STATUS OF THESIS

Title of thesis:	DISTRIBUTIVE AND SELF-SUSTAINABLE SCHEDULING ALGORITHMS FOR WIRELESS SENSOR NETWORKS

I MUHAMMAD AMAN SHEIKH .

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

[ ] Confidential

[√] Non-confidential

If this thesis is confidential, please state the reason:

____

____

The contents of the thesis will remain confidential for __ years.

Remarks on disclosure:

____

____

Endorsed by

DR. NOOHUL BASHEER ZAIN ALI
Senior Lecturer
Electrical & Electronic Engineering

Signature of Author	Signature of Supervisor
Permanent address:	Dr. Noohul Basheer Zain Ali .
Mandian, Abbottabad, Pakistan	
Date : 12 - 08 - 2013 .	Date : 12 - 08 - 2013

DISTRIBUTIVE AND SELF-SUSTAINABLE SCHEDULING ALGORITHMS FOR WIRELESS SENSOR NETWORKS

by

MUHAMMAD AMAN SHEIKH

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fullfilment of the requirements for the degree stated.

Signature: DR. NOOHUL BASHEER ZAIN ALI Senior Lecturer Electrical & Electronic Engineering

Main Supervisor: Dr. Noohul Basheer Zain Ali

Signature: Dr. AZLAN AWANG Lecturer Electrical & Electronic Engineering Department Universiti Teknologi PETRONAS 31750 Seri Iskandar, Perak Darul Ridzuan Tel: 605-368 7840 Fax: 605-365 7443 12/08/2013

Co-Supervisor: Dr. Azlan Awang

Signature:

Head of Department: Dr. Rosdiazli Bin Ibrahim

Date: 12/8/13

Assoc Prof Dr Rosdiazli Bin Ibrahim Head of Department Electrical & Electronic Engineering Department Universiti Teknologi PETRONAS

DISTRIBUTIVE AND SELF-SUSTAINABLE SCHEDULING ALGORITHMS FOR WIRELESS SENSOR NETWORKS

By

MUHAMMAD AMAN SHEIKH

A Thesis

Submitted to the Postgraduate Studies Programme

as a requirement for the degree of

MASTER OF SCIENCES

ELECTRICAL AND ELECTRONIC ENGINEERING

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR

PERAK, MALAYSIA

APRIL 2013

DECLARATION OF THESIS

Title of thesis	DISTRIBUTIVE AND SELF-SUSTAINABLE SCHEDULING ALGORITHMS FOR WIRELESS SENSOR NETWORKS

I, MUHAMMAD AMAN SHEIKH

hereby declare that the thesis is based on my original work except for quotations and citations, which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

DR. NOOHUL BASHEER ZAIN ALI
Senior Lecturer
Electrical & Electronic Engineering

Signature of Author

Permanent address:

Mandian, Abbottabad, Pakistan

Date: 12 - 08 - 2013

Signature of Supervisor

Dr. Noohul Basheer Zain Ali

Date: 12 - 08 - 2013

DEDICATION

To my beloved family, specially to my father Sheikh Khuda Bakhsh (Late)

ACKNOWLEDGEMENT

All praise are due to Allah (SWT), created man and taught what he knew not "Who has taught (the writing) by pen", and taught man which was not possible without Allah's guidance. Peace and prayers be upon the messenger Mohammed (SAAWS). First, I am very thankful to Almighty Allah, for enhancing my courage for completion of this research work gracefully.

Secondly, it is matter of great honor and pleasure for me to express deepest gratitude and sincere thanks to my respectable supervisor Dr. Noohul Basheer Zain Ali and co-supervisor Dr. Azlan Awang for their valuable guidance, intellectual activities and advice for completion of this work.

I am also thankful to management of Universiti Teknologi PETRONAS, Malaysia, a number of individuals for their assistance and cooperation in accomplishing this research work. I am also greatly obliged to my brother Dr. Sheikh Tahir Bakhsh and all my friends especially Mr. Omer Chughtai who helped me whenever needed.

ABSTRACT

Wireless Sensor Networks (WSNs), due to their vital importance, are emerging as a ubiquitous networking arena which pervades some of the old applications and also enables many new ones. The credit for the rapid growth in WSN technology goes to its self-organising and self-configuring abilities. Generally, the distributed environment of WSNs with little or no predetermined infrastructure, mobility, lack of bandwidth and scalability are the issues that affect network performance. In WSNs, lifetime is considered as the key challenging issues because all of the sensors are battery powered. Physically, it is infeasible to recharge or replace the battery. Most of the energy in WSN is wasted due to idle listening, collision, message overhearing, and message overhead. It has been found that the part of overall network functionality known as the medium access control (MAC) protocol handles issues regarding energy efficiency, fairness, collision, and reliable access to the medium. The question of how to design an optimal MAC algorithm has been extensively studied; still a design for an optimal distributed scheduling algorithm has remained a challenge. To cope up with this challenge, distributive and self-sustainable MAC scheduling algorithms are proposed in this thesis. According to the proposed scheduling algorithms, each node schedules its time slot in such a way that the same slot is not reserved by any of its conflicting nodes, this leads to conflict-free and collision-free scheduling. The scheduling through proposed algorithms is in a heuristic manner and helps to overcome all of the anomalies that hamper to prolonging the network lifetime. Along with the scheduling, the proposed algorithms also obviate the weaknesses of traditional algorithms with several unique features. First, they optimise energy through reserving conflict-free slots and can easily adapt their transmission schedule in response to the topology changes without reconstructing a whole network transmission schedule. Second, transmissions can be executed dynamically by a node in each schedule slot; as a result, it may adjust to workload more effectively and efficiently. Furthermore, the proposed algorithms outperform existing technique in terms of run time, message overhead, energy consumption, and limited memory requirements to be adjustable for resource constrained devices.

ABSTRAK

Rangkaian sensor tanpa wayar (WSNs), kerana kepentingan itulah telah muncul sebagai arena rangkaian sentiasa ada yang meliputi beberapa aplikasi lama dan juga aplikasi yang baru. Kredit untuk pertumbuhan pesat dalam teknologi WSN adalah kerana kebolehannya untuk beroperasi dan mengkonfigurasi dengan sendiri. Secara umumnya, suasana penyebaran WSNs dengan infrastruktur terhad atau tidak ditetapkan, kebolehan untuk bergerak, kekurangan jalur lebar dan skala adalah isu-isu yang memberi kesan kepada prestasi rangkaian. Dalam WSNs, kebolehan untuk bertahan dianggap sebagai isu-isu utama yang mencabar kerana semua sensor dijanakan oleh tenaga bateri. Secara fizikal, tenaga ini tidak boleh dicas semula atau menggantikan bateri. Kebanyakan tenaga dalam WSN adalah sia-sia kerana hanya mendengar, perlanggaran, mesej yang tidak perlu, dan mesej terlalu banyak. bahawa sebahagian daripada fungsi keseluruhan rangkaian dikenali sebagai kawalan akses protokol sederhana (MAC) mengendalikan isu-isu mengenai kecekapan tenaga, keseimbangan, perlanggaran, dan akses dipercayai kepada medium. Persoalan bagaimana untuk reka bentuk algoritma MAC optimum telah dikaji secara meluas, tetapi reka bentuk diedarkan algoritma penjadualan yang optimum kekal menjadi cabaran. Untuk menghadapi cabaran ini, pengedaran dan kebolehan untuk mengadaptasi dengan persekitaran MAC algoritma penjadualan yang dicadangkan di dalam tesis ini. Menurut cadangan algoritma penjadualan, setiap nod menjadual slot masa supaya tidak dikhaskan oleh mana-mana nod yang bercangah, ini akan membawa kepada tiada konflik-percanggahan dan bebas dari perlanggaran. Penjadualan melalui cadangkan algoritma adalah secara heuristik dan membantu untuk mengatasi semua anomali yang menghalang untuk memanjangkan hayat rangkaian. Bersama-sama dengan penjadualan, algoritma yang dicadangkan juga menyingkirkan kelemahan algoritma tradisional dengan beberapa ciri-ciri unik. Pertama, mereka mengoptimumkan tenaga melalui menempah slot konflik bebas dan senang menyesuaikan diri dengan jadual penghantaran mereka sebagai tindak balas kepada perubahan topologi tanpa membina semula keseluruhan jadual penghantaran

rangkaian. Kedua, penghantaran boleh dilakukan secara dinamik oleh nod dalam setiap slot jadual; hasilnya, ia boleh menyesuaikan diri dengan beban kerja yang lebih berkesan dan cekap. Tambahan pula, algoritma cadangan mengatasi teknik yang sedia ada dari segi masa pengoperasian, mesej yang tidak perlu, penggunaan tenaga, dan keperluan memori yang terhad supaya dapat disesuaikan terhadap sumber peranti yang`terhad.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

WPAN	Wireless personal area network
WSNs	Wireless sensor networks
MEMS	Micro-Electromechanical systems
MAC	Medium access control
CPU	Central processing unit
DRAND	Distributed randomised scheduling algorithm
CSMA	Carrier sense multiple access
CSMA-CA	Carrier sense multiple access with collision avoidance
TCL	Tool command language
PHY	Physical
S-MAC	Sensor-MAC
RTS	Request to send
CTS	Clear to send
ACK	Acknowledgment
T-MAC	Timeout-MAC
FRTS	Future request to send
B-MAC	Berkeley medium access control
CCA	Clear channel assessment
LPL	Low power listening
TDMA	Time-division multiple access
RR	Reservation request
RC	Reservation confirmation
TRAMA	TRaffic-Adaptive medium access
NP	Neighbor protocol
SEP	Schedule exchange protocol
AEA	Adaptive election algorithm
Z-MAC	Zebra MAC
LCL	Low contention level
HCL	High contention level
ECN	Explicit contention notification
TR	Transmission range

TF	Time frame
MH	Multi-hop
E	Edges
H1	One-hop
H2	Two-hop
N1	1-hop neighbourhood
N2	2-hop neighbourhood
IR	Interference range
NB	Neighbouring list
DSSA	Distributive and self-sustainable scheduling algorithm
S1	Slot in 1-hop
S2	Slot in 2-hop
IDSA	An improved distributive scheduling algorithm
NS	Network simulator
OTcl	Object-oriented tool command language
NAM	Network animator

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The progressive nature and exploration in wireless sensor networks (WSNs) have experienced unprecedented development and is considered to be the most preferable medium for communication [1-3]. Nowadays global businesses are mobile and distributed. Consequently, WSNs provide a seamless bridge to permeate the gap between distance and movement. Therefore, the emergence of a high data rate, low power consumption, and small sized sensor network applications due to the rapid advancement in micro-electromechanical systems (MEMS) has increased the demand for wireless network services [4]. Typically, in WSNs, the sensor nodes are not isolated rather they are geographically-distributed and may significantly vary up to thousands of sensor nodes depending upon the application [5, 6]. Figure 1.1 shows that each node is capable of sensing environmental variations and responding accordingly to those changes. Each sensor node immediately reacts on the environmental variations and routes the environmental changes to the sink node. Finally, the sink node translates and routes the aggregated data of all of the sensor nodes to the end user via a wireless radio interface [7, 8].

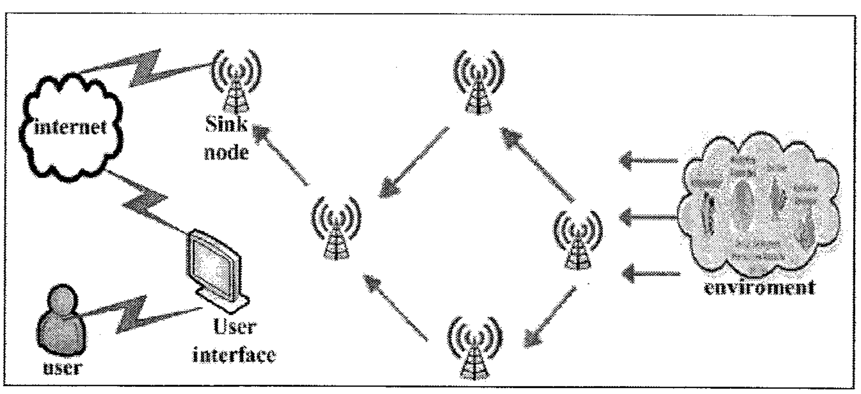


Figure 1.1 Data flow of WSNs from environment to end user

1.2 MOTIVATION

WSNs are envisioned enablers for a wide range of applications, ranging from health monitoring, environmental surveillance, home appliances, industrial process monitoring, and inventory tracking, to even providing networking facilities in or around a human body such as in wireless body area networks WBANs [9-11]. WSNs' sensing and monitoring phenomena is as diverse as speed, moisture, temperature, and any particular location by means of optical motion, piezoelectric and thermistor detectors[12, 13]. For all of these applications, a WSN shares some of the common attributes. The sensor nodes are battery powered and have a limited power, however, it is infeasible to recharge or replace the battery [14]. Consequently, this research work has investigated that, for optimal utilization of energy in WSNs the entire protocol stack must be energy-efficient. Based on this criterion in WSNs, this research has proposed two distributive scheduling algorithms.

Existing WSNs differ a lot from traditional communication networks in many aspects. The distributed environment, with little or no predetermined infrastructure, mobility, lack of bandwidth and scalability are challenges and issues that distinguish the existing WSNs from traditional WSNs [15]. In the past, many researchers have focused on designing tiny sensors, limited power consumption processors, cost effective and energy-efficient protocols to minimize energy consumption and prolong network lifetime [16, 17]. However, in widespread distributive WSNs, the single, most important challenge of energy-efficient protocols has remained a challenge for the last four decades [18]. This dissertation has investigated the requirement for an innovative medium access control (MAC) to utilize network resources in a more efficient and effective manner by the proliferation of advanced computing devices.

Hence, it has been observed that the distributed interactions, and self-organizing nature of WSNs bring about challenges in predicting the performance of current network technologies. Whenever, conflicting nodes simultaneously start transmission it results in a collision. Collision is one of the sources of energy wastage. After each data collision, packets are retransmitted that consumes at least twice of the energy for transmission of the same data packet. Thus, to deal with these challenges, this current research has required the presentation of distributed scheduling MAC algorithms.

Therefore, to design a large and scalable network, the scheduling algorithms should be computationally distributed as well as simple to resolve contention by providing collision-free scheduling.

This research has addressed scheduling techniques and has provided adaptive topology independent and distributed MAC scheduling algorithms to resolve the challenging issues related to scheduling. In this research work novel and heuristic scheduling algorithms are presented which are simple, distributive and performance optimal. According to the proposed scheduling algorithms each node reserves a conflict-free slot for its transmission in a heuristic manner that has not been reserved by any node in its two-hop neighbourhood to overcome all of the anomalies that hampers prolonging the network lifetime.

1.3 PROBLEM STATEMENT

In WSNs, a common medium is shared for communication. Therefore, simultaneous transmission of conflicting nodes results in a collision. Collisions degrade network performance because it not only wastes the bandwidth of the network but also the power resources of individual devices. After each data collision, packets are retransmitted that creates overhead in terms of extra control packets, energy consumption, and computational time; these are all anomalies that have a direct influence on network lifetime. [8, 19, 20]. Moreover, the distributed environment of WSNs with little or no predetermined infrastructure, mobility, lack of bandwidth and scalability are the issues that affect network performance.

In WSNs the term "Energy optimization" is the main challenge for WSNs and it has a direct impact on a network's lifetime. Figure 1.2 shows the basic architecture of sensor nodes is composed of several units such as: micro controller, Central Processing Unit (CPU), transmitter, receiver, mobiliser and a location finding system [21]. In fact, all of these units rely on a power unit to carry out all of the primitive operations, as WSNs are battery operated to accomplish all the tasks with little or no scope to replace or recharge batteries. In the past, intensive works were carried out and different methods were proposed to overcome the energy scarcity. Broadly, the

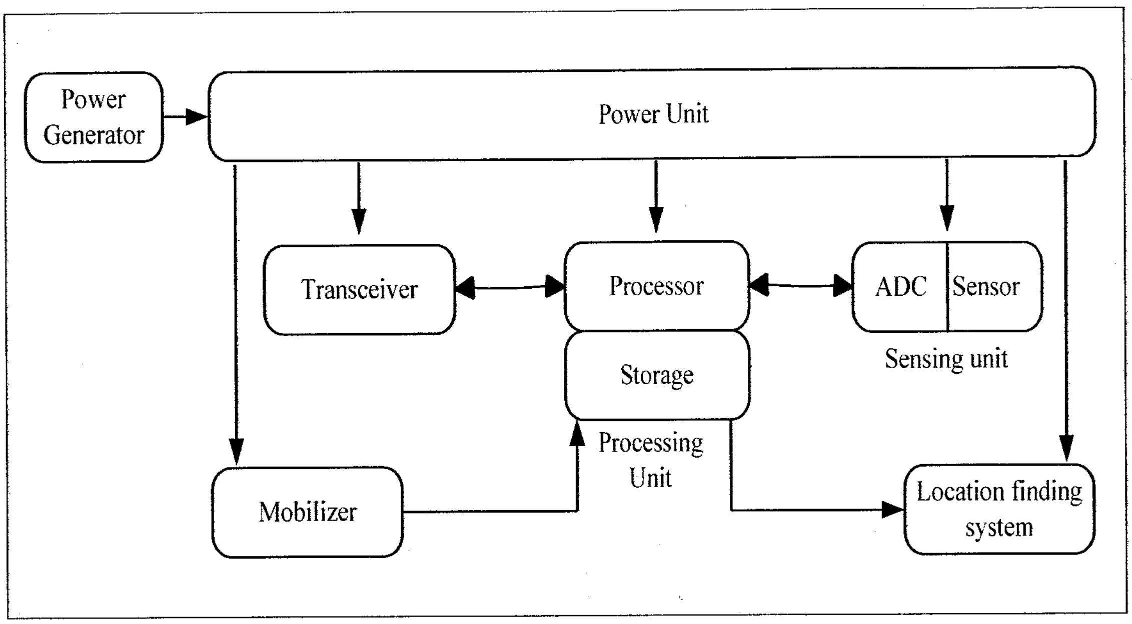


Figure 1.2 Basic architecture of a wireless sensor node

energy efficiency can be achieved at two levels: hardware design and MAC algorithm designs [22, 23]. However, MEMS has led to the creation of tiny micro processor, low power consumption, multi-functional and miniature sensing devices. On the other hand, many individual have contributed through designing energy efficient algorithms but unfortunately commercially no in depth study has been carried out on optimal protocol stack designs [24, 25].

