodes towards the sink node. Due to mobility, node types may change during the lifetime of the network. Four types of communications (see Figure 2.7) that have to be supported by the MAC protocol:

1. Sensor node to sensor node communication: This type of communication is used for local operations, like clustering of the network, route discovery and security authentication communication.

2. Sensor node to relay node communication: This type of communication is used to transfer sensed data from sensor node to relay node and it is unicast in nature.

3. Relay node to sensor node communication: Request for data and signaling messages is done through relay node to sensor nodes and normally this type of communication is multicast.

4. Relay node to relay node communication: Relay node to relay node communication is done to forward data towards the sink node. This type of communication is often unicast.

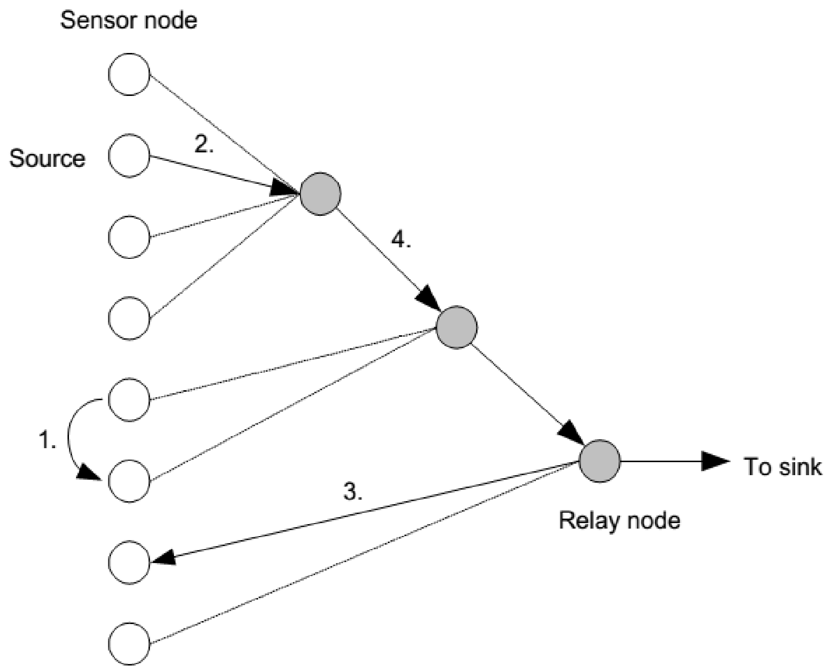


Figure 2.7 Types of network nodes and their communication.

2.7 Sensor Network Scenario

There may have different sensor network scenarios [20] based on the types of sources and sinks, number of sources and sinks and nature of mobility of sources and sinks. Network scenario may also vary depending on hop distance from sources to sink(s).

2.7.1 Types of Sources and Sinks

A source is an entity in the network that can provide information and is called a sensor node; it could also be an actuator node that provides feedback about an operation.

Whereas a sink is the entity where information is required. There are three probable options for a sink: it could be in the sensor networks as such other nodes and be just another sensor/actuator node or it could be an entity outside this network (Figure 2.8 shows sources and main types of sinks in direct communication).

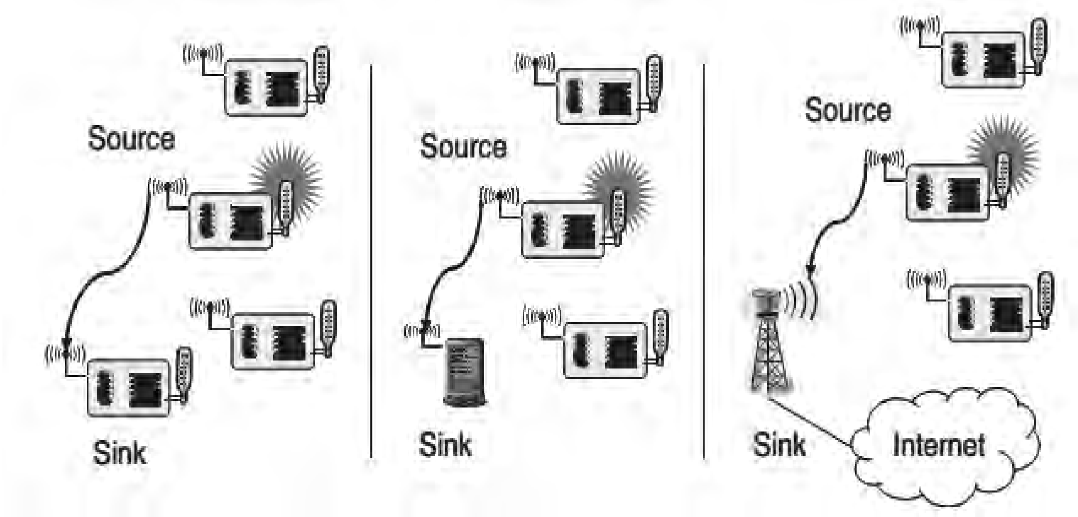


Figure 2.8 Three types of sinks in a very simple, single-hop sensor network.

For the second case, the sink could be a handheld device or PDA to interact with the network. It could also be a gateway to another larger network and it also could be the Internet from where the actual request for information comes from and indirectly connected to such a sensor network.

2.7.2 Single-hop versus Multihop Networks

Due to the power limitations of the radio communication, communication between sender and receiver is possible for a feasible distance and so that in WSNs, it could not be always possible of direct communication between source and sink. To overcome such limited distance problem, one or more intermediate nodes between source and sink are used as relay node. This concept is called multi-hop because packet travels multiple hops from the source to the sink and is illustrated in Figure 2.9.

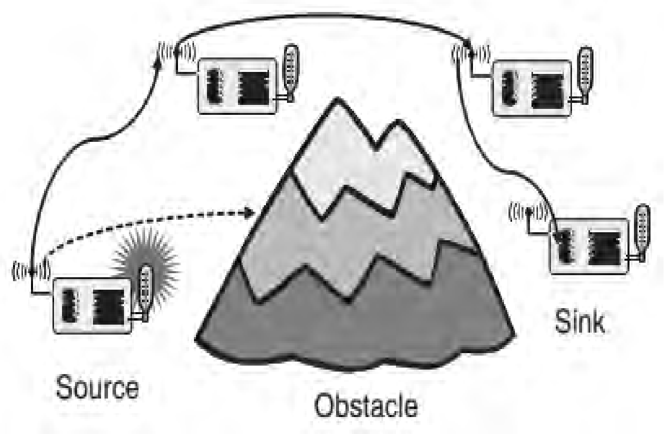


Figure 2.9 Multihop networks: since the direct communication is not possible due to the distance and/or obstacles, so the multihop communication can circumvent the problem.

2.7.3 Multiple Sinks and Sources

In real network scenario generally there have multiple sources single/multiple sinks present. In case of single sink, multiple sources should send information towards the single sink and in case of multiple sinks, multiple sources should send information to multiple sinks and all or some of the information may reach to all or some of the sinks.

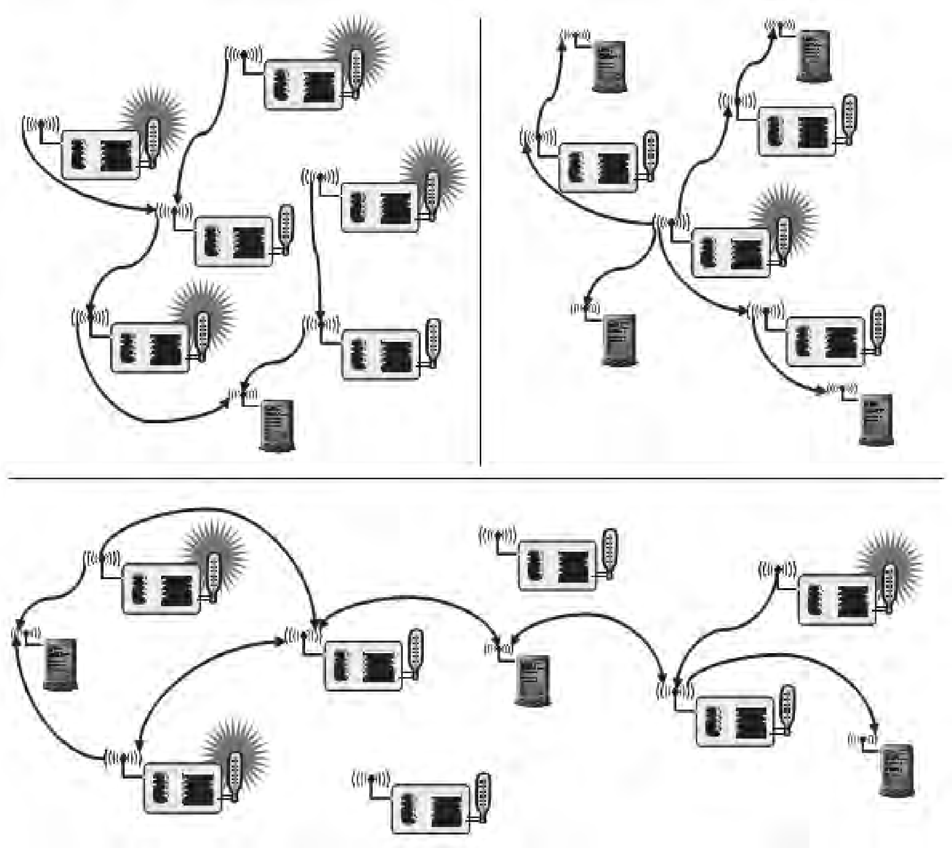


Figure 2.10 Multiple sources and/or multiple sinks. Note how in the scenario in the lower half, both sinks and active sources are used to forward data to the sinks at the left and right end of the network.

2.8 Types of Mobility

One of the main characteristics of wireless communication is its ability to support the mobility of the participants. In wireless sensor networks mobility can be three types [20]: Node mobility, Sink mobility and Event mobility.

2.8.1 Node Mobility

For many applications of wireless sensor networks, wireless sensor nodes can be mobile. For example, workers in disaster recovery scenes [22] and oil extraction and refinery areas [23] can carry sensing devices to avoid dangerous situations; biomedical sensor nodes can be attached to the bodies of patients [24] and nurses [25] to monitor their activities; mobile sensor nodes can also be employed to report or debrief soldiers the events encountered during a mission [26]. In case of nodes mobility sensor network should be able to recognize and function correctly.

2.8.2 Sink Mobility

The information sinks can also be mobile (See Figure 2.11). For example, a user requested information via a PDA while walking in an intelligent building. Despite the movement of the requester and with the assistance of the requester, the network must be able to provide the data to the requester [27].

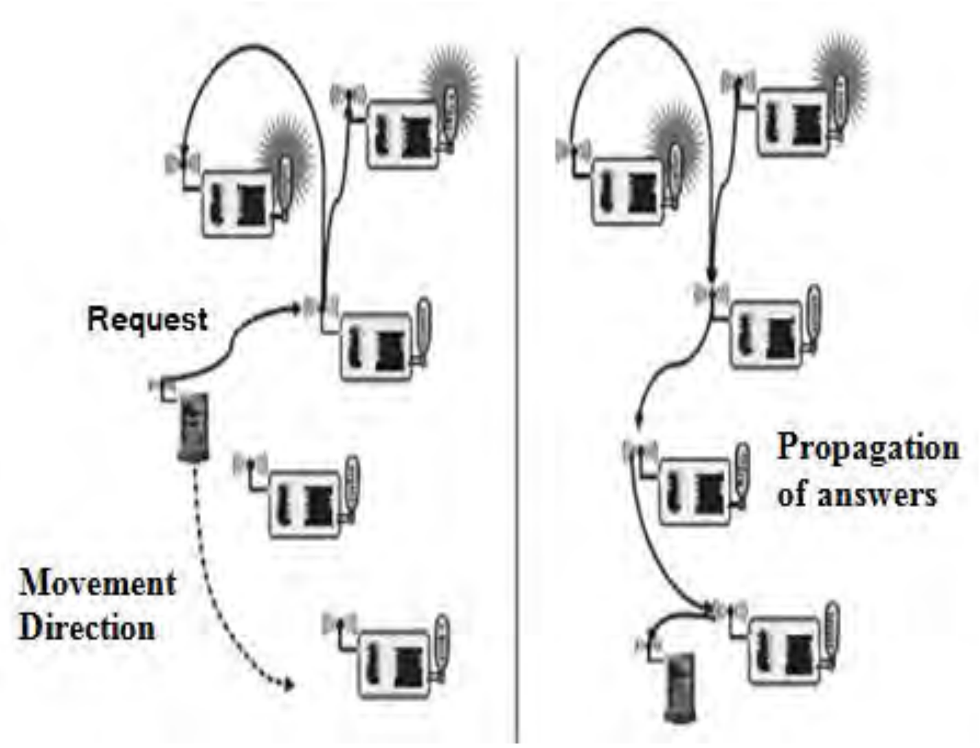


Figure 2.11 A mobile sinks moves through a sensor network as information is being retrieved on its behalf.

2.8.3 Event Mobility

In some applications like event detection and tracking applications, the cause of event or the object to be tracked can be mobile. In such cases it is important that the observed event is covered by a sufficient number of sensors all the time. Similar event detection application scenario is presented in Figure 2.12, where the task is to detect a moving elephant and to observe it as it moves around. It is done by the use of frisbee model [28], which also describes algorithms for handling the “wakeup wavefront” i,e sensors will wake up around the object and busy in the higher activity region to observe the object and then go back to sleep.

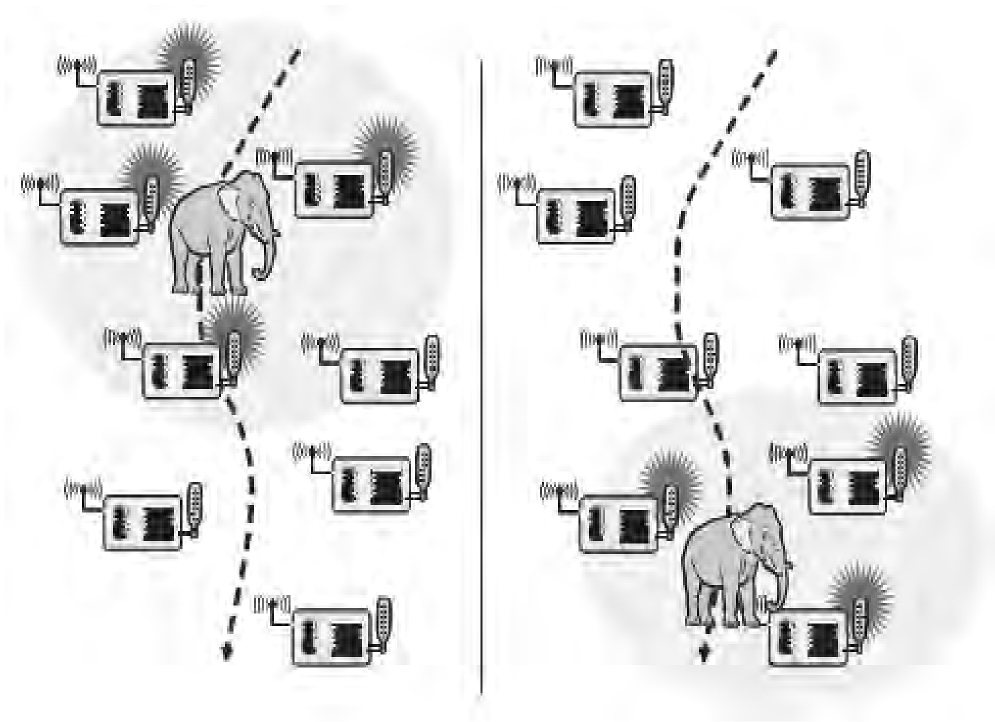


Figure 2.12 Sensor nodes area of detecting an event – an elephant that moves through the network (dashed line indicates the path followed by the elephant and shaded ellipse is the activity area passing or following by the elephant) [29].

2.9 Gateway Concepts

Sensing data from network by sensor nodes is not the main objectives of using wireless sensor networks but provide sensing data to other information device is must. Hence the concept of gateway arises [20].

2.9.1. The Need for Gateways

For practical applications sensor networks has to be able to provide sensing data with other information devices. This information device can be a PDA holding by a user during moving in the coverage area of the network or with a remote user and trying to interact with the sensor network via the Internet (for example, a user accessing the Internet and while traveling may need to read the temperature sensors in one's home). Figure 2.13 shows this networking scenario.

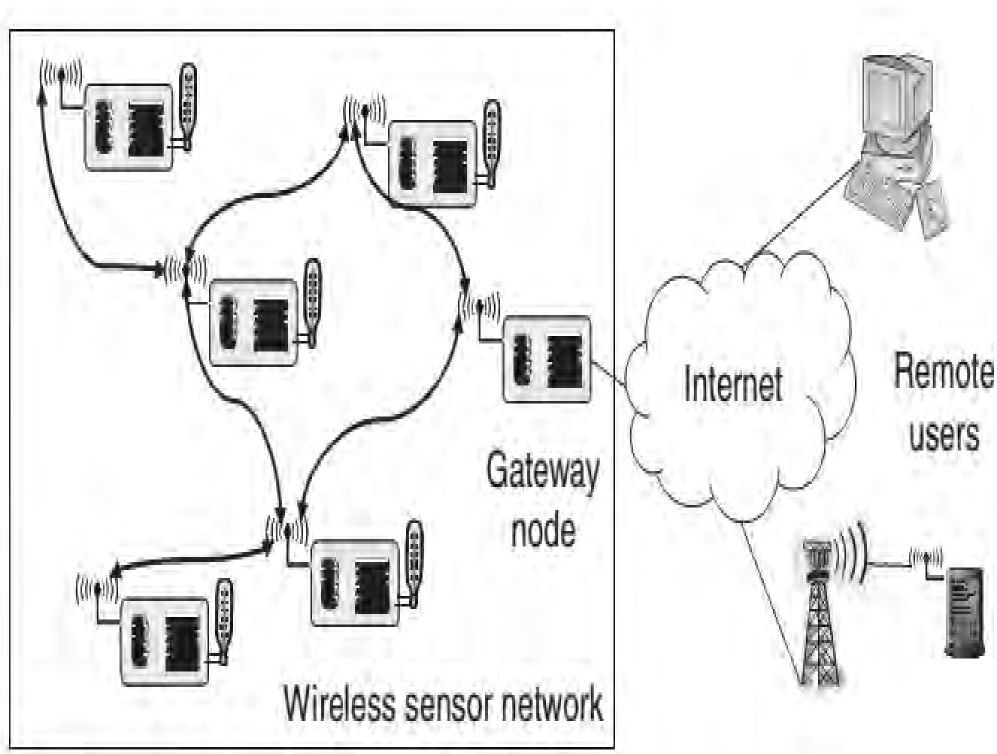


Figure 2.13 A wireless sensor network with a gateway node, enabling access to remote clients via the Internet.

To do the above job, the WSN has to be able to exchange data with such kind of mobile device or with a gateway, which provides the connection to the Internet. This can be done either by using a radio transceiver as used in the WSN with the mobile device/the gateway or by using some nodes in WSN that support standard wireless communication technologies.

2.9.2 WSN to Internet Communication

When a sensor node in the WSN wants to deliver an alarm message to some Internet host needs to find a gateway among one or many gateways in the WSN, and if not all the Internet hosts are not reachable via each gateway or if some gateway node should preferred for a given destination host. Then how to handle several gateways, each capable of IP networking, maintaining communication among them and find a suitable one for a given destination host? Possible solution of the above problem is to build an IP overlay network on top of the sensor network [30]. Figure 2.14 shows an example of WSN to Internet communication, where a sensor node needs to deliver a message to Alice's desktop resides in the Internet.

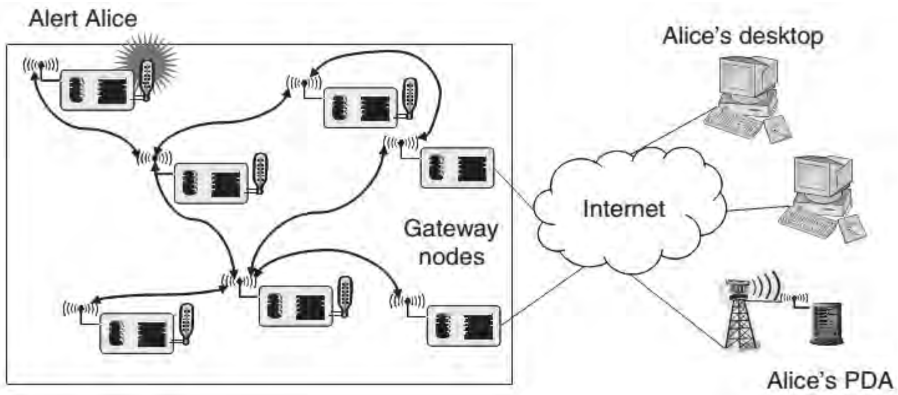


Figure 2.14 An event notification to "Alice" needs decisions about the gateway choice, mapping "Alice" to a concrete IP address, and translating an intra-WSN notification message of an event to an Internet application message.

How does a sensor node map a semantic notion "Alert Alice" to a concrete IP address? Although the sensor node does not need to process the IP protocol, at least it includes sufficient information in its own packet; the remaining task is done by gateway by extracting this information and translating it into IP packets. Here the gateway node in a sense has to perform the similar tasks of a network address translator (NAT) device [31].

2.9.3 Internet to WSN Communication

Figure 2.15 shows the case, where a host in the Internet trying to access services of a WSN. It will be simple if the requesting mobile terminal is equipped with a WSN transceiver and able to directly communicate with the WSN and also has all the necessary protocol components.

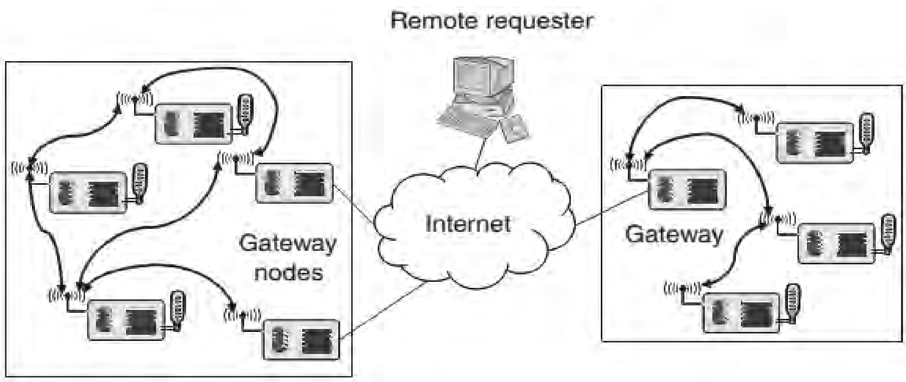


Figure 2.15 Requesting sensor network information from a remote terminal entails choices about which network, which gateway of that network to address, and how and where to adapt application-level protocol in the Internet to WSN-specific protocols.

In another case, if the requesting terminal is far away and requesting the service of WSN requiring the assistance of an application level gateway node which uses an application-level protocol. The gateway translates this request into the proper intrasensor network protocol interactions. The remote located requesting terminal also using an application-level protocol can send a properly formatted request to the gateway. The gateway can then coordinate the data-centric data exchange within the network behind an identity-centric exchange used in the Internet. The use of application-level protocol by remote requester and gateway is more suitable for communication over the Internet than the use of actual sensor network protocols. A general-purpose service interface to WSNs is commonly accepted [32] and will have a clear impact on how to access WSN services from afar as well.

2.9.4 WSN Tunneling

In addition to the above described interactions between WSN and Internet terminal, the gateway terminal can also act as simple extensions of one WSN to another WSN to build a larger-virtual WSN by tunneling all protocol messages between these two networks and simply using the Internet as a transport network [32]. Figure 2.16 shows such a scenario.

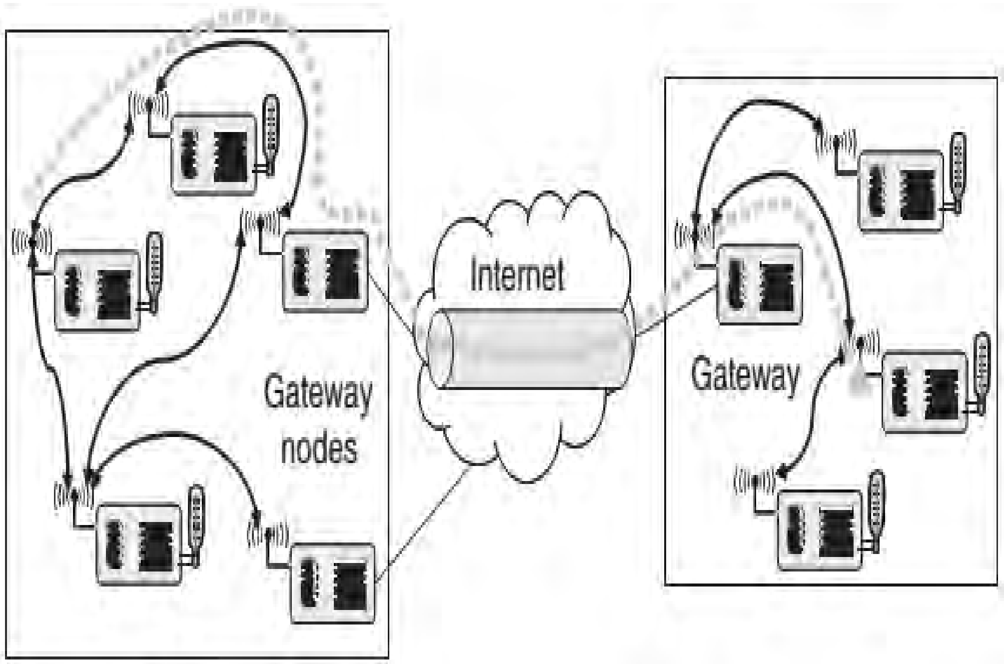


Figure 2.16 Connecting two WSNs with a tunnel over the Internet.