Although there have been hardware design improvements by MEMS, still the lack of optimal energy efficient algorithms have motivated this research to propose MAC algorithms in order to tackle all of the anomalies of energy wastage. The question of how to design an optimal MAC algorithm has been extensively studied in the past and many scheduling algorithms have been proposed. Although, some of these are in practice, still the design of optimal distributed scheduling algorithm has remained a challenge because most of the exiting algorithms are either contention based or dependent upon central controller for scheduling.

Generally, WSNs are distributed and self-organising. Therefore, a scheduling algorithm is a fundamental design problem to resolve contention and allocate resources among different entities in scattered and distributive WSNs. In this thesis, self-sustainable and adaptive topology independent heuristic scheduling algorithms have been presented, which are computationally simple, distributed and performance optimal. According to the proposed scheduling algorithms, each node reserves a conflict-free slot for its transmission without any central controller. Further, these scheduling algorithm trim downs the parameters that affect network performance during setup phase such as:

- Number of rounds,
- Communication overhead,
- Run time, and
- Energy consumption.

1.4 OBJECTIVES

The MAC layer is responsible for the allocation of resources among different entities. It helps nodes to access a channel for data transmission. Furthermore, the performance and lifetime of WSNs can be improved by resolving the contention among all of the conflicting nodes within the network. The objectives of this research work are to propose an energy-efficient, computationally simple, optimal, and distributed MAC scheduling algorithms. The propose MAC algorithms should not require any time synchronisation and must be robust against any dynamic topology changes. The algorithms must easily adapt to the changes in the topology explicitly without reconstructing the global transmission schedule with minimum message overhead. Furthermore, the propose algorithms should provide optimal collision-free transmission schedule while utilizing minimal system resources. To achieve the above mentioned aims, following are the main objectives that ought to be accomplished in this thesis:

- To propose distributive and self-sustainable MAC scheduling algorithm to enhance DRAND algorithm, through minimizing the chances of unsuccessful cycle and obviates its weakness.
- To propose an improved distributed scheduling algorithm that should allow each node to decide its own schedule according to the local information, rather than each time depending on neighboring nodes.

- To verify the effectiveness of propose scheduling algorithms over DRAND in terms of run time, message overhead, energy consumption, and number of rounds.

Simulation results validate that the performance of proposed scheduling algorithms by comparing with the distributed randomised scheduling algorithm (DRAND) [26].

1.5 THESIS ORGANIZATION

This thesis introduces MAC algorithms for distributive and self-sustainable WSNs. These scheduling algorithms present the solution to resolve the contention among all of the conflicting nodes and provide collision-free transmission in an energy efficient manner by avoiding idle listening. The objectives and contributions of the thesis are summarized as follows:

Chapter two provides an overview to outline a comparative study of various MAC algorithms. It has been found that most of the existing algorithms are centralized to establish and manage the network. The centralized techniques are expensive and inflexible approaches for the resource-constrained network. This enabled to analyze different categories of scheduling algorithms and gain a perspective in order to reshape the objectives of this research.

Chapter three addresses the approaches and methodologies adopted for the design of the new distributive scheduling algorithms. In WSNs, a common medium is shared among all of the nodes for transmission. To avoid conflict among the nodes, the distributive scheduling algorithms assign a conflict-free time slot to achieve collision-free and reliable transmission. Beside this, the proposed algorithms prolong network lifetime by avoiding idle listening and handling dynamic topology changes without reconstructing whole network.

Chapter four focuses on the simulation results. The performance of the proposed algorithms were verified and compared against the original DRAND algorithm in terms of run time, message overhead, energy consumption, and number of rounds for optimal and collision free transmission.

Finally, chapter five summarizes and concludes the thesis along with some seminal ideas for future work based on the facts and figures carried out in this research.

1.6 THESIS CONTRIBUTION

The research work presented in the thesis has made a significant contribution in investigating various perspectives of the MAC scheduling algorithm and have presented two distributive and topology independent MAC scheduling algorithms. As identified earlier all sensor nodes in a network share a common medium for transmission. If many contenders contend for a medium and simultaneously starts their transmission it results in collision. The only way to avoid collision is by assigning conflict-free schedule among the entire conflicting nodes for their transmission. Moreover, the propose scheduling algorithms can also tackle with the distributive and self-configuring nature of WSNs and efficiently support scalability to overcome anomalies such message overhead, run time, and energy wastage.

The major contribution of this thesis is the development of distributive and self-sustainable MAC scheduling algorithms to resolve challenging issues related to scheduling. In this thesis, novel and heuristic scheduling algorithms have been presented. The first one is distributive and self-sustainable scheduling algorithm (DSSA) and the second is an improved distributive scheduling algorithm (IDSA). Both of the scheduling algorithms are computationally simple, distributed and performance optimal. According to the proposed scheduling algorithms, each node reserves a conflict-free slot for its transmission in a heuristic manner that has not been reserved by its two-hop neighbouring node to overcome all of the anomalies that hamper prolonging the network lifetime.

In the distributive and self-sustainable scheduling algorithm (DSSA), each sensor node collaborates with its one-hop ($H1$) and two-hop ($H2$) to maintain and generate its own transmission schedule. Entire network scheduling through the DSSA is simple and local because every node collaborates up to its two-hop neighbouring nodes only. None of the single nodes in the network is aware of global information, such as, the membership or transmission schedule or the entire network size. Each node directly interacts with its $H1$ but the nodes apart from that up to $H2$ causes interference. Nodes, for their transmission schedule in the DSSA, are indirectly aware of the schedules of the nodes up to their $H2$ and reserve a slot that is not reserved by any of the nodes in their $H1$ and $H2$ to avoid a collision.

According to an improved distributive scheduling algorithm (IDSA), sensor nodes do not collaborate with their $H1$ and $H2$ for their schedule; each node instead maintains and generates its own transmission schedule based on its own record. The scheduling in the IDSA is quite simple because each node maintains a record of its neighbouring node schedule in the $H1$ and $H2$ queue. Finally, when a node schedules its own slot it has to look up to its own record rather than requesting its neighbouring nodes and reserves or proposes the slot which is not reserved by any of the nodes in the $H1$ and $H2$ queue. Scheduling through the IDSA is based on local information because none of the nodes is aware of the entire network transmission schedule or size.

The presented MAC scheduling has been presented in the recognized conferences and journal articles as follows:

1 Sheikh, Muhammad Aman, Zain Ali, Noohul Basheer and, Azlan Awang (2013). ***"Queue Based Distributed Scheduling for Wireless Sensor Networks", In*** Annual Postgraduate Conference *(APC 2013) Malaysia (Submitted).*

2 Sheikh, Muhammad Aman, Zain Ali, Noohul Basheer and, Azlan Awang (2013). **"Distributive *and Self-Sustainable Scheduling Algorithm for Wireless Sensor Networks",*** In IEEE The 22nd Wireless and Optical Communication Conference (WOCC 2013) 2013, Chongqing, China.16-18May 2013.

3 Sheikh, Muhammad Aman and Drieberg , Micheal and Zain Ali, Noohul Basheer (2012). ***"An Improved Distributed Scheduling Algorithm for Wireless Sensor Networks",*** In Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on Intelligent and Advanced Systems, 2012, pp. 274-279.

4 Sheikh, Muhammad Aman and Drieberg , Micheal and Zain Ali, Noohul Basheer (2011). ***"Fair Scheduling Algorithm for Wireless Sensor Networks",*** In National Postgraduate Conference (NPC) 2011, 19-20 September, 2011, Universiti Teknologi PETRONAS.

CHAPTER 2

BACKGROUND

This chapter provides significant knowledge about MAC layer and its importance followed by basic innovative work in context of WSN. This chapter thoroughly discusses the problems and challenging issues in the existing WSNs and important features of MAC layer, by summarizing some of the MAC layer categories and the approaches adopted to tackle the key challenging issues. Further, the chapter describes the system model, performance parameters and problem formulation in WSNs through a network and interference model. Finally, the chapter ends with conclusion and the critical research problem within the existing scheduling technique based on which this research is carried out.

2.1 PROBLEMS/ CHALLENGES IN WSNs

A number of challenges, which are not present or are present in various different forms in conventional wired networks, exist in the design and deployment of wireless sensor networks [27]. Some of such challenges are presented as follows:

Self-organization: Each individual node in a WSN has to possess the capability of attaching to and detaching from a network, without the necessity for any fixed infrastructure. There is a necessity for protocols with the ability to offer support for and facilitate the construction of topology, possible re-configuration, and overall maintenance. Moreover, they must support and facilitate admission control, traffic monitoring and routing.

Scalability of the network: This is in regards to the capability of maintaining certain parameters for performance regardless of any small or large change in the number of nodes that are set up in any particular network. Scalability is dependent upon how much overhead is present at the various layers (medium access control, physical, transport, networking/routing) of the protocol stack of the network.

Delay: This parameter is quite critical in certain kinds of applications; for example, in applications regarding health care where patients with serious and urgent medical conditions must be continuously monitored for important health variables by way of EEG, ECG, or other probes. Another application is in military usage such as in the detection and monitoring of troop movement or communications on the battlefield. Scheduling, bandwidth reservation can provide low delays; however, scheduling is a mechanism that need a coordinator or controller for monitoring and prevention of congestion and collision in a network.

Throughput: This performance target is the most important in a several collaborative, distributed computing applications. It is also vital in regards to mobile access to the Internet; this might include very large amounts of multimedia traffic. Throughput could become hindered at the physical (PHY) layer level; this could be a result of noise and interference causing packet errors. Throughput can also be hindered by collisions that occur at the Medium Access Control (MAC) level. There are two possible causes: one is when using a contention-based medium access mechanism and the other is because of unfairness when either a scheduling-based access mechanism or bandwidth reservation is utilized. Cross-layer optimization that takes into account at least some but preferably, all of those effects might be necessary if high throughput is to be achieved.

Fairness: This is another important factor among applications, various nodes and/or users that how fairly the system resources are utilized within a network.

Power management: Generally, WSNs are battery power and physically it is infeasible to recharge or replace the battery. Therefore, battery power should be efficiently utilized.

Automated or simple to use: All tasks regarding maintenance in WSN should be, if possible, automated or at least easy enough to be carried out by human operators with only basic knowledge such as laptop/computer and PDA owners.

2.2 MAC

The part of the overall network functionality known as the medium access control (MAC) protocol handles issues regarding efficiency, fairness, and reliable access to the medium that is simultaneously used by variety of devices [28]. The role of MAC layer protocols in wireless networks is of particular importance as these networks are quite different from their wired counterparts in many ways. The most important among these differences comes from the nature of the wireless communication medium itself; this difference is that devices do not need to be connected directly to communicate with each other. Rather, it is enough for the devices to be within the same range of the radio transmission.

For example, if two or more packets are received at the same time, problems could be encountered by the receiver. The best case would be that the unwanted packets are handled as noise that hinders the reception of the intended packet but could possibly be filtered out. The worst case would be that the wanted packet could be damaged in such a way that prevents repair causing the receiver to be unable to understand it; this refers as a collision. Collisions not only waste the bandwidth of the network but also the power resources of individual devices (receivers and transmitters). Therefore, measures must be actively undertaken to guarantee that the likelihood of collisions is reduced.

In wired networks detection and avoidance are among the traditional techniques utilized for minimizing collisions. Collision detection is widely used in wired networks, where the simple act of listening while transmitting is involved. However, this is not feasible in wireless communication, where not many devices possess the necessary ability. Moreover, packet collision situations not possible in a wired network that can take place in wireless networks; some cases are such as the so-called hidden and exposed terminal problems [29, 30].

MAC protocols for wireless networks have to depend on various techniques for avoiding collisions because of its lack of collision detection. These techniques include bandwidth reservation, explicit scheduling and only attempting to transmit packets after listening to the medium. Although other names may sometimes be found for this last procedure, it is usually called the clear channel assessment [31].

It is obvious that MAC protocols in networks that are wireless, face not only the usual challenges found in wired networks but also the new ones that come from using a wireless communication medium. The most important components of a MAC protocol in WSNs can be summarized by the following which are the aim of our research.

- A distributed operation of the protocol is necessary, if possible without a dedicated central controller. When using a controller is unavoidable, it is important that it is only temporary. Moreover, any device possessing suitable capabilities should be allowed to take over the role of controller for a specific amount of time.

- The protocol must possess scalability capable of forming a large network.

- An efficient utilization of the available bandwidth is vital. This includes minimizing packet collisions as well as the overhead necessary for monitoring and controlling a network's operation. In particular, the protocol must have the ability to minimize the effects of problems resulting from hidden and exposed nodes.

- The protocol must guarantee fairness in its bandwidth allocation to each and every node. It is preferred that the current level of congestion in the network is considered by the fairness mechanism.

- Power management policy/policies should be incorporated by the MAC protocol. This is necessary for minimizing the consumption of power not only in individual nodes but also in the entire network.

- Quality of service (QoS) support must be provided by the protocol for real-time traffic wherever possible. In this sense, real-time refers to data traffic with predefined performance bounds; included in these bounds could be delay, delay jitter, throughput, and/or other indicators of performance.

2.3 CLASSIFICATION OF MAC PROTOCOLS

Before some of the important MAC protocols for WSN are presented, a brief overview of some of the possible criteria for classification of the protocols will be given. This will aid the reader in understanding the key features of various MAC protocols and identifying the important differences /similarities among them.

Mechanism for accessing the medium: The way in which the medium is accessed is potentially the most intuitive among the classification criteria. As shown in Figure 2.1 medium access is achieved with three main techniques: contention-based, schedule-based and hybrid [32-36].

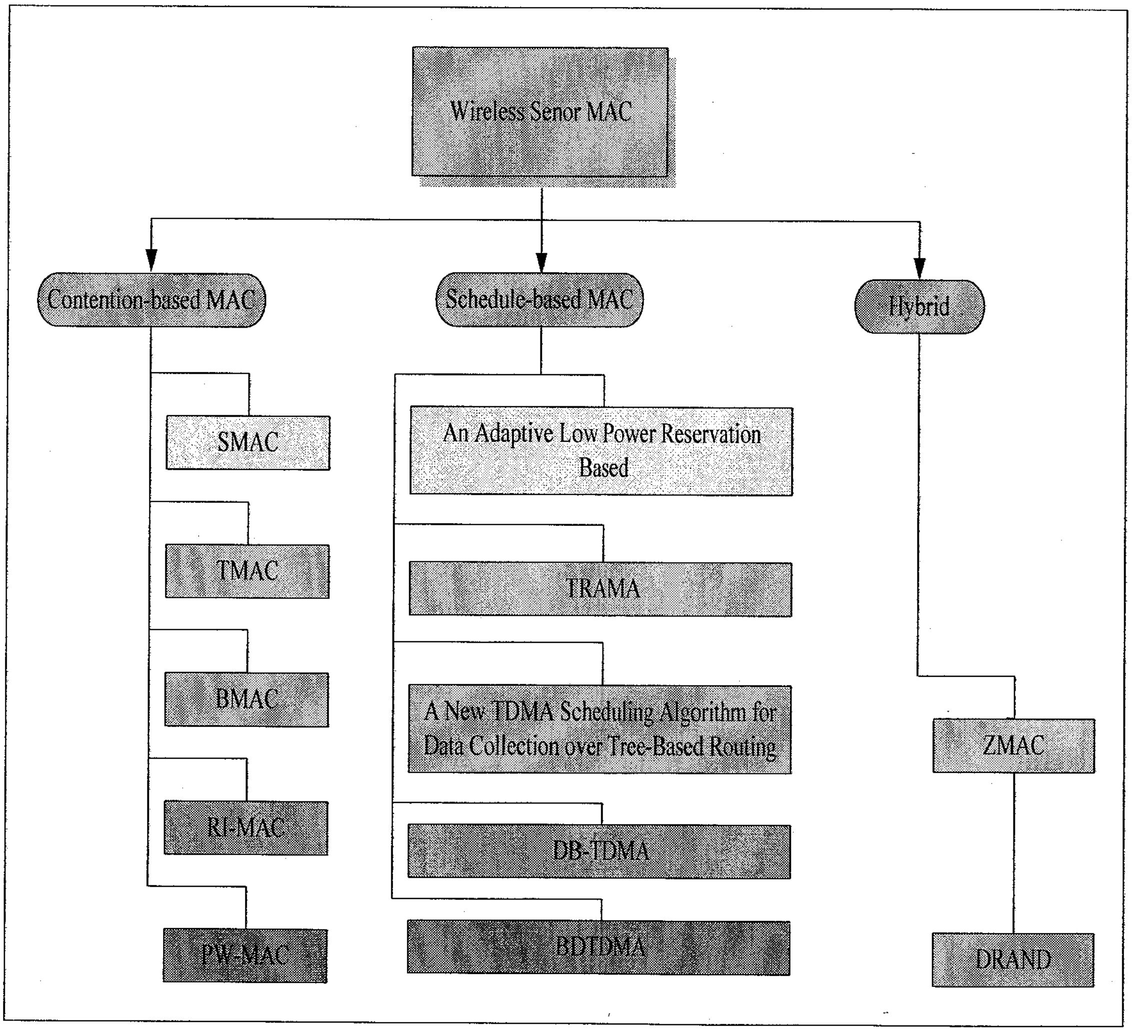


Figure 2.1 Categories and some examples of medium access control in WSNs

2.3.1 Contention-Based MAC Protocols

Variations of carrier sense multiple access (CSMA) techniques are utilized in contention-based protocols [37]. With CSMA, the basic characteristic is that the nodes listen to the shared transmission medium of the network before trying to transmit any packet. When a transmission in progress is detected the node will wait for the end of that transmission, and only then attempt to transmit its packet. Four sources of energy consumption are minimized by contention-based protocols, such as S-MAC and T-MAC. *Idle listening* is the first source; this is where a node is kept in a state wakefulness so that it can actively listen for any traffic that may or may not be present. *Overhearing* is the next source and happens when a broadcast packet, addressed to another node, is picked up by an ideally listening node. *Collisions* are another source; they make the node transmit its data again which uses up at least two times the amount of the energy for the same data. The last source is *protocol overhead.* This source uses up not only energy but also resources because of the transmission and reception of large control packets.

2.3.1.1 S-MAC

Sensor-MAC (S-MAC) is one of the first protocols specifically designed for a wireless sensor network [38]. Three methods are utilized in this protocol to minimize the energy consumption. *Periodic listen and sleep* is the first method. As shown in Figure 2.2 the whole period in SMAC is periodically scheduled into a sleep and active mode. When going to sleep, the node switches off its radio and then sets a timer to wake up after a predefined duration. Upon waking, the node starts on listening for the nodes in its range to communicate and continues listening until the time for it to return to the sleep mode. In this way, both idle listening and overhearing are minimized by S-MAC. In the beginning, each node will listen in SYNC phase to find out what its neighbors sleep schedules are. If the node receives a schedule from a neighbor before it has chosen its own schedule, it will take the neighbor's schedule. However, if it does not receive any of the neighbor's schedules, then the node will randomly choose its own sleep schedule and transmit it to its neighbors. On the other hand, if a schedule is received by the node after having chosen a schedule of its own,

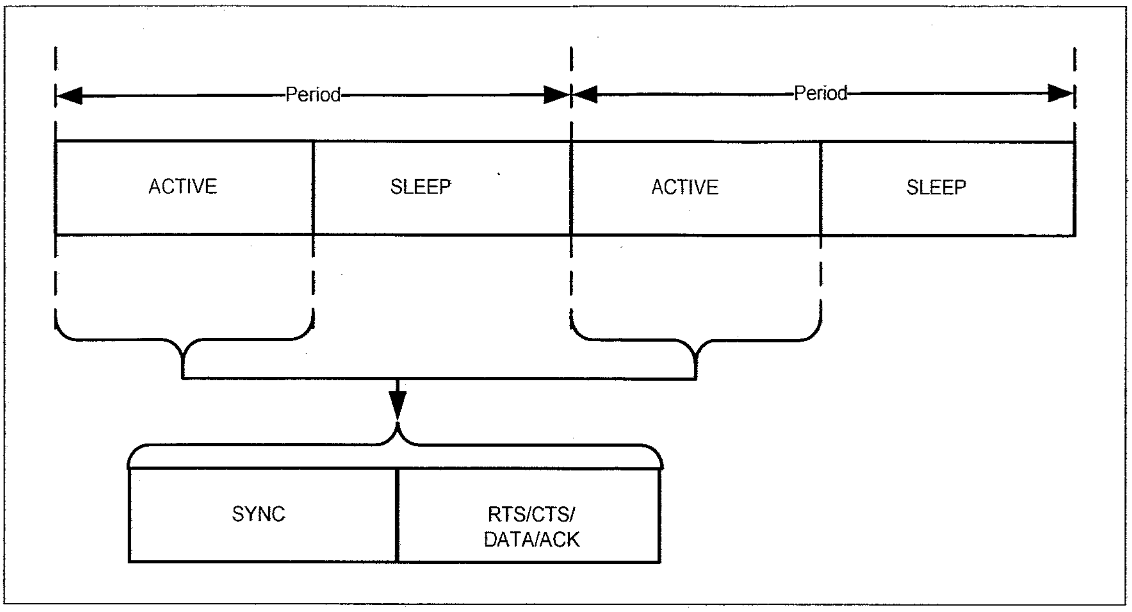


Figure 2.2 SMAC alternatively turns ON and OFF the radio in active and sleep period adopted by [38]

it will join them into one schedule. This allows *virtual clusters* of nodes to be formed among neighbors with the same schedule. In this way, broadcasting is more efficient and there is no need for a schedule to be kept for each individual neighbour. Moreover, periodical synchronisation of schedules is carried out among neighbours so that long-term clock drift can be prevented and adjustments can be made for any alteration in the WSN. Furthermore, as a transmission is more important than the sleep schedule of a node, the node will remain awake until the transmission has finished.

The second method considers avoidance of collision and overhearing. A contention-based scheme similar to IEEE 802.11 is adopted by S-MAC. As shown in Figure 2.2 this scheme includes not only virtual and physical carrier senses but also a request to send (RTS)/clear to send (CTS) exchange in an active period so that collisions can be avoided. A duration field is included in each transmitted packet of the virtual carrier sensing; it indicates how much time remains until completion of the transmission. In that way, the receiving node is aware of how long it must wait before it can transmit its own packet. In addition, physical carrier sensing is performed by each node by listening to the medium for any transmission that may be in progress. The node is free to transmit only if both the virtual and physical carrier sense indicates that there is no transmission in progress. A node goes to sleep mode when it hears an RTS or CTS packet among the other nodes, and thus overhearing is minimized. Therefore, the neighbouring nodes avoid the much longer data packets and receive only the small RTS/CTS control packets.

The last method utilized by S-MAC is known as *message passing*. In this method, long messages are transmitted efficiently. Whenever a single packet is used to send a long message, there is always the risk that it will become damaged in some way; this would result in the packet needing to be resent. On the other hand, a longer delay results from breaking the message into smaller fragments as it causes large control overhead. However, in the case of S-MAC, long messages are broken up into smaller fragments and then sent in a burst. By doing so, S-MAC reserves the medium for all the fragments utilizing only one RTS and CTS packet [39]. As each fragment is sent, the node that sent the fragment waits for an acknowledgment (ACK) from the node that received it. If no ACK is received, the fragment will be sent again and the reserved transmission time will be extended in the duration field to allow for the retransmission. A neighboring node, using overhearing avoidance, will go into sleep mode when it hears an RTS or CTS packet. It will remain asleep until all the fragments have been sent; therefore, the switching control overhead will be reduced. If a node wakes up during the transmission of the fragments, it will know how long it must go back to sleep for, based on the amount of time given in the duration field of the particular fragment being sent.

S-MAC is very successful at minimizing the energy consumption of a node. However, the throughput and latency are decreased. As the nodes are not able to send packets while in the sleep mode, throughput is decreased. Moreover, when a node is sleeping resulting in an increased delay because it must be queued until the node wakes up. In addition, with an increase in the network size, there are more schedules to be maintained by each node and this will causes more overhead which in turn results in the lifespan being shorter. Another drawback is that S-MAC is unfair; nodes with larger packets to transmit are allowed to have control over the medium and nodes with shorter packets must wait until the medium is free.

2.3.1.2 T-MAC

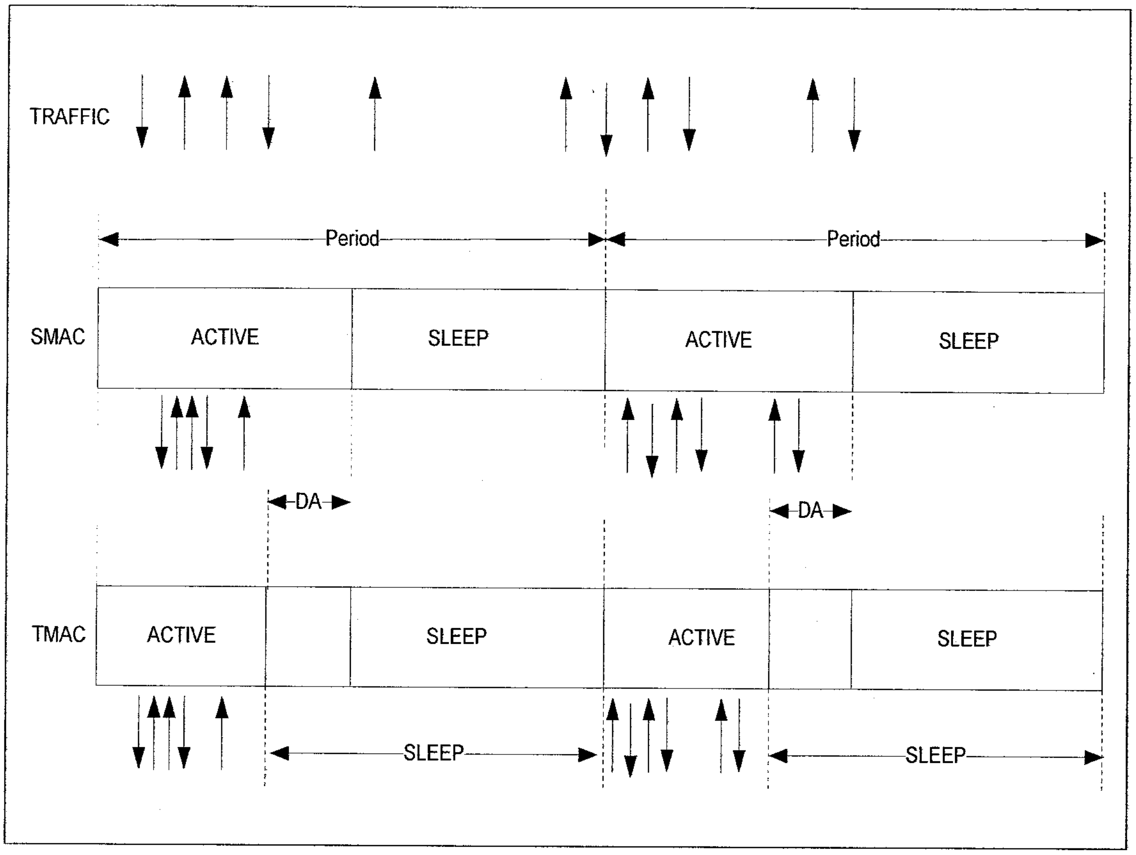


Figure 2.3 TMAC prematurely downsizes active period of SMAC adopted by [40]

The Timeout-MAC (T-MAC) protocol enhances S-MAC in regards to idle listening [40]. It assumes that message rates vary but latency requirements and buffer space are usually fixed. Under these assumptions, the periodic listen and sleep cycle of S-MAC is no longer optimized. To adjust for the varying message rate, messages are sent by T-MAC nodes in bursts of different lengths.

The initialization of T-MAC is, similar to S-MAC, until all of the nodes have sleep schedules. Figure 2.3 shows the key idea of TMAC and the comparison among TMAC and SMAC, where each node predicts activities in an active period so that it should switch off its radio if no task to be accomplished in that active period. It can be observed from the Figure 2.3 that the node in TMAC turns OFF its radio even in an active period if no operation is found in order to conserve energy and remains in sleep state for longer period of time as compared to SMAC. On the other hand in SMAC the node has to be in an active period and turn ON its radio throughout the whole active period. Normally, node will wake up at a set time in order to communicate with its neighbors; it will remain in this state until any *activation event* has completed for a certain amount of time. Such events include the reception of data, the firing of a periodic frame timer, the sensing of communication on the radio, the end-of-transmission of a node's own data packet or an acknowledgement, or that a neighbor's data exchange has ceased.

A contention-based scheme is utilized by T-MAC to avoid collisions; however, it does not make use of the usual technique of increasing the contention interval. The medium becomes saturated and the traffic load stays rather high as a result of each node sending its queued messages in a burst as soon as it wakes up. Because of this, the request to send (RTS) of a transmitting node starts by listening for an unspecified time with a contention interval that is fixed; this takes place even when a collision has not happened. If there is no clear to send (CTS) reply received by the node, it retransmits the RTS. If there is again no CTS reply, the transmitting node stops transmitting and returns to sleep mode. However, when maximum throughput is necessary, overhearing avoidance is not utilized by T -MAC. This is because, if a node sleeps when it hears an RTS or CTS packet, other control packets might not be heard which would cause a reduction in the maximum throughput.

Figure 2.4 shows that the Node 1 is interested to transmit data to Node 2. Node 1 will first transmit RTS, and if in response it receives CTS from node 2 than node 1 transmits the actual data and receives feedback through acknowledgement (ACK) packet. A drawback to T-MAC is that it is susceptible to an *early sleeping* problem when a unidirectional path is used by the traffic. This problem is noticed when a third hop node (Node 3), supposed to be the next relay of an ongoing transmission, prematurely goes to sleep. As a result, the node must wait until the next contention cycle to transmit its packet. Two solutions are available for this problem. In the first solution as shown in Figure 2.4, the node transmits a *future request to send* (FRTS) packet upon being hindered by neighbor nodes. In this way, neighbor node spends more time waiting so that it can avoid the corruption of its message by the FRTS packet. Simultaneously, the neighbor receives the FRTS packet and is aware that it must remain awake.

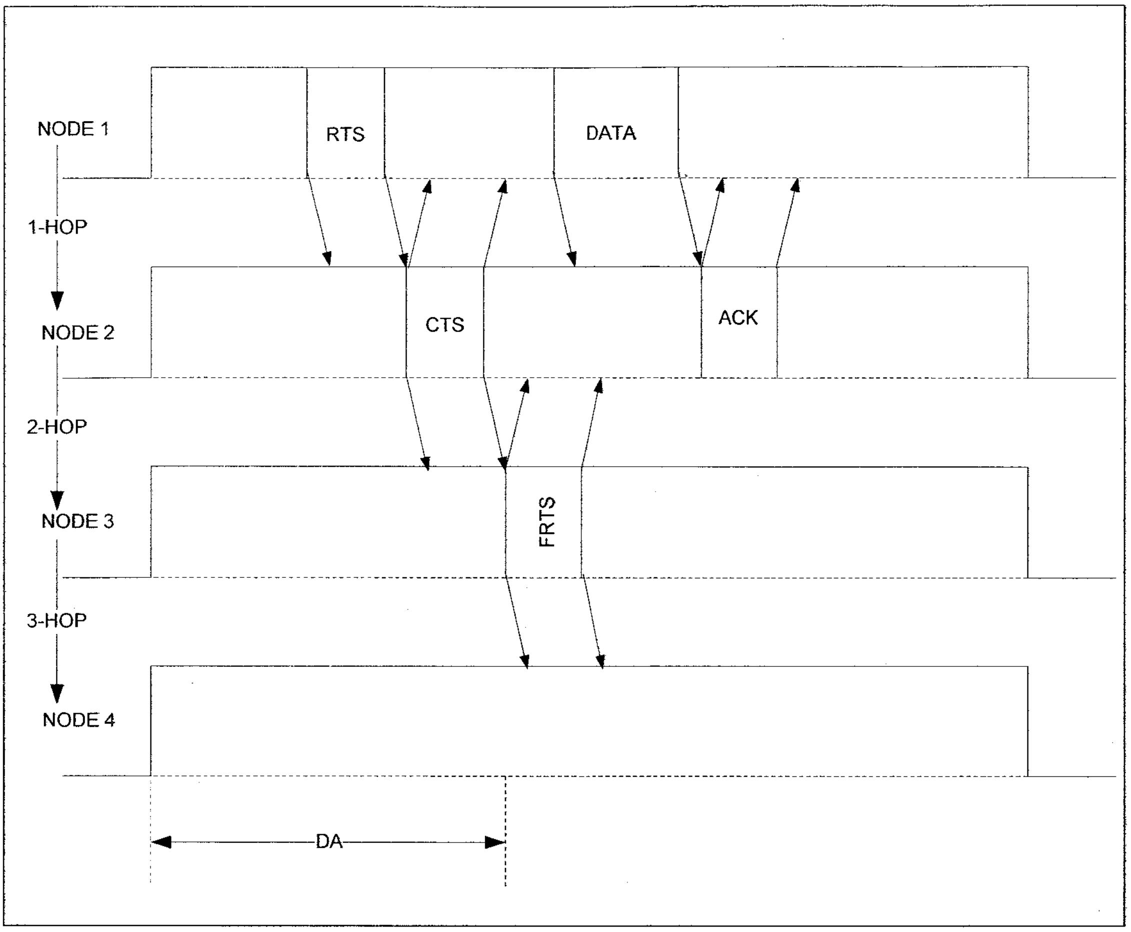


Figure 2.4 Future request to send message to avoid packet corruption adopted by [40]

In the second solution, a node that has been trumped is allowed to re-trump the original node. If the buffer node almost full and an RTS are received, the node replies with request to transmit (RTS) instead of clear to send (CTS). In this way it gets priority to transmit its packet in order to empty its buffer. However, this function must not be used carelessly as its usefulness could cause negation leading to a large number of collisions. It is specified by T-MAC that a node can only use this technique if another node has trumped it twice.

While it is true that T-MAC uses energy more efficiently than S-MAC, it does it at a cost to not only throughput but also latency. In addition, it is no better off in terms of scaling as S-MAC; this is because its overhead increases as the network size increases.

2.3.1.3 B-MAC

While the energy limitations of WSNs are improved with S-MAC and T-MAC, they were created to handle generic traffic loads. On the other hand, the Berkeley Medium Access Control also known as the B-MAC protocol was created on the assumption that WSN data is transmitted periodically in short packets [41]. However, B-MAC is merely a link protocol; as such, it requires that higher applications control other services. In this way, the applications of the node itself are totally responsible for the optimization of the consumption of power, latency, fairness, throughput, and reliability. Moreover, B-MAC is able to adapt in a more efficient manner to a dynamic topology and can tolerate network conditions which are altered.

Clear channel assessment (CCA) is utilized by B-MAC to know whether a channel is clear or not. A node makes use of CCA, to estimate the noise floor through the analysis of various signal strength samples of a channel assumed to be free; an example would be right after transmission of a packet is completed. Upon being ready to send a packet, the node monitors the energy of the channel and searches for outliers that are quite below the noise floor. If an outlier exists, it is assumed that the channel is clear and valid data packets are received. On the other hand, if after five samples no outliers were found, the channel is considered to be busy. If it finds that the channel is not busy, the node will send a random *backoff*, and then run CCA again. If it finds that the channel is busy, the node will again go through a random *backoff*; otherwise transmission will be initialized by the node.

Low power listening (LPL) is implemented by nodes in order to use less energy. In LPL, a node goes through stages and tests the channel periodically. A node is in sleep mode in the first stage. Then, the node is woken up by a timer. As shown in Figure 2.5 upon waking, the receiver turns ON its radio and begins to start carrier sensing in check interval for any activity (preamble) on the channel.

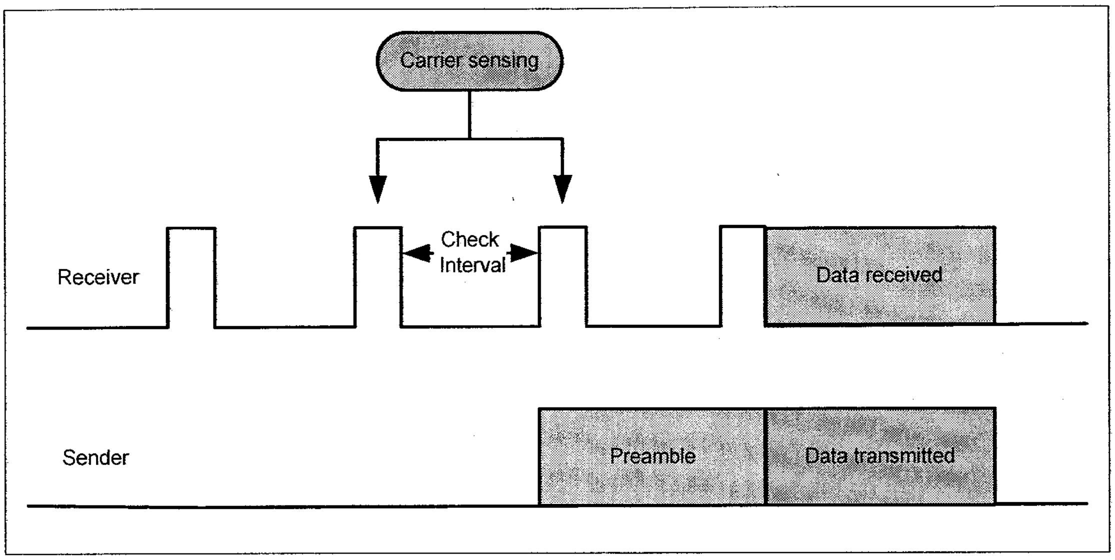


Figure 2.5 BMAC introduces preamble for transmission and reception by [41]

If there is no detection of any activity (preamble), the receiver is put into sleep mode again by a timer. If there is detection of any activity, the receiver stays awake until it has finished receiving the incoming packet. When transmission has completed it goes back into sleep mode. The maximization of the interval between the times that LPL tests the channel is performed in order to prevent any idle listening.