Such tunnels need not to be like fixed network connections, mobile nodes can also be considered as intermediate interconnections of WSNs [33].

2.10 WSN Application Examples

Depending on the sensing and actuating capabilities of different sensors technology which have the ability to sense different physical parameters like: acoustic, temperature, pressure, humidity, vibration and mechanical stress etc., in combination with computation and communication abilities, many different kinds of applications [20] can be constructed with very different types of nodes, even of different kinds within one application.

Disaster Relief Applications

Most disaster relief application is wildfire detection in which sensors are deployed over a wildfire, for example in a forest, from an airplane, where nodes are equipped with thermometers and can collectively produce a temperature map of the area with high temperature that can be accessed. Some other disaster relief applications are control of accidents in chemical factories and military applications, where sensors should detect enemy troops rather than wildfire.

Environment Control and Biodiversity Mapping

WSNs can be used to control the chemical pollutants in garbage dump sites. WSNs can also be used to control the environment by gaining an understanding about the number of plant and animal species that live in a given habitant.

Intelligent Buildings

In a building large amount of energy wastes by inefficient humidity, ventilation, air conditioning usage. Using WSN, a better high-resolution and real time monitoring of humidity, airflow, temperature and other physical parameters can increase the comfort level of inhabitants and reduce the energy consumption. In seismically active zones sensor nodes can also be used to monitor the mechanical stress levels of buildings.

Facility Management

WSNs can be used to facility management in a large company sites by checking which person is allowed to enter which areas of a larger company site. Similar examples is the detection of intruders, for example of vehicles that pass a street outside of normal business hours. WSN could track such a vehicle’s position and alert security personnel. WSNs can also be used in a chemical plant to scan for leaking elements.

Machine Surveillance and Preventive Maintenance

Sensor nodes could be used in difficult-to-reach areas of machinery, where they can detect the unexpected vibration patterns that indicate the need for maintenance.

Precision Agriculture

By placing humidity/soil composition sensors into the field of agriculture allows precise irrigation and fertilization.

Medicine and Health Care

The use of WSN in health care applications is a potentially very beneficial by directly attaching sensors to patients and could be monitored by doctors resides in away.

Logistics

If sensors are equipped with goods or individual parcels, allows a simple tracking of these objects during transportation.

Telematics

Sensors could be embedded in the streets or roadsides called “intelligent roadside”, to gather information about traffic conditions at a much finer grained resolution, it could also be used to interact with the cars to exchange traffic jams ahead or danger warnings about road conditions.

2.11 Types of Applications

Many of the above discussed applications share some basic characteristics. In WSNs, the actual nodes that sense data is called source and nodes where the data should be delivered to is called the sink node. The interaction patterns between sources and sinks show some common patterns. The most relevant ones are [20]:

Event Detection

When sensor nodes detect the occurrence of a specified event, should report to the sink(s) nodes.

Periodic Measurements

Depending on the application requirements, sensor nodes periodically should report the measured data to the sink node after a certain duration of time.

Function Approximation and Edge Detection

WSNs can be used to approximate a physical value like the changing of temperature from one place to another can be regarded as a function of position. One similar application example is to find the border of the actual fire by detecting the isothermal points in a forest fire.

Tracking

Since the source of an event can be mobile for example, an intruder in surveillance scenarios; WSNs can be used to report updates of the event source's position with speed and direction to the sink.

2.12 Challenges for WSNs

To accommodate the following characteristics with new mechanisms is the major challenges [20] of the vision of wireless sensor networks.

2.12.1 Characteristic Requirements

The following characteristics are common among most of the wireless sensor networks application examples discussed above.

Type of Service

In WSN, it is expected to provide purposeful information and/or activities about a given task.

Quality of Service

Providing quality of service support is a challenging issue due to highly resource constrained nature of sensor nodes, unreliable wireless links and harsh operation environments. In some cases, very high reliability requirements exist; in some cases, low latency is important; in some other cases, only occasional delivery of a packet can be more than enough. So, quality of service is also application dependent.

Fault Tolerance

During network lifetime, there may arise some faults like: node damage, run out of energy, interruption of communication among two or more nodes or any others. In all cases, WSN as a whole should be able to tolerate such faults and function accurately.

Lifetime

Since nodes in WSN are small, cheap and in most cases are deployed in an easily unreachable area, so nodes will have to rely on a limited supply of energy. Hence, the lifetime of a WSN becomes prominent.

Scalability

Since, WSNs may contain a large number of sensor nodes, so the network architectures and protocols must be able to scale these numbers.

Wide Range of Densities

The number of nodes per unit area may vary among applications, even within a given application. The network should adapt to such variations.

Programmability

Nodes should be flexible to react on changes in their tasks. So nodes should be programmable because their programming should be changeable during operations when new tasks become important.

Maintainability

As both the sensor networks and it may changes, the system has to adapt. So, it is important to monitor and take necessary actions regarding the changes. To guarantee the expected quality and to ensure its extended operation, it could also be able to interact with external maintenance mechanisms [52].

2.12.2 Required Mechanisms

To realize the above requirements, new architectures, protocol concepts and innovative mechanisms for a communication network have to be found. Some of the mechanisms are listed below:

Multihop Wireless Communication

Since the use of intermediate nodes as relays in long distance communication can reduce the total required power, so for many WSNs multihop communication will be a necessary ingredient.

Energy-Efficient Operation

In order to increase the lifetime of a WSN, system should accommodate necessary techniques for energy-efficient operations among nodes.

Auto-Configuration

In WSNs, sensor nodes will have to configure most of its operational parameters autonomously without the need for external configuration.

Collaboration and In-network Processing

In some WSN applications, data of single node is not enough to take decisions but data of several sensors have to process to take the decisions. In such cases, sensors have to collaborate to detect an event. An example is to determine the highest or the average temperature within an area and to report that value to the sink.

Data Centric

Unlike traditional communication networks where communication is address centric, in WSN it is important to get the answers and values of the query, not which node has provides them. Hence, the designing of architecture and communication protocols is important for efficient implementation of data-centric paradigm rather than address-centric paradigm.

Locality

Since sensor nodes in WSNs are very limited in resources, so it should attempt to limit the processing with the hope that this will allow the network to use a large numbers of nodes without having to rely on huge processing at each single node.

Exploit Trade-offs

During system/protocol design, WSN will have to rely to a large degree of inherent trade-offs between contradictory goals and at runtime as well. Example for such trade-offs is- higher energy consumption allows higher accuracy.

2.13 Mobile Ad Hoc Networks versus Wireless Sensor Networks [20]

Even though there have some similarity between Mobile Ad-hoc NETworks (MANET) and WSN, there are some major differences between the two concepts ensures a distinction between them and confirms the need for separate attention for research efforts.

Applications and Equipment

Application area of MANET is somewhat different than WSN. Equipment used between this two networks are also different. In MANET, equipment used in the terminal can be powerful (usually a laptop or PDA) with a comparably large battery than the equipment used in WSN.

Application Specific

In WSN many different application scenarios becomes possible and it is unlikely that there will be a "one-size-fits-all" solution for all these potentially very different possibilities. For example in WSNs node densities in the application areas may vary from very sparse to very dense, which demands adaptive protocols. Although this diversity is present in MANET, but not as large as in WSNs.

Environment Interaction

MANETs is human-driven forms of networks and are used to support more conventional applications (Web, voice and so on) with well understood traffic characteristics. On the other hand WSNs have to interact with the environment and their traffic characteristics are very different are likely to exhibit very low data rates over a low timescale, but can have very bursty traffic when something happens.

Scale

In WSNs number of entities may be so large (thousands or perhaps hundreds of thousands) than ad-hoc networks. So requiring different and more scalable solutions.

Energy

Although in both cases, energy is a scarce resource but recharging or replacing a WSNs node battery is much more difficult than in MANETs.

Self-Configurability

Self-configurability is required for both types of networks but the differences in traffic, energy trade-offs could require different solutions.

Dependability and QoS

In MANETs each individual node should be fairly reliable than in WSNs and the QoS service issues in MANETs are dictated by traditional applications than in WSNs where the energy explicitly must take into account.

Data Centric

Normally the operation in MANETs is address-centric. This types of networks are typically centered on the transfer of data between two specific devices, each equipped with one network address. On the other hand redundant deployment will make data-centric protocols attractive in WSNs.

Simplicity and Resource Scarceness

Since sensor nodes are simple and energy supply is rare, the operating and networking programs must be kept simpler compared to today's laptop and desktop computers which are normally used in MANETs.

Mobility

In both network type's mobility problem can be happen if nodes are moving around, but in WSNs two additional aspects of mobility are to be considered. First, Since WSNs can be used to detect and observe a physical phenomenon in an application area. What happens if the phenomenon moves about? In WSN, this mobility problem is to be considered. Second, if sink node(s) in the network moves around which can cause some difficulties for protocols that operate efficiently in fully static scenarios should also take into consideration.

2.14 Summary

A detail overview of wireless sensor networks are discussed throughout this chapter. This chapter covers different sensor nodes property that constitutes sensor networks. The integration of WSNs in large network contexts, for example, to allow Internet-based hosts to access WSN services are also discussed in this chapter. However, it is very difficult to design of a uniform, general-purpose service interface for the large diversity of WSNs applications and network types.

CHAPTER 3

FUNDAMENTAL ISSUES OF MAC DESIGN

3.1 Introduction

In the protocol stack of any network architecture, Medium Access Control (MAC) is one of the most important layer/sublayer. The performance of any network is largely dependent on the performance of the MAC layer. MAC layer protocol defines the functionalities of the MAC layer. To get a better network performance from any network use of a suitable MAC protocol is important. So it is crucial to design a good MAC protocol. Before designing a good MAC protocol it is very important to know the role of MAC layer in network architecture, multiple access technique in MAC layer and the design constraints for wireless MAC protocol. Moreover, different issues and challenges that should also be considered before designing a good MAC protocol. In this chapter it has been tried to discuss about the fundamental issues of MAC design and different aspects related to MAC layer. This chapter also includes a detail overview of some schedule-based MAC protocols like: EMACS, LMAC, AI-LMAC and MC-LMAC. This is because these protocols are closely related to the proposed protocol.

3.2 Role of MAC Layer in Network Architecture

Within the OSI reference model, data link layer (DLL) is considered as two sublayers. The upper sublayer is responsible for flow and error control is called the logical link control (LLC) layer. Flow control is used to control the rate of transmission to prevent a slow receiver from being overwhelmed by faster transmission and error control ensures correctness of transmission and take necessary actions in case of transmission errors. The lower sublayer is mainly responsible for multiple access resolutions called the MAC layer [34]. The role of MAC layer is to control the access of a shared medium among all the networking stations in such a way that certain application specific performance requirements are satisfied. Some of the common performance requirements are delay, throughput, fairness and energy efficiency [20]. For the dedicated link, lower sublayer is not needed.

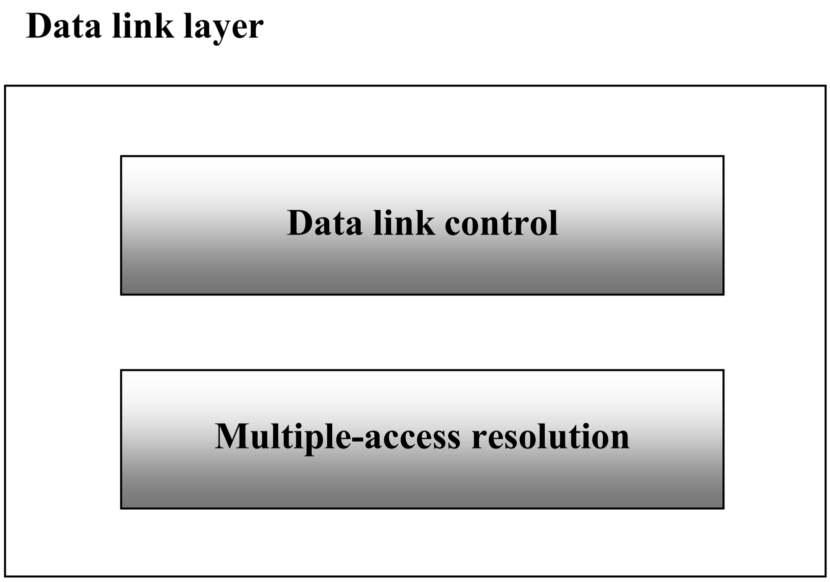


Figure 3.1 Data link layer divided into two functionality-oriented sublayers [34].

3.3 Multiple Access Techniques in MAC Layer

In wireless communication system, the transmission medium is broadcast in nature. So a wireless node cannot transmit its data frame haphazardly. Several multiple access schemes are used to control the access of the shared medium to share simultaneously a finite amount of radio spectrum to achieve high capacity by simultaneously allocating the available bandwidth to multiple users. It must be done without severe performance degradation for ensuring the quality of the communication.

In wireless communication systems, frequency division multiple access (FDMA), time division multiple access (TDMA), and code division multiple access (CDMA) are the three major multiple access techniques used to share the available bandwidth. Space division multiple access (SDMA) is another multiple access techniques, in which, the medium is controlled by controlling the radiated energy [35].

3.3.1 Frequency Division Multiple Access (FDMA)

In FDMA technique, total available bandwidth is divided into multiple channels and each wireless stations are allowed to send its data all the time through its assigned channel. Different station send data through different channel. Each station uses a bandpass filter to capture the desired transmitted frequency. Channels are separated by strips of unused bandwidth called guard bands to prevent signal from overlapping and interference.

3.3.2 Time Division Multiple Access (TDMA)

TDMA systems divide the radio spectrum into timeslots, and all the wireless stations share the bandwidth of the channel in time. In this system each user is allowed to use a specific timeslot in a time frame (total number of slots constitutes a time frame) and is allowed to send its data during that timeslot in each time frame. This timeslot assignment can be done in fixed or demand basis. The main problem of TDMA system is time synchronization among multiple channels, i,e how each station knows the starting and ending of a slot. To denigrate this synchronization problem unused time called guard time, is used between two consecutive timeslots.

3.3.3 Code Division Multiple Access (CDMA)

In CDMA systems, each wireless stations are allowed to operate independently with no knowledge of the other stations using the whole bandwidth and may transmit data simultaneously without time sharing. To send data each station use its own pseudorandom codeword which is approximately orthogonal to all other codewords using by other stations. For the detection of the data signal, the receiver needs to know the codeword used by the transmitter to perform a time correlation operation to detect only the specific desired codeword.

3.3.4 Space Division Multiple Access (SDMA)

SDMA controls the radiated energy for each stations in space by using spot beam antennas instead of omnidirectional antenna used in TDMA, FDMA or CDMA systems. Different regions can be served using the same frequencies (in a TDMA or CDMA system) or different frequencies (in a FDMA system). This technique mainly used in cellular mobile systems and satellite communication systems where, a satellite can reuse the same frequency to cover many different regions of the earth surface.

3.4 Design Constraints for Wireless MAC Protocols

Since the transmission in wireless communication is broadcast in nature and if two nodes are out of reach, so they cannot hear each other. Then the basic access mechanisms gives rise to the well-known problem- called hidden-terminal problem and exposed-terminal problem [36].

3.4.1 Hidden-Terminal Problem

In wireless network, a hidden terminal is a wireless station that is out of the range of another wireless nodes or collection of nodes. The hidden-terminal problem occurs mainly for the class of carrier sense multiple access (CSMA) protocols, where a node sense the medium before starting to transmit a packet. If the medium is found to be busy, the node waits and sense the medium at a later time to avoid a collision. Consider the Figure 3.2, where we have three nodes A, B and C that are arranged such that node B and node C both are within the transmission range of node A, but B and C cannot hear each other. Assume that B starts to transmit a packet to A and at the same time or some time later node C also decides to start a packet transmission to node A. A carrier sensing operation by C shows an idle medium since C cannot hear B's signals. When C starts its packet transmission, the signals collide at A and both packets are useless thus leads to useless collisions and results decrease of network throughput.

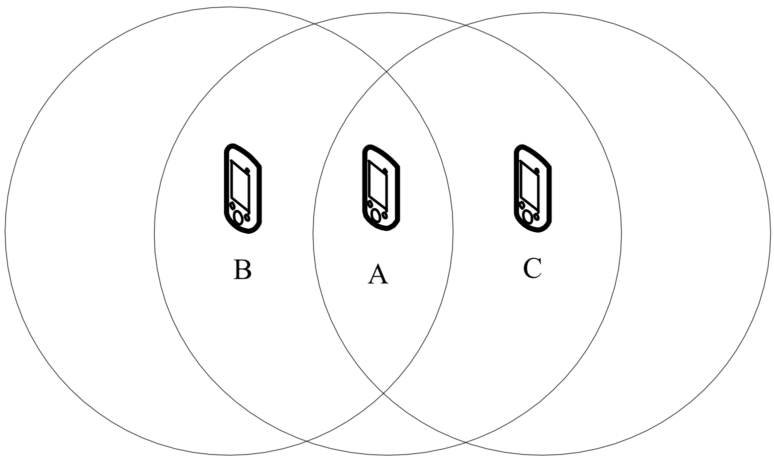


Figure 3.2 Hidden-terminal scenario where circles indicate the transmission and interference range. B, A, C

3.4.2 Exposed-Terminal Problem

In the exposed-terminal scenario, A transmits a packet to B, and some moment later, C wants to transmit a packet to D. Although this would be theoretically possible since both B and D would receive their packets without distortions, the carrier sense operation performed by C suppresses C's transmission thus leads to needless waiting and bandwidth is wasted. Hence, it is said that transmission from node A to node B exposed to node D.

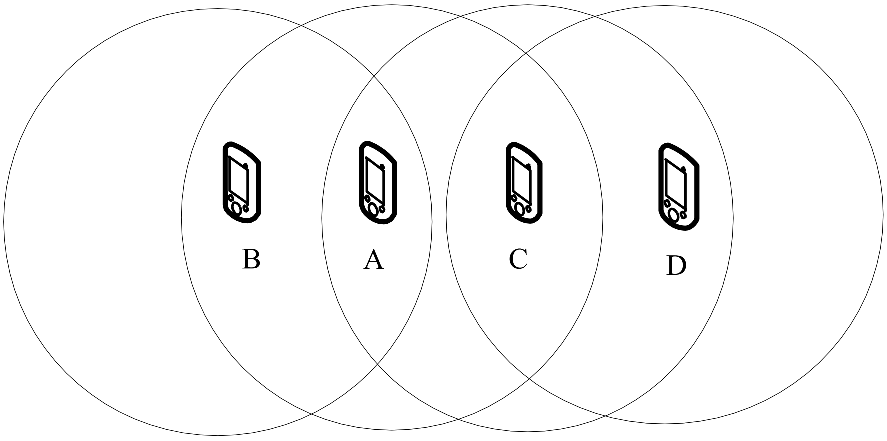


Figure 3.3 Exposed-terminal scenario where circles indicate the transmission and interference range. B, A, C, D.

Two solutions to the hidden-terminal and exposed-terminal problems are busy-tone solutions [36] and the RTS/CTS handshake used in the IEEE 802.11 WLAN standard [37] and first presented in the MACA [38] protocol.

3.5 Classification of Wireless MAC Protocols

According to the network architecture, wireless MAC protocols are classified into two broad categories, distributed and centralized. Based on mode of operation further classification can be done into random access protocols, guaranteed access protocols and hybrid access protocols. Based on the intelligence at the base station, the hybrid access protocols are classified into random reservation access (RRA) protocols and demand assignment (DA) protocols [39].

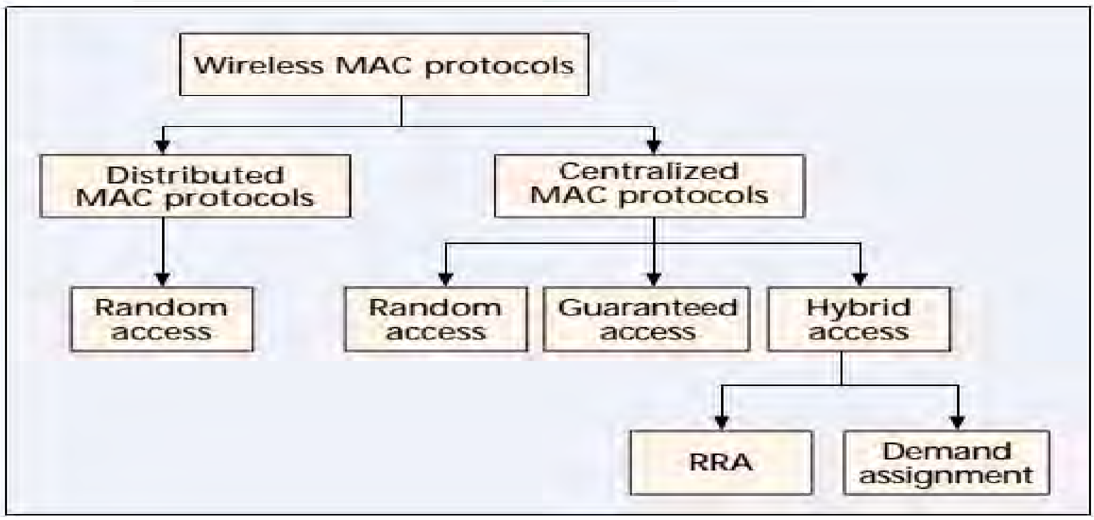


Figure 3.4 Classification of wireless MAC protocols.

3.5.1 Distributed MAC Protocols

To transmit data packets, with the exception of ALOHA, all distributed MAC protocols first sense the carrier by listening to the medium to detect any ongoing transmission. If it find the channel in idle then it send the data packet, otherwise it waits and sense again at a later time. This carrier sense operation is performed to avoid collisions.

3.5.1.1 Distributed Random Access Protocols

In distributed random access protocols, Anode that has data to send, for a certain duration of time it senses the channel before sending. If the channel is busy, the node waits a random period of time to transmit. If the channel is idle, the node makes an attempt to acquire the channel. After acquiring the channel transmission of the data packet will happen. If the acquisition attempt results in a collision, then the colliding nodes try to solve the problem of collision in an orderly manner. After packet transmission the destination station acknowledge the packet receiving.

3.5.2 Centralized MAC Protocols

In centralized MAC protocols the complexity and arbitration are moved into the base station. It has the explicit control on whom and when they access the medium. Since the base station (BS) is located at the center, it is assumed that all nodes can talk to and hear from the BS. Therefore, the hidden and exposed nodes do not exist. However, all communications must go through the base station.

3.5.2.1 Random Access Protocols

In random access protocol, nodes compete to access the medium. Successful packet delivery will happen when only one node makes a transmission attempt and when multiple nodes make a transmission attempt a collision occurs. Nodes resolve the collisions in an orderly manner according to rules defined by the contention resolution algorithm (CRA). ALOHA is a classic example of a random access protocol and it was the first proposed protocol for packet radio networks.

3.5.2.2 Guaranteed Access Protocols

In a guaranteed access protocol, nodes access the medium in a round-robin fashion. These protocols can be implemented in two ways. One is a master-slave configuration, where the master node polls other nodes and the node sends data in response to the poll. This type of protocols are called polling protocols. Another one is to operate in a distributed manner by exchanging tokens and the station with the token can transmit data.

After transmitting data, each station passes the token to the next station. This type of protocols are called token-passing protocols.

3.5.2.3 Hybrid Access Protocols

Hybrid access MAC protocols combined the best features of the above two protocols to make a more efficient MAC protocols. Most of the hybrid access MAC protocols are based on request-grant techniques, where each node sends a request to the base station using a random access protocol, indicating how much time or bandwidth is required to send its data. For actual data transmission, the BS then allocates and sends a concession to the node pointing that timeslot. Based on the intelligence at the base station, hybrid access protocols are classified into RRA protocols and DA protocols.

3.5.2.3.1 Random Reservation Access protocols

In a RRA protocol, the BS has implicit rules for reserving upstream bandwidth. For example a rule is: a successful request results a periodic reservation of an upstream slot.

3.5.2.3.2 Demand Assignment protocols

In a DA protocols, the BS controls upstream data transmissions according to their quality of service requirements, collects all the requests from the nodes and uses scheduling algorithms to make bandwidth allocations.

3.6 Issues in Designing a MAC Layer Protocol

To design a good MAC protocol the various design issues for sensor network environment which should be considered are [40] [41] [42]:

3.6.1 Energy Efficiency

The sensor nodes are battery operated and it is often very difficult to change or recharge batteries for these sensor nodes. On certain occasions, it is beneficial to replace the sensor node rather than recharging them.

3.6.2 Latency

The second one is latency and the latency requirement basically depends on the application. In most of the sensor network applications, to take the appropriate actions immediately, the detected events must be reported to the sink node in real time.

3.6.3 Throughput

Throughput requirement also varies with different applications. Some of the sensor network application requires information sampling with fine temporal resolution. In such sensor network applications it is better that sink node receives more data.

3.6.4 Fairness

In many sensor network applications when bandwidth is limited, it is necessary to guarantee that the sink node fairly receives data from all the sensor nodes. However among all of the above aspects the efficient use of energy and throughput are the major aspects. Energy efficiency can be increased by minimizing the energy wastage.