By utilizing reconfiguration, feedback and interfaces with higher-layer applications, B-MAC has been seen to exceed S-MAC's and T-MAC's performances. Furthermore, with B-MAC, applications are not forced to increase their overhead with synchronization and state maintenance. B-MAC outperforms S-MAC and T-MAC in terms of throughput, latency, and energy consumption utilizing only the default B-MAC parameters with no additional data required.

2.3.1.4 RI-MAC

In most of the existing protocols, preamble techniques were introduced which may engage the transmission medium for a much longer time than the actual data. These preamble technique protocols prevent all of the neighbouring nodes from transmitting with pending data. As the neighbouring nodes have to wait for longer periods of time till the medium is occupied, the network experiences a high delay. To overcome the issues caused by preamble technique protocols, the Receiver-Initiated MAC (RI-MAC) was introduced [42].

The RI-MAC minimises the medium occupying the time between a sender and its intended receiver by finding a rendezvous time in order to exchange data packets, while decoupling the duty cycle of the sender and receiver's as scheduled in B-MAC. Figure 2.6, provides the detail overview of the RI-MAC operation, in which the intended receiver initiated the DATA frame transmission.

In RI-MAC, all of the nodes maintain their schedule and wake up periodically based on the schedule. Upon waking, the nodes listen to the medium, if the medium is found to be free, the node announces its status that it is ready to receive DATA. The nodes which previously have pending DATA to send start their DATA frame transmission. As shown in Figure 2.6, node A with the pending DATA listens to the beacon messages from node B in order to start its pending transmission. Upon receiving a beacon messages, node A will start its DATA transmission. For the correct data reception, node B will respond to node A by an acknowledge (ACK) message. Note that the ACK message has two indications for a transmitter: first, the acknowledgement for the reception of the correct DATA, and second, the invitation of a new DATA for the same receiver. If a node does not have more DATA to send it goes into the sleep state.

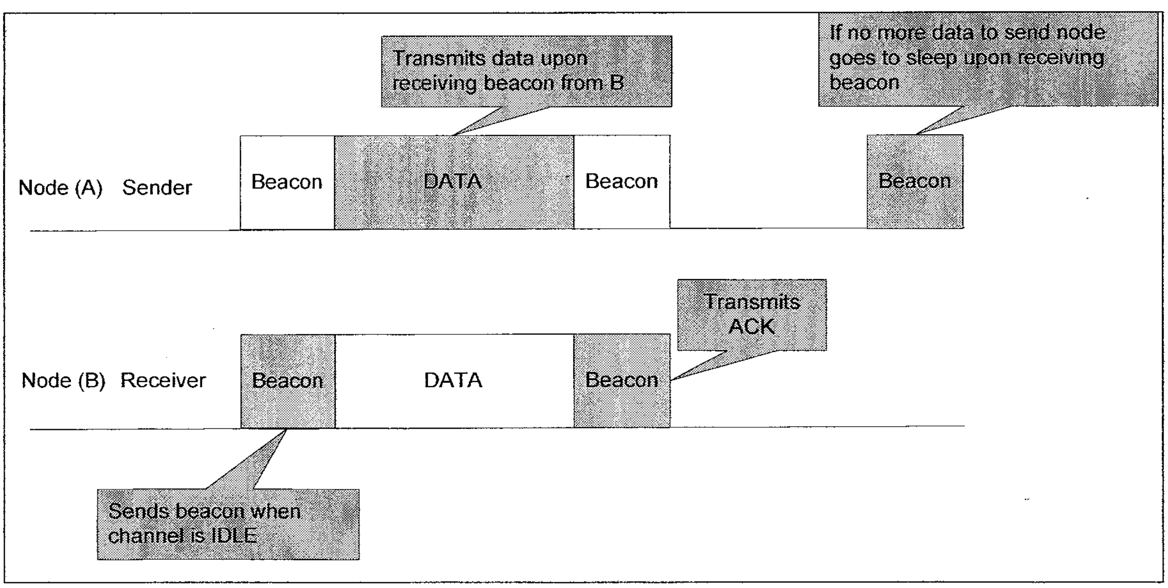


Figure 2.6 Data transmission and reception concept in RI-MAC adopted by [42]

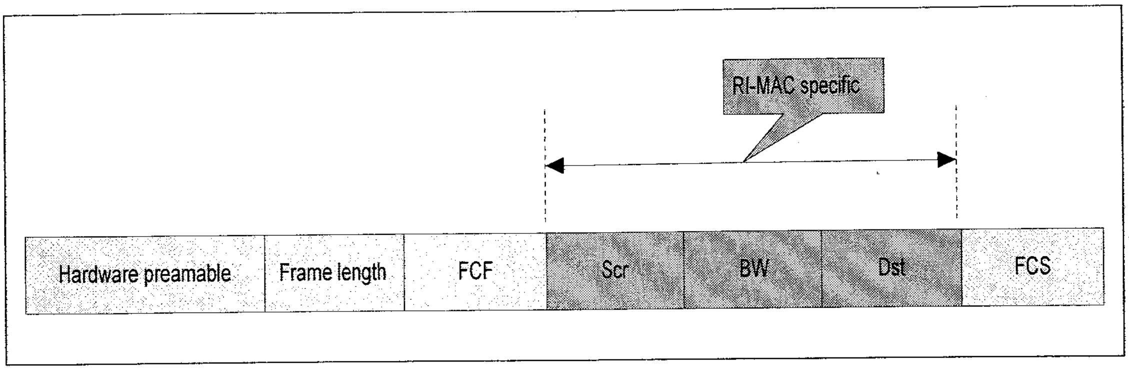


Figure 2.7 Format of beacon frame in RI-MAC [42]

As shown in Figure 2.7, the RI-MAC beacon frame consists of three fields: Scr field, BW field, and Dst field while the frame check sequences (FCS), frame control field and frame length are the fields of IEEE 802.15.4. Here, the Scr field and Dst field refer to the address of the source and destination while BW is the backoff window size. Scr is a compulsory field while Dst and BW are optional fields. Upon receiving a beacon, the node can compute which field exists in the beacon message. There are two major roles of the RI-MAC beacon: the first is the indication of the packet reception and the second one is the request for the transmission of the next DATA frame. If the receiver node experiences that the channel is busy, it takes a random back off and sends a beacon after some time.

RI-MAC aims to minimise the cost of active time and collision detection in order to provide power efficient transmission. In RI-MAC, a beacon frame is employed for coordination. If more than one node has pending DATA to transmit to a contender receiver, then there are chances of collision. In these circumstances, BW plays an important role to avoid collision. As shown in Figure 2.8, if nodes A and C transmit there DATA to node B without receiving a beacon from node B then a collision occurs at node B. In order to avoid the collision, the receiving node, B, transmits the beacon message to the sender nodes, A and C, with different BW sizes in order to avoid collision. Hence, nodes A and C after receiving the different BW, transmit their DATA at different times to avoid a collision. If there is no BW in the beacon message, a node can immediately start its transmission after receiving the beacon messages.

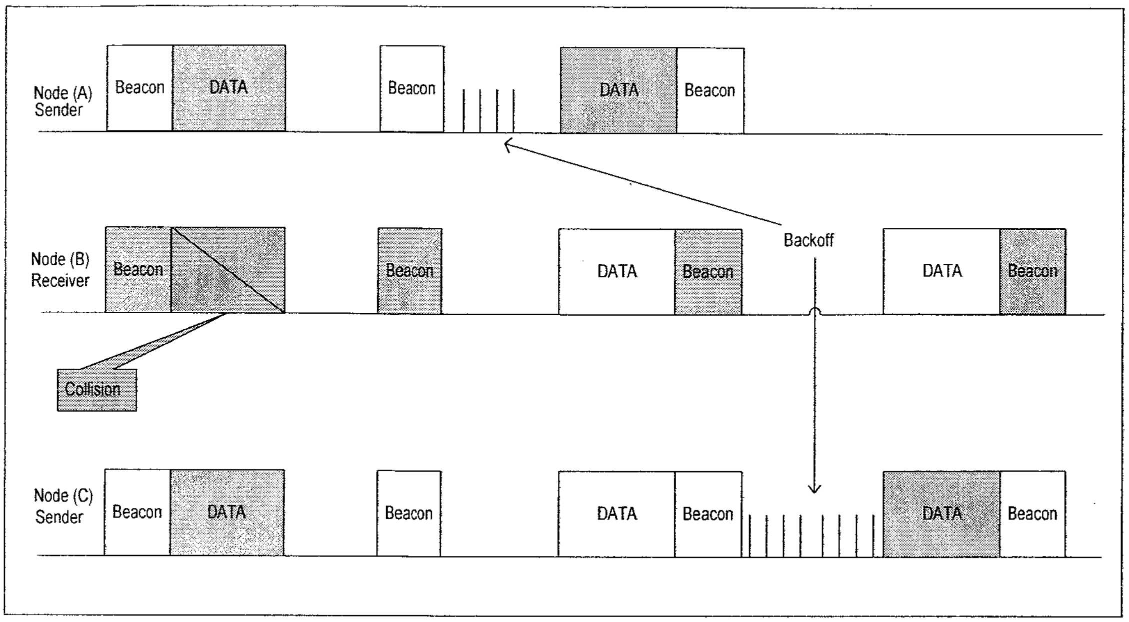


Figure 2.8 RI-MAC DATA frame transmission by back off mechanism [42]

2.3.1.5 PW-MAC

Predictive-Wakeup MAC (PW-MAC) is a contention-based MAC algorithm and minimises energy consumption by enabling sender nodes to predetermine the wakeup time of the receiver node [43]. For accurate time prediction, PW-MAC has introduced an on-demand prediction mechanism that accurately address the time changes due to clock drift, unpredictable hardware, and operating system delays. PW-MAC, even in busty traffic, achieves high energy efficiency through the prediction based retransmission mechanism.

The aim of the PW-MAC is to switch ON the sender radio before the wakeup time of the intended receiver. For collision avoidance, PW-MAC strives to enable each sender sensor node to predict the intended receiver wakeup time. In PW-MAC, the wakeup time is computed by utilising a pseudo-random wakeup-schedule rather than a fixed regular schedule. The pseudo-random wakeup-schedule avoids the simultaneously wakeup times of all the conflicting nodes, as such, a consistent and simultaneous wakeup gives rise to collisions and it mostly occurs in fixed wakeup schedules.

The PW-MAC predictive-wakeup mechanism is easily applicable to receiver-initiated MAC protocols (e.g., RI-MAC) and also to sender-initiated MAC protocols (e.g., WiseMAC). The duty cycle of the receiver-initiated provides good performance for the state prediction of a node. Therefore, PW-MAC has adopted the design of receiver-initiated for the predictive-wakeup mechanism. Pseudo-random numbers are generated by different techniques but for the sake of simplicity, PW-MAC has introduced a linear congruential generator (LCG).

$$X_{n+1} = (aX_n + c) mod\ m \tag{2.1}$$

Each node in a network has different parameters to avoid conflict among the generation of their pseudo-random number. As m, a, c and X_n are the indication of pseudo-random numbers and if sender node (S) learns about a pseudo-random number of a receiver node (R), S can figure out all the future predications in order to avoid conflicts.

Figure 2.9 illustrates the overview of the predictive-wakeup mechanism in PW-MAC. In most of the contention-based protocols, nodes wake up periodically after a fixed interval which is resolved by RI-MAC, in which each node wakes up on the basis of a beacon as shown in the figure. PW-MAC adopts the same technique of RI-MAC but based on a pseudo-random number. As shown in Figure 2.9

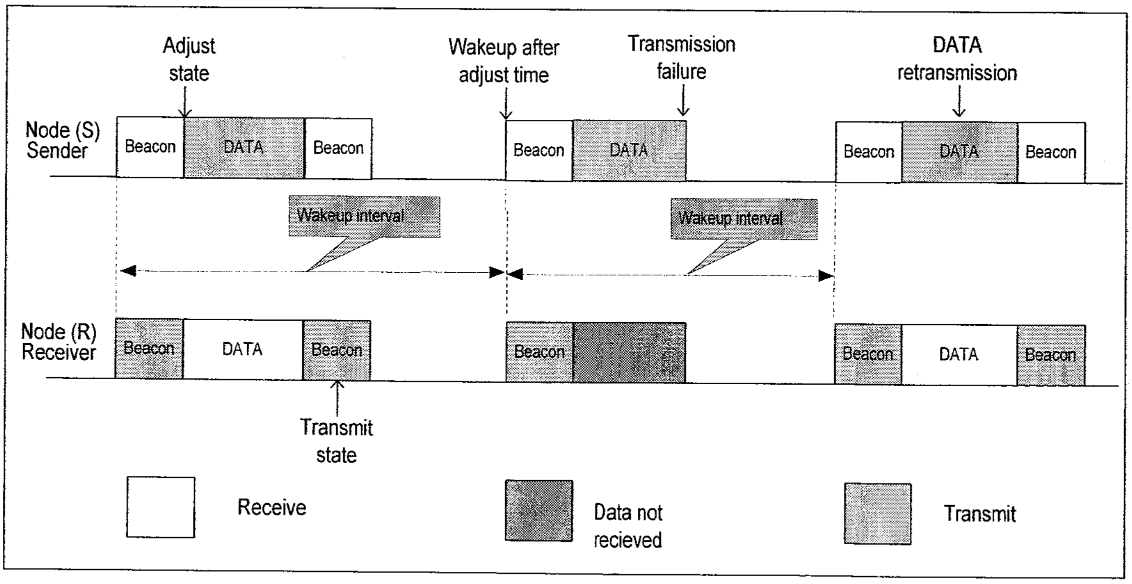


Figure 2.9 PW-MAC predictive-wakeup mechanisms for transmission referred by [43]

the receiver node, R, transmits its state in a beacon signal to the sender node, S, which upon receiving the beacon signal from R becomes aware of its schedule. Node S sets its flag and S turns OFF upon the calculation of the next wakeup time. When S wakes up after the pseudo-random number interval, it transmits DATA to R. Node R upon receiving DATA will acknowledge S by an acknowledgement packet. If the acknowledgement packet is not received by S, it will again transmit DATA in the next wakeup duration. Once node R successfully receives DATA from S, it predicts its S state and computes its current time. It will then send back an ACK for the successful reception of DATA and also request of S the transmission of the next packet. As soon as S receives the ACK from R it computes the time difference of R and gets aligned to that time in order to minimise the system delay.

The ACK beacon information helps node S to predict its future wakeup duration and time by the calculation of the time difference between the S and the R times. If node S has any pending DATA for R, in the future, S wakes up before R and transmits the DATA, as illustrated in the figure above. In contrast to earlier techniques (e.g., RI-MAC), node S in PW-MAC wakes just before R and transmits its DATA without any delay; while in RI-MAC, each time the sender node's DATA transmission depended on R which resulted in a higher delay. Hence, once S predicts the state in PW-MAC through the beacon, it reduces the delay and the idle listening almost to 0. This results in greater energy efficiency as well.

Collision is one of the major sources of energy wastage and it also leads to message overhead because after every collision, DATA packets are retransmitted. Mostly, WSNs are battery powered and have a limited lifespan; therefore, DATA transmission should be efficiently handled in WSNs. PW-MAC includes a prediction-based retransmission which achieves better energy efficiency even under busty traffic. In PW-MAC according to the prediction-based retransmission when node S transmits DATA and does not receive any ACK from R, S recognises that either the DATA transmission has failed or the ACK message has failed. Node S turns OFF its radio and wakes up at the next receiver's predicted time and retransmits the DATA while minimising energy and delay.

The prediction error is the time difference between the predicted wakeup time and the actual wakeup time of receiver R. The prediction error should be tackled carefully because if the intended receiver wakes up earlier or much later than the actual predicted time, it will miss matching the timing between R and S which will increase delay and also the duty cycle. The wakeup timing miss match could be easily handled through clock drift. Sender node S and receiver node R both in PW-MAC can adjust their wakeup time by making use of the on-demand prediction correction mechanism.

2.3.2 Schedule-Based MAC Protocols

Time-division multiple access (TDMA) is the basis on which schedule-based protocols are developed [44]. They conserve energy by making use of both scheduling and reservations. In this way, collision-free communication is guaranteed without the contention-introduced overhead [45]. This is because slots are scheduled for each node. Idle listening is also decreased which results in considerable energy savings [46]. However, making use of a TDMA protocol necessitates that nodes create real communication clusters instead of the virtual ones normally present in CSMA protocols. It is not an easy job to manage inter-cluster communication and interference job. Determining which slot is assigned to which node, the high initial overhead when setting up and distributing a schedule for throughout in the WSN, and accurate time synchronization that can prevent clock drifts in order to keep the time slots of the nodes from overlapping are some of the challenges when using the TDMA protocol. Furthermore, it is not easy for a TDMA-based protocol to change its schedule without resending overhead packets when the number of nodes in a cluster is altered. Therefore, scalability is better in contention-based protocols.

2.3.2.1 An Adaptive Low Power Reservation Based

These schedule-based MAC protocols uses TDMA scheme as the primary channel access method. It is self organization wireless sensor network protocol, where each node maintain a TDMA-like frame, known as superframe [47]. In order to avoid interference between two adjacent links, different channels are assigned. Each node gets the information about the transmission time of its next packet and piggybacks this information over its current DATA packet. Low reservation based MAC has hierarchical structure similar to that of LEACH [48, 49]. Every node within the cluster wants to contend the roll of a cluster-head, so the node that first captures medium become cluster head. In a situation where the node receives packets from multiple cluster-head, node chooses the one with largest signal power. Inside cluster the cluster-head is responsible for maintaining the synchronization.

Low reservation based MAC protocol uses single shared channel, which is divided into synchronized superframes. As shown in Figure 2.10, super-frame is further divided in four parts: Control slot, Reservation Request (RR) window, Reservation Confirmation (RC) slot, and Data slots. Control slots have information which is broadcast by the cluster-head such as requests for cluster-head rotation and length of the next superframe. Nodes which are willing to send data will access the medium by sending Reservation Request in RR window. RR packet contains information about source and the intended destination. When RR packets are successfully received by cluster-head, it reserves data slot for transmission of Data packet in the Data window slot. All other nodes wake up when they transmit RR packet. In the RC slot, the cluster-head broadcasts the RC packet, which contains the data packet transmission schedule of all the nodes whose RR packets were successfully received during RR window.

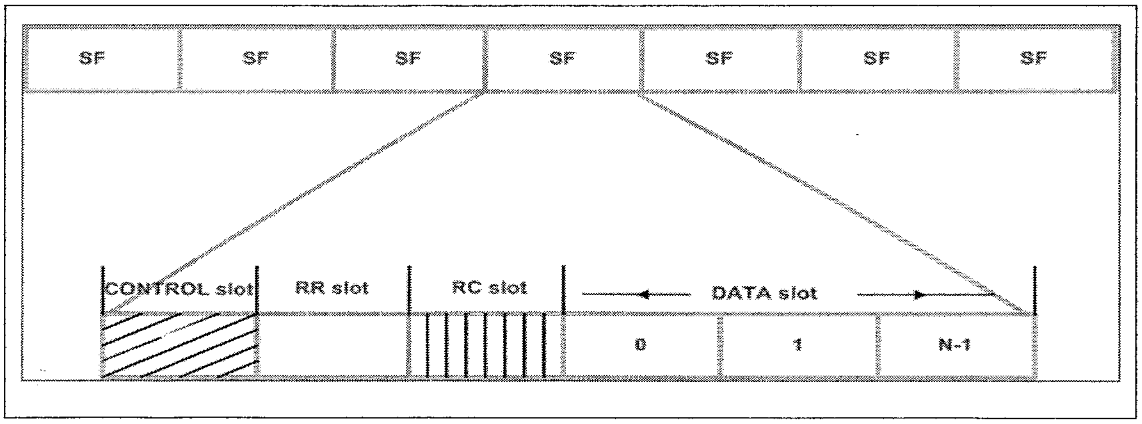


Figure 2.10 Superframe format of Low reservation based MAC proposed in [47]

All nodes wake-up in RC slot in order to receive the RC packet. Thereafter, each source node goes into a sleep mode and wakes up only in its designated Data slot. After successfully completion of data transmission by one cluster-head decides to relinquish its role. For the next round new node adopts role of cluster-head by going through several steps mentioned above.

2.3.2.2 TRAMA

TRaffic-Adaptive Medium Access or TRAMA is different from previously discussed MAC protocols because it supports traffic that is unicast, broadcast, or multicast[50]. It is, by nature, collision-free because of the use of TDMA. It makes use of a dynamic method to change nodes to a lower power on the basis of the pattern of the traffic. It is made up of three elements: the *Neighbor Protocol* (NP), *Schedule Exchange Protocol* (SEP), and *Adaptive Election Algorithm* (AEA). The information and schedules of the neighbors are exchanged by the first two elements. The third element uses the neighbors' information to choose receivers and transmitters for a given time slot; this lets all of the other nodes go into sleep mode, thereby, collision-free transmissions are achieved.

TRAMA's NP shares one-hop neighbor information while being initialized. As shown in Figure 2.11 each node must contend with its neighbors to send packets which contain incremental neighborhood updates in a *signaling slot* that is chosen randomly. In this way, nodes become aware of the one-hop neighbors of their one-

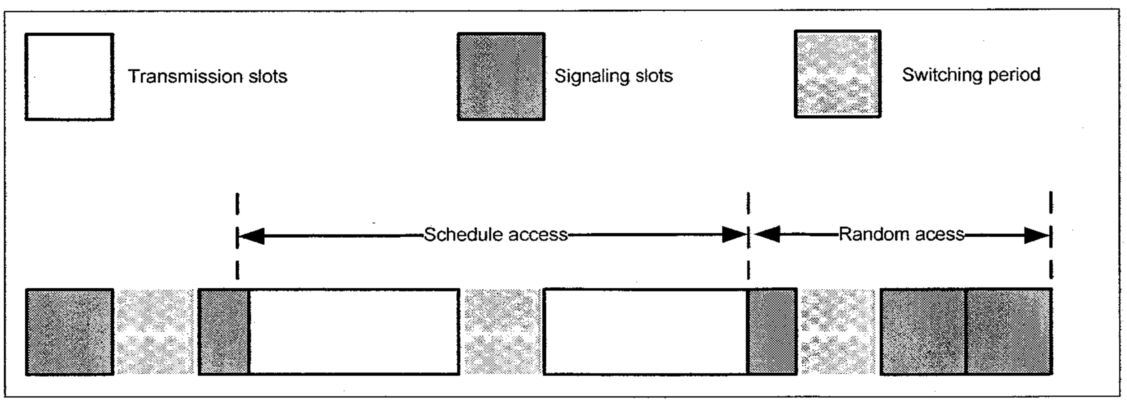


Figure 2.11 Time slot organization [50]

hop neighbors. This allows for two-hop neighbor information to be generated across the whole of the network. A neighbor is taken off from a node's neighborhood list if the node does not hear from the neighbor after a certain amount of time. Nodes will transmit signaling packets during its time slot so that the premature removal of active nodes is prevented. This takes place whether there are any updates or not.

When information is known about a two-hop neighbor, traffic-based schedule information amongst the neighbors is generated and maintained by TRAMA's SEP. A schedule is generated by each node by comparing an interval of slots with its two-hop neighbors. The slots for which the node has the highest priority are the slots in which transmissions can be made by that node. The node broadcasts a schedule packet that contains a bitmap representing each one-hop neighbor in order to inform to its neighbors that it plans to make a transmission. A neighbor is an intended receiver if the corresponding bit in the bitmap is set. In the case of a transmitting node not having enough packets to use its reserved slots, the node informs its neighbors of the situation and gives the slots up to be used by them. At the end, the individual nodes save the last reserved slot for transmitting their schedules for the next interval. A node's schedule is transmitted with every data packet so that the schedule can be maintained. There is a timeout associated with each schedule and a node cannot make any changes to the schedule until the timeout has ended. This guarantees that there is consistency among all of the one-hop neighbors. Each individual node which updates the schedule by utilizing the information transmitted with every data packet. Furthermore, all of the nodes listen during a *ChangeOver* slot; this is the slot after which all reserved slots are unused so that schedules can be synchronized.

Neighborhood and schedule information from NP and SEP are utilized by AEA to choose receivers and transmitters for the present time slot, allowing all of the other nodes to go into sleep mode; in this way collision-free transmissions are achieved. Each node initiates AEA so that it can make a decision as to whether it should receive, transmit, or sleep based upon the announced schedules from one-hop neighbors and the present node priorities. A node will only send packets when it possesses the highest priority among its two-hop neighbors and has information to be transmitted. A node decides to receive packets after it has analyzed the schedule of the sending node and has determined that it is an intended receiver. If the node is not an intended receiver it will go into sleep mode. A hidden node problem can be avoided if each node, before going into sleep mode, accounts for the two highest-priority transmitting nodes. Otherwise, a node with a highest priority does not have any packet for transmission it goes to sleep modes. If at the same time any node from two-hops transmits the packet for highest priority node, this would result in no reception of the packet by the highest priority node because it is in sleep mode.

A 40% higher throughput than S-MAC is achieved by TRAMA. Moreover there is a considerable amount of energy savings because it is a schedule-based. However, since it is schedule-based, an increased delay is incurred. Therefore, it is more suitable for use in applications that are more tolerant of delays and in need of a guaranteed reliable delivery as well as energy efficiency.

2.3.2.3 A New TDMA Scheduling Algorithm for Data Collection over Tree-Based Routing in Wireless Sensor Networks

In WSNs, data gathering applications may have variable traffic demand. Therefore, due to a random data rate and optimal delivery duration, TDMA scheduling algorithms are preferred over contention-based. A new TDMA Scheduling Algorithm for Data Collection over a Tree-Based Routing in WSNs is a slot scheduling algorithm that enables sensor nodes to allocate time slots proportionally to the traffic demand level [51]. According to the algorithm, each node in a network is sorted as per priority. The node closer to the sink is given a higher priority over the node away from the sink. The scheduling in the algorithm is accomplished on the basis of the congestion degree of a sensor node in the network.

Congestion is one of the problems in WSNs that should be taken into consideration for the data collection process. The congestion mostly exists with the nodes closer to the base station of the sink node because the sink node not only generates data for the base station but is also responsible for forwarding the data of all of the nodes in its routing tree table. Each node in the network can generate its own data as well as gather and forward the data of its neighbouring nodes. At any particular time, some of the nodes do not forward any data but at any other instant they forward a lot of data packets. Therefore, the nodes that forward more data packets experience more congestion that over all leads to network delay. A new TDMA Scheduling Algorithm for Data Collection over Tree-Based Routing in WSNs aims to reduce congestion in a network based on the prioritisation among the nodes. The node with the higher priority is most likely given more time slots to transmit its data. Figure 2.12 shows the node priority which is calculated based on the number of the degree of a node (number of child nodes attached to a node).

Node colouring in a network is carried out in such a way that none of the two nodes that conflict with each are assigned the same colour. The slot assignment of the algorithm is a bit complicated because each time, at the beginning, the congestion degree is computed and after that all of the nodes in the network are sorted as per the decreasing degree of congestion. If two or more sensors nodes in a network have the same congestion degrees, then the nodes are further sorted according to their conflict degree and the node with the higher conflict degree is prioritised.

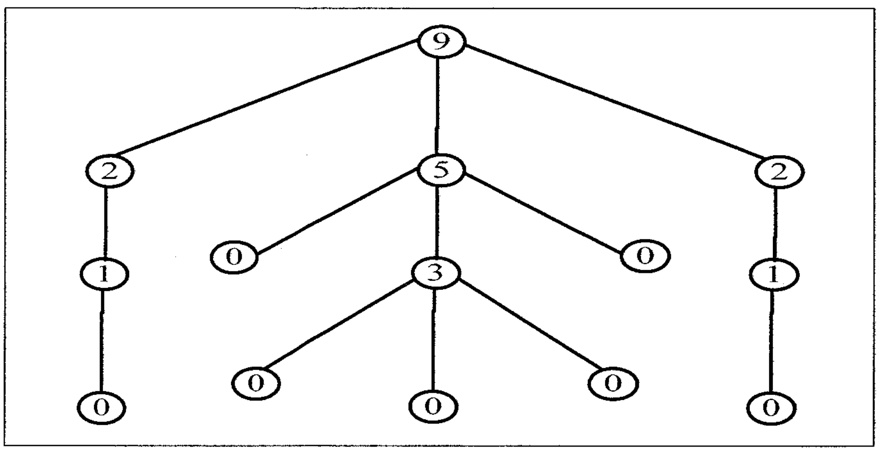


Figure 2.12 Degree are assigned to nodes as per child node from [51]

2.3.2.4 A Depth-based TDMA Scheduling for Clustering Sensor Networks

The depth-based TDMA Scheduling (DB-TDMA) is a TDMA scheduling based hybrid protocol for WSNs. The object of DB-TDMA is to support the fluctuating demand of the data gathering in a network in a power efficient manner. According to DB-TDMA, each node is scheduled a time slot in a network by breaking down the network into a clustering hierarchy. Nodes, during their time slot, can receive or transmit data while in other time slots, the nodes are permitted to go into the sleep mode in order to conserve energy. The DB-TDMA protocol is composed of three parts: depth-based scheduling for inter-cluster scheduling, TDMA for intra-cluster scheduling, and fluctuated demand support.

According to DB-TDMA a network is broken down into small clusters and each cluster further organises the cluster nodes in a tree hierarchy [52]. In each cluster, one node acts as a cluster head that is responsible for managing the whole cluster. In DB-TDMA, the cluster head is selected on the basis of its depth (level) and each cluster head wakes up sequentially in a tree as per its depth. Let A be the area of the network and R be its radius then the number of the depth of the network could be determined as:

$$D = \frac{M\sqrt{2}}{R} \tag{2.2}$$

The whole time frame is broken into small time slots and each node is assigned the time T_S . The time T_S is long enough for the nodes to carry out all of the primitive operations within this duration. The cluster head's wake up is set as:

$$S_{head}(d) = T_{\frac{period}{T_S}}\left(1 - \frac{d}{D}\right) \tag{2.3}$$

Where, $S_{head}(d)$ is the wake up of cluster node, T_{period} is the sampling period and d is the depth of a particular cluster head. So, the cluster head wakes up at $S_{head}(d)$ and this time is assumed to be long enough that the cluster head can manage to process and receive the data from the entirety of its child nodes.

Actually, TDMA slot scheduling is carried out during the intra-cluster scheduling. The cluster head, after recognising its child and member nodes, establishes the TDMA schedule for all of the nodes in its cluster. The schedules of the nodes should differ from each other to avoid the conflict in the network. Thus, the schedule of the child nodes is computed as:

$$S_{Transmit}(d) = S_{TDMA} + S_{head}(d) \tag{2.4}$$

There are three types of applications according to the sensed data fuse rate. The first is the aggregation where the data fuse rate is 100%; this means that all of the data should be fused perfectly. The second is the acquisition where the data fuse rate is 0; this means that the data could not be fused and all of the sensed data packets should be sent to the base station (BS). The third is the partial aggregation where the data fuse rate is between 0 and 1; this means that only a few parts of the sense data can be fused. Each node in a DB-TDMA is assigned at least one time slot for its operation. If any of the nodes require more slots, it transmits the request for the extra time slots to the cluster head in the buffer queue. Upon receiving the extra time slot request, the cluster head recomputed the time slots and assigns extra time slots to that particular node.

2.3.3 Hybrid Protocols

Hybrid Protocols are a combination of schedule-based and contention-based protocols; by utilizing them together energy savings can be achieved and at the same time their respective weaknesses can be offset [53]. Simplicity, flexibility and robustness are all offered by protocols that are contention-based and do not need a lot of infrastructural support. These advantages took a large number of trial and error. Additionally, because of the problem with hidden nodes, packet collisions can happen within any two-hop neighborhood of a node. Minimisation of these collisions can be achieved by making use of RTS/CTS; however, it results in a high overhead consuming 40% - 75% of the capacity of the channel's. On the other hand, the problem of hidden nodes can be solved by schedule-based protocols by scheduling the nodes of the neighbors' to make their transmissions at different times; however, these protocols have drawbacks of their own. It is not easy to create a schedule that is efficient as there is a need for each node to take care of the clock synchronization. When the synchronization is tighter, the overhead required is also higher because there are more frequent exchanges among nodes. Furthermore, alterations to the WSN topology necessitate alterations in the schedules which induce more overhead.

2.3.3.1 Z-MAC

Zebra MAC (Z-MAC), as shown in Figure 2.13 a hybrid protocol created on the basis of CSMA, maintains high channel utilization by making use of TDMA during intervals of high contention and CSMA during intervals of low contention [54] . In its worst case, Z-MAC performs identical to CSMA. It is made up of four sequential procedures: Neighbor discovery, slot assignment, local frame exchange, and global time synchronization. These procedures only function at the time of the initialization period of the WSN or after considerable changes have been made to its topology.

During neighbor discovery, a ping message is periodically broadcast by each node to its one-hop neighbors. This message contains an updated one-hop neighbor list. In this way, a list of the nodes' two-hop neighbors can be created by each node. One-hop and two-hop neighboring list is used by Z-MAC with the DRAND algorithm to allocate a time slot to each node, ensuring that there are no two-hop neighbors sharing the same slot, Detail explanation of DRAND is provides below in section 2.3.3.2 [26]. A *time frame* is then created by each node; this is the interval during which the node can use its time slot. It is ideal the same time frame is shared by all the nodes of each two-hop neighborhood. However, in dynamic WSNs, each alteration in the topology would necessitate that updated time frames be generated throughout the entire network resulting in energy being wasted. To account for alterations to the topology, the *time frame rule* of Z-MAC lets each individual node maintain its own local time frame which fits its two-hop neighborhood; however, conflicts among all contending neighboring nodes are avoided up to large extend. After the time frame

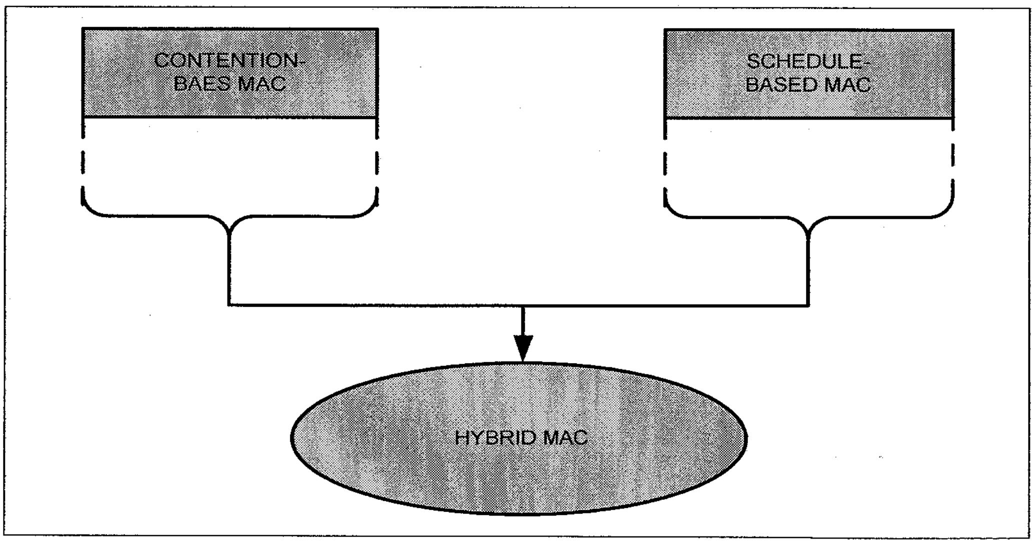


Figure 2.13 Hybrid MAC protocols switching between contention-based and schedule-base

and slot number of each node has been determined, the node then broadcasts this data to its two-hop neighborhood and their time slots are synchronized to slot 0. Each node maintains its own time slot periodically transmits a synchronization message that contains its present clock value.

In Z-MAC a node can operate in a *low contention level* (LCL) or a *high contention level* (HCL) mode. A node has to compete in order to make a transmission in the present slot only when the slot belongs to it or the node is a one-hop neighbor to the node that owns the slot while in the HCL mode. However, when in the LCL mode, a node has to compete in any slot. On the other hand, when in either mode, it is the slot's owner that has the highest priority compared to the other nodes. A slot can be used by other nodes if it does not have an owner or there is no data to be transmitted by the owner. When a node receives an *explicit contention notification* (ECN) message from a two-hop neighbor within a given time, it enters the HCL mode. While ECN functions in a way that is similar to RTS/CTS, it avoids collisions by making use of information about the topology and slots. An ECN message is transmitted by a node when the node has determined that there is high contention among the nodes as measured by the noise level of the channel.