3.6.5 Scalability

Scalability is another important criterion in designing a MAC protocol. The sensor network should adapt to the changing network size, topology and node density. Also some nodes may join, some may move to different locations and some nodes may die overtime. A good MAC protocol should accommodate these changes to the network.

3.6.6 Bandwidth Utilization

Bandwidth available for communication is limited. While designing a MAC protocol, the control overhead must be kept minimal. Bandwidth efficiency can be defined as the ration of the used bandwidth for actual data transmission to the total available bandwidth. A MAC protocol must try to maximize the bandwidth efficiency.

3.6.7 Synchronization

Synchronization in access to medium must be maintained among the wireless nodes. Synchronization is very important for bandwidth efficiency. A MAC protocol should not exchange excessive control packets in achieving synchronization.

3.6.8 Error Prone Shared Broadcast Channel

A node should be given access to the medium only when its transmission does not affect any ongoing data transmission. Since wireless medium is broadcast in nature, collisions among multiple packets may take place. A MAC protocol should grant anode access to wireless channel in such way that overall collisions are minimized.

3.6.9 Lack of Central Coordination

In ad hoc network, nodes do not have any central coordinators. Therefore, Access to medium must be handled in distributed manner. This may take some additional control overheads which could be the cause of system throughput degradation.

3.6.10 Quality of Service (Qos)

Nodes in infrastructure less WLAN may be mobile with time. In mobile environment providing proper quality of service, i.e., throughput, end to end delay etc. is very difficult. A MAC must take the mobility of the wireless medium and nodes into consideration. Services for the real time application must be addressed properly.

3.6.11 Mobility of Nodes

Nodes in ad hoc network are mobile most of the time. If the node mobility is very high, control information among the nodes may be wasted. Thus the throughput of the system would be decreased. A MAC protocol design should take this mobility factor into consideration that the performance of the wireless network system is not significantly affected by node mobility.

3.6.12 Traffic Adaptivity

In many sensor network applications, where the rate of data flow through the network is not fixed and may change abruptly. So the MAC layer should take the necessary actions to that rapid change of data flow for better performance.

3.6.13 Complexity

During the MAC protocol design it should also take under consideration that the algorithm should not be so complex that it requires plentiful resources for sensor node and becomes costly.

3.7 MAC Layer Properties for WSN

In WSNs, because of sensor nodes, in a limited coverage area, a large number of sensor nodes are deployed to cover the target area successfully. The main objectives of the WSN-MAC layer is to create the sensor network infrastructure in order to establish the links between sensor nodes, when nodes access the shared wireless medium to send/receive to meet the certain application requirements. The most important performance required for MAC protocols are [41] [43] high throughput, scalability, fairness, less delay, low overhead and energy efficiency. Among them for WSN-MAC protocol fairness, throughput, delay and scalability are the secondary requirements, Common objective of wireless sensor network is to maximize the network lifetime. So, energy efficiency is the primary requirement. Owing to large number of sensor nodes in sensor network, it is impractical to change the sensor nodes battery in a frequent manner. Another thing is the user of sensor networks are expect cost effective sensor network and hence demands a long lifetime of the sensor network becomes prominent. Therefore, the issue of energy conservation becomes more important.

3.8 Major Source of Energy Wastes

A transceiver of a node in a WSN can be in one of the four operational states [44]: In the transmitting state only the transmit part of the transceiver is active with its associated antenna, similarly in the receiving state only the receive part is active. Both states consume almost same energy. The receiver in the idle state receives nothing but consumes significant energy. On the other hand, in the sleep state, receiver consumes almost no energy, but results a deaf node.
The following situations cause major energy wastes [41] [45] [46]:

i. Collisions: A transmitted packet may collide with other packet and may be discarded due to bad channel conditions and needs to be retransmitted, as a result costs and latency increases.

ii. Overhearing: When a node picks up packets for which it is not destined causes overhearing.

iii. Idle Listening: The receiver in the idle state is ready to receive but nothing is receives, consumes significant energy.

iv. Protocol Overhead: For sending actual data, protocol may needs to send extra control frame like RTS, CTS, beacon signal, header, trailers which causes overhead and results extra energy consumptions.

v. Over Emitting: Over emitting problem is caused when message is transmitting but destination node is not ready to receive or sending rate is higher than receiving rate.

3.9 MAC Performance Metrics

To evaluate and compare the performance of energy aware MAC protocols, the following metrics are being considered by the research community [47].

(i) Energy Consumption per bit: Energy efficiency of sensor nodes can be defined as the energy consumed/bits transmitted. Unit of energy efficiency is joules/bit. Lesser value of energy efficiency refers to the better efficiency of a protocol in transmitting the information. This performance matrices value gets affected by all the sources of energy waste in wireless sensor network such as collisions, overhearing, idle listening, overhearing and control packet overhead.

(ii)Average Delivery Ratio: The average packet delivery ratio is the averaged value of number of packets received to the number of packets sent by all the nodes in the network.

(iii) Average Packet Delay: This is the average time taken by the packets to reach from source nodes to the sink node in the network.

(iv) Network Throughput: The network throughput means, total number of packets delivered at the sink nodes per time unit.

3.10 WSN Energy Efficient MAC Protocol

For WSNs, a large number of energy efficient MAC protocols are introduced in the literature. A few of them are discussed in this paper with their essential properties. These protocols can be classified into four categories [48].

3.10.1 Contention based MAC protocol

Contention-based MAC protocols allow many users to share the same radio channel without pre-coordination and are based on Carrier Sense Multiple Access (CSMA) technique and Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) technique. In case of contention based MAC protocols for a given transmit opportunity every neighbor node of the receiver can try its luck at the risk of collisions. Here the channel is accessed randomly so there is no need of coordination. Chance of collision can be decreased if a station follows the sense before use policy, but if channel is not free it will wait for random amount of time. This principle of CSMA is called “sense before transmit”.

Example of contention based MAC protocols are: S-MAC, T-MAC, U-MAC etc.

3.10.2 Schedule Based (TDMA) MAC Protocol

TDMA methods are based on scheduling in which an implicit collision-free scheme is used to transmit or receive data messages to every node and compared with contention-based MAC protocols, unique timeslots assignments is ensured. In TDMA clashing between adjacent nodes is avoided. By this collision avoidance, energy wastage due to collision is reduced. Without using of additional data message (overhead), TDMA can also solve the hidden terminal problem because in TDMA, neighboring nodes transmit at different timeslots.

In fixed TDMA the timeslots are assigned permanently to each terminal. Even though, this process ensures a simple implementation but the timeslot is wastage when there is no data to send.

In dynamic TDMA, timeslot assignment is done by a central station based on the requests from any terminal for slots. A separate signaling slot is used to broadcast terminal requests.

TDMA-based MAC protocols are μ-MAC, DEE-MAC, SPARE MAC. A survey on some schedule based MAC protocols were presented in [14].

3.10.3 Hybrid MAC Protocol

It is integration of contention-based MAC and TDMA-based MAC. Here all the advantages of these two methods are taken into consideration and make a better solution which is called hybrid MAC. In Hybrid MAC protocols the channels are divided into two parts, channel control packets are transmitted using random access policy whereas data packets are transmitted using scheduled access policy. The hybrid protocols can save higher energy and can provide better flexibility and better scalability in comparison with CSMA based and TDMA based MAC protocols. Some example of Hybrid protocols are IEEE 802.15.4, A-MAC and Z-MAC.

3.10.4 Cross layer MAC Protocol

So far the discussion about the WSN-MAC protocols covers; most of them are based on the single MAC layer design that means using only the useful information for MAC layer. But without considering the correlation of all layers in WSN, good flexibility and high efficiency cannot be obtained. So the need of cross layer MAC protocols, where the relation between each layer for improving consumption of energy is maintained.

In MAC-CROSS protocol, energy efficiency is improved by combining the characteristics of routing layer and MAC layer.

In CLMAC protocol, for simplification the transport layer and the network layer is removed from the protocol stack and this two layers functionality is combined into Application layer and the MAC/Physical layers.

3.11 Contention Based MAC Protocol Challenges

Since, using contention based MAC protocols more than one neighboring node may try their luck for sending their data through a shared medium, so it may face the following challenges [20]:

- Due to lack of scheduling it becomes very difficult to avoid collisions, overhearing and idle listening problems.
- Energy efficient MAC protocol design is difficult because of energy consumption by collisions, overhearing and idle listening problems.
- Since two or more neighboring nodes try their luck to get the medium, hidden-terminal situations is a common problem in contention-based MAC protocols.

3.12 Schedule Based MAC Protocol Challenges

Even though the schedule-based MAC protocols can avoid idle listening, collisions and overhearing, there have some drawbacks also [20]:

- During network setup and topology changes, the maintenance of scheduling involves traffic signaling which results protocol overhead.
- A time synchronization between the neighboring nodes is required which involves some extra traffic signaling, and due to clock drift of oscillators and mobility of nodes, resynchronization is required.
- Schedule adaptation becomes difficult with the change of network traffic load.
- The nodes require significant amount of memory to keep its and its neighbors schedule.
- Conflict free asignment of TDMA schedules is difficult [49].

So, the schedule-based MAC protocols are facing a challenge to minimize the problems stated above.

3.13 EMACS

The EYES MAC (EMAC) [2] is a schedule-based MAC protocol designed with the objectives of reducing the inefficiencies from most of the sources of inefficient use of energy like: idle listening, collision and protocol overhead in exchange of some increase in latency, but only for those sensor services that can tolerate such latency.

This protocol is scalable, can dynamically configure itself, and provides support for routing functionality by providing some topology information.

The basic mechanisms applied and combined in this protocol are: connection oriented operations, scheduled operations, collision avoidance, grouping send and receive and assistance for routing.

- **Scheduled operations:**
  In many sensor network applications for a long time, nodes are allowed to be in idle state without transmit or receive, which results a wastage of valuable energy. EMAC protocol uses the concept of periodically listen and sleep in which most of the time a node is allowed to be in sleep mode so reduces the listen time. At the scheduled time node wake up to announce its presence, and monitor new communication requests from other nodes for the scheduled operation. This protocol maintains time synchronization over the whole network, so that nodes transceivers can be turned on at the time when needed for performing an operation.

- **Collision avoidance:**
  The protocol is schedule based, so to maintain time synchronization among all nodes in the network, scheduled information is completely distributed over the whole network by each node without any central controller. Each node maintains a schedule table that stores the schedules of all its immediate neighbors and adds its own schedule, such that it will not occupy slots in use by those nodes. Collisions are only possible in a small portion of time, when nodes do requests to each other or inform their neighbors of their existents.

- **Grouping of send and receive:**
  The transition time of a wireless transceiver between the various power states like power on and off, and between sending and receiving (and vice versa) states can be significant. In this protocol it has been tried to group such operations as much as possible, such that after a node powered on can both send and listen.

- **Assist routing protocols:**
  In a multihop network a source node as well as an intermediate node has to decide to which neighboring node an incoming packet should be passed on so that it eventually reaches the destination. Knowledge of the actual topology is necessary to efficiently route the packets over the network and deliver at the destination.

The responsibility to determine these routing tables is of the routing algorithm with the help of the routing protocol that allow messages to be routed over the network. Storing routing tables in a static network might be an option, but when nodes are mobile the resulting frequent updates will be too expensive. The MAC layer contains a certain amount of (local) topology knowledge that should be made available to routing protocols.

- **Connection oriented:**

  EMAC uses two operational modes for the communication module of a sensor node: dormant mode and communication mode.

  In the dormant mode, the sensor node merely monitors its neighbors, keeps track of changes in schedule or topology and maintains synchronization among neighboring nodes.

  In the communication mode, the sensor node has established a connection with one of its neighbors, and agreed on a communication channel between them using timeslots and periods of communication.

  In this way the connection can be tuned to the requirements of the service requesting the communication.

EMAC Frame Format

In the EMAC protocol time is divided into frames and each frame is divided into a number of timeslot and each node allows to take control over a timeslot in a frame. Using this slot it can transmit its data without having contended for the medium. Based on the communication that should take place in its timeslot, the network node decides whether it denies or accepts requests from other nodes. Timeslots are assigned within the cell (directly connected nodes constitute a cell) and same timeslots can share with other nodes that are not within reach of each other, i,e in another cell. Each node in the network maintains a table called schedule table in which it stores its cell's schedule and also the schedules of its neighbors.

In EMAC, a timeslot is further divided into three parts: Communication Request (CR) section, Traffic Control (TC) section and Data section (see Figure 3.5).

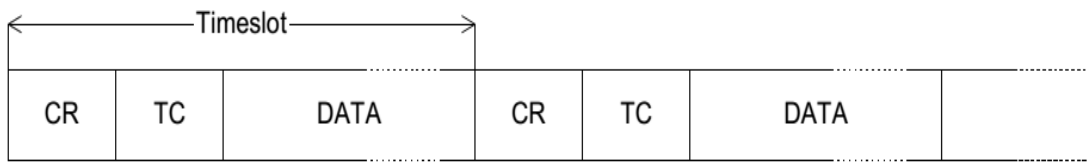


Figure 3.5 EMACS Frame Format.

In the CR section other nodes can ask for data or to notify the availability of data for the timeslot controller node. Nodes that do not have a request during the current slot owner, will keep their transceiver on during the entire duration of the CR section.

The owner of the slot always transmits its schedule for its data section and broadcast the schedule table in the TC section. When a timeslot is not owned by any node, all nodes will remain in sleep state during the whole duration of that slot.

If a node is not addressed in the TC section, not its request was approved, the node will resume in the standby state during the entire duration of the data section. After the TC section actual data transmission follows either downlink or uplink.

Although high occurrence of collisions is not expected, collisions can occur in the communication request section. A collision handling mechanism in the EMACS protocol is done by the slot owner. When it detects a collision notifies its neighbor nodes that a collision has occurred and after a random, but limited backoff time, the collided nodes retransmit their request in the data section. To prevent the distortion of ongoing requests carrier sense is applied.

Sleep modes in the EMACS protocol

Since in WSN idle listening, transmitting and receiving results in a significant amount of power consumption so as often as possible the networking nodes should turn off their transceivers. The EMACS protocols supports two sleep modes of the network nodes:

1. **Standby mode:** When for certain time duration no transmissions are expected the node releases its slot and to keep up with the network starts periodically listening to a TC section of a frame and this mode of a node is called standby mode. When the node has to transmit some data (typically an event driven sensor node), it can just request a transmission (both upload and download) in a CR section of another network, complete its transmission and go back to sleep. Depending on the communication needs, it will start owning a timeslot.

2. Dormant mode: The sensor node goes to low power dormant mode for some amount of time agreed by the upper layer. In this mode the synchronization with the network will be lost and communication with other nodes becomes impossible. So after wakes up, synchronizes with the network and performs the communication.

When nodes are in one of the sleep modes, does not own a timeslot, because when a node is in a sleep mode does not transmit in a TC section every frame. But, an event driven nodes may not relieve their right to get control over a timeslot. A drawback of not controlling a timeslot is that the node will not receive any messages directly addressed to it, but can only be able to receive multicast messages. Transmitting data to nodes that own a timeslot has no problem.

Distributed power control

Scalability of the network is done by controlling the transmission power of the node. Here nodes have the ability to control the transmission power and using this mechanism nodes can dynamically change the cell size, and try to establish a certain node density in its radio range. For example, when the number of timeslots is reaching the frame size, then a node might consider lowering its transmission power, and thus in effect, reducing the number of neighboring nodes and if needed increasing transmission power will results increasing the number of neighboring nodes.

3.14 LMAC

The lightweight medium access control (LMAC) [3] protocol has been designed based upon the scheduled-based access where each node periodically gets a time interval in which it is allowed to control the wireless medium according to its own requirements and needs. Beyond this interval, by receiving a short message, nodes are notified if it is the intended receivers. Whenever a node is not needed for communication, it switches its transceiver to standby and is, therefore, able to conserve energy. As all the node gets its own turn in using the medium, so there will be little idle-listening and collision of messages which are in other types of MAC methods —such as CSMA– and results for energy waste.

The key features of the protocol are its self-organizing in terms of timeslot assignment and synchronization and includes efficient routing of messages to designated gateways in the network at low additional costs.

Frames and Timeslots

In the LMAC protocol time is divided into a fixed length of frames and each frame is divided into a (integer) n number of timeslots. The scheduling principle in the LMAC protocol is very simple (hence its name lightweight): every node gains control of one timeslot to carry out its transmission. The schedule is repeated every frame, i.e. a node transmitting in —for example– timeslot 5, uses the same timeslot in the next frame. When a node has some data to transmit, it waits until its timeslot comes up, addresses one or more neighboring node and transmits the packet without causing interference or collision to other transmissions. Unlike traditional TDMA based systems, the timeslot assignment among the networking nodes are not done by central controller. Instead, LMAC protocol use a distributed algorithm called localized scheduling algorithm as is described in [2].

In order to be capable of receiving messages, nodes always listen at the beginning of each timeslots of other nodes to determine whether they are addressed either by node ID or by broadcast address.

In the LMAC protocol a timeslot is further divided into two parts of unequal length: control message (CM) and data message (DM).

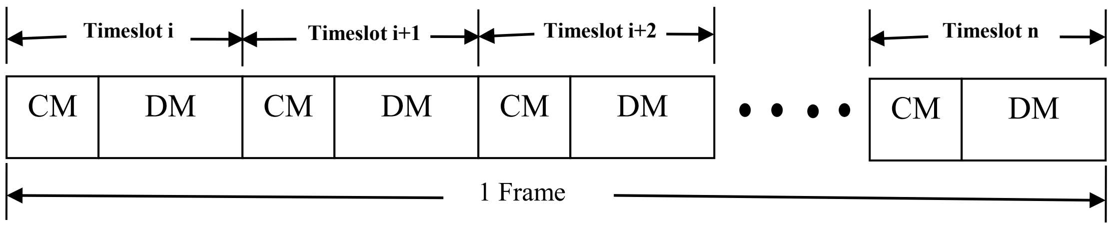


Figure 3.6 Timeslot and its contents for LMAC protocol.

Control Message

For each node, if it does not have any data to send, current slot controlling nodes starts its slot by sending a control message (CM). The control message (CM) has a fixed size and is used for several purposes like for maintaining synchronization, and for neighbor discovery. The control message of LMAC protocol contains the following contents:

Identification

The identification number (ID) of the slot controller node is included in the control message (CM). A node always has knowledge about its neighboring nodes by listening to all of the CMs. This is valuable information for routing towards the gateway node.

Current Slot Number

This number indicates the position of the slot in the current frame and is used by new (i.e. unsynchronized) nodes in the network.

Occupied Slots

In the LMAC protocol, each node use a bit vector to keep track of controlled timeslots by its neighbors including itself. The length of bit vector is equal to the length of number of timeslots in a frame. A timeslot is represented by each position in the bit and when a node detects energy in the wireless medium or receives a CM successfully, it updates the bit vector by setting 1 at that timeslot position; otherwise a 0 will be inserted in that position. A node inserts a 1 at its controlling timeslot. This bit vector plays a vital role, since, it lets nodes to know which timeslot does not interfere with other nodes between two hop distances, and thus can be used.

Distance to Gateway

For each node, this number represents the hop distance to the closest gateway in their vicinity. This number is also used for synchronization and shortest path routing of messages to the gateways.

Collision in Slot

When two or more nodes choose the same timeslot for its own control and hence for sending, then collision problem may occur. This can happen during network setup, mobility of nodes, variations in link quality etc.

The nodes that caused the collision cannot detect the collision by themselves; because they are transmitting when the event occurs. So they need to be informed by their neighboring nodes. From colliding packets, neighboring nodes cannot extract any information like: the ID of the node that caused the collision. So neighboring nodes can only report the colliding node using the collision in slot field by setting the timeslot in which they detected a collision. The neighbors on their turn check their controlled timeslot against the transmitted collision information. If it matches, they decide that they are in collision with another node, so release their timeslot and for obtaining a non-interfering one initiate the procedure.

Destination ID

This field is used for addressing neighboring nodes. The protocol also supports broadcast messages (message that should be received by all neighbors). When a node is addressed in this field, it will try to receive the upcoming DM after the end of CM time period. If they are not the intended receiver, or that the transmitting node simply has no data to transmit, they immediately turn off their power-consuming receiver and wait until the next timeslot to turn on the receiver. This allows the protocol to be energy efficient, i.e. no idle-listening.

Data Size (bytes)

This field is used to inform the intended receiver about the data message size in bytes that will be send following this control message.

Data Message

After sending control message (CM), the data message (DM) portion of the timeslot contains the data that a node needs to send. If a node is addressed, keep its receiver on until the data packet has been completely received and then goes back to standby mode for the remainder of the timeslot.

Network Setup

During the network setup phase, to be synchronized, the gateway nodes take the initiative by controlling a timeslot. Using that timeslot the gateway nodes send a CM message (CM includes: Sender ID, Destination ID , Current Slot Number, Occupied Slots, Collision in Slot, Distance to Gateway, Data Size etc.) to its one-hop neighbor. Then these neighbors synchronized their clocks with the gateway and pick a random timeslot to control except the already occupied ones (only this timeslot can be used for sending by this node but any other slots of its neighbors can be used for receiving), by a localized scheduling algorithm, which ensures that the same timeslot will not be used between the two hop neighbor for avoiding well known hidden terminal problem. The timeslot occupancy is efficiently encoded by a bit vector, which lengths equal to the number of timeslots in a frame. Each nodes can start controlling a timeslot when the slot is considered to be free by all its neighbors and thus, for all nodes it is to maintain a table to collect information about their neighborhood. Continue this process by sending CM to its next neighbors until all the nodes of the network get synchronized and occupy timeslot. After the network setup the network is ready for actual data collection from the sensor nodes to the gateway.

Routing to Gateway Nodes

To efficiently route the data message towards the gateway node, this protocol use ideas from [3]. Since each node keeps the record of its hop distance to the gateway, so when a node needs to send a data message (either generated by itself or received from another node) the node will look in its neighbor table to find a neighbor that is closer to the gateway than itself and will pick this node as destination for the message. In case of multiple neighbor with same minimum distance to the gateway, the node will randomly pick one from the candidates. Continue this process until it reaches the gateway and eventually the message will arrive at the gateway.

3.15 AI-LMAC

An Adaptive, Information-Centric and Lightweight MAC (AI-LMAC) [6] is a TDMA based adaptive and information-aware version of LMAC protocol for WSN. Unlike LMAC which allows a node to own only single timeslot, whatever the data traffic will flow through that node, In AI-LMAC a node is allowed to own more timeslot depending on the amount of data traffic that is expected to flow through it. This protocol define fairness as follows: only nodes that are able to service an incoming query or nodes that have children, which can service a particular query, should be given the chance to transmit their data. Nodes that are not involved in current query should be given the lower priority. Therefore, the priority given to a node is directly proportional to the amount of data a node is expected to transmit. So, this protocol provides fairness in the sense that bandwidth is allocated to the node depending on the data traffic it will encounter to flow i,e giving every node in the network an equal opportunity to transmit its data.

AI-LMAC protocol maintains a parent child relationship where the gateway of the network can be considered to be the highest parent in the hierarchy. The process of giving advice starts at the gateway node of the tree when a query is first injected into the network. This process then continues down the branches of the tree towards the leaf nodes. When a query is injected in the network the parent node use it's Data Distribution Table (DDT) (every node maintains a DDT which records the data history for different query) to predict the amount of data their children will generate based on the current query and history. If a parent node realizes that a subset of its immediate children's are going to transmit a large volume of data. Then it advice the children's for taking multiple slots (obviously on free basis between two-hop neighbor) under the current condition. Then, it is up to the child nodes to follow the advice as closely as possible.

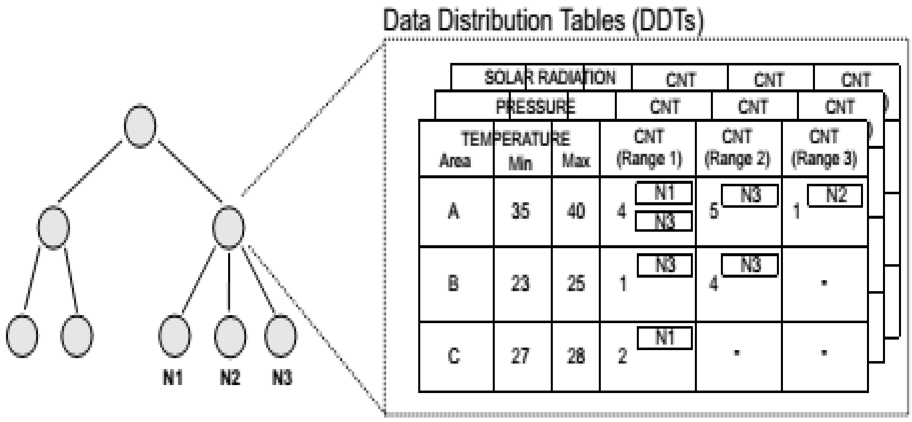


Figure 3.7 Format of the Data Distribution Table.

Since, this protocol provides fairness in the sense that bandwidth is allocated to the node depending on the data traffic it will encounter to flow. In AI-LMAC protocol this issue of fairness is done by two-dimensional fairness (Horizontal and Vertical). Horizontal fairness mechanism ensures that all sibling nodes (at the same level) of a certain parent are allocated slots fairly. Vertical fairness mechanism ensures that the total number of assigned slots to the immediate children node does not exceed the number of controlled slots by the parent.