The CCA, backoff and LPL interfaces of B-MAC are utilized by Z-MAC so that LCL and HCL can be implemented. Upon being ready to transmit data, a node checks to see whether it possesses the slot or not. If the slot belongs to the node, it takes a random backoff for a certain amount of time. As soon as the backoff timer ends, CCA is utilized by the node to sense the channel; it then sends the data if the channel is not busy. If the channel is busy, it goes through the process until the information can be transmitted. If the slot does not belong to the node and the node is in LCL, or if the node is in HCL and the slot does not belong to its two-hop neighbors, a random backoff is taken by the node within a contention window and otherwise the node performs as was described previously. If the slot does not belong to the node and the node is in HCL as a result of a neighbor transmitting an ECN, the node goes into sleep mode. It remains that way until a slot arrives that belongs to it or does not belong to a two-hop neighbor. At that time, it wakes up and goes through the previous process. The B-MAC's LPL mode is utilized for the nodes to receive packets.

Z-MAC has a performance that is no worse than CSMA at rates of low transmission. However, when the rates of transmissions increase, Z-MAC outperforms B-MAC in terms of fairness, energy efficiency and throughput. However, they both possess similar latency, matter what the transmission rate may be.

2.3.3.2 DRAND

DRAND is the most famous distributive scheduling algorithm and a ZMAC relies on DRAND for scheduling purpose. It enhances bandwidth utilization by combining the strength of the time division multiple access (TDMA) MAC and the carrier sense multiple access (CSMA) MAC protocols. It switches between TDMA and CSMA with correspondence to the contention-level in order to use the bandwidth more effectively. According to DRAND algorithm, initially all the nodes are in IDLE state. Each node tries to access the medium. All the nodes in a network toss a coin, whose probability of getting head or tail is *1/2*. If a node gets head, it tries for lottery, which has some preset probability of success. If a node loses the lottery, after T_a time (where $T_a = 3d_a$ and d_a estimate one message delay) it will again try to win a lottery. As shown in Figure 2.14, node H after successfully winning a lottery moves to REQUEST state and broadcast REQUEST message [21]. If all the neighbours of node H are in IDLE state, they will send back a GRANT message to node H. The GRANT message has a record of the slots reserved by 2-hop neighboring nodes of node H. For collision-free scheduling, it is important to know the slot reserved by the nodes within 1-hop and 2-hop neighbors. Hence, node H on receiving the GRANT message is familiar about the slot reserved by its 1-hop and 2-hop neighboring nodes. After receiving the GRANT message, node H will reserve the slot for transmission which is not reserved by its 1-hop and 2-hop neighboring nodes. Finally, node H broadcast RELEASE message, which contains information about the slot reserved by node H and goes back to IDLE state.

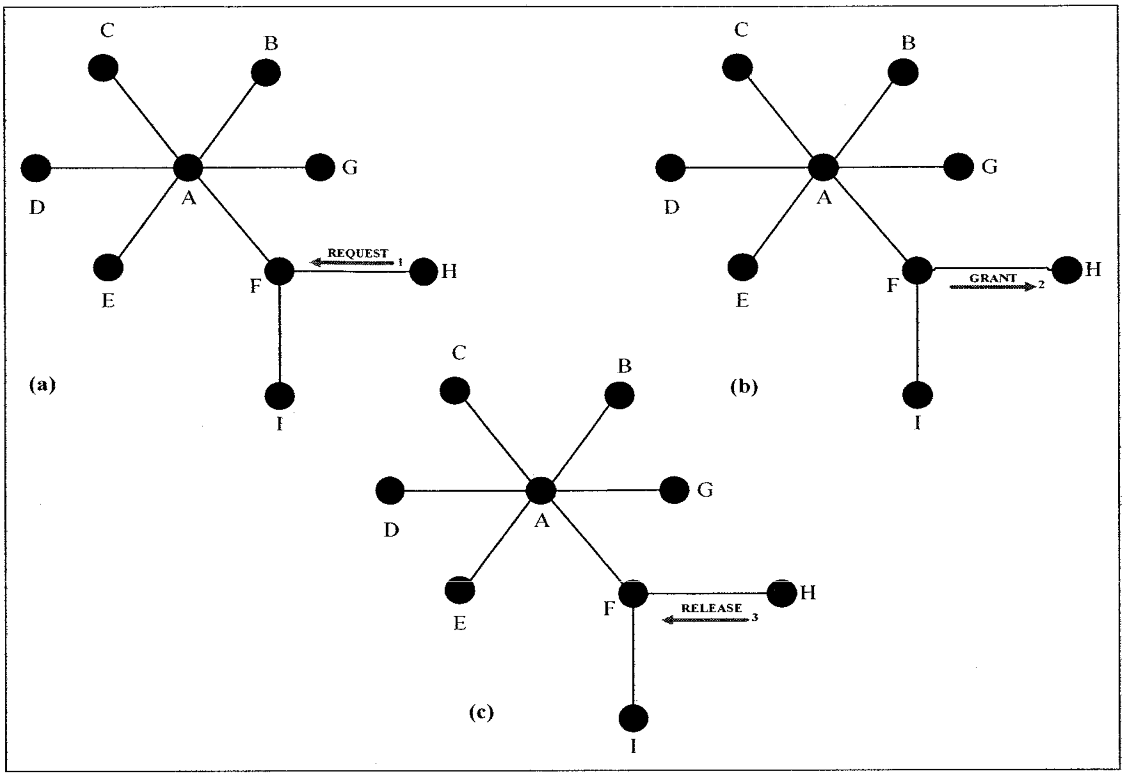


Figure 2.14 Successful scheduling cycle of DRAND

Node H will successfully get GRANT only if all its neighboring nodes are in IDLE or RELEASE state [26]. For an example as shown in Figure 2.15, if node F has already given a GRANT message to node I before receiving REQUEST message from node H, it will send back a REJECT message to node H. The REJECT message states that node F has already given a GRANT message to any of its neighboring node. On receiving the REJECT message, node H will broadcast a FAIL message to its entire 1-hop neighboring node, which states that node H is not reserving any slot at this particular cycle. After that node H will go to IDLE state and when T_a time expires it will again retransmits REQUEST message. Within time T_a node I reserve a slot for itself and broadcast RELEASE message. If node A after T_a time REQUEST to node F for GRANT before node H, node F gives GRANT message to node A. This case states the unsuccessful cycle of DRAND because node F has no record about node H, that it has requested before node I, so first GRANT should be given to node H. Node

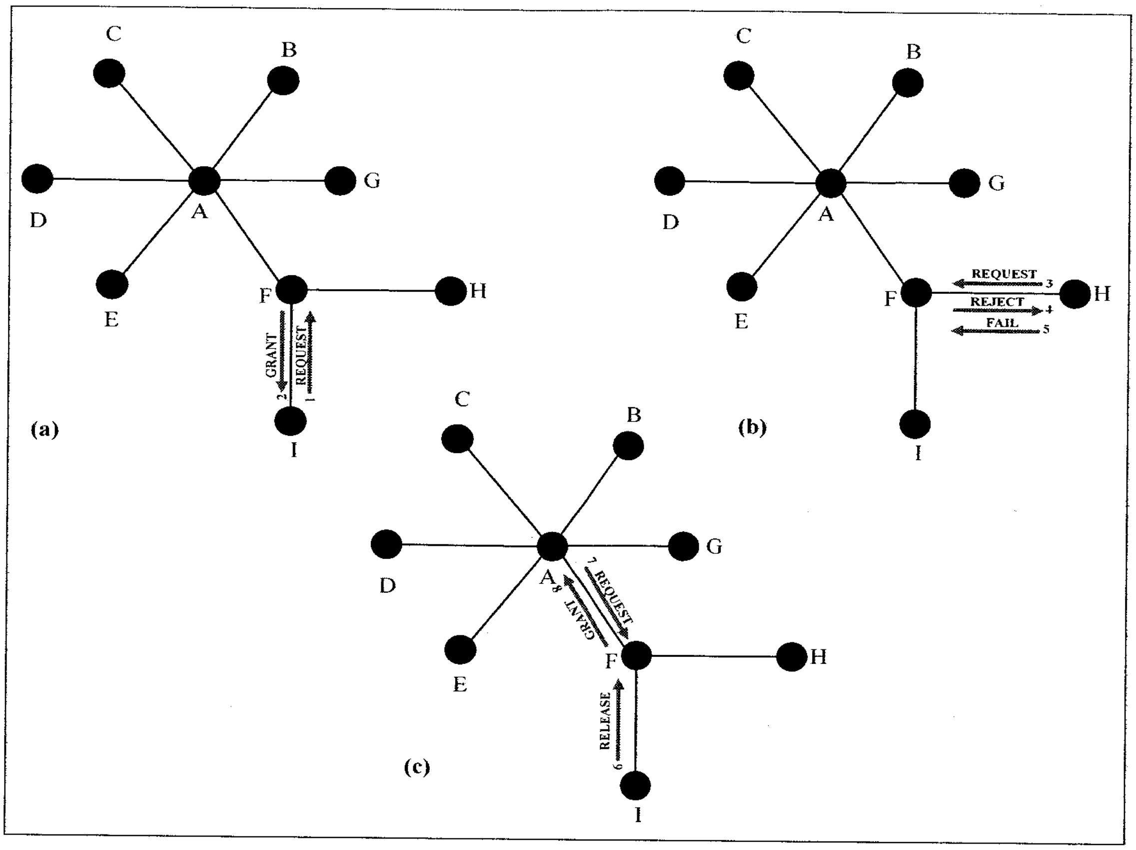


Figure 2.15 Unsuccessful scheduling cycle of DRAND

F once again will send REJECT message to node H. On receiving REJECT message node A will again broadcast FAIL message. Although, DRAND slot assignment presents reservation scheme but each time REQUEST, REJECT and FAIL message transmission leads to unnecessary message overhead that increases latency, message overhead and energy consumption, which motivates our work.

2.4 NETWORK MODEL AND COMMON NOTATIONS

In this thesis, the major focus is on single-hop as well as on multi-hop WSNs. N number of sensor nodes were randomly deployed in an $A \times A\ m^2$area and equipped with identical radio transceivers. All the nodes have same transmission range (TR), which helps the nodes to identify the set of nodes in its communication range. Typically, a multi-hop (MH) WSN is represented by a graph $G = (V, E)$, whereas, $V = \{v_1, v_2, v_3, v_4 \ldots\ldots\ldots\ldots\ldots, v_n\}$ represents the set of nodes in a network and $E = \{e_1, e_2, e_3, e_4, \ldots\ldots\ldots\ldots\ldots, e_n\}$ represents the distance (edges) between the links

[34]. Node u and v are adjacent if both are within TR of each other and their distance is less than Euclidean distance, where $v = (u, v) \mathcal{E}\, V$ and the edge $e = (u, v)\, \mathcal{E}\, E$. More generally, the set of links are defined in equation 2.5:

$$E = \{(u, v)\ \mathcal{E}\ V.V\ |dist_{EU}\ (u, v) \leq TR\} \qquad 2.5$$

Where $dist_{EU}$ is the Euclidean distance.

DEFINITION:

Node u and v are in a one-hop $(H1)$ neighbourhood if both u and v are in TR of each other and there exists a link, $(u, v)\mathcal{E}\ E$.

$$\forall\ (u, v)\ \mathcal{E}\ H1, there\ exists\ a\ link, (u, v)\mathcal{E}\ E \qquad 2.6$$

Node u and v are in a two-hop $(H2)$ neighbourhood if there does not exist a direct link $(u, v)\mathcal{E}\ E$ but alternatively, there exists a link, $(u, w), (w, v)\ \mathcal{E}\ E$.

$$\forall(u, v)\mathcal{E}H2, there\ exists\ a\ link, (u, w), (w, v)\mathcal{E}\ E \qquad 2.7$$

Similarly, in this case node, u and v are connected through node w, which plays the role of a bridge between node u and v.

For collision-free transmission in WSNs, none of the nodes from $N2(u)$(two-hop neighbourhood of node u) can schedule the same time slot for itself that has been already reserved by any of the nodes from $N2(u)$, where $N2(u) = H1(u) \cup H2(u)$. $H1(u)$ and $H2(u)$ represent the set of nodes in one-hop and two-hop, respectively. $N2(u) \supseteq N1(u)$, where $N1(u)$ is in the 1-hop neighbourhood of node u.

Correspondingly, k-hop neighbours of node v can be represented by $\Delta_k(v)$, which is a set of all of the sensor nodes whose distance to node v is at most k. Thus, the number of k-hop neighbours of node v is denoted by $\delta_k(v)$, i.e, $\delta_k(v) = |\Delta_k(v)|$, and the maximum k-hop neighbours are given as $\delta_k(v) = \ max_k \delta_k(v)$.

2.5 INTERFERENCE MODEL

In WSNs, several sensor nodes in a network are wirelessly linked and equipped with Omni directional antennas. Due to the broadcast nature of the WSN, if all of the sensor nodes within the same TR broadcast simultaneously, this results in collisions, each collision make the node to transmit its data again which uses up at least two times the amount of the energy for the same data to be transmitted. Collisions and interference hinders throughput of the network. Generally, there are two types of interference in WSNs: primary interference and secondary interference [44]. Typically, the primary interference occurs when a node in the same time slot is assigned multiple tasks (sending, receiving from multiple transmitters) to be carried out. Secondary interference takes place when parallel transmissions are going on at the same time and in the same collision domain (i.e., nodes are in the interference range) where each destination node is tuned to one particular source node.

In WSNs, each node has a fixed interference range (IR) and transmission range (TR). Conventionally, $IR > TR$ and the ratio between them is $\gamma = \mathrm{IR}/TR$. Practically, $2 \leq \gamma \leq 4$ and the transmission will be successful at time τ between v_i to v_j if none of the nodes v_k within the same TR is transmitting at τ. More formally, it can be explained as:

$$v_i\ (\tau)\ \mathcal{E}\ \{0, 1\} \qquad 2.8$$

$v_i\ (\tau) = 1$ if none of the nodes from $H1(v_i)$ and $H2(v_i)$ is transmitting at time τ and

$v_i\ (\tau) = 0$ if any of the nodes from $H1(v_i)$and $H2(v_i)$ are transmitting at time τ.

Consequently, the two nodes u and v are adjacent and conflicting, i.e., $v\ \mathcal{E}\ N_u$ and $u\ \mathcal{E}\ N_v$, where N_v and N_u are neighbouring nodes of u and v. The focus of the research is to tackle the primary and secondary interference by scheduling distributive time slots.

2.6 PERFORMANCE METRICS/PARAMETERS

The main focus of this research is to indicate the network performance measured in a distributive manner. As scheduling is carried out during setup phase, therefore, the setup parameters are measured during setup phase until the network reaches steady state. The performance metrics are described below (See appendix A):

Running time: The time required by the nodes in a network to allocate collision-free time slots.

Message overhead: The number of messages exchanged during the assignment phase.

Energy consumption: Amount of battery power utilized to set up the scheduling process.

Number of Rounds: The number of rounds that nodes require to acquire a time slot.

Table 2.1 Current and Power Consumption Rates of CC1101 radio transceivers [55]

Node State	**Current (I)**	**Power (P = V I)**
Transmitting (at 0 dBm)	16.8mA	50.40mW
Receiving	18.3mA	54.9mW
Listening	1.8mA	5.4mw
Sleeping	0.0004mA	0.0012mW

Table 2.1 shows the current and power consumption rate of CC1101 radio transceiver [55]. The table indicates current and the respective power consumption rate in each mode. Power consumption is achieved simply multiplying current with battery voltage used. Further, energy consumption in each state can be computed by power consumed by a sensor node in that particular state to the proportions of time that node's spends in that state. Therefore, the total power and energy consumption will be the sum of powers and energies consumed by the network in each mode of operation.

Performance evaluation of DSSA and IDSA has been conducted to show that the proposed algorithms are optimal, topology independent and simple in terms of implementation. For subsequent comparisons, DRAND and the proposed algorithms were implemented in NS-2. The algorithms were tested by configuring various network topologies with increasing number of nodes in the network. The parameters used for the simulations are listed in Table 2.1.

In wireless networks, to establish and maintain the link connectivity among the neighbors' beacon messages are periodically exchanged which helps to select the best route and forward the data packet along that path. In WSNs, nodes are battery power so energy wastage should be avoided as much as possible because energy wastage shorten the life span of the network. Therefore, the working of WSNs should be under low power constraint and all the resource of energy wastage should be avoided as much as possible [56]. Moreover, in WSNs sensor node are wireless linked therefore the connectivity of links is too volatile because the low power radio of WSNs is vulnerable due to interference from other higher power radio [57, 58]. Additionally, mobility makes far more difficult to maintain the link connectivity and estimate the best route [59]. Thus, to estimate the best neighbor information is of great significance. Therefore, in this work TwoRayGround is selected as propagation model because it gives the best neighboring information with minimum overhead and energy consumption [60, 61] (See Appendix B).

The power consumption reported in the table are acquired from the standard CC1101 radio transceiver [55].The CC1101 provides functions for PHY layer, MAC layer and its radio operates at low power, that results in reliable wireless communication. Besides this the radio also supports multiple data rate, power consumption, and channels [62]. For a simulation scenario, the network topology was varied by randomly deploying sensor nodes on a $150m \times 150m$ surface area. Whereas, the TR of all the nodes was set at $40m$. Bandwidth and data rate are set to be the same that are equal to 250 kHz and 250 kbps, respectively. The data rate 250 kbps is the one rate offered by CC1101 radio transceiver. From the data rate and packets lengths transmission time can be easily computed which will be further used to compute the energy dissipation in each state of mode. The network density was increased by varying the number of nodes from 5 to 70. The simulation time is set to 700s, and the initial energy of each node was set to 100 Joule. The simulation is run

15 times and the results reported in chapter 4 were obtained after 15 repetitions of trials.

Table 2.1 Simulation parameters

PARAMETERS	VALUES
PHY Parameters	
Number of nodes	70
Simulation area	150m* 150m
Transmission range	40 m
Sensor node deployment	Random Deployment
Sensor node transmit power	1mW
Propagation model	TwoRayGround
Packet size(Request, Grant, Release)	50 byte
Bandwidth	250 kHz
Data rate	250 kbps
Packet reception-power threshold	-60dBm
TwoRay Propagation Model	
Tx and Rx Antenna Gain	1
Lambda	0.3279m
Simulation parameters	
Initial energy	100 J
Simulation time	700 sec

2.7 SUMMARY

This chapter discusses some of the most prominent techniques of MAC layer protocols that are responsible for regulating and sharing a common medium. Significant work was done by previous techniques to ensuring and overcome contention in a network by avoiding interference among conflicting nodes. The three broader categories of MAC layer protocols; contention based, schedule based, and hybrid MAC algorithms are examined by evaluating their performance and access methods. Lastly and the most important, existing distributive scheduling technique is discussed in detail which is the motivation of our research work. The next chapter presents proposed scheduling algorithms for a distributive, scalable and self-configurable network in order to ensure fair bandwidth allocation for collision-free transmission.

CHAPTER 3

DISTRIBUTIVE AND SELF-SUSTAINABLE SCHEDULING ALGORITHMS FOR WSNS

Based on the identification and challenging issues via literature, in this chapter scheduling algorithms have been proposed; both of these are capable of generating distributive scheduling schemes. The operation of these algorithms is not affected by the size of the network; moreover, it can accommodate random network changes and distributive slot allocation explicitly without reconstructing the global transmission schedule.

3.1 INTRODUCTION

In general, WSNs are distributive, self-configurable, and scalable with little or no predefined infrastructure [63]. WSNs are also highly correlated and may undergo topology changes due to node failures. Therefore, traditional scheduling algorithms are not preferable approach in order to ensure fair bandwidth allocation and to handle topology changes. This chapter provides the methodologies used in this work. The focus of this chapter is on the considerations and the approaches to be adopted based on which this research has been carried out. Further, the chapter continues with the detailed description of the proposed techniques which are divided into four phases: (1) Neighbour discovery, (2) Slot scheduling procedure, (3) Update procedure, and (4) Local framing. Figure 3.1, presents a sequence wise elucidation of these phases. Finally, the chapter is concluded by the contributions and design of the proposed algorithms which guarantee conflict-free scheduling schemes.

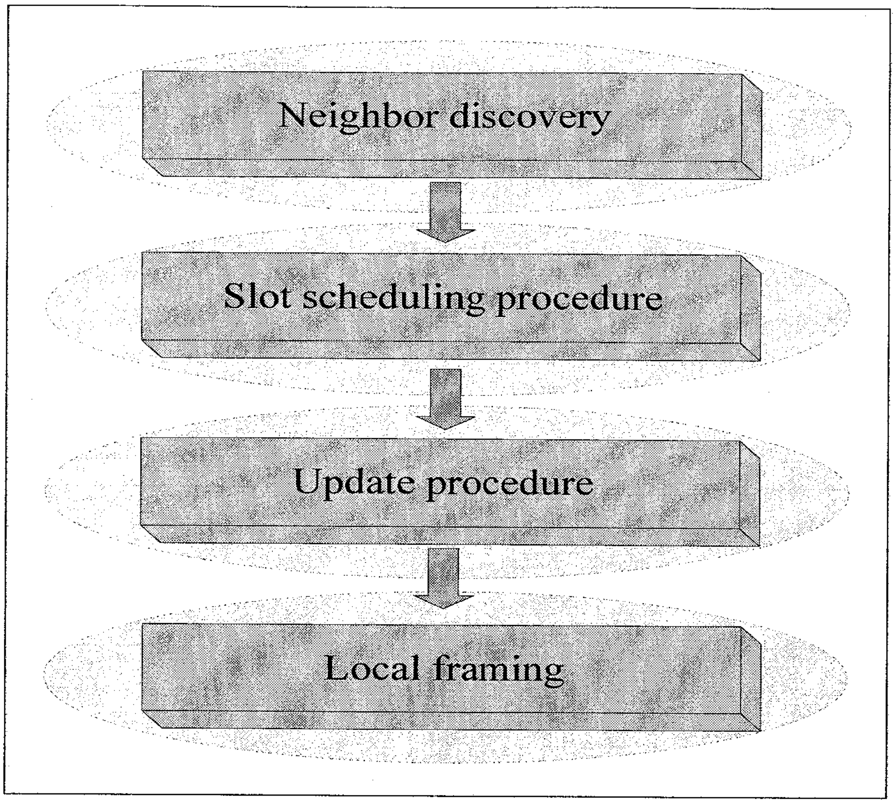


Figure 3.1 Sequence wise elucidation of the phases in the propose algorithms

3.2 DESIGN CONSIDERATION

The broadcast nature of WSNs is readily supported by the radio channel. Mostly, sensor nodes are equipped with Omni-directional antennas for broadcast communication, and the packet transmission will be successful if the packet is received by all of the nodes in the transmission range (TR) without any error. For successful transmission, the capture effect of the transmitter has to be observed. When the transmitter broadcasts, it should prohibit and block all of the other communication among its neighbouring nodes except for the intended transmitter. The capture effect of the transmitter is complicated and has no information of the nodes beyond its one-hop neighbours (conflicting nodes) which gives rise to hidden terminal problems [64, 65]. Communication can be classified into different categories depending upon designated receivers: unicast, broadcast, and multicast. For the most general case the multicast or broadcast nature of a network can be viewed with an arbitrary subset of neighbouring nodes up to a two-hop distance, whereas in real

WSNs, the communication is a combination of both the unicast and the broadcast. Therefore, the available bandwidth should be fairly and distributively shared among the nodes to control and manage all of the activities in a network in a precise manner without any conflict.

Generally, the transmission schedule in WSNs is equivalent to a graph colouring problem, where each time slot is represented by a unique colour. The unicast transmission schedule can be represented by edge colouring, whereas, the broadcast scheduling can be represented by the node colouring and the multicast scheduling can be represented by multiple edge colouring [66, 67]. The transmission schedule is the combination of edge colouring and node colouring. For optimal scheduling, the conflict-free transmission and colouring constraints must be considered. Optimal scheduling (optimal bandwidth efficiency is measured through the minimum TDMA slots used) is directly or incrementally NP-complete [68]. However, in a dense and mobile network, the bandwidth-efficiency is of highly concern with least redundancy. Mobile networks have a fragile nature and the transmission schedule may be corrupted with the movement of the nodes that leads to collision among all of the conflicting nodes. Therefore, the changes in a network topology should be updated frequently after every change. This refers to scheduling maintenance and should be performed in a cost-efficient manner. Compared to other networks, WSNs have limited bandwidth and computational power. It is desirable for generation of transmission schedules that the communication and computational overheads should be as low as possible. A force that completely tears downs the existing transmission schedules is the change in a network topology because after a topology changes apparently the transmission schedule is regenerated. A new transmission schedule reflects the change in topology and is redundant and costly, when a small portion of a schedule has to be outdated from the existing topology. For schedules, regeneration is evolutionary, or an incremental approach will be a more feasible approach. These approaches are more economical as compared to completely regenerating schedules because only the outdated part is rescheduled, due to node failure or mobility. Due to the self-configuring and dynamic nature of WSNs, distributive scheduling techniques are more preferable to handle scalability and robustness in an efficient manner. Nodal density and mobility could be partitioned down a network in many smaller networks, where each smaller network operates by itself. This requires the communication protocol and algorithms to handle scalability and dynamic topology changes; i.e., it can perform equally well both in a dense or in a sparse network.

For optimal and rapid schedule updates or regeneration, the focus should be on the local information of the network rather than global information. The transmission of a node can be affected by all of the conflicting nodes up to its two-hop neighbouring nodes. Hence, for conflict-free scheduling, it is significant for a node to have knowledge of the nodes up to two-hop neighbouring nodes, only. The local information is enough to design conflict-free schedules. Recently, hybrid MAC protocols have introduced some of the distributive scheduling algorithms for slot reservation [69-71]. Although these scheduling algorithms provide distributive scheduling techniques, the nodes have to repeatedly proceed through many primitive states and a lot of control messages are exchanged. Moreover, these techniques demands more scheduling time which will not be an efficient approach for a more sensitive and speedy network [72]. The anomalies and the drawbacks of the preceding techniques, which are not taken in account are the motivation of this research with the intention of not only designing conflict-free scheduling algorithms but the proposed algorithms should also be optimal, scalable, dynamic and fully-distributed conflict-free.

3.3 FRAMEWORK OF PROPOSED ALGORITHMS

Our conflict-free MAC scheduling algorithms follow the approach of the DRAND technique; where each time frame is divided into a fixed length depending upon a two-hop neighbourhood size. As shown in Figure 3.2, each frame is further divided into small equal parts known as time slots. The time slot refers to the time required by the sensor node to transmit or receive a message. In the proposed approaches, there is

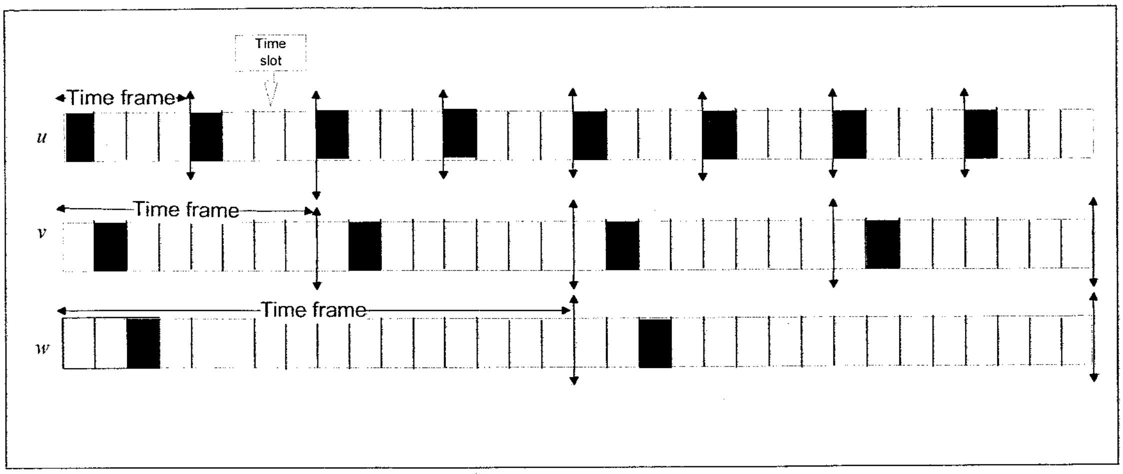


Figure 3.2 Time frame based on a number of two-hop neighbouring nodes for data transmission

no need of frame alignment at various nodes. In the Figure 3.2 u, v and w refers to the nodes in a network with neighborhood size of 3, 8 and 16. Therefore, each node in a time frame reserves a time slot for itself for transmission and reuses that slot for its transmission in each frame. However, the frame size of the nodes may differ depending upon two-hop neighbourhood size. Traditionally, most of the MAC scheduling protocols are frame aligned and are time synchronized.

Node failure and movement affects the network topology, therefore, existing scheduling may cause collisions. In order to avoid a collision among all of the conflicting nodes, the proposed scheduling approaches schedule a time slot in such a way that none of the nodes within the two-hop neighbourhood is assigned the same slot based on local information in an optimal way. To guarantee conflict-free slot reservation and dynamically adjust to workload more effectively, each node can recycle its time slot in its own frame as per the time frame (TF) rule. The detailed explanation of the TF rule is discussed in this chapter. The network remains in a setup phase until all of the nodes in a network's schedules their conflict-free time slot based on the local information up to a two-hop neighbour record.

In this research work two distributive scheduling algorithms has presented with minimal message passing, i.e., exchanges of control messages between interfering neighbours. Both of the scheduling algorithms' skeletons are quite different from each other but the center of attention is the same, where each node has to decide its own conflict-free slot based on local information. This research considered discrete time slots, i.e., each node decides its schedule based on local information and reserves a conflict-free time for its transmission time $\tau \in Z+$ where τ represents time of any particular slot and Z refers to the time frame. All the conflicting nodes at least up to two-hop neighbouring nodes are assigned a time step $\tau \in Z+$, for a collision-free transmission. Each node assigns $\tau = \tau + 1$ based on their neighbours. Hence, to ensure a successful transmission, node i attempts to transmit in its reserved slot at time τ such that none of its neighbours attempt to transmit, simultaneously, i.e.

$v_i(\tau) = 1$ when only node i transmits at time τ and 3.1

$v_j(\tau) = 0$ none of the neighbouring nodes $j \in N(i)$ (neighbours of i) is transmitting at time τ. 3.2

The appropriate skeleton of these techniques are described in detail in Section 3.5 and 3.6.

3.4 CONTENTION PHASE

Slot scheduling is carried out in the contention phase or setup phase. In the proposed algorithms, the focus is on the set-up phase. During the set-up phase, several operations run continuously, such as [73]:

- Neighbour discovery
- Slot assignment
- Local framing
- Synchronisation

In the proposed algorithms, these operations run once during the set-up phase to avoid complexity during transmission. These operations do not run until any significant topology change in the network takes place. All conflicting nodes are assigned a unique colour, where each colour represents a time slot in a frame. To avoid interference and collision among all of the conflicting nodes, each node up to $H2$ is assigned a unique colour and the same colour can be assigned to a node beyond $H2$.

3.4.1 Neighbor discovery

Neighbour discovery for both of the proposed algorithms are carried out at the beginning in the HELLO state. During the HELLO state, control messages are exchanged in a distributive and scalable manner throughout the network and each node discovers its $H1$ and $H2$. The important element is to properly recognise $H1$ and $H2$ to avoid future conflicts. For example, there may be a possibility that node A is aware of its neighbouring node B but node B is unaware of node A due to an asymmetric link. In order to accomplish the conflict-free time slot scheduling, three different methods are adopted through the HELLO state listed below. For simplicity, an assumption is made on the two nodes, A and B.

3.4.1.1 Broadcast

Both nodes, A and B, periodically broadcast hello messages to each other. If node A receives a HELLO message from node B, it will update its $H1$ and record the B's address in it.

3.4.1.2 Three-way-hand-shake

Suppose node A broadcasts the HELLO message and node B on receiving the HELLO message, will reply with a unicast HELLO message to node A using its address. Similarly, on receiving the reply from node B, node A updates its $H1$ and records B's address. Finally, node A replies to B with a unicast message and node B also updates its $H1$ and records A's address.

3.4.1.3 One-way and two-way list method

Figure 3.3 shows the algorithm used for neighbour discovery in the proposed techniques. At the beginning, all of the nodes broadcast HELLO messages to become aware of their $H1$ and $H2$neighbouring nodes. Here, flag is an indication for a sender that the receiver is aware of its neighbouring nodes.

Initially, for the neighbour discovery, simple broadcast HELLO messages were used in this work; it was quite simple but could not guarantee exact neighbour information under asymmetric links. Then, the simple broadcast HELLO messages were replaced by a three-hand-shake technique. Although the three-hand-shake technique supports the network under an asymmetric link, it is an expensive technique because it results in a large number of messages, energy and time for neighbour discovery. Additionally, if any of message acknowledgements out of the three-hand-shake fails, the whole process has to be rerun, which is quite expensive and is not supportive for an energy constraint network.

```
node A receives HELLO message from B
if A is in the  one-way-list of B
    if B is in the one-way-list of A
        add B into the two-way-list of A
        discard B from the one-way-list of A
        else if B is in the two-way-list of A
            set flag
        else if  Add B in the two-way-list of A
else if A is in the two-way-list of B
    if B is in the  two-way-list of A
        add B into the two-way-list of A
        discard B from the one-way-list of A
                set flag
        else if B is in the two-way-list of A
            do nothing
        else if Add B in the one-way-list and the two-way-list of
A
                do nothing
else if A is not in the one-way-list and the two-way-list of B
    if B is in the one-way-list of A
        add B into the one-way-list of A
    else if B is in the one-way-list of A
            set flag
        else if B is in the two-way-list of A
                do nothing
```

Figure 3.3 Pseudo code of neighbour discovery used by the algorithms

For the proposed algorithms, one-way and two-way schemes were devised. Figure 3.3, shows the working flow of the one-way and two-way techniques which work without any problem even under asymmetric nodes. One-way messages are for asymmetric node information. For example, if for the first time node A receives a message from node B, node A will record node B's address in a one-way array. And, the second type of array is the two-way array used for the symmetric node. For instance, if node B receives a message from node A, node B on receiving the message from node A, records A's address in its two-way array. Both of these arrays enable information of the symmetric one-hop neighbour to be obtained. One-way and two-way schemes guarantee the same result as the three-hand-shake but with less message overhead, almost to the level of the broadcast technique.

3.4.2 The Allocation Phase

In the allocation phase, each node reserves a time slot for itself, and its neighbouring nodes are informed about the schedules through transmitting schedule update packets. Slot scheduling is more sensitive, due to the conflicting nodes; a node which acquires a channel in the i^{th} round may be able to reserve its slot in the i^{th} round. However, the rounds for slot allocation are more likely to be increased with an increasing number of nodes in the network. Nodes, after scheduling their slots, broadcast schedule update packets. On receiving the schedule update packets, the neighbouring nodes become aware of the schedule of their neighbouring nodes and update their neighbouring list. Nodes only communicate in their designated slot while in other slots they listen for schedules transmitted by their neighbours.

3.5 OVERVIEW OF THE PROPOSED SCHEDULING ALGORITHMS

In the proposed scheduling algorithms, the nodes are allowed to distributively assign transmission schedules among themselves based on the network composition. Slots are reserved in the setup phase through exchanging the control messages in order to support unicast, multicast, and broadcast transmissions. Furthermore, with the change in a network's topology, the schedules adjust accordingly to maintain conflict-free transmission schedules. Both of the algorithms cope with network topology changes in a distributive and incremental manner.

All the nodes in a network equally participate in the scheduling process. The scheduling process is simultaneously executed across the entire network. Each node tries to contend a medium in order to reserve a slot, and several nodes may contend for a medium to acquire a free medium and simultaneously reserve their transmission schedule. Overall, this reduces the network degradation parameters and enhances the robustness. Basically, each sensor node has to maintain its own transmission schedule. That is because, for a conflict-free and collision-free transmission a node can only reserve a slot which is not reserved within its one-hop and two-hop neighbours. If any node in a network suffers a slot confliction due to some topological change, through the attachment or detachment of nodes, the node learns this from the local information and reschedules its own slot to avoid a collision. Due to the local nature of the proposed algorithms, it is neither sensitive to the network size nor affected by the network partition. The proposed algorithms are suitable for large, distributive and self-configurable WSNs.

For the subsequent comparisons with existing DRAND scheduling algorithms, following assumptions were considered which were similar to that of DRAND, such as:

1. All of the nodes in a network have a unique ID.
2. Initially, all of the nodes are synchronised at slot 0.
3. The network is fully connected and each node is familiar with its $H1(u)$ and $H2(u)$.
4. Multiple nodes in a same TR cannot communicate at the same time.