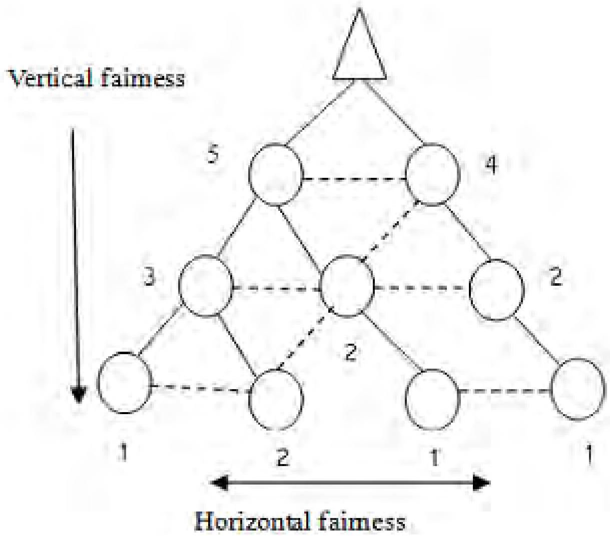


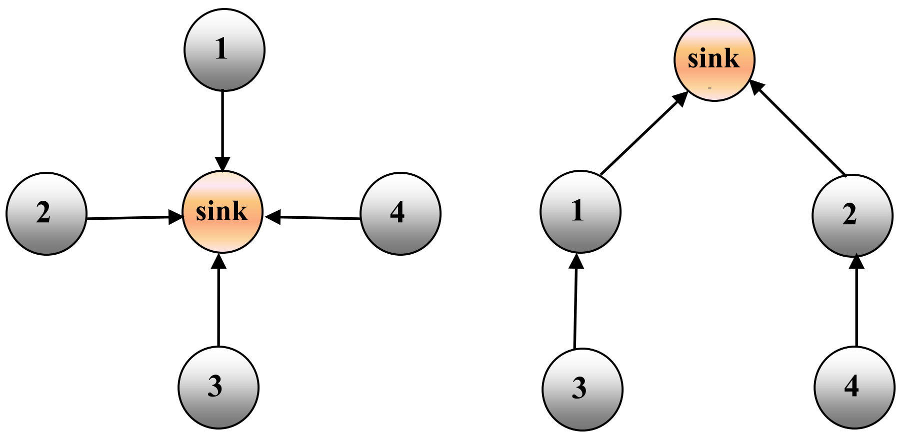
Figure 3.8 Horizontal and Vertical fairness in AI-LMAC protocol.

3.16 MC-LMAC

Although in traditional wireless sensor network applications where primary consideration is energy efficiency and secondary consideration is bandwidth utilization and throughput maximization, some recent wireless sensor network applications, such as structural health monitoring, require faster data collection. A Multi-Channel MAC Protocol (MC-LMAC) [7] for WSN which is a schedule-based MAC protocol designed based on single channel LMAC protocol with the objective to increase the throughput by coordinating transmissions over multiple channels and hence increase the bandwidth utilization.

Efficiency of Multiple Channel

To show the efficient use of wireless shared medium and to increase the network throughput by using the concept of multi-channel, Figure 4.5 presents a benchmark scenario. Figure 3.9(a) presents a single-hop scenario where all the source nodes can reach the sink node directly. If W be the capacity of the shared medium. In an ideal case aggregate throughput would be W, and each source node should transmit with a capacity of W/4. After considering the multi-hop scenario which is shown in Figure 3.9(b), and if there is no interference then using a suitable scheduling mechanism, W/4 throughput by each source node can be achieved. If all the nodes interfere with each other (since here all the nodes are in maximum two-hop distance and it is known that in wireless medium nodes that are in two-hop distance can interfere), then each source node can get only W/6 capacity. So, if nodes can use different channels to transmit then interference can be eliminated and the source nodes can reach the W/4 capacity.



a) **b)**

Figure 3.9 a) Single-hop topology; b) Multi-hop topology.

Timeslot and Channel Selection

Timeslot and channel selection in MC-LMAC protocol is done by using localized scheduling algorithm with the same manner in LMAC protocol with an exception- In LMAC which ensure that, a node does not use same timeslot within its two-hop neighborhood and in MC-LMAC, which ensure same channel/timeslot pair is not use in the two-hop neighborhood.

Since LMAC protocol is a single channel protocol where timeslot selection is done in single channel, so for a node to choose a free timeslot which is not used by any other nodes within the two-hop neighbor such that it does not interfere the communication between other nodes in the network. In LMAC protocol new node performs XOR operation among all the received occupied slots vector from its direct neighbor to exclude the timeslots which are used between its two-hop neighbors. In MC-LMAC, nodes keep occupied slots vectors per channel and select a timeslot to be used on a particular channel. In MC-LMAC, when a node is trying to get the control of a timeslot, executes the “bitwise OR” operation over each occupied slots vector for each channel and discovers the free slots of that channels. Similar to single channel LMAC, this method guarantees that the same “channel/timeslot” pair is not used in the 2-hop neighborhood. So here timeslot which is used by another node within 2-hop neighbor can be selected by a node but in different channel.

To understand the timeslot and channel selection procedure, Figure 3.10 shows an example where the number of timeslots per frame is 5 and the number of frequencies is 2. The node marked with “?” is searching for a slot of any channel and other nodes are marked by channel/timeslot pair that they are using. The node without a timeslot receives the occupied slots information (the position of a bit in the vector is the timeslot number: 1 means the timeslot is occupied and 0 means free and since, number of timeslot in a frame is 5 so the vector length is 5-bit) from the neighbors, executes the OR operation among received occupied slots vector and discovers that on F1 (frequency 1) all the slots are occupied. However, there are free slots 1, 2, 3 and 5 on F2. The node selects timeslot 5 (which is not occupied by the direct neighbors and also not by the neighbor of direct neighbors) on F2.

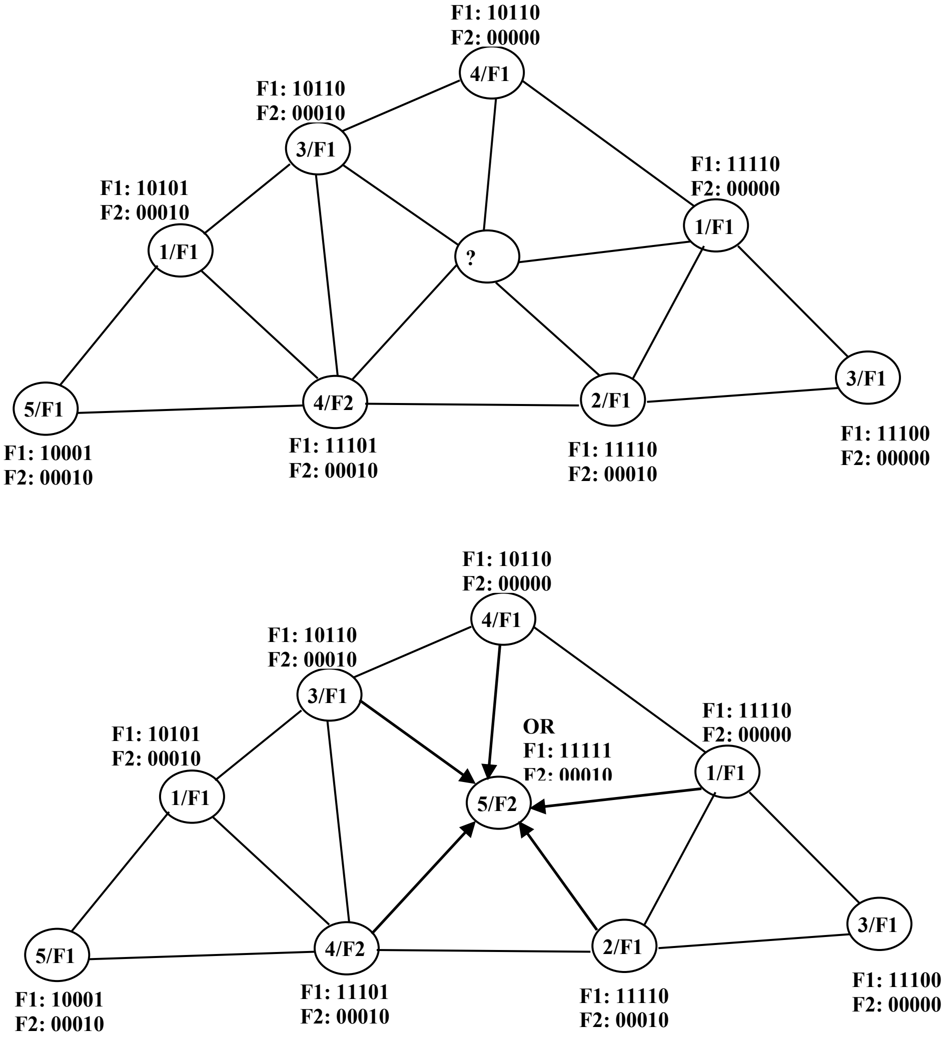


Figure 3.10 MC-LMAC timeslot and channel selection.

MC-LMAC Timeslot Structure

In MC-LMAC to support multi-channel communication, nodes select a timeslot-channel pair. Timeslot structure of LMAC is extended in MC-LMAC by adding a small slots called common frequency (CF) periods with control message (CM) and data transmission (DATA) period which is shown in figure 3.11. Communication during CF period is also based on scheduled access. Each CF slot is indexed by a channel number and the number of CF slots is equal to the number of channels. A sender controlling the current timeslot addresses the destination by sending only the intended destination ID during the CF slot which is reserved for the channel number it controls. Other nodes which are not the controller of the current timeslot listen during the whole CF period in order to be informed about the intended destinations. If a node is addressed during a CF slot then it confirms that it is the intended receiver and switches its transceiver on the sender's associated frequency for further receiving the upcoming control message and data message. If not, the node switches its transceiver to standby for the remainder of the timeslot to conserve energy.

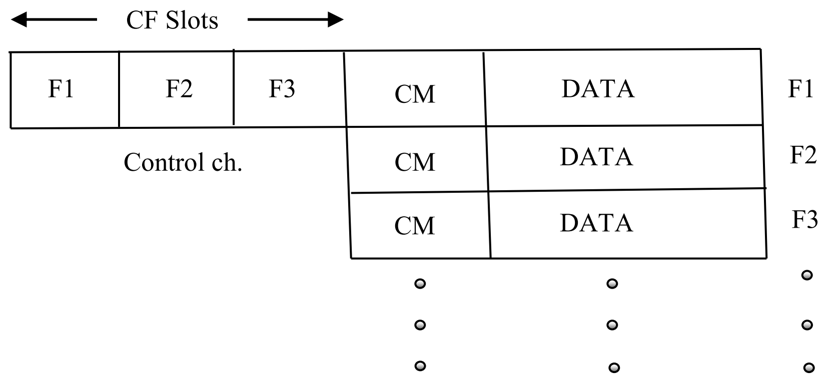


Figure 3.11 MC-LMAC timeslot structure.

After the common frequency (CF) period the receiver switches on the sender's channel and the timeslot owner transmits CM which includes control information, followed by the DATA section.

The contents of the control message transmitted during CM period and they are as follows: ID represents the sender node id, Destination ID is the receiver's id or it can be a broadcast address. Occupied Slots Vector represents the bit vector for the occupied slots in the neighborhood and this Occupied Slot Vector updated and sends for each channel. Collision in Slot represents the slot number during which a collision has been detected and Collision in frequency field is added to detect the colliding channel. Current Slot represents the slot number and it is used for synchronization by the new joining nodes. Hops to Gateway field lets the other nodes to be informed about senders hop distance to the sink node and it is used for synchronization. Acknowledgement bit vector has equal length to the number of timeslots per frame. Nodes keep track of the slots during which they receive data. At the beginning the bit vector is filled with 0's. Whenever a message is received, logic 1 is added at the position of the respective slot in the acknowledgement field.

Medium Access

A simple example to show overall medium access coordination using multiple channel is shown in Figure 3.12. The initial part of the figure shows the topology: the numbers inside the circles represent the ids of the nodes. It is assumed that there are 3 channels available, represented as F1, F2, and F3 and there are 3 CF slots. Sender 1send destination id 4 to addresses node 4 on the first CF slot to communicate on channel F1, sender 2send destination id 5 to addresses node 5 on the second CF slot to communicate on F2 and sender 3send destination id 6 to addresses node 6 on the third CF slot to communicate on F3. CF section takes place on the control channel which is F1 and is called default channel. At the end of CF slots, nodes tune their transceivers to the related channels: node 1 and 4will tune on F1, node 2 and 5 on F2, node 3 and 6 on F3 for transmitting and receiving upcoming CM and DATA messages in remaining time of current slot. Note that, due to interference these three parallel transmissions would not be possible if there was only a single channel available. Using multi-channel parallel transmission is possible without interference among the nodes.

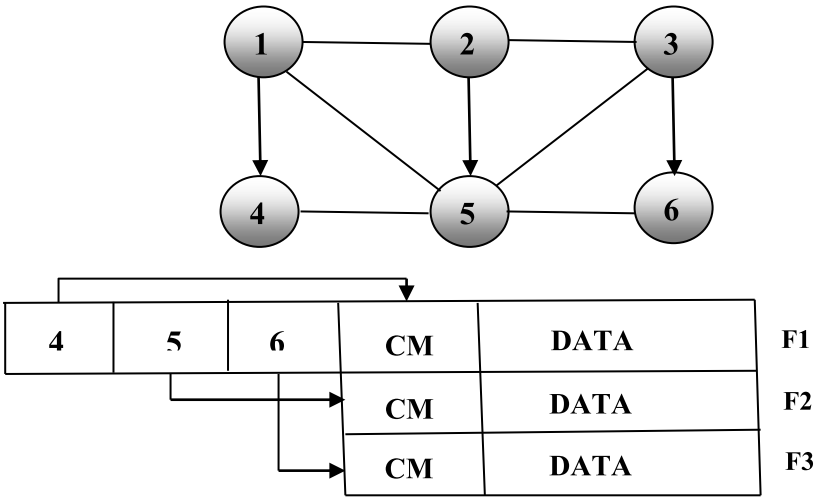


Figure 3.12 MC-LMAC coordination scheme.

Although MC-LMAC protocol increase the network throughput by transmitting simultaneously using different frequencies within a collision domain, but delay problem of traditional schedule-based MAC protocols still exists- because if a node needs to send data message just after ending its controlled timeslot, it has to wait until the same timeslot of next frame which results a significant delay.

3.17 Summary

This chapter gives a brief overview of the role of MAC layer in network architecture, different types of multiple access techniques in MAC layer, problems that may encounter during MAC protocol design, different classes of MAC protocols and the issues that should be considered during MAC protocol design. This chapter also discuss the challenges of different types of wireless sensor network MAC protocols that also should take consideration during MAC protocol design. Finally, this chapter concludes by the description of some important schedule-based MAC protocols.

CHAPTER 4

LOW LATENCY MULTICHANNEL LMAC DESIGN

4.1 Introduction

This chapter presents the proposed schedule-based MAC protocol named "Low Latency Multichannel LMAC (LL-MCLMAC) protocol for Wireless Sensor Networks". A detail overview of the proposed protocol are presented in this chapter, including its network setup procedure, channel and timeslot selection procedure, scheduling mechanism for medium access, data routing procedure to gateway node and finally the operation cycle of LL-MCLMAC is also presented in this chapter.

4.2 Proposed LL-MCLMAC Protocol

LL-MCLMAC, low latency multichannel LMAC is a scheduled-based energy efficient MAC protocol designed for delay sensitive WSN applications which is based on the existing single channel energy efficient LMAC protocol and multichannel MC-LMAC protocol. Here multichannel concept is used like MC-LMAC protocol. However, two timeslots of a channel has been assigned to a node here unlike MC-LMAC depending on the free timeslots of a channel by ensuring the two hop neighbor timeslots conflict avoidance over the same channel. The free timeslots of each channel are calculated by performing the OR operation between its local vector and the vector found from its neighbor. A node is allowed to transmit by using its controlled timeslots over controlled channel and can receive from any timeslot of any channel by switching the interface between different channels as well. Therefore, the transmitter has the option to choose the best timeslot between the two and hence the delay of packet forwarding decreases dramatically.

4.2.1 Scheduling Mechanism for Medium Access

This section presents a scheduling mechanisms of proposed Low Latency Multichannel LMAC (LL-MCLMAC) Protocol for Wireless Sensor Networks. In this protocol scheduling algorithm allows each node to choose two different timeslots with a channel, which is not interfering with the communication between other nodes in the network. This channel has been referred as ControlledCh and two timeslots as ControlledSlot1 and

ControlledSlot2. Since a frame consists of a integer number of timeslots and each node transmit packets in its controlled timeslot and frame is repeated periodically. So to indicate the period of timeslot in a frame, the variable CurrentSlot is introduced to indicate the current slot position in the frame. During the timeslots of other nodes a node receives packets. Figure 4.1 illustrates the scheduling mechanism. For any nodes in the network if CurrentSlot equal its ControlledSlot1 or ControlledSlot2, then this node can send packets during this timeslot. Other nodes can receive packets. Using default channel, CurrentSlot controlled nodes first send initial_control (Ini_CTL) message to notify the intended receiver using which channel it will send further control and data message during this timeslot.

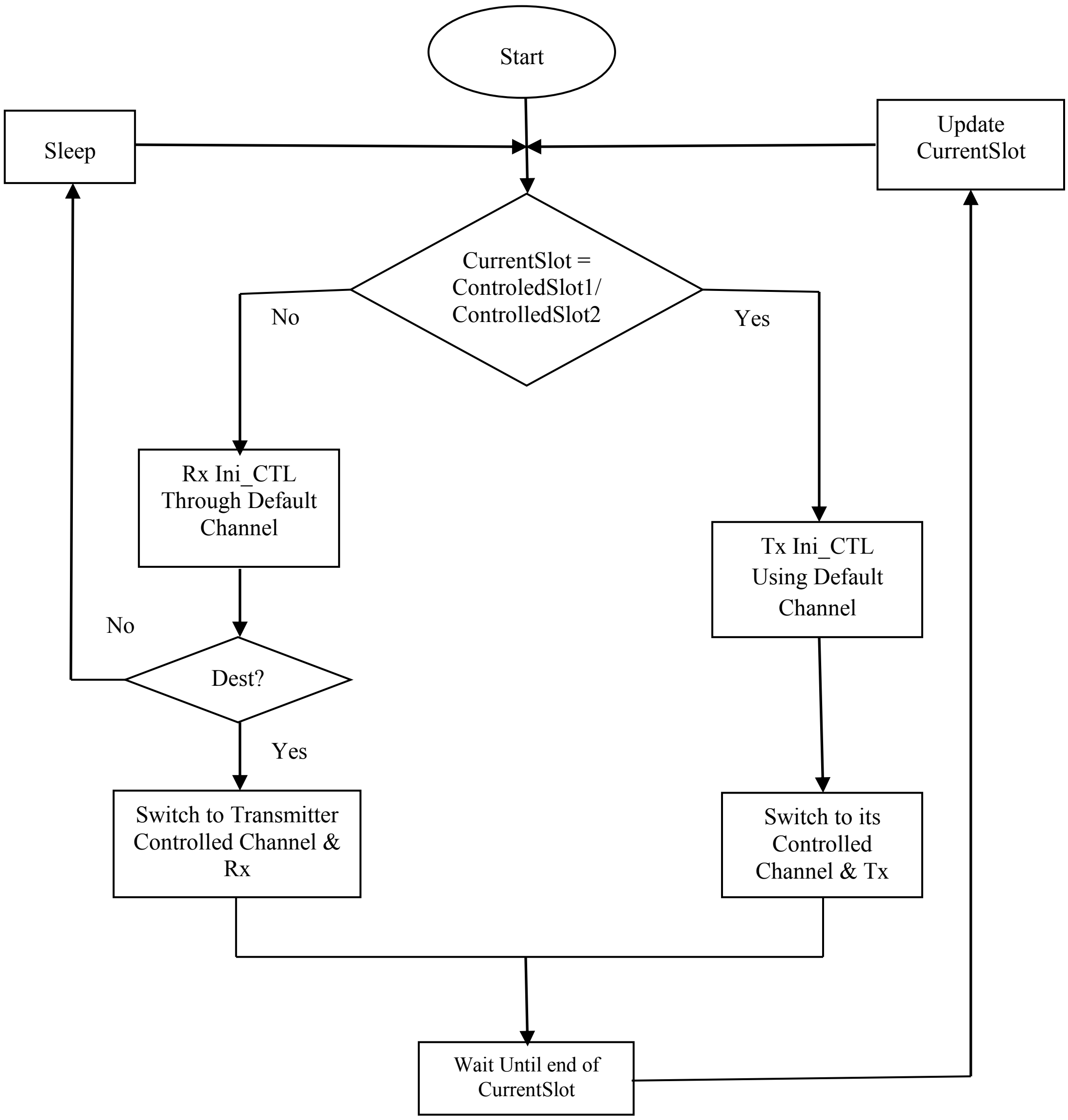


Figure 4.1 Flow chart for medium access scheduling principle.

Then CurrentSlot controlled nodes switch its channel from default channel to its controlled channel ”ControlledCh” and send control message and then data message. On the otherhand other nodes which are not the owner of this CurrentSlot receives initial_control (Ini_CTL) message, decides and if it is the intended receiver switch its transceiver to Ini_CTL message senders ControledCh for further receiving control message and data message, otherwise it will go back to sleep without receiving control and data message. After the end of CurrentSlot duration variable of CurrentSlot value is updated. Continue this process again and again.

4.2.2 Bit Vectors of Occupied Timeslots

To avoid timeslot conflict and collision, For a node it is necessary to choose timeslots that is not using between its two hop neighbors which is done by using OR operation among list of the timeslots which are in use by its neighbors. This list of timeslots can be represented using bit vectors to separate used and unused timeslot in the list. Figure 4.2 shows a sample bit vector in which 0 indicates that the node did not detect any transmission in the respective timeslot and 1 indicates this slot is using by node between two hop neighbors.

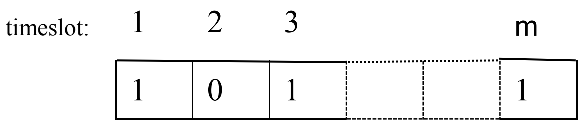


Figure 4.2 Bit vector of occupied slots. 0 indicates slot is free and 1 indicates slot is in use by its first or second order neighbors.

4.2.3 Channels and Timeslots Structure

Figure 4.3shows the channel and timeslot structure.

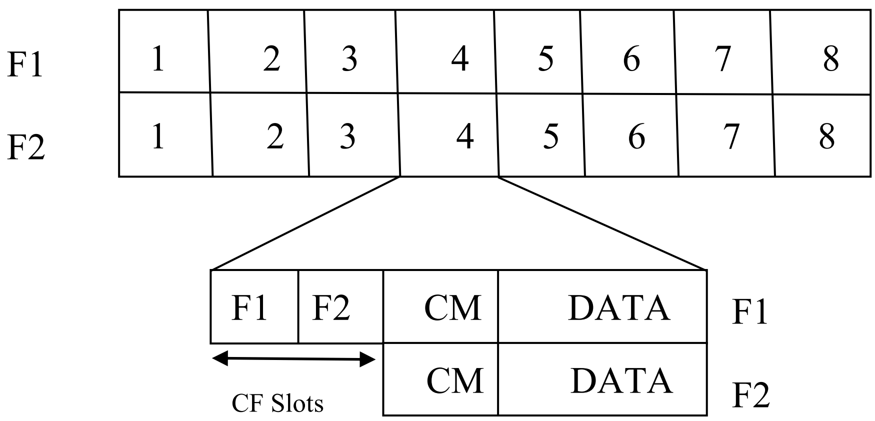


Figure 4.3 LL-MCLMAC channel and timeslot structure.

The timeslot and channel structure of the proposed MAC protocol is similar to that of MC-LMAC protocol. For example, a timeslot incorporates a control message (CM) period and a data message (DM) period here as well. Moreover, communication during common frequency (CF) period is also based on schedule access and appears on small slots called CF slots which is equal to the number of channels. Here, for simplicity figure shows two channels, where each channel is divided into 8 timeslots.

4.2.4 Contents of Initial_control message and Control message

When sender needs to send data packet, it first send initial_control message which includes only destination ID of intended receiver to instruct to switch it's channel to senders controlled channel. Table 4.1 shows the contents of initial_control message.

Table 4.1 Contents of the initial_control message

Description	**Size(bytes)**
Destination ID	2

After channel switching sender sends control message (CM) to that intended receiver. This control message includes- occupied channel of sender, senders controlled timeslot1 and timeslot2 named "Occupied Slot1" and "Occupied Slot2", sender hop distance to gateway node is also included in control message and occupied slots vector for each channel (here number of channel is 2) is also included in control message. Table 4.2 summarizes the parameters.

Table 4.2 Contents of the control message (CM)

Description	**Size(bytes)**
Occupied Channel	2
Occupied Slot1	2
Occupied Slot2	2
Distance to Gateway	2
Occupied Slots vector1	2/4
Occupied Slots vector2	2/4
Total	**12/14**

4.2.5 Network Setup

Initially when the nodes are powered on, the nodes are unsynchronized. Each node randomly chooses a channel and two timeslots of that channel to control and updates their local vector for that channel. At this time timeslot selection is done by trying to select two timeslots as they are half frame time separated for reducing end-to-end delay using initial slot selection algorithm. Algorithm-1 explains the procedure how it has been tried to separate two timeslots by half frame duration during initial slot selection by each node.

Algorithm 1: Initial Slot Selection

Input: Node Id (node_index), total number of slots in a frame (num_slots) and number of reserved slots (reserved_slots).

Output: timeslot1 (slot1) and timeslot2 (slot2).

```
if(num_slots>reserved_slots) then
    if((num_slots - reserved_slots)>=2) then
        slot1 = node_index%(num_slots - reserved_slots);
        slot2 = (node_index + floor((num_slots - reserved_slots)/2))
        %(num_slots - reserved_slots);
    end if
    if ((num_slots - reserved_slots) = 1) then
        slot1 = 0;
        slot2 = -1;
    end if
else
    slot1 = -1;
    slot2 = -1;
end if
```

The gateway (sink) node takes the initiative to become synchronous with other nodes of the network. Gateway node selects one of its controlled timeslot and using the default channels (usually 1) common frequency (CF) slot for its channel, the gateway node broadcast initial control message and instructs its one-hop neighbors to switch their interface to the senders controlled channel which will be used by the gateway to further receive control message (CM). After channel switching by the one-hop neighbors, the gateway node broadcast control message (CM) during the control message (CM) period. This CM is received by the gateway's one-hop neighbor and synchronizes their clocks with the gateway.

Then receiving nodes updates their local vector for that channel, neighbor table and distance to gateway (dtg) (using algorithm-2) and confirms that its controlled channel controlled timeslots is not using by any other nodes between its two-hop neighbors. If one or both of its controlled channel timeslots are using by any other nodes between its two hop neighbors, then it choose another one/two free timeslots by performing bitwise OR operation between it and its one-hop neighbors local vector of that channel. This process is continued for sufficient number of frames as all the nodes get free timeslots and become synchronous with other nodes. Node can only select timeslots when the timeslots of its channel are free between its entire two-hop neighbors, so all the nodes require to maintain bit vectors for each channel to keep record about used and unused timeslots of each channel by its neighbors. In this way the hidden terminal problem is avoided. During the network setup phase, nodes send only control messages. After the end of setup phase, nodes can send control messages and data message as well.

4.2.6 Distance to Gateway Calculation

Initially gateway node sets its distance to gateway (dtg) value zero. Then it send this value through control message. Nodes those receive this dtg value from gateway set their dtg to 1. When a node receives a dtg>0 and receiving nodes dtg=0 and if its not gateway, then it set its dtg=received dtg+1. Again if a node receives a dtg>0 and receiving nodes dtg>0, and received dtg<nodes own dtg, then it set its dtg=received dtg+1.Algorithm 2 illustrates the distance to gateway calculation after a node receives a control packet from any other node.

Algorithm 2: Distance to gateway calculation after a node receives a control packet from any other node.

Input: Node Id (myId), Current distance to gateway(myDtg), Sink address(sinkAddr), Address of control packet sending node(SrcAddr), Distance to gateway from control packet sending node(getDtg).

Output: Update distance to gateway of a node (myDtg).

```
if (myId = sinkAddr) then
   |   myDtg=0;
end if
if (srcAddr() = sinkAddr || (getDtg > 0 &&myDtg =0)) then
   |   myDtg = getDtg +1 ;
end if
if ((getDtg>0) &&(myDtg != 0) && (getDtg <myDtg)) then
   |   myDtg = getDtg + 1;
end if
```

4.2.7 Channel and Timeslot Selection Procedure

When a node joined in the network, how it will select its channel and timeslots? Figure 4.4 illustrates how channel and timeslots are selected by a node joining in the network. The node marked with "?" is seeking for a channel and two timeslots. Other nodes are tagged with engaged slots vector for each channel. For simplicity, Here 2 channels and 8 timeslots for each channel are considered. The node without a timeslot receives the engaged slots vector information (the position of a bit in the bit vector is the timeslot number: 1 means- timeslot is engaged and 0 means- timeslot is free) from the neighbors. First, a channel is chosen randomly from two channels. For example, it chooses channel F2. Now this node has to execute bitwise OR operation on all of its neighbor's occupied slots vector (10001000, 10111011 and 00110011) to find two free timeslots of channel F2. The result of OR operation is 10111011. So, node proceeds to free timeslots of 2 and 6 of channel F2 for its use. Algorithm 1explains the procedure how it has been tried to separate two timeslots by half frame duration during initial slot selection by each node.

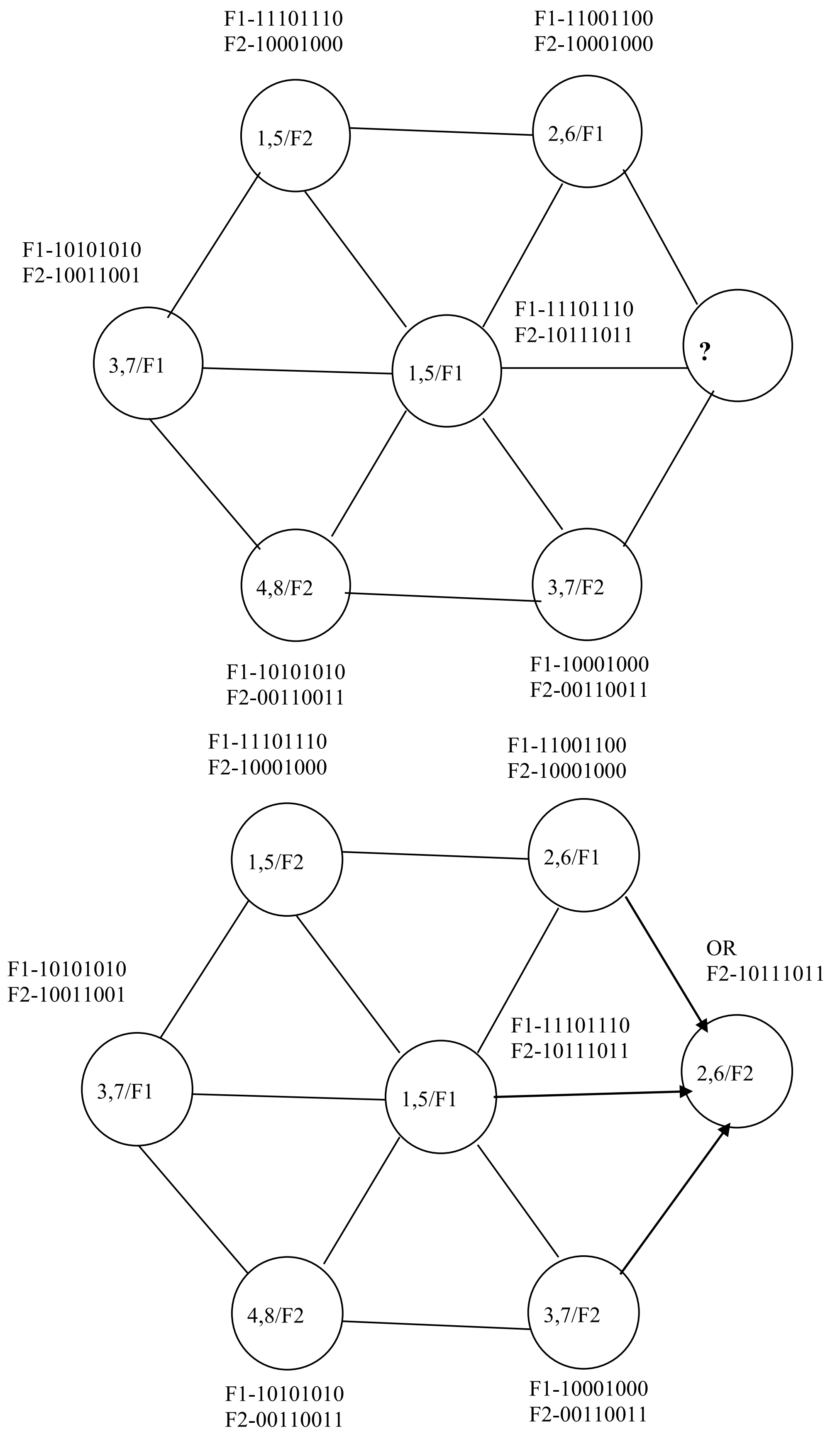


Figure 4.4 Channel and timeslot selection procedure of LL-MCLMAC.

4.2.8 Routing to Gateway Node

When a node need to send a packet to designated gateway i.e., if it's macQueuesize is greater than zero, it has to wait until its controlled timeslot1 or timeslot2. This message may be either generated by itself or received from another node. For efficiently route this message towards the gateway node, each node keeps track of its hop-distance to a designated gateway node and broadcasts this information efficiently in its control message. Using its neighbor table, the sending node choose the most close neighbor node to the gateway as the destination for the message. The node add this node address as destination address and send this address through initial_control message (in case of multiple neighbors closer to the gateway, the node will randomly pick one from the candidates) in it's allocated common frequency (CF) slot using default channel F1 and switch its channel to its control channel for further transmitting control message (CM) and data message (DM). A simple distance to gateway scenario is shown in Figure 4.5.

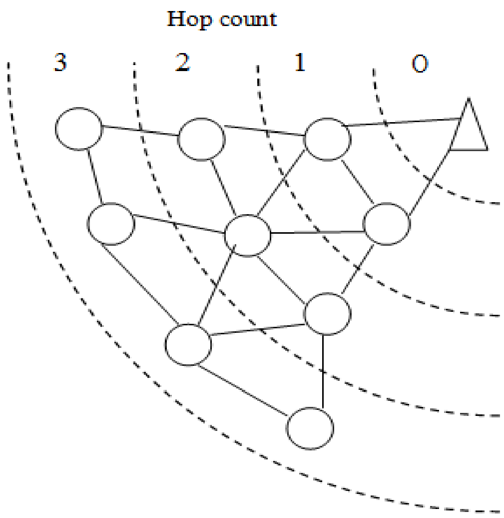


Figure 4.5 Nodes keep track of their hop distance to the closest the gateway. This information is used for efficient routing to the gateway.

Then the nodes which receives this initial_control message decides by getting destination address which was included in initial_control message. If it is the intended receiver, switch its channel to senders associated channel and waits to further receive control message (CM) and data message (DM). If it is not the intended receiver it goes to sleep to conserve energy. After channel switching, sender sends its control message through it's controlled channel and timeslot. After receiving this control message, the receiver node update channel vector for that channel and value of distance to gateway (dtg). After sending CM, current timeslot controlled node sent DM and intended receiver receives this DM. Continue this process until the message arrives at the gateway.

4.2.9 Operation Cycle of LL-MCLMAC

The operation cycle of LL-MCLMAC is shown in Figure 4.6 and described in detail.

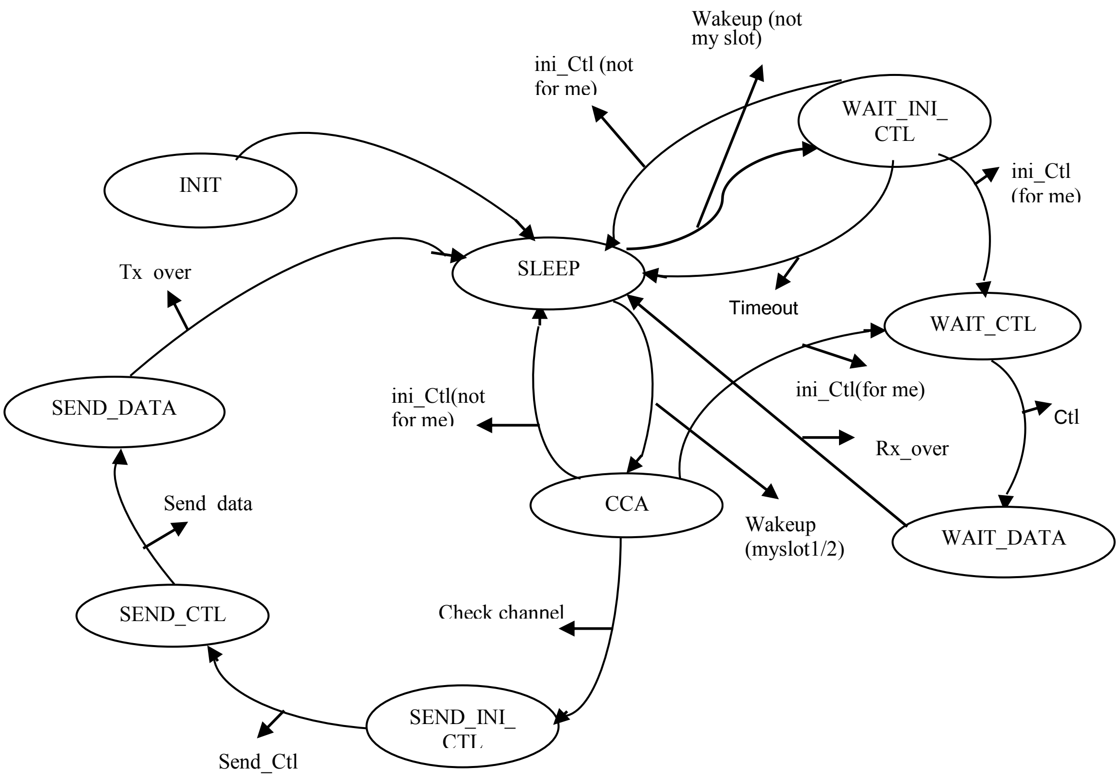


Figure 4.6 Operation cycle of LL-MCLMAC.

INIT: This is the initial MAC state of each node. When nodes are powered on every nodes MAC state starts from this INIT state and goes back to sleep to awake in the next timeslot.

SLEEP: After awaking from sleep state in the next timeslot, nodes check if is it its controlled timeslot or not? If it is then node goes to CCA state to check the channel and hence to send initial control message otherwise it will go to WAIT_INI_CTL state to receive initial control message.

CCA: In this state current timeslot controlled node check the channel to see is it idle or busy? If it is free then goes to SEND_INI_CTL state to send initial control message otherwise it decides it is interfering timeslot and change the slot.

WAIT_INI_CTL: Here, node is waiting to receive initial control message at the CF slot. After receiving initial control message MAC goes to WAIT_CTL state.

WAIT_CTL: In this MAC state node is waiting to receive control message after the CF slot. After receiving control message MAC goes to WAIT_DATA state.

WAIT_DATA: After receiving control message MAC is waiting in WAIT_DATA state to receive data and after completion of data receiving MAC goes to SLEEP state.

SEND_INI_CTL: In this MAC state node transmits initial control message through CF slot to notify the intended receiver about the upcoming control and data message.

SEND_CTL: In this MAC state node transmits control message using CM section of its controlled channels controlled timeslot.

SEND_DATA: After sending control message node sends data message using DM section of its controlled channels controlled timeslot. After completion of data sending node goes back to sleep.

4.3 Summary

In this chapter, the proposed LL-MCLMAC- "Low Latency Multichannel LMAC protocol for Wireless Sensor Networks" is fully depicted. The scheduling mechanisms for medium access, channel and timeslot structure, different message contents, channel and timeslot selection procedure using algorithms, data routing to gateway nodes, operation cycles of the proposed LL-MCLAMC are also lucidly described throughout this chapter.

CHAPTER 5

SIMULATION RESULTS AND DISCUSSION

5.1 Introduction

This chapter includes the simulation results for different simulation scenarios what have been done in order to evaluate the performance of the proposed MAC protocol, an implementation of the LL-MCLMAC protocol was done in a custom WSN simulator build in discrete event simulator OMNeT++ [50] with modeling framework MiXiM [51]. Several experiments are conducted to investigate the performance of LL-MCLMAC protocol as well as to compare the performance with existing LMAC and MC-LMAC protocols.

5.2 OMNeT++

OMNeT++ [50] is an component based object-oriented modular discrete event simulation framework. OMNeT++ is not a simulator by itself, rather it provides infrastructure and tools for writing simulations. In OMNeT++, simulation models are assembled from reusable components termed modules and modules can be connected with each other via gates and combined to form compound modules. Gate is the input and output interface of modules, through which data can be send to or receive from another nodes. Module parameters can be used to customize modules behavior. Modules communicate with each other through message passing technique, where messages may carry a variety of data structures and this message can be passed along predefined paths using gates and connections, or directly to their destinations the later is mainly useful for the case of wireless simulations. Some of the main features of OMNeT++ are the following:

- Hierarchical nested modules
- Modules are instances of module types
- Modules communicate via message passing through channels
- Highly flexible module parameters
- Topology description language i,e NED.

5.3 MiXiM

MiXiM [51] is an OMNeT++ modeling framework created for fixed and mobile wireless networks (ad-hoc networks, wireless sensor networks, vehicular networks and body area networks etc.) which offers customizable models of interference estimation, wireless MAC protocols, radio wave propagation and radio transceiver power consumption. Actually MiXiM framework is designed to concentrates on the lower layers of the protocol stack.

5.4 Network Model

In MiXiM using NED editors one can represent the physical topology of any network. NED editors can also be used to define the position of the entity and interconnection between entities, (e.g. node and links). Node model is used to describe the characteristics of each node and a node can also be either fixed or mobile. Actual communication channel conditions can be implemented by customizing the link property. Figure 5.1 shows a simple network model for 10 nodes in the simulation area.

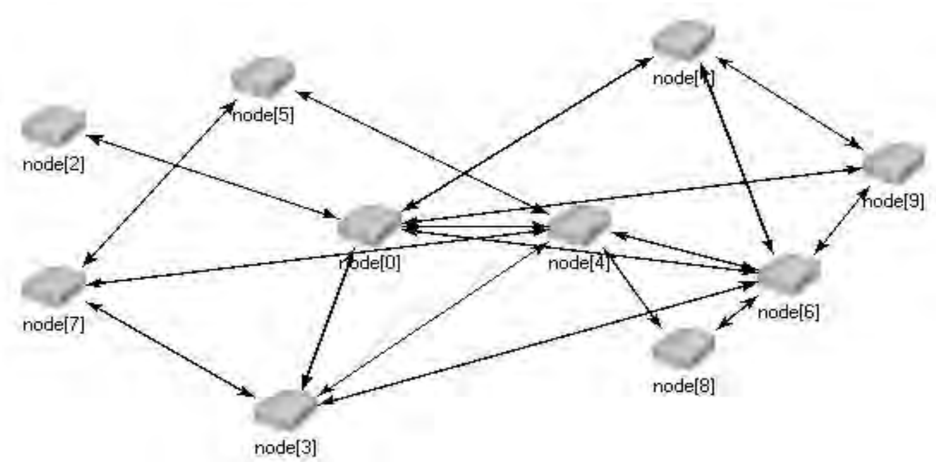


Figure5.1 A Sample Network Model for LL-MCLMAC Protocol.

5.5 Node Model

At the network level, the communicating devices are created and interconnected, whereas, at the node level, these devices are to be specified. Node models are conveyed by interconnected modules and each modules in a node model has its own functionality. Figure 5.2 shows the node model for each node in the network model. Node models ***appl*** module is used to generate packets. For designing the node module an***appl*** module has been used which generate packets addressed to a sink. ***Netwl*** module is used to implement network layer functionality. ***Nic*** module consists of two submodule: ***phy*** and ***mac***, shown in Figure 5.3. The lower module phy represents the physical layer which consists of a receiver and transmitter. Physical layer also implement analog model for path loss calculation and decider to consider receiver threshold before receiving. It also consider power consumption for Tx, Rx, SLEEP and radio switching state. ***Mac*** module is used to implement MAC layer functionalities and mobility of each node is controlled by mobility module. The purpose of arp module is to translate between layer 2 and layer 3 addresses. Battery is a simple battery module which provides updated battery related information to ***batteryStats*** module. ***BatteryStats*** module is used to collect and format statistical data from battery.

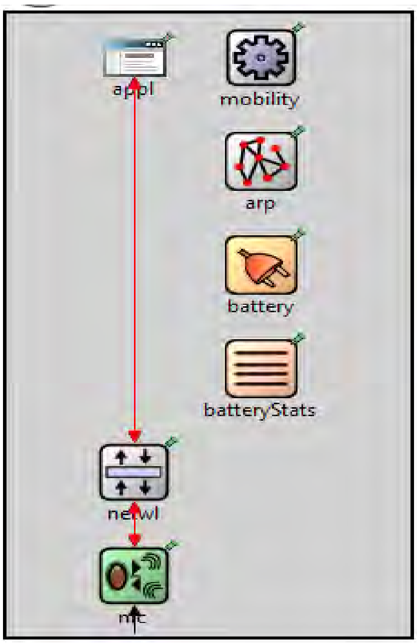


Figure 5.2 A Sample Node Model

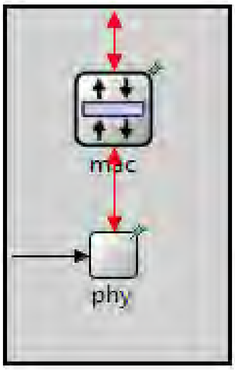


Figure 5.3 Components of Nic Module

5.6 Experiment Setup

In the experiment design, a node has been placed as a sink node and place it at the center of the simulation area and other nodes are placed randomly in the area. In this experiment, all nodes can generate data except the sink node, which can only receive data. The same simulation parameters shown in Table 5.1, have been set for LL-MCLMAC as in LMAC and MC-LMAC protocol. In this simulation a physical layer with energy model is also implemented to take into account the sending, receiving and sleeping energy consumptions of each nodes.

5.7 Simulation Parameters

For the performance evaluation of the proposed technique, the simulation model has been defined using the following simulation parameters:

Table 5.1

Simulation Parameters

Parameter	**Value**
Simulation Time	60 sec
Playground Size	500 x 500 x 500 m^3
Number of Nodes	25 to 200
Nodes Distribution	Randomly
Transmission Power	100 mW
Sensitivity	-84 dBm
Data Rate	100kbps
Carrier Frequency	2.4 GHz
Number of Channel	2
Number of Slot	16/32
Slot Duration	0.1 S
No. of Battery Per Node	1
Battery Capacity	1000 mAh
Sleep Current	0.02 mA
Rx Current	16.4 mA
Tx Current	17 mA
Data Packet Size	16 Bytes
Mobility Type	Stationary Mobility
Application Type	1 packet/sec

5.8 Evaluation Metrics

The main focus in this study is on exploiting the benefits of multiple timeslots of multiple channels in proposed LL-MCLMAC in terms of latency, throughput, energy consumption as well as network lifetime. As a performance metrics here the number of received packets is used for throughput, end-to-end delay for latency, energy consumption for each packet received by sink node and network lifetime.

5.8.1 Number of Received Packets

This performance metric is used for throughput by calculating the number of packet received packets by sink node from other nodes of its network in a certain amount of time.

5.8.2 End-To-End Delay

This performance metric is used to measure the average end-to-end delay of data packet transmission from any node of the networks to the desired destination node. The End-to-End delay is the average time difference between a packet initially sent by the source and successfully receiving the packet at the destination.

5.8.3 Energy Consumption

Here this metric is used to measure the amount of energy consumed per node per second for successfully receiving a packet by gateway node. Used power of a node is calculated by taking into account Sleep Current, Rx Current and Tx Current etc.

5.8.4 Network Lifetime

There are many ways to calculate the network lifetime. In this thesis work, the network lifetime has been calculated as the duration between the network startup time and the time when all the networking nodes get die due to lack of energy of all nodes.

5.9 Simulation Results

This subsection presents the simulation results. Here data calculated in the form of: number of received packets for throughput, end-to-end delay, and average energy consumption in a second for each packet received by sink node and network lifetime for five different network topologies and average the results. Some simulation code is shown in Appendix A, and some sample snapshot of simulation results are shown in Appendix B. Analysis of the performance of LL-MCLMAC protocol, with comparing LMAC and MC-LMAC protocol for above mentioned performance metrics are given below:

Number of Received Packets

Figure 5.4 presents the number of successfully received packets by the gateway node for 16 slots in a frame during the whole simulation time from different number of nodes in the network area. As can be seen from the figure, the number of received packets while using the proposed LL-MCLMAC protocol is higher than the existing LMAC as well as MC-LMAC protocols. This is due to the benefits of multiple timeslots of multiple channels exploited in the LL-MCLMAC protocol whereas in existing MC-LMAC protocol, a node is allowed to take control over one timeslot only from multiple channels.

The LL-MCLMAC protocol provides significant improvement in throughput from 25 nodes to 125 nodes i.e., when the traffic load increases and maximum throughput of around 370 packets is achieved at 125 node. However, the number of received packets for proposed MAC protocol and also for LMAC and MC-LMAC protocol decreases when the traffic load increases further. This is because the number of slots in a frame are limited, all nodes are not guaranteed more slots in such high traffic situation. Overall, the proposed LL-MCLMAC protocol outperforms the existing MC-LMAC protocol and better than LMAC protocol throughout the experiment as expected.

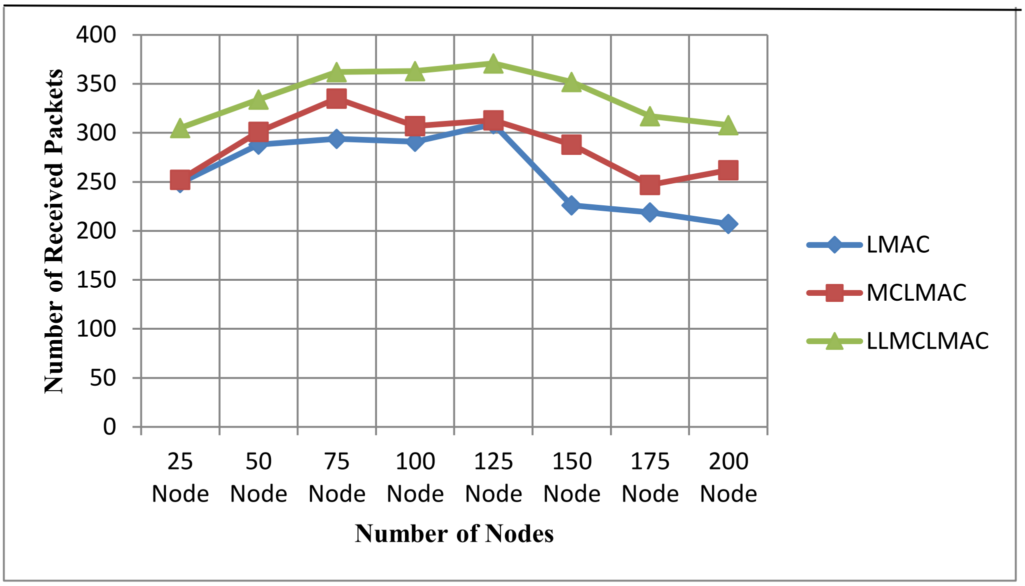


Figure 5.4 Number of Nodes vs. Number of Received Packets for 16 Slots in a Frame.

Figure 5.5 shows the number of successfully received packets by the gateway node for 32 slots in a frame during the whole simulation time from different number of nodes in the network area. From figure, it can be seen that, number of received packets are increased for all three protocols with the increasing number of nodes (from 25 nodes to 125 nodes) in the network area and maximum 338 packets are received for proposed MAC protocol at 125 nodes and then number of received packets for all three protocols are gradually decreases with the increase of number of nodes in the network area for the same reason of 16 slots in a frame. It can also be seen from the figure, for node 25 to 125, number of received packets for LMAC is greater than MC-LMAC and lower after 125 nodes. This is because now number of slots in a frame is 32 and both LMAC and MC-LMAC protocol uses a single slot for transmitting, so for low dense networks, chance of unavailable timeslot for sending is low and for that reason sending over single channel is enough. But with more increasing the number of nodes in the network, sending over single channel is not enough. For that reason now MC-LMAC performs better than single channel LMAC protocol. However, in all cases, the proposed LL-MCLMAC protocol outperforms the existing MC-LMAC and LMAC protocols throughout the experiment as expected.

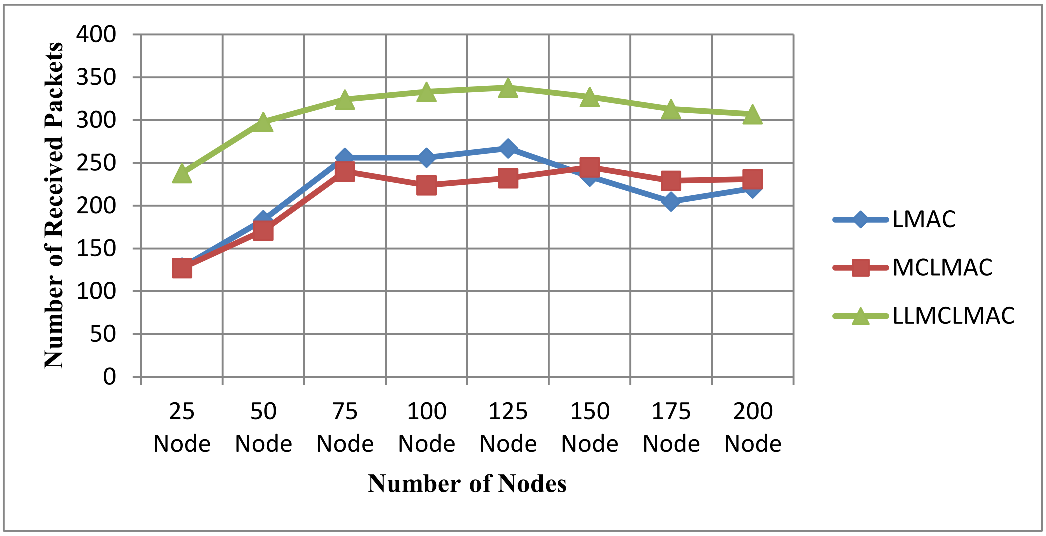


Figure 5.5 Number of Nodes vs. Number of Received Packets for 32 Slots in a Frame.

Figure 5.6 represents the number of received packets by gateway node for LL-MCLMAC protocol for both 16 slots in a frame and 32 slots in a frame during the whole simulation time with increases the number of nodes in the simulation area. Figure shows that, number of received packets for both 16 slots and 32 slots increases for 25 nodes to 125 nodes in the simulation area and then gradually decreases for both types. Figure 5.6 also reveals that the number of received packets for 16 slots in a frame always higher than the number of received packets for 32 slots in a frame. This is because for 32 slots in a frame, sending nodes have to wait more time to send than 16 slots in a frame.

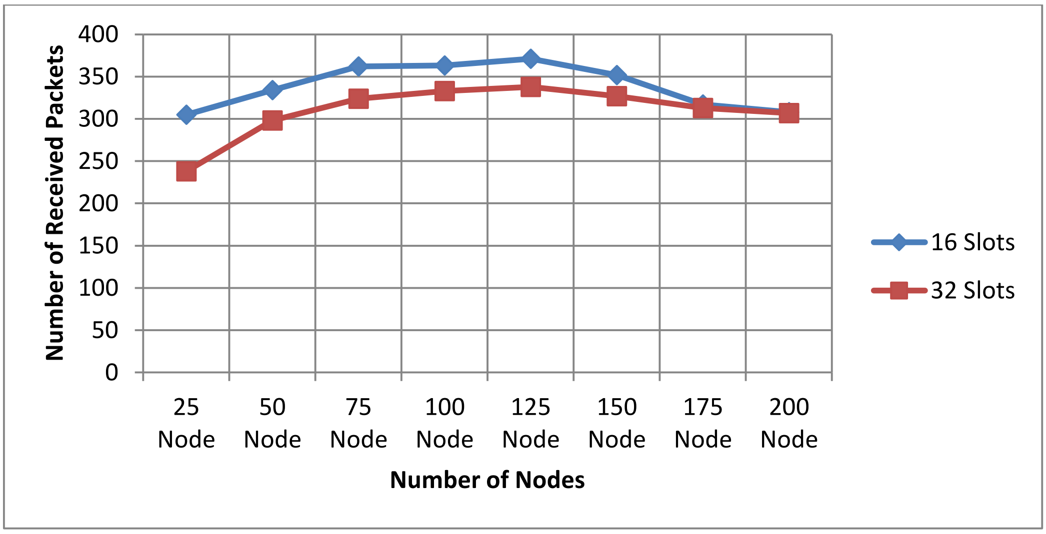


Figure 5.6 Number of Nodes vs. Number of Received Packets of LL-MCLMAC Protocol for 16 and 32 Slots in a Frame.

End to End Delay

The performance result illustrated in Figure 5.7 and Figure 5.8 (Figure 5.7 for 16 slots in a frame and 5.8 for 32 slots in a frame) shows the effect of employing multiple slots (two slots for this example) for each node on reduction of End-to-End delay. In both figure it is shown that the average End-to-End delay performance has been improved significantly while employing the proposed LL-MCLMAC protocol compared to the existing LMAC and MC-LMAC protocols. This is because multiple timeslots of a channel has been assigned to a node in this research whereas a node is allowed to transmit using only one timeslot for both the MC-LMAC as well as LMAC protocols. However, there is a gradual rise in the data latency of proposed MAC protocol with the increase of traffic load. For similar reasons as explained in the previous experiment, LL-MCLMAC is observed to have lower performance at high traffic load. In summary, it is observed that the delay levels for both LMAC and MC-LMAC protocols remain much higher than the proposed MAC protocol throughout the experiment. It also shows that although the existing MC-LMAC improves throughput performance of LMAC, the delay problem has not been resolved enough in MC-LMAC. However both figure confirms that the LL-MCLMAC always achieves a higher performance in lessening End to End delay compared to the existing MC-LMAC and LMAC protocols.

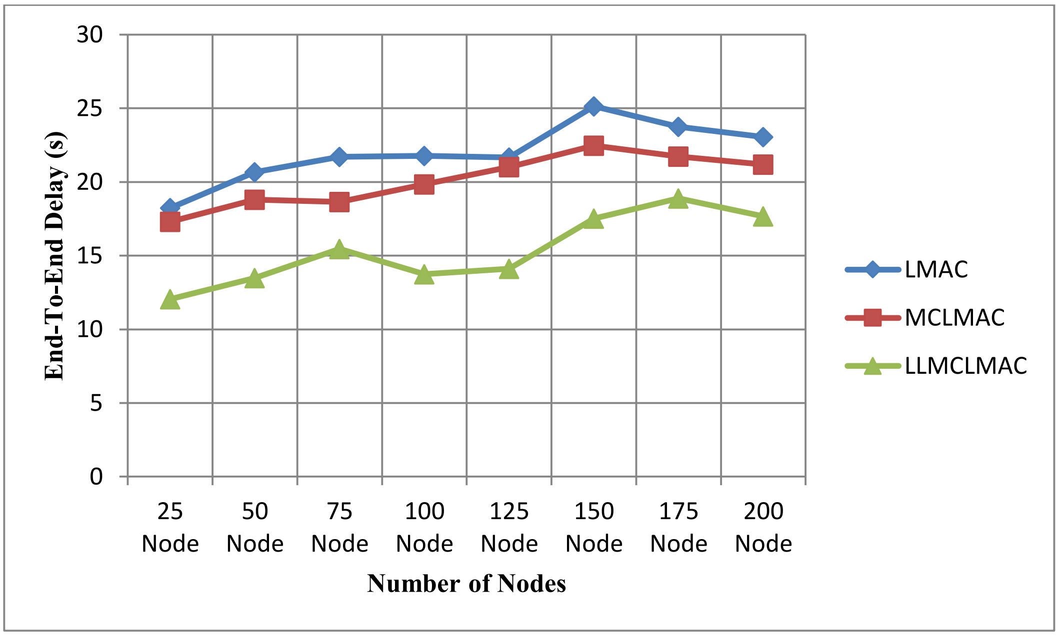


Figure 5.7 Number of Nodes vs. End-To-End Delay for 16 Slots in a Frame.

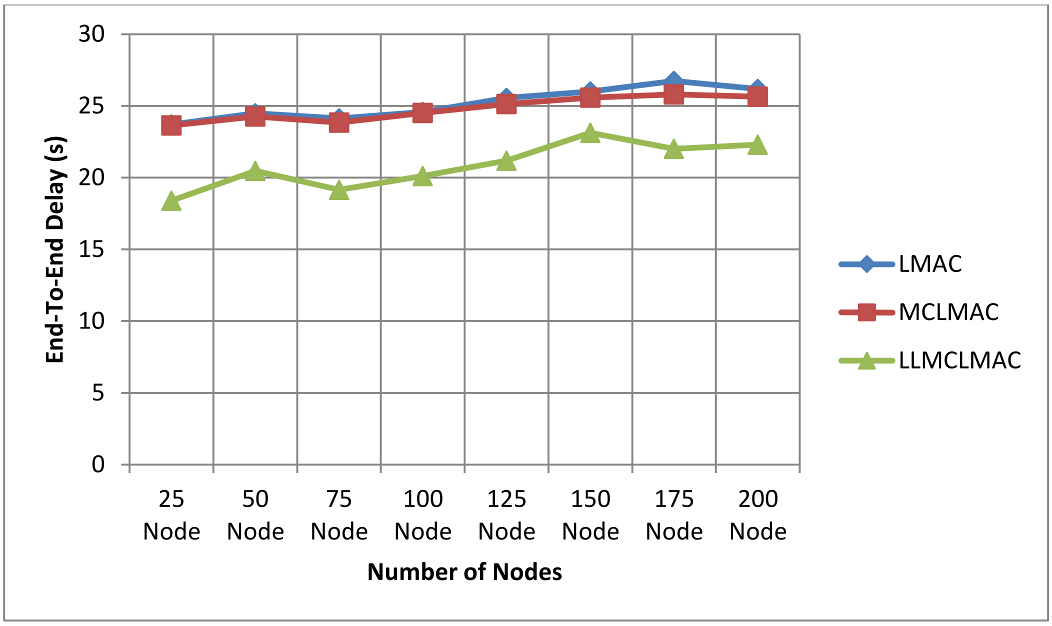


Figure 5.8 Number of Nodes vs. End-To-End Delay for 32 Slots in a Frame.

Figure 5.9 illustrates end-to-end packet delay of proposed LL-MCLMAC protocol for both 16 slots and 32 slots in a frame with increasing the number of nodes in the simulation area. Figure confirms that end-to-end delay for 16 slot in a frame is enough lesser than 32 slots in a frame for the same reason as explained for figure 5.6, Since for 32 slots in a frame, sending nodes have to wait more time to send than 16 slots in a frame as expected. Figure also shows that, end-to-end delay for both cases increases gradually with the increase of number of nodes in the simulation area.

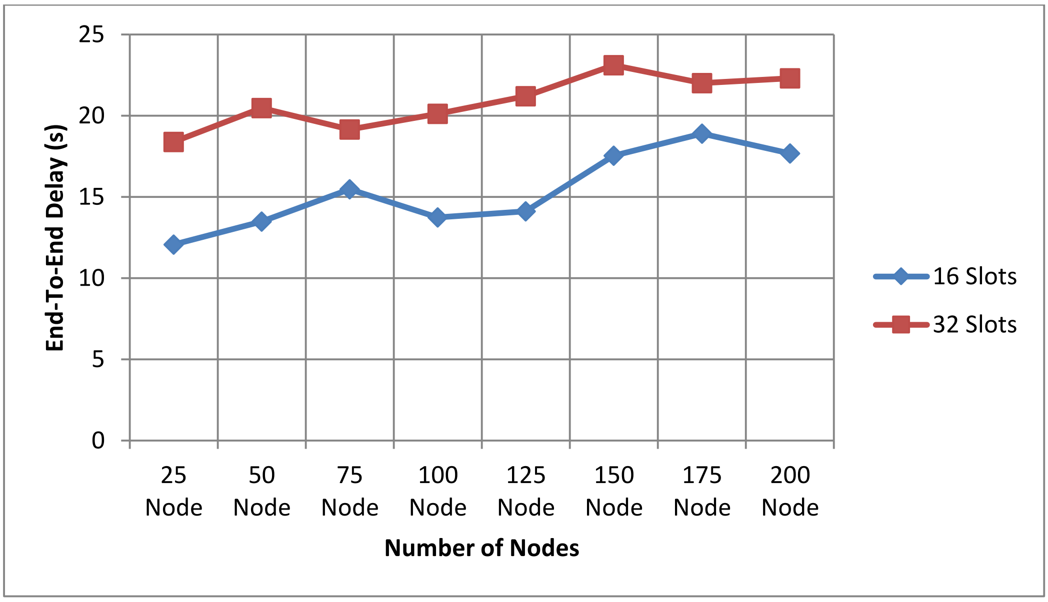


Figure 5.9 Number of Nodes vs. End-To-End Delay of LL-MCLMAC Protocol for 16 and 32 Slots in a Frame.

Energy Consumption

In Figure 5.10 and 5.11 (5.10 for 16 slots in a frame and 5.11 for 32 slots in a frame), The performance of the proposed MAC protocol are analyzed in terms of energy consumption of each node in a sec for successfully receiving a packet by sink node. Once again, the performance result presented in both figure confirms LL-MCLMAC's scalability with respect to energy consumption. It is observed that while the proposed MAC protocol uses the benefits of multichannel multislot concept to improve delay and throughput performance, in both cases (16 Slots and 32 Slots), for each packet received by sink node, it performs little more energy efficient operation (in terms of energy consumption for each packet received by sink node) than existing single channel LMAC protocol and a considerable improvement in energy consumptions compared to existing multichannel LMAC protocol throughout the experiment. In summary, for each packet received by sink node, multichannel LMAC protocol consumes much energy than single channel LMAC protocol and single channel LMAC protocol consumes more energy than proposed multichannel multislot LMAC protocol. So in conclusion, for each packet received by sink node, the proposed LL-MCLMAC is more energy efficient than existing LMAC and MC-LMAC protocol.

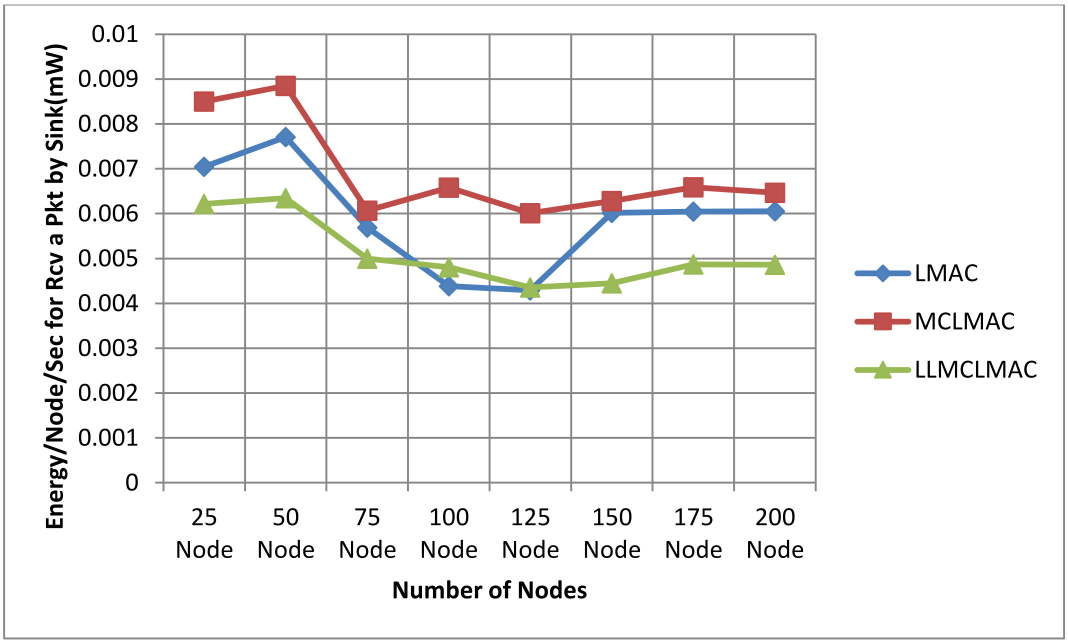


Figure 5.10 Number of Nodes vs. Energy Consumed per Node per Sec for Receiving a Packet by Sink for 16 Slots in a Frame.

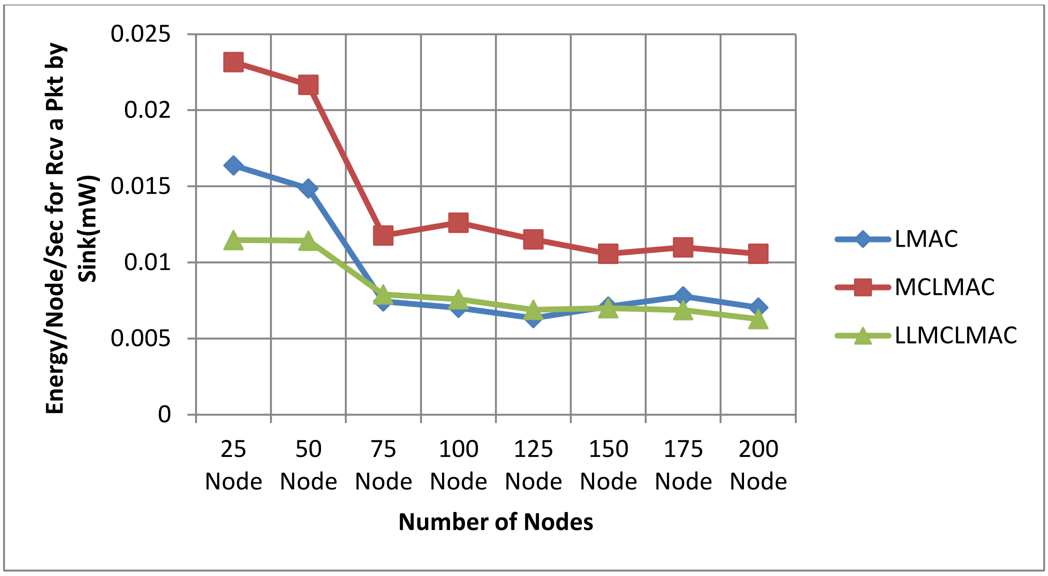


Figure 5.11 Number of Nodes vs. Energy Consumed per Node per Sec for Receiving a Packet by Sink for 32 Slots in a Frame.

The comparison of energy consumption of each node in a sec for successfully receiving a packet by sink node using proposed LL-MCLMAC protocol for 16 slots and 32 slots in a frame is illustrated in figure 5.12. Figure shows that energy consumption for 16 slots in a frame is less than 32 slots in a frame throughout the experiment. This is because in case of 16 slots in a frame, sending nodes have to wait less time than 32 slots in a frame, so number of received packet for 16 slots in a frame is greater than 32 slots in a frame. Hence, for 16 slots in a frame, energy consumption for receiving each packet is always less than energy consumption for receiving each packet for 32 slots in a frame.

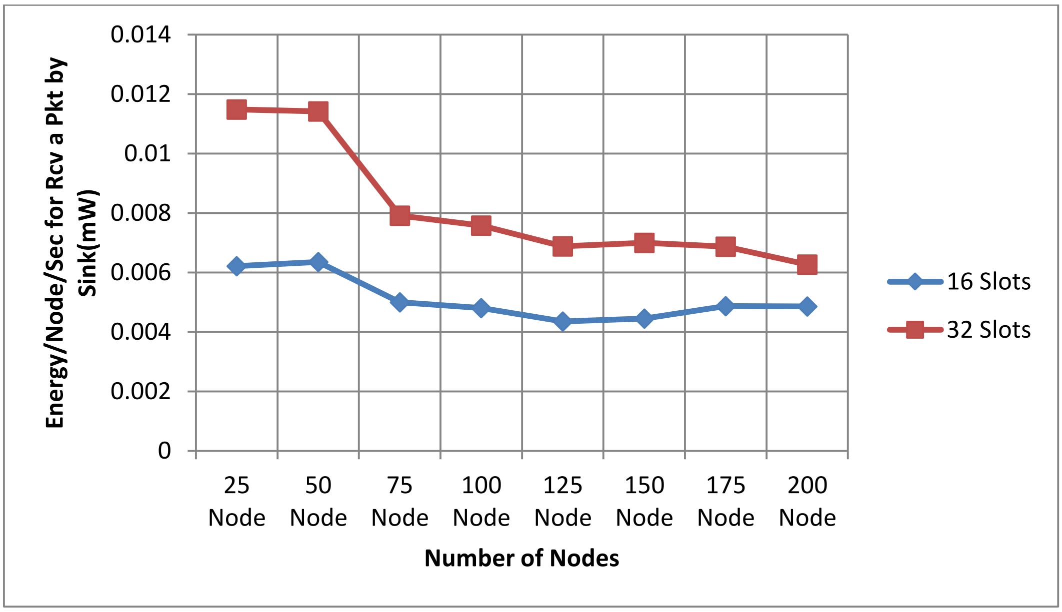


Figure 5.12 Number of Nodes vs. Energy Consumption of LL-MCLMAC Protocol for 16 and 32 Slots in a Frame.

Network Lifetime

For different node densities, using existing LMAC, MC-LMAC and proposed LL-MCLMAC protocol, the network lifetime for 16 slots and 32 slots in a frame is shown in figure 5.13 and figure 5.14 respectively. Both figure represents that, using LMAC protocol, the network lifetime is greater than multichannel MC-LMAC and LL-MCLMAC protocol. This is because, multichannel MC-LMAC and LL-MCLMAC protocol consumes higher energy than single channel LMAC protocol. But in spite of using two slots per frame, in proposed LL-MCLMAC protocol, it consumes less energy than existing MC-LMAC protocol. It is because, LL-MCLMAC protocol ensures less end-to-end delay than MC-LMAC protocol. So for same traffic load, LL-MCLMAC protocol ensure to receive packets within short time (less delay and less waiting time) than using MC-LMAC protocol, which results less energy consumption of using LL-MCLMAC protocol than MC-LMAC protocol. So the network lifetime of LL-MCLMAC protocol is greater than the existing MC-LMAC protocol but less than existing LMAC protocol. Figures also shows that, with increasing the number of nodes in the network, the network lifetime is gradually increasing for all three protocols as expected.

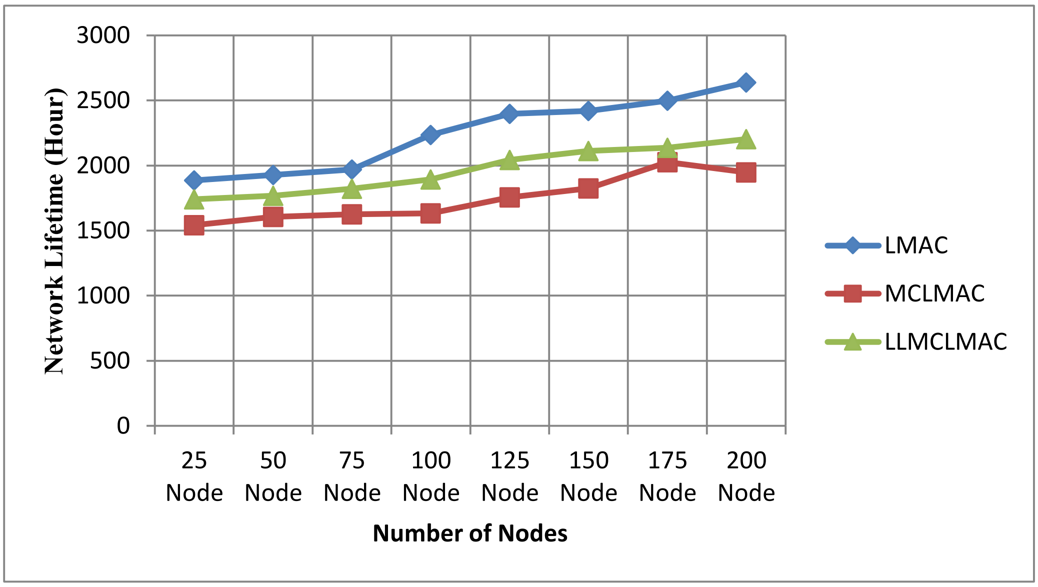


Figure 5.13 Number of Nodes vs. Network Lifetime for 16 Slots in a Frame.

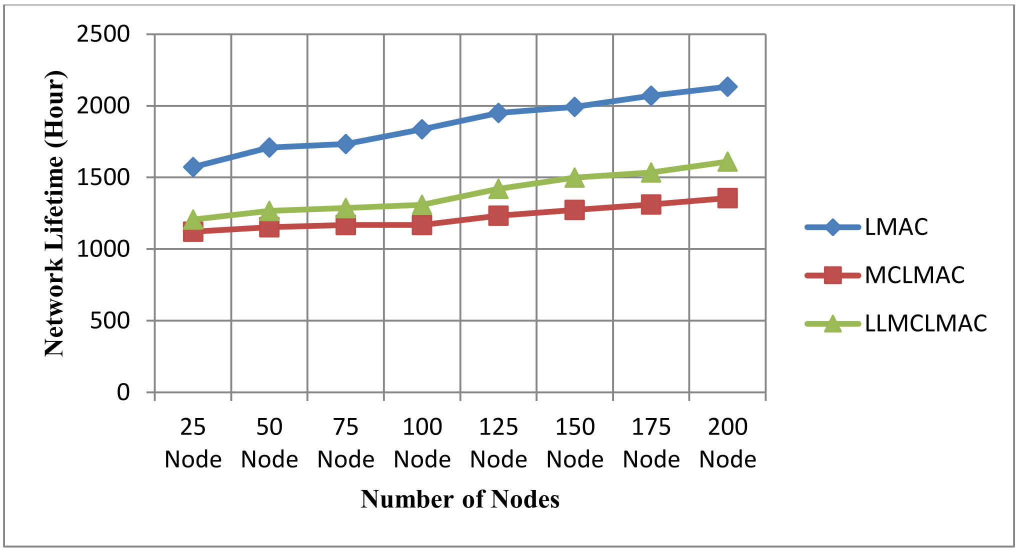


Figure 5.14 Number of Nodes vs. Network Lifetime for 32 Slots in a Frame.

Figure 5.15 shows the network lifetime of proposed protocol for 16 slots and 32 slots in a frame. Figure represents that, for same traffic load, the network lifetime of using 16 slots per frame is always greater than 32 slots per frame and the network lifetime is increasing with the increase of number of nodes in the network. Energy consumption for 16 slots in a frame is lesser (greater lifetime) than 32 slots in a frame. Because using 16 slots in a frame ensures less end-to-end delay than 32 slots in a frame (see figure 5.9) and results less energy consumption than 32 slots in a frame (see figure 5.12). Which results long network lifetime using 16 slots in a frame than 32 slots in a frame.

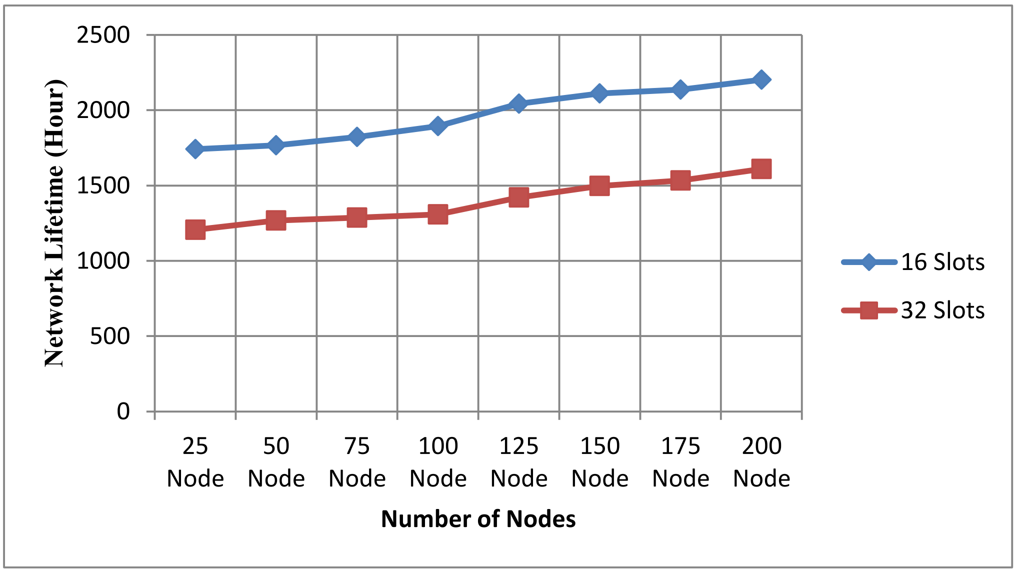


Figure 5.15 Number of Nodes vs. Network Lifetime of LL-MCLMAC Protocol for 16 and 32 Slots in a Frame.

5.10 Summary

The performance of the proposed LL-MCLMAC protocol have been analyzed and establish a comparison with existing LMAC and MC-LMAC protocol in terms of number of received packets for throughput, end-to-end delay, energy consumption for each packet received by sink node and network lifetime throughout this chapter. The analysis shows that the proposed LL-MCLMAC protocol performs better than existing LMAC and MC-LMAC protocol by increasing number of received packets, decreasing end-to-end delay and energy consumption for each packet received by sink node. However, the network lifetime of proposed protocol is less than existing single channel LMAC protocol but greater than existing multichannel MC-LMAC protocol.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, LL-MCLMAC, "Low Latency Multichannel Lightweight Medium Access Control" protocol have been presented for wireless sensor networks. Since, sensor nodes are generally battery powered and may not recharge easily, so to prolong the lifetime of the nodes is an important issue while designing a MAC protocol. However, the proposed LL-MCLMAC protocol has good energy efficiency because it uses scheduled operations in which sensor nodes are allowed to sleep for significant amount of time. There is a direct trade-off between energy efficiency and latency, because lowering the energy consumption may result in high latency and this types of high latency MAC protocols may not suitable for delay sensitive WSN applications. Another category of WSN applications which requires high throughput to fulfill the application requirements. However, for almost all types of WSN applications energy efficiency is must. By keeping these in mind, in this thesis work, low latency multichannel LMAC protocol, LL-MCLMAC protocol is presented that exploits the benefits of multiple slots of a channel from multiple channels to improve end-to-end delivery latency as well as throughput while ensuring energy efficient operations. Since, the proposed protocol is based on LMAC and MC-LMAC protocol, using different performance parameters, the performance of the proposed LL-MCLMAC protocol with the existing LMAC and MC-LMAC protocols has been analyzed in chapter 5. Performance analysis was carried out in terms of number of received packets for throughput, end-to-end delay, energy consumption for each packet received by sink node and network lifetime. Performance result confirms LL-MCLMAC's scalability to improve delay and throughput performance while ensuring an energy efficient operation compared to existing MC-LMAC and LMAC protocols. However, the network lifetime of proposed protocol is less than existing single channel LMAC protocol but greater than existing multichannel MC-LMAC protocol. Some of the limitations of proposed protocol may include highly application dependency and need multichannel transceiver to be used in all nodes which increase the price of each node.

Finally, it can be said that, the proposed LL-MCLMAC protocol outperforms the existing LMAC and MC-LMAC protocols and will be well suited for wireless sensor network applications that require low latency and high throughput such as intrusion detection, security and tactical surveillance applications or sampling the environment with fine temporal resolution.

6.2 Recommendation for Future Works

Proposed LL-MCLMAC protocol can be analyzed for more channels and by allocating more timeslots for each nodes. This proposed protocol may be improved to increase more energy efficiency and hence to increase the network lifetime. Performance of proposed protocol can also be improved by incorporating new techniques to minimize co-channel and adjacent channel interference. Performance of this protocol can also be tested for many network scenarios to find others applications for which using it will be suitable. In future, the performance of the proposed protocol can also be compared with some related recent protocols.

Finally, the proposed protocol can be tested in testbed and practical WSN applications as well.

REFERENCES

[1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," Proceedings of the 33rd Hawaii International Conference on System Sciences, January 2000, pp. 174–185.

[2] T. Nieberg, S. Dulman, P. Havinga, L. V. Hoesel and J. Wu, "Collaborative Algorithms for Communication in Wireless Sensor Networks," Ambient Intelligence: Impact on Embedded Systems, Kluwer Academic Publishers, ISBN 1-4020-7668-1, November 2003.

[3] L. F. W. van Hoesel and P. J. M Havinga, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches," Proceedings of the 1st International Conference on Networked Sensing Systems (INSS), Tokyo, Japan, June 2004.

[4] G. Lu, B. Krishnamachari and C. S. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering Sensor Networks," 4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks held in conjunction with the IEEE Conference 18th International Parallel and Distributed Processing Symposium, 2004. pp. 1-12.

[5] R. Kalidindi, L. Ray, R. Kannan and S. Iyengar, "Distributed Energy Aware MAC Layer Protocol for Wireless Sensor Networks," Proceedings of the International Conference on Wireless Networks (ICWN03),Las Vegas, NV, June 2003, pp. 282-286.

[6] S. Chatterjea, L. van Hoesel, and P. Havinga, "AI-LMAC: An Adaptive, Information-centric and Lightweight MAC Protocol for Wireless Sensor Networks," Proceedings of the Intelligent Sensors, Sensor Networks, and Information Processing, December 2004, pp. 381–388.

[7] O. D. Incel, P. Jansen, S. Mullender, "MC-LMAC: A Multi-Channel MAC Protocol for Wireless Sensor Networks," Technical Report of TR-CTIT-08-61; Centre for Telematics and Information Technology, University of Twente: Enschede, the Netherlands, 2008, pp. 1381-3625.

[8] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Multichannel Scheduling for Fast Convergecast in Wireless Sensor Networks," University of Southern California, Technical Report CENG-2008-9, September 2008.

[9] Md. A. Hamid, M. A. A. Wadud and I. Chong, "A Schedule-based Multi-channel MAC Protocol for Wireless Sensor Networks," Journal of Sensors (Basel, Switzerland), vol. 10, Issue. 10, pp. 9466-9480, October 2010.

[10] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "A Dynamic Multichannel Energy-Efficient MAC Protocol for Wireless Sensor Networks," Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'11), Paris, France, May 2011.

[11] G. Wang, J. Yu, . Yu, H. Yu, L. Feng, P. Liu, "DS-MAC: An Energy Efficient Demand Sleep MAC Protocol with Low Latency for Wireless Sensor Networks," Journal of Network and Computer Applications, vol. 58, pp. 155-164, December 2015.

[12] S. Gajjar, N. Choksi, M. Sarkar, K. Dasgupta, "LEFT: A Latency and Energy Efficient Flexible TDMA Protocol for Wireless Sensor Networks," International Journal of Computer Network and Information Security, DOI: 10.5815/ijcnis.2015.02.01, vol. 7, no. 2, pp. 1-14, January 2015.

[13] T. N. Quynh, V. T. Quang, Q. N. Trung, "A Low Latency Communication Protocol for Target Tracking in Wireless Sensor Networks," EURASIP Journal on Wireless Communications and Networking, DOI:10.1186/s13638-016-0517-4, January 2016.

[14] A. K. Azad, Md. H. Kabir and Md. B. Hossain, "A Survey on Schedule-Based MAC Protocols for Wireless Sensor Networks," IJCSN, International Journal of Computer Science and Network, vol. 2, Issue. 6, pp. 120-128, November 2013.

[15] P. Radmand, A. Talevski, S. Petersen and S. Carlsen, "Comparison of Industrial WSN Standards," 4th IEEE International Conference on Digital Ecosystems and Technologies, IEEE DEST, 2010, pp. 632-637.

[16] T. Lennvall, S. Svensson and F. Hekland, "A Comparison of Wireless HART and Zigbee for Industrial Applications," IEEE International Workshop on Factory Communication Systems, WFCS 2008, pp. 85-88, May 2008.

[17] Analysis of Wireless Industrial Automation Standards: ISA-100.11a and WirelessHART. Available online: https://www.isa.org/standards-publications/isa-publications/intech-magazine/2012/december/web-exclusive-analysis-wireless-industrial-automation standards-isa-100-11a-wirelesshart/. (Last accessed on- 16 July, 2016).

[18] P. D. Marco, "Protocol Design and Implementation for Wireless Sensor Networks," KTH Electrical Engineering, Stockholm, Sweden, April 2008.

[19] Protocol Stack for Wireless Sensor Networks (WSNs). Available online: https://sindarku.wordpress.com/2010/08/13/protocol-stack-for-wireless-sensor-networks-wsns. (Last accessed on- 16 July, 2016).

[20] H. Karl and A. Willig, "Protocols and Architectures for Wireless Sensor Networks," England: John Wiley & Sons, Ltd. ISBN: 0-470-09510-5, 2005.

[21] S. Dewangan, A. K. Pandey, N. Verma and D. Xaxa, "A Comparative Assessment of Topologies and Their Issuesin Wireless Sensor Networks," International Journal of Engineering Sciences& Research Technology, vol. 5, Issue. 5, pp. 388-393, May 2015.

[22] K. Lorincz, D. J. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor Networks for Emergency Response: Challenges and Opportunities," IEEE Pervasive Computing, vol. 3, Issue. 4, pp. 16-23, October 2004.

[23] X. Chao, W. Dargie, and G. Lin, "Energy Model for h2s Monitoring Wireless Sensor Network," Proceedings of the 2008 11th IEEE International Conference on Computational Science and Engineering, IEEE Computer Society, Washington, DC, USA, 2008, pp. 402-409.

[24] S. Dagtas, Y. Natchetoi, and H. Wu, "An Integrated Wireless Sensing and Mobile Processing Architecture for Assisted Living and Healthcare Applications," Proceedings of the 1st ACM SIGMOBILE International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments, New York, NY, USA, June 2007, pp. 70-72.

[25] M. Cheng, M. Kanai-Pak, N. Kuwahara, H. I. Ozaku, K. Kogure, and J. Ota, "Dynamic Scheduling Based Impatient Nursing Support: Applicability Evaluation by Laboratory Experiments," Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems, New York, NY, USA, May 2009, PP. 48-54.

[26] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless Sensor Network Design for Tactical Military Applications: Remote Large-scale Environments," Proceedings of the 28th IEEE Conference on Military Communications, Piscataway, NJ, USA, October 2009, pp. 911-917.

[27] C. C. Shen, C. Srisathapornpat, and C. Jaikaeo, "Sensor Information Networking Architecture and Applications," IEEE Personal Communications, vol. 8, Issue. 4, pp. 52-59, August 2001.

[28] A. Cerpa, J. Elson, D. Estrin, L. Gird, M. Hamilton, and J. Zhao. "Habitat Monitoring: Application Driver for Wireless Communications Technology," Proceedings of the ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, San Jose, Costa Rica, April 2001, pp. 20-41.

[29] C. Intanagonwiwat R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networks," IEEE/ACM Transactions on Networking, vol. 11, Issue. 1, pp. 2-16, February 2003.

[30] M. Zuniga and B. Krishnamachari, "Integrating Future Large Scale Sensor Networks with the Internet," Technical Report CS 03-792, University of Southern California, Los Angeles, Department Computer Science, 2003.

[31] K. Egevang and P. Francis, "The IP Network Address Translator (NAT)," RFC 1631, May 1994.

[32] M. Sgroi, A. Wolisz, A. Sangiovani-Vincentelli, and J. M. Rabaey, "A Service-Based Universal Application Interface for Ad-Hoc Wireless Sensor Networks," White Paper, Private Communication, November 2003.

[33] N. Glance, D. Snowdown, and J. L. Meunier, "Pollen: Using People as a Communication Medium," Computer Networks, vol. 35, Issue. 4, pp. 429-442, February 2001.

[34] B. A. Forouzan, "Data Communications and Networking, Fourth Edition," Ubiquitous Computing and Communication Journal, ISBN 978-0-07-296775-3, vol. 4, no. 3, August 2009.

[35] T. S. Rappaport, "Wireless Communications- Principles and Practice," Prentice Hall, Upper Saddle River, 2nd ed., NJ, ISBN:0130422320, 2002.

[36] F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II the Hidden Terminal Problem in CSMA and Busy-Tone Solutions," IEEE Transactions on Communications, vol. 23, Issue. 12, pp. 1417–1433, 1975.

[37] C.K. Toh,"A Novel Distributed Routing Protocol to Support Ad Hoc Mobile Computing," Proceedings of IEEE 15th Annual International Conference on Computers and Communications, 1996, pp.460–486.

[38] P. Karn, "A New Channel Access Method for Packet Radio," Proceedings of the ARRL/CRRL Amateur Radio 9th Computer Networking Conference, September 1990, pp.134–140.

[39] A. Chandra, V. Gummalla and John O. Limb, "Wireless Medium Access Control Protocols," IEEE Communications Surveys and Tutorials, vol. 3, Issue. 2, pp. 2–15, November 2009.

[40] S. Gowrishankar, T. G. Basavaraju, D. H. Manjaiah and S. K. Sarkar, "Issues in Wireless Sensor Networks," Proceedings of the World Congress on Engineering 2008, ISBN:978-988-98671-9-5, London, U.K. vol. 1, July 2008, pp. 176-187.

[41] W. Ye, J. Heidemann and D. Estrin, "An Energy Efficient MAC Protocol for Wireless Sensor Networks," IEEE INFOCOM, New York, vol. 3, pp. 1567-1576, June 2002.

[42] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC Protocols for Wireless Sensor Networks: a Survey," IEEE Communications Magazine, vol.44, Issue.4, pp. 115-121, April 2006.

[43] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," IEEE/ACM Trans. Net, vol. 12, no. 3, pp. 493–506, June 2004.

[44] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-Aware Wireless Microsensor Networks," IEEE Signal Processing Magazine, vol. 19, Issue. 2, pp. 40–50, March 2002.

[45]T. V. Dam and K. Langendoen, "An Adaptive Energy Efficient MAC Protocol for Wireless Networks," Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, November 2003, pp. 171-180.

[46] S. Agarwal, V. Jain and K. Goswami, "Energy Efficient MAC Protocolsfor Wireless Sensor Network," International Journal on Computational Sciences & Applications (IJCSA), DOI:10.5121/ijcsa.2014.4115,vol.4, no.1, pp. 153-160, February 2014.

[47] R. Yadav, S. Varma and N. Malaviya, "ASurveyof MAC Protocolsfor Wireless Sensor Networks," UbiCC Journal, vol.4, no.3, pp. 827-833, August 2009.

[48] Z. Haiyang, "Classic Efficient-Energy MAC Protocols for Wireless Sensor Networks," Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on Digital Object Identifier: 10.1109/WICOM.2010.5601365, 2010.

[49] I. Cidon and M. Sidi, "Distributed Assignment Algorithms for Multihop Packet Radio Networks," IEEE Transactions on Computers, vol. 38, Issue. 10, pp. 1353–1361, October 1989.

[50] OMNet++ network simulator. Available online: http://www.omnetpp.org/omnetpp/cat_view/17-downloads/1-omnet-releases. (Last accessed on- 16 July, 2016).

[51] MiXiMmodeling framework. Available online: http://www.http://sourceforge.net/projects/mixim/files/mixim/MiXiM%202.3. (Last accessed on- 16 July, 2016).

[52] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," Proceedings of the 1st ACM Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, September 2002, pp. 88-97.

APPENDIX A

Thesis Simulation Code

LL-MCLMAC Simulation Code:

omnetpp.ini file:

```
[General]
cmdenv-express-mode = true
ned-path = ../../src;..
network = LLMCLMACExampleNetwork
sim-time-limit = 60s
record-eventlog = false
eventlog-file = ${resultdir}/${configname}-1.elog
##############################################################
#                    Simulation parameters                   #
##############################################################
tkenv-image-path = ../../images;
**.coreDebug = false
**.debug = true
**.playgroundSizeX = 500m
**.playgroundSizeY = 500m
**.playgroundSizeZ = 500m
##############################################################
#                    WorldUtility parameters                 #
##############################################################
**.world.useTorus = false
##############################################################
#                    channel parameters                      #
##############################################################
**.connectionManager.sendDirect = false
**.connectionManager.pMax = 100mW
**.connectionManager.sat = -84dBm
**.connectionManager.alpha = 3.0
**.connectionManager.carrierFrequency = 2.412e+9Hz
################ PhyLayer parameters #####################
**.node[*].nic.phy.usePropagationDelay = false
**.node[*].nic.phy.recordStats = true
**.node[*].nic.phy.thermalNoise = -100dBm
**.node[*].nic.phy.useThermalNoise = true
**.node[*].nic.phy.analogueModels = xmldoc("config.xml")
**.node[*].nic.phy.decider = xmldoc("config.xml")
**.node[*].nic.phy.timeRXToTX = 0s
**.node[*].nic.phy.timeRXToSleep = 0s
**.node[*].nic.phy.timeTXToRX = 0s
**.node[*].nic.phy.timeTXToSleep = 0s
**.node[*].nic.phy.timeSleepToRX = 0s
**.node[*].nic.phy.timeSleepToTX = 0s
**.node[*].nic.phy.sensitivity = -84dBm
**.node[*].nic.phy.maxTXPower = 100.0mW
```

```
**.node[*].nic.phy.initialRadioState = 0
**.node[*].nic.phy.initialRadioChannel = 1
################ MAC layer parameters ####################
**.node[*].nic.mac.headerLength = 24bit
# the length of the queue - if exceeded, new packets are dropped
**.node[*].nic.mac.queueLength = 50
**.node[*].nic.mac.bitrate = 100000bps
**.node[*].nic.mac.txPower = 100
**.node[*].nic.mac.defaultChannel = 0
# duration of one slot
**.node[*].nic.mac.slotDuration = 0.1s
# number of slots
**.node[*].nic.mac.numSlots = 16
**.node[*].nic.mac.reservedMobileSlots=0
**.node[0].mobility.initialX = 200m
**.node[0].mobility.initialY = 200m
**.node[0].mobility.initialZ = 100m

**.node[1].mobility.initialX = 300m
**.node[1].mobility.initialY = 300m
**.node[1].mobility.initialZ = 200m

**.node[2].mobility.initialX = 225m
**.node[2].mobility.initialY = 160m
**.node[2].mobility.initialZ = 300m

**.node[3].mobility.initialX = 150m
**.node[3].mobility.initialY = 200m
**.node[3].mobility.initialZ = 400m

**.node[4].mobility.initialX = 250m
**.node[4].mobility.initialY = 250m
**.node[4].mobility.initialZ = 250m

**.node[5].mobility.initialX = 350m
**.node[5].mobility.initialY = 300m
**.node[5].mobility.initialZ = 100m

**.node[6].mobility.initialX = 230m
**.node[6].mobility.initialY = 180m
**.node[6].mobility.initialZ = 200m

**.node[7].mobility.initialX = 150m
**.node[7].mobility.initialY = 200m
**.node[7].mobility.initialZ = 300m

**.node[8].mobility.initialX = 250m
**.node[8].mobility.initialY = 300m
**.node[8].mobility.initialZ = 100m

**.node[9].mobility.initialX = 300m
**.node[9].mobility.initialY = 220m
**.node[9].mobility.initialZ = 200m

**.node[10].mobility.initialX = 350m
```

```
**.node[10].mobility.initialY = 250m
**.node[10].mobility.initialZ = 300m

**.node[11].mobility.initialX = 325m
**.node[11].mobility.initialY = 240m
**.node[11].mobility.initialZ = 250m

**.node[*].applicationType = "SensorApplLayer"
**.appl.trafficType = "exponential"
**.appl.trafficParam = 1s
**.appl.nbPackets = 50
**.appl.initializationTime = 0.1s
**.appl.destAddr = 4
**.node[*].nic.mac.sinkAddr = 4  #destAddr and sink both shuld be the same address.
**.node[4].appl.broadcastPackets = true
**.appl.stats = true
**.node[*].mobilityType = "StationaryMobility"
**.node[*].mobility.debug = false
**.node[*].mobility.initFromDisplayString = false
**.node[*].netwl.stats = false
**.node[*].netwl.headerLength = 32bit
**.batteryStats.detail = true
**.batteryStats.timeSeries = false
**.battery.nominal = 1000mAh
**.battery.capacity = 1000mAh
**.battery.voltage = 3.3V
**.battery.resolution = 0.1s
**.battery.publishDelta = 0
**.battery.publishTime = 0.5s
**.battery.numDevices = 1
**.nic.sleepCurrent = 0.02mA
**.nic.rxCurrent = 16.4mA
**.nic.decodingCurrentDelta = 0mA
**.nic.txCurrent = 17mA
**.nic.setupRxCurrent = 8.2mA
**.nic.setupTxCurrent = 8.2mA
**.nic.rxTxCurrent = 17mA
**.nic.txRxCurrent = 17mA
```

LLMCLMAC.ned file:

```
package org.mixim.examples.llmclmac;
import org.mixim.base.modules.BaseNetwork;
import org.mixim.modules.node.HostLLMCLMAC;
networkLLMCLMACExampleNetwork extends BaseNetwork
{
parameters:
int numNodes; // total number of hosts in the network
    numNodes @prompt("Enter Total Number of Nodes:");
@display("bgb=519,339");
submodules:
    node[numNodes]: HostLLMCLMAC {
parameters:
@display("p=147,95;i=device/smallrouter;is=s");
    }
```

```
connectionsallowunconnected:
}
```

HostLLMCLMAC.ned file:

```
package org.mixim.modules.node;
module HostLLMCLMAC extends WirelessNodeBattery
{
parameters:
	nicType = default("org.mixim.modules.nic.NicLLMCLMAC");
	arpType = default("org.mixim.modules.netw.ArpHost");
@display("i=misc/node2_vs");
}
```

WirelessNodeBattery.node file:

```
package org.mixim.modules.node;
import org.mixim.modules.power.battery.BatteryStats;
import org.mixim.modules.power.battery.SimpleBattery;
module WirelessNodeBattery extends WirelessNode
{
parameters:
	applicationType = default("BurstApplLayerBattery");  //type of the application layer
@display("i=misc/node2_vs");
submodules:
	batteryStats: BatteryStats {
@display("p=140,240;i=block/table,#FF8040");
	}
	battery: SimpleBattery {
@display("p=140,170;i=block/plug,#FF8000");
	}
}
```

WirelessNode.ned file:

```
package org.mixim.modules.node;
import org.mixim.base.modules.IMobility;
import org.mixim.modules.nic.IWirelessNicUpperCtrl;
import org.mixim.base.modules.IBaseArp;
import org.mixim.base.modules.IBaseApplLayer;
import org.mixim.base.modules.IBaseNetwLayer;
module WirelessNodeNetwl
{
parameters:
string networkType = default("BaseNetwLayer");//type of the network layer
string mobilityType = default("ConstSpeedMobility"); //type of the mobility module
string arpType = default("BaseArp"); //type of address resolution module
string nicType;
@display("i=misc/node2_vs");
gates:
input radioIn; // gate for sendDirect
submodules:
	arp: <arpType>like IBaseArp {
@display("p=140,100;i=block/network2");
	}
	mobility: <mobilityType>like IMobility {
```

```
parameters:
@display("p=140,30;i=block/cogwheel");
        }
        nic: <nicType>like IWirelessNicUpperCtrl {
parameters:
@display("p=70,380;i=block/wrxtx,#008000");
        }
        netwl: <networkType>like IBaseNetwLayer {
parameters:
@display("p=70,310;i=block/layer");
        }
connectionsallowunconnected:
        nic.upperLayerOut --> netwl.lowerLayerIn;
        nic.upperLayerIn <-- netwl.lowerLayerOut;
        nic.upperControlOut --> { @display("ls=red;m=m,70,0,70,0"); } --> netwl.lowerControlIn;
        nic.upperControlIn <-- { @display("ls=red;m=m,70,0,70,0"); } <-- netwl.lowerControlOut;
        radioIn --> nic.radioIn;
}
module WirelessNode extends WirelessNodeNetwl
{
parameters:
string applicationType = default("BurstApplLayer");  //type of the application layer
@display("i=misc/node2_vs");

submodules:
    appl: <applicationType>like IBaseApplLayer {
parameters:
@display("p=70,30;i=app");
        }
connections:
        netwl.upperLayerOut --> appl.lowerLayerIn;
        netwl.upperLayerIn <-- appl.lowerLayerOut;
        netwl.upperControlOut --> { @display("ls=red;m=m,70,0,70,0"); } -->appl.lowerControlIn;
        netwl.upperControlIn <-- { @display("ls=red;m=m,70,0,70,0"); } <--appl.lowerControlOut;
}
```

NicLLMCLMAC.ned file:

```
package org.mixim.modules.nic;
module NicLLMCLMAC extends WirelessNicBattery
{
parameters:
            macType = default("LLMCLMacLayer");
            phy.initialRadioChannel = default(1);
            phy.nbRadioChannels     = default(15); //1 to 14, 0 is invalid
            phy.decider      = default(xmldoc("NicLLMCLMac_Decider.xml"));
}
```

WirelessNicBattery.ned file

```
package org.mixim.modules.nic;
module WirelessNicBattery extends WirelessNic
{
parameters:
      phyType = default("org.mixim.modules.phy.PhyLayerBattery");
double sleepCurrent @unit(mA);
```

```
double rxCurrent @unit(mA);
double decodingCurrentDelta @unit(mA);
double txCurrent @unit(mA);
double setupRxCurrent @unit(mA);
double setupTxCurrent @unit(mA);
double rxTxCurrent @unit(mA);
double txRxCurrent @unit(mA);
}
```

WirelessNic.ned file:

```
package org.mixim.modules.nic;
import org.mixim.base.phyLayer.IWirelessPhy;
import org.mixim.base.modules.IWirelessMac;
module WirelessNic like IWirelessNicUpperCtrl
{
parameters:
string connectionManagerName = default("");   //name of the ConnectionManager module
gates:
input upperLayerIn; // to upper layers
output upperLayerOut; // from upper layers
output upperControlOut; // control information
input upperControlIn; // control information
input radioIn; // radioIn gate for sendDirect
submodules:
	mac: <macType>like IWirelessMac {
@display("p=52,51;i=block/layer");
	}
	phy: <phyType>like IWirelessPhy {
@display("p=52,125;i=block/process_s");
	}
connections:
	mac.upperLayerOut --> { @display("ls=black;m=m,25,50,25,0"); } --> upperLayerOut;
	mac.upperLayerIn <-- { @display("ls=black;m=m,15,50,15,0"); } <-- upperLayerIn;
	mac.upperControlOut --> { @display("ls=red;m=m,75,50,75,0"); } --> upperControlOut;
	mac.upperControlIn <-- { @display("ls=red;m=m,85,0,85,0"); } <-- upperControlIn;
	phy.upperLayerOut --> { @display("ls=black;m=m,25,50,25,0"); } --> mac.lowerLayerIn;
	phy.upperLayerIn <-- { @display("ls=black;m=m,15,50,15,0"); } <-- mac.lowerLayerOut;
	phy.upperControlOut --> { @display("ls=red;m=m,75,50,75,0"); } --> mac.lowerControlIn;
	phy.upperControlIn <-- { @display("ls=red;m=m,85,0,85,0"); } <-- mac.lowerControlOut;
	radioIn --> phy.radioIn;
}
```

InitialLLMCLMACPkt.msg file:

```
cplusplus {{
#include "MacPkt_m.h"
}}

class MacPkt;
class LAddress::L2Type extends void;
message InitialLLMCLMacPkt extends MacPkt
{
	int dstAdr;
}
```

LLMCLMacPkt.msg file:

```
cplusplus {{
#include "MacPkt_m.h"
}}
class MacPkt;
class LAddress::L2Type extends void;
message LLMCLMacPkt extends MacPkt
{
        int ownCh;
        int mySlot1;        // the node's current slot number
        int mySlot2;
        int dtg;
        LAddress::L2Type occupiedSlots1[]; // currently known occupied slots for ch 1
        LAddress::L2Type occupiedSlots2[]; // currently known occupied slots for ch 2
   }
```

LLMCLMacLayer.ned file:

```
package org.mixim.modules.mac;
import org.mixim.base.modules.BaseMacLayer;
simpleLLMCLMacLayer extends BaseMacLayer
{
parameters:
int sinkAddr;
bool debug = default(false);
double slotDuration @unit(s) = default(100ms);
double queueLength = default(10);
double defaultChannel = default(0);
double bitrate @unit(bps) = default(19200bps);
int reservedMobileSlots = default(2);
int numSlots = default(32);
double txPower = default(50);
@class(LLMCLMacLayer);
}
```

LLMCLMacLayer.h file:

```
#ifndefLLMCLMAC_LAYER_H
#defineLLMCLMAC_LAYER_H
#include<list>
#include "MacToNetwControlInfo.h"
#include "NetwToMacControlInfo.h"
#include "MiXiMDefs.h"
#include "DroppedPacket.h"
#include "BaseMacLayer.h"
#include "PhyUtils.h"
#include "SimpleAddress.h"
classLLMCLMacPkt;
class MIXIM_API  LLMCLMacLayer : public BaseMacLayer
{
private:
LLMCLMacLayer(constLLMCLMacLayer&);
LLMCLMacLayer&operator=(constLLMCLMacLayer&);
public:
LLMCLMacLayer()
```

```
        : BaseMacLayer()
        , SETUP_PHASE(true)
, my(true)
        , slotChange()
        , macState()
        , radioState()
        , slotDuration(0)
        , controlDuration(0)
, mycontrolDuration(0)
        , centerFreq(0)
, ownCh()
, channel()
        , mySlot(0)
        , mySlot1(0)
        , mySlot2(0)
        , numSlots(0)
        , currSlot()
        , reservedMobileSlots(0)
        , macQueue()
        , queueLength(0)
        , wakeup(NULL)
        , timeout(NULL)
        , sendData(NULL)
        , initChecker(NULL)
        , checkChannel(NULL)
        , start_llmclmac(NULL)
        , send_control(NULL)
, send_initial_control(NULL)
        , bitrate(0)
        , droppedPacket()
        , nicId(-1)
        , txPower(0)
    {}
virtual~LLMCLMacLayer();
virtualvoidinitialize(int);
virtualvoidfinish();
virtualvoidhandleLowerMsg(cMessage*);
virtualvoidhandleUpperMsg(cMessage*);
virtualvoidhandleSelfMsg(cMessage*);
virtualvoidhandleLowerControl(cMessage *msg);
virtual MacPkt* encapsMsg(cPacket*);
protected:
typedef std::list<LLMCLMacPkt*> MacQueue;
enum States {
INIT,
SLEEP,
CCA,
WAIT_CONTROL,
WAIT_DATA,
SEND_INITIAL_CONTROL,
SEND_CONTROL,
SEND_DATA
    };
enum TYPES {
LLMCLMAC_CONTROL=167,
```

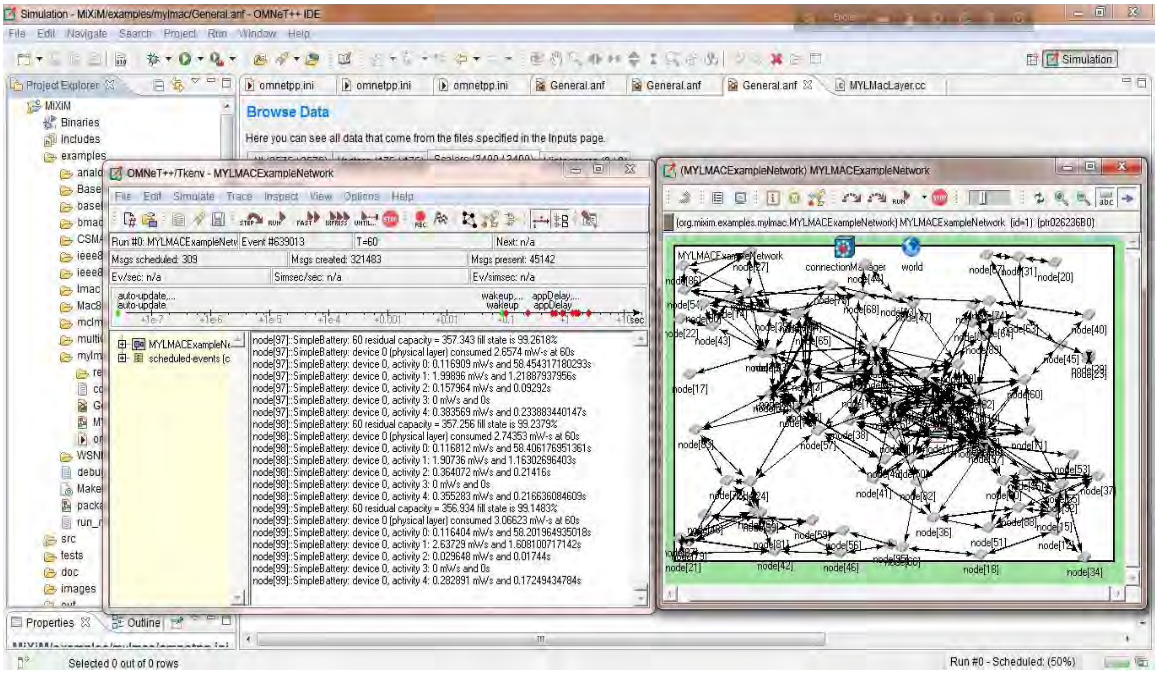
```
LLMCLMAC_TIMEOUT=168,
LLMCLMAC_WAKEUP=169,
LLMCLMAC_SEND_DATA=170,
LLMCLMAC_SETUP_PHASE_END=171,
LLMCLMAC_CHECK_CHANNEL=172,
LLMCLMAC_DATA=174,
LLMCLMAC_START_LLMCLMAC=175,
LLMCLMAC_SEND_CONTROL=176,
LLMCLMAC_INITIAL_CONTROL=177,
LLMCLMAC_SEND_INITIAL_CONTROL=178
   };
staticconst LAddress::L2Type LLMCLMAC_NO_RECEIVER;
staticconst LAddress::L2Type LLMCLMAC_FREE_SLOT;
bool SETUP_PHASE;
bool my;
   cOutVector* slotChange;
   States macState;
   Radio::RadioState radioState;
double dura;
simtime_t mytime1;
simtime_t mytime2;
simtime_t mytime3;
int coll;
double centerFreq;
int ownCh;
int channel;
int mySlot;
int mySlot1;
int mySlot2;
int dtg;
bool myVar;
int sinkAddr;
int frame_num;
double slotDuration;
double controlDuration;
double mycontrolDuration;
int numSlots;
int currSlot;
int dstadr;
int smalldtg;
LAddress::L2Type neighbortable[200];
   LAddress::L2Type occSlotsDirect1[32];
   LAddress::L2Type occSlotsAway1[32];
   LAddress::L2Type occSlotsDirect2[32];
   LAddress::L2Type occSlotsAway2[32];
int reservedMobileSlots;
   MacQueue macQueue;
unsigned queueLength;
   cMessage* wakeup;
   cMessage* timeout;
   cMessage* sendData;
   cMessage* initChecker;
   cMessage* checkChannel;
   cMessage* start_llmclmac;
   cMessage* send_control;
```

```
cMessage* send_initial_control;
double bitrate;
voidfindNewSlot();
   DroppedPacket droppedPacket;
int nicId;
voidattachSignal(MacPkt *macPkt);
Signal* createSignal(simtime_t_cref start, simtime_t_cref length, double power, double bitrate);
double txPower;
};
#endif
```

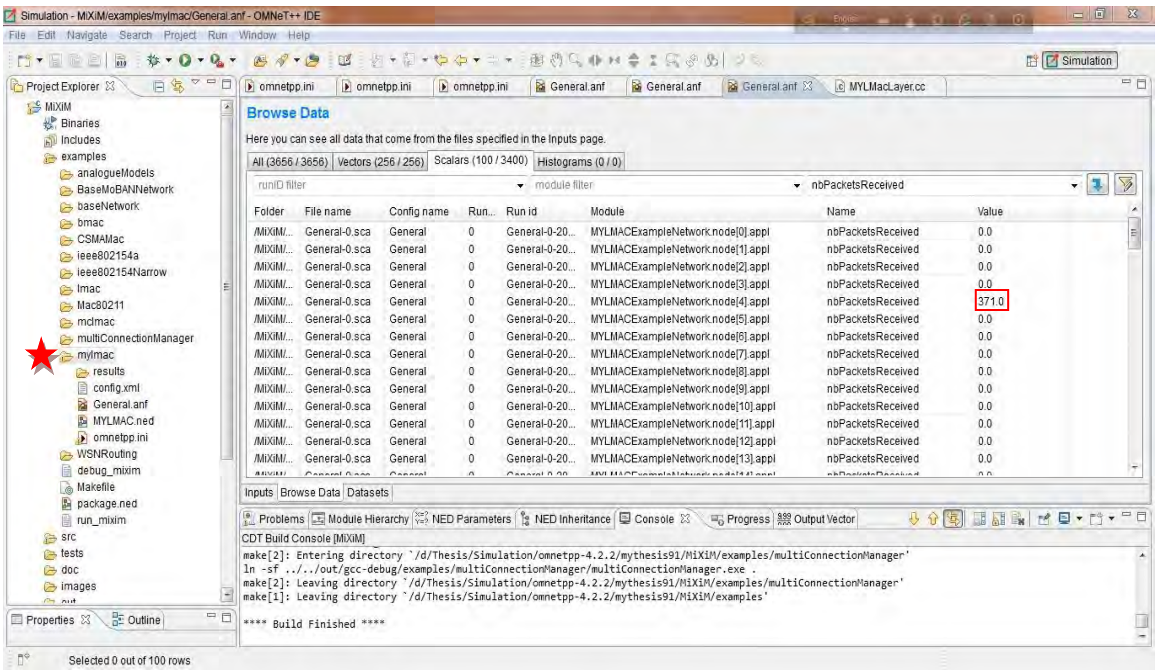
APPENDIX B

Snapshot of Simulation Results

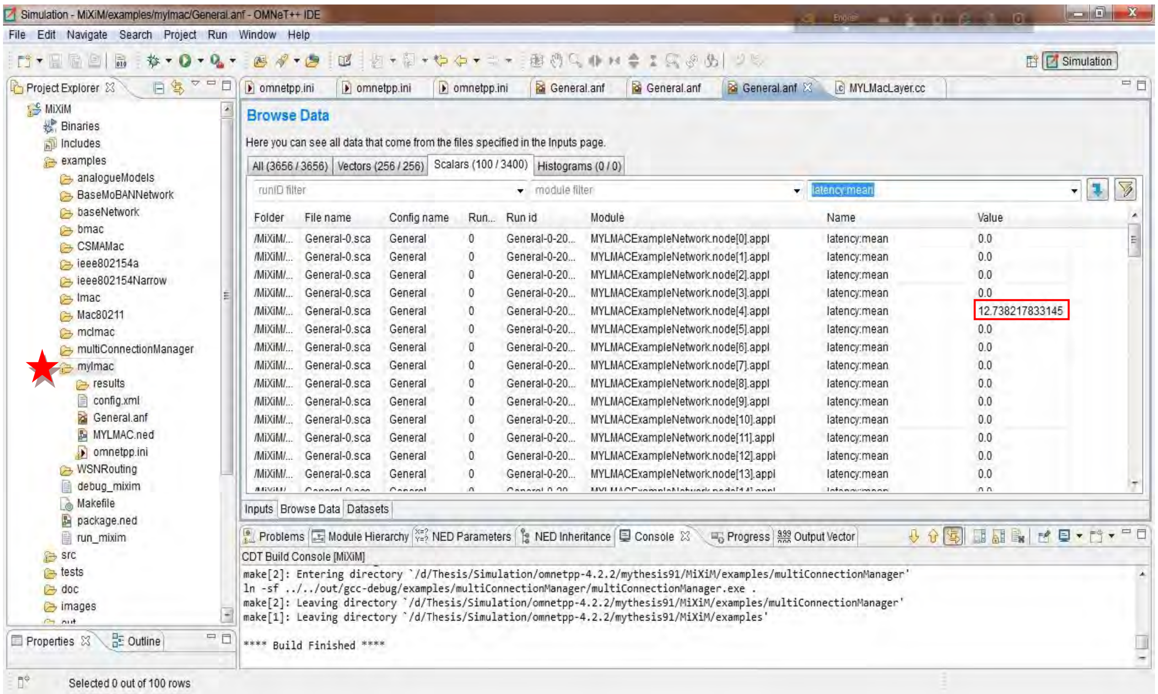
- **Snapshot of network model of proposed protocol for 100 nodes in the network.**



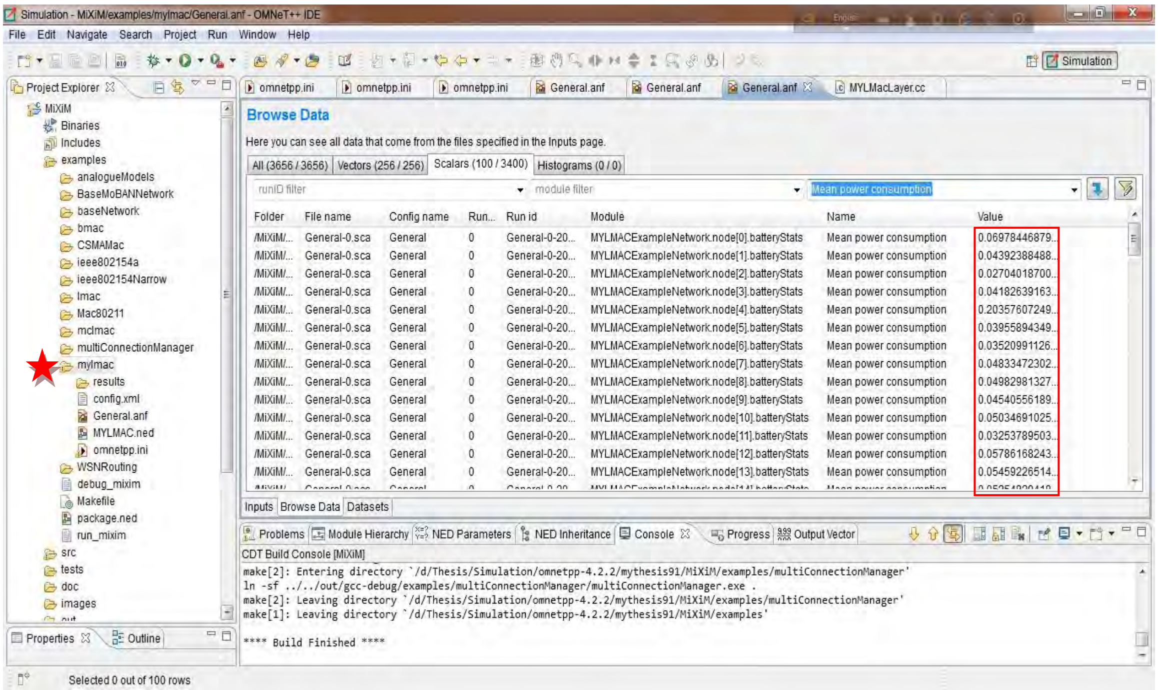
- **Snapshot of simulation result of proposed protocol for number of received packets of topology-1 for 16 slots in a frame where the number of node is 100.**



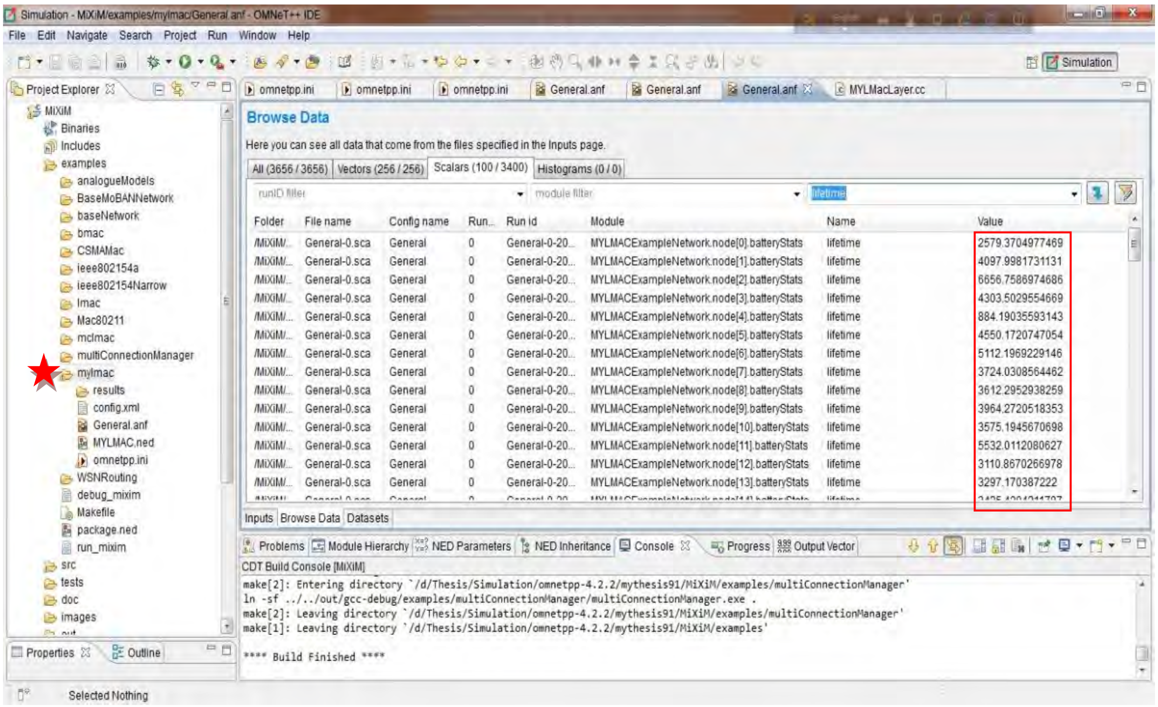
- **Snapshot of simulation result of proposed protocol for end-to-end delay of topology-1 for 16 slots in a frame where the number of node is 100.**



- **Snapshot of simulation result of proposed protocol for energy consumption of topology-1 for 16 slots in a frame where the number of node is 100.**



- **Snapshot of simulation result of proposed protocol for network lifetime of topology-1 for 16 slots in a frame where the number of node is 100.**



N.B. In simulation, proposed LL-MCLMAC protocol was renamed as MYLMAC.