3.6 DISTRIBUTIVE AND SELF-SUSTAINABLE SCHEDULING ALGORITHM (DSSA)

Distributive and self-sustainable scheduling algorithm (DSSA) is a new scheduling approach that allocates a transmission schedule in a collision-free manner. DSSA provides a distributive scheduling technique as well as obviates the weaknesses of traditional algorithms. DSSA not only resolves scheduling issues, but also focuses on to trim down the network degradation parameters as much as possible. In the DSSA, each sensor node collaborates with its $H1$ and $H2$ to generate its own transmission schedule. Scheduling through DSSA is simple and local because every node collaborates up to its two-hop neighbouring nodes only. Each node directly interacts with its $H1$ but the nodes apart from that up to $H2$ causes interference. Nodes, for their transmission schedule in the DSSA, are indirectly aware of the schedules of the nodes up to their $H2$ and reserve a slot that is not reserved by any of the nodes in their $H1$ and $H2$ to avoid a collision. For example, three nodes $(u, v \text{ } and \text{ } w)$ have to schedule their time slot, where (u, v) and (v, w) are in $H1$of each other, similarly (u, w) are in $H2$. If node u schedules $(S0)$ for itself, then node v reserves $(\text{S}0 + 1)$. Similarly, node w reserves $(\text{S}0 + 2)$ to avoid a collision. Nodes after scheduling their own slot, broadcast this information and the nodes in $H1$ update this information in their record and later on pass this information to $H2$ to avoid future conflicts.

Scheduling is always carried out on the basis of existing topology. At the completion of scheduling, neighbouring nodes are updated about the schedule. A schedule is modified when a network suffers a collision due to topology changes. Schedules for new nodes are carried out on the basis of local information in such a way that it should not conflict with any of the nodes within its IR. This results in evolving scheduling with the changing topology over time. Figure 3.4 show the flow of DSSA and the description of the DSSA is illustrated in more detail in Figure 3.5.

3.6.1 Algorithm description

The idea behind the DSSA is to minimize the chances of unsuccessful cycles because each unsuccessful cycle results in additional computation. In the DSSA, each node decides its own time slot based on the local information gathered by its two-hop

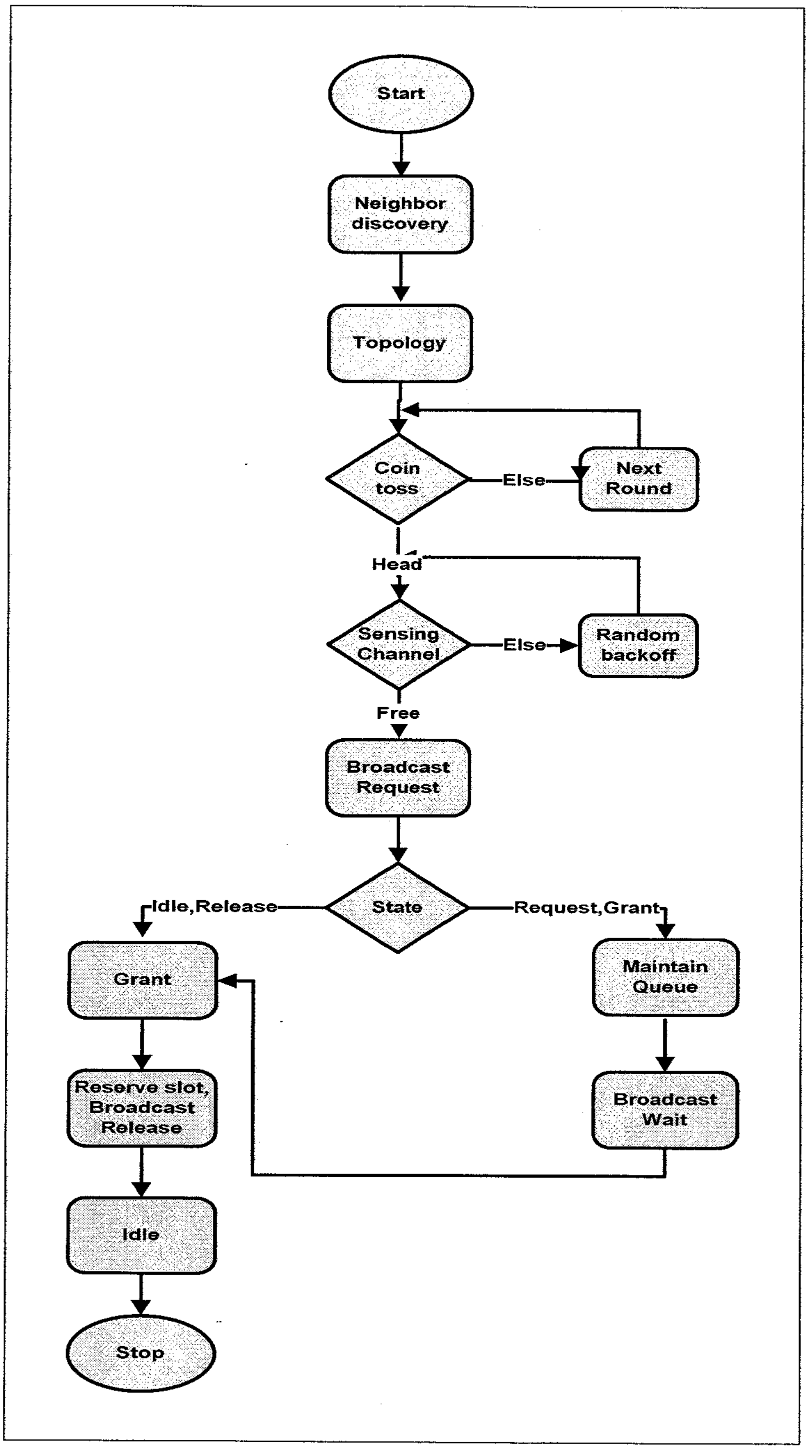


Figure 3.4 Flow sequence of the DSSA

neighbouring nodes ($N2$). The DSSA algorithm runs in rounds where each node reserves a conflict-free slot in a heuristic manner. The DSSA algorithm runs in parallel and consists of the following two policies:

- Scheduling policy
- Update policy

3.6.1.1 Scheduling policy

During the scheduling policy, each node reserves a conflict-free time slot in a distributive manner. The DSSA runs in rounds and derives heuristic time slots among all of the nodes in a simpler and more optimal way by avoiding any additional computation. In the DSSA algorithm, at the beginning, all of the nodes are in the IDLE state and each node competes to schedule a conflict-free slot. Each node, for the first time, has to proceed through a coin toss and lottery phase as explained in the DRAND technique. If the node successfully passes the coin toss and lottery phase, then it reserves the minimum unassigned slots among its $H1(u)$ and $H2(u)$. Figure 3.5(a), shows the slot assignment procedure of the DSSA in which, initially, node I successfully proceeds through the coin toss and lottery phase to broadcast a REQUEST message to the entire $H1(\mathrm{I})$. If all $H1(\mathrm{I})$ are in the IDLE or RELEASE state, they will reply with a GRANT message. After sending a GRANT message, $H1(I)$ waits for the RELEASE message of node I that which slot it will reserve. As shown in Figure 3.5(b), if at the same time node H and then node A comes in with a REQUEST message, node F will not send a REJECT message to node H and A as in the DRAND case; instead of a REJECT message, it will reply with a WAIT message. Actually, the WAIT message is an indication for node H and A that node F is aware of their REQUEST and has maintained them in its QUEUE. So, there is no need to send a REQUEST message repeatedly as when node F receives a RELEASE message

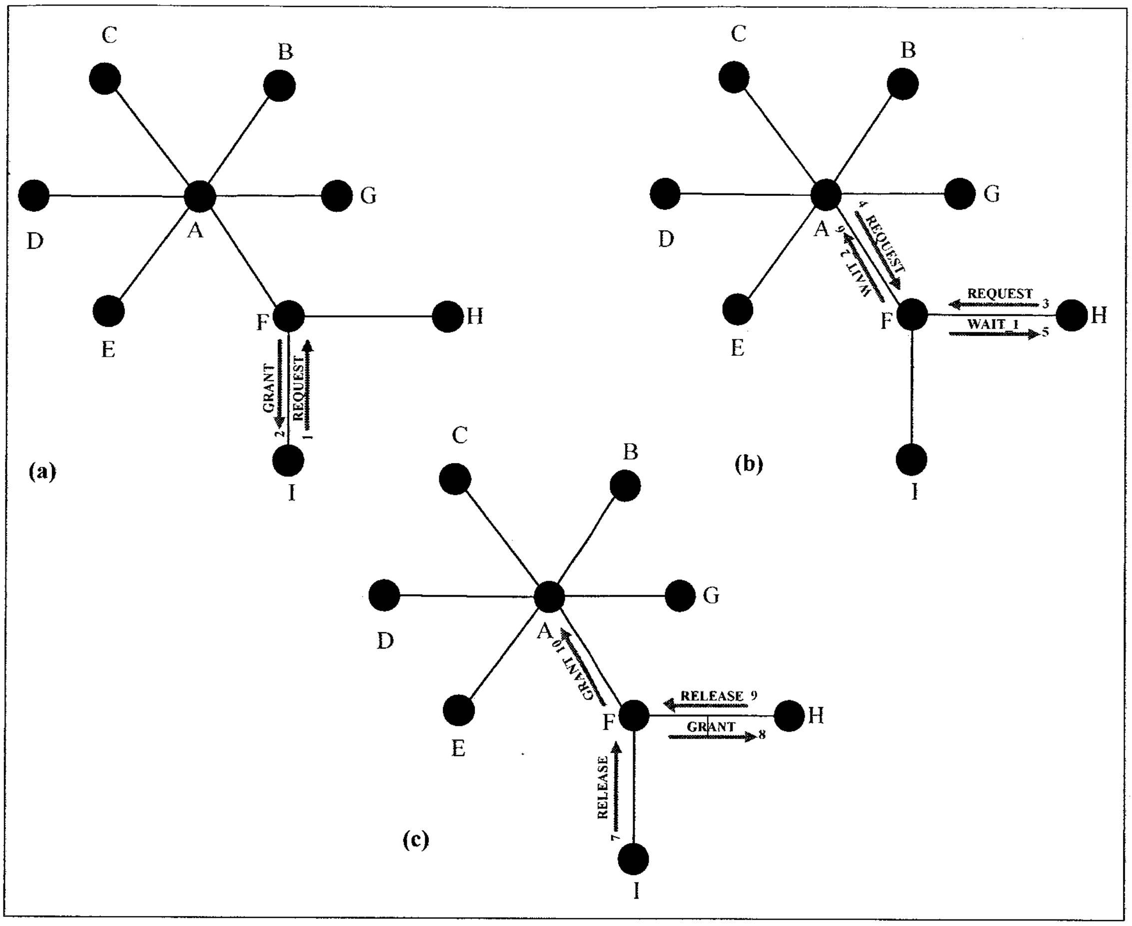


Figure 3.5 Slot assignments procedure in DSSA

from node I, it will send a GRANT message to the nodes among the priority of its QUEUE. As shown in Figure 3.5(c), first node F will send a GRANT message to node H and then to node A.

3.6.1.2 Update policy

The update policy is executed in parallel along with the scheduling policy to minimize the additional computation. In the update policy, each node maintains a record of S1 (where S1 represents the slot reserved in H1) through the RELEASE message. The RELEASE message has information of S1. In order to reserve a conflict-free time slot, each node must have information of S1 and S2 (where S2 represents the slot reserved by H2) where S2 is achieved through the GRANT message. Node I, after getting heads and winning the lottery, broadcasts a REQUEST message to get the update about S2(I). In response, node $H1$(I) will reply with a

GRANT message which has the information of $S2(\mathrm{I})$. Node I will go through S1(I) and S2(I) and finally reserve the minimum unassigned slots in N2(I); if at the same time, node H and A come on with a REQUEST message, node F will send a WAIT message to both of them and record their REQUEST in the WAIT queue. Each wait message is assigned a WAIT message and as soon as node F gets a RELEASE message from node I, it will send a GRANT message to node H and A based on the priority of the WAIT message.

3.7 DETAILS OF AN IMPROVED DISTRIBUTIVE SCHEDULING ALGORITHM (IDSA)

In an improved distributive scheduling algorithm (IDSA), sensor nodes do not collaborate with their $H1$ and $H2$ for scheduling their schedule; each node instead maintains and generates its own transmission schedule based on its own record. The scheduling in the IDSA is trivial because each node maintains a record of its neighbouring node schedule in the H1 and $H2$ queue. Finally, when a node schedules its own slot it has to look up to its own record rather than requesting its neighbouring nodes and reserves or proposes the slot which is not reserved by any of the nodes in the H1 and $H2$ queue. Scheduling through the IDSA is based on local information because none of the nodes is aware of the entire network transmission schedule or size.

Scheduling is IDSA is also carried out on the basis of the existing topology. After the scheduling, the neighbouring nodes are updated about the schedule. Whenever a network suffers topology in a network the schedule is regenerated in the affected part of the network. Schedules for new nodes are carried out on the basis of local information in such a way that it should not collide with any of the nodes within its IR. This results in evolving scheduling with the changing topology over time. A description of the IDSA is illustrated in next section.

3.7.1 Algorithm specification

The idea behind the IDSA is to overcome the drawbacks of DRAND. The IDSA allows each node to decide its own schedule according to the local information, rather than each time transmitting a REQUEST message to its neighbouring nodes. This local information includes the 1-hop and 2-hop neighbours' IDs, the slots reserved by 1-hop and 2-hop neighbouring nodes, the distance between them, and whether the neighbouring nodes are scheduled or not. The IDSA algorithm runs in rounds and is composed of the following procedures:

- Slot scheduling procedure
- Update procedure

3.7.1.1 Slot scheduling procedure

According to the proposed algorithm, at the beginning, all of the nodes are in the IDLE state. Each node contends to reserve a conflict-free slot for itself. Thus, each node has to toss the coin and then go through the lottery process as mentioned in DSSA. If a node wins the lottery, it will move to the PROPOSE state. In the PROPOSE state, the node schedules the minimum numbered unassigned slots for itself based on its $H1(u)$ and $H2(u)$ neighbouring record. Here, in the IDSA, the node looks up its own record rather than each time sending a REQUEST message to all of its 1-hop neighbouring nodes as in DRAND. In DRAND the REQUEST, GRANT, RELEASE, REJECT and FAIL messages incur high message overhead, which increases energy consumption, latency and less probability of reserving a particular slot. The drawbacks in DRAND have motivated this research. Finally, after scheduling a slot, the neighbouring nodes in the IDSA are informed through the Update procedure.

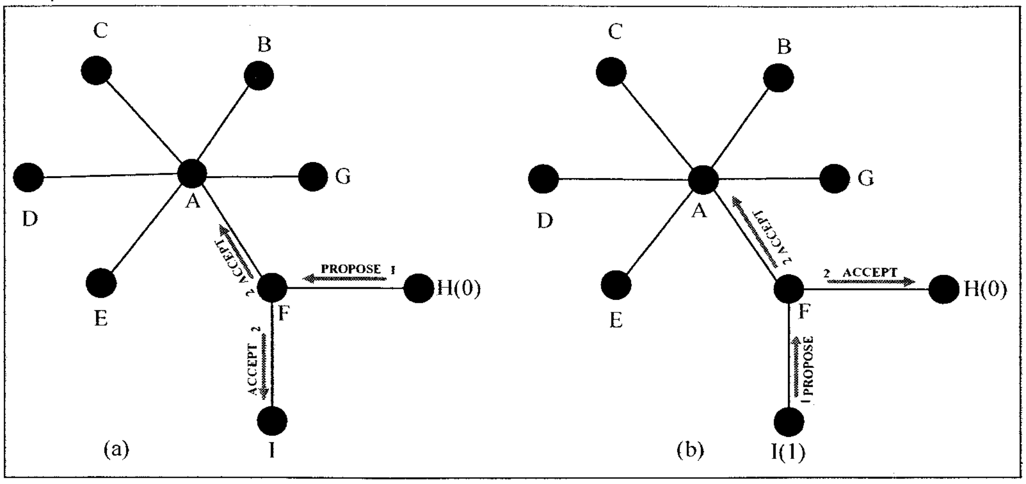


Figure 3.6 IDSA scheme

3.7.1.2 Update procedure

As shown in Figure 3.6(a), node H contends to reserve a slot, so it attempts to reserve slot (0), i.e., $S(H) = \{0\}$ (where S is the slot). Once node H gets the schedule of its own slot, it broadcasts the "PROPOSE" message. On receiving the "PROPOSE" message, each node $H1(H)$ will update the 1-hop record, which ensures that the node is in $H1(F) = \{H\}$ and has the slot scheduled $S1(F) = (0)$. In the next step, node F will broadcast the "ACCEPT" message. Nodes which have received the "ACCEPT" message, update the $H2(u)$ record which ensures that the node in their 2-hop neighbour $H2(A) = \{H\}$ and $H2(I) = \{H\}$ have the scheduled $S2(0)$. So, when any node in a network contends to reserve a slot for itself, it only looks up its own $H1(u), S1(u)$,$H2(u), S2(u)$ record. As shown in Figure 3.6(b), node I contends to reserve a slot. It checks its $H1(I)$, $S1(I)$,$H2(I)$, $S2(I)$ records and reserves the minimum unassigned slots for itself which are not reserved within its own record , which is S(I) = {1}. After reserving the slot, "PROPOSE" messages and "ACCEPT" messages are broadcast in the same way as described above.

3.8 LOCAL FRAMING

After scheduling a time slot, the node has to decide on the period for its transmission. This transmission period is known as the time frame (TF). To avoid conflict among the nodes, all of the nodes, at the same time, are conventionally synchronised to slot 0. In the proposed algorithms, new nodes are assigned time slots based on local information while maintaining the existing schedule. (Note that if the whole network has to be rescheduled while scheduling a small portion of the network, it will incur high overhead which may lead to many other drawbacks). Here, a new technique has been presented that fits the local time frame based on contending. The main idea is explained in detail in the following section.

3.9 TIME FRAME RULE (TF RULE)

After scheduling a slot, each node is familiar with its own time in which it has to start the transmission. The period in which a node starts its transmission is known as the time frame (TF). Conventionally, the TF is kept constant depending on the maximum slot schedule within $N2(u)$ in a network. Let L_u be the length of TF and $L_u = 2^i$, where "*i*" is an integer satisfying the condition:

$$2^{i-1} \leq L_u \leq 2^i - 1 \qquad 3.7$$

In order to make full use of the TF each node "*i*" guarantees the reusability of slot S_i in L_u, i.e., $l * L_i + S_i$ where, *l=1, 2, 3……n.* Let node *u* schedule slot S_u depending on the information within its $N2(u)$.

Theorem 3.1: If any node u schedules and uses a time slot $l * 2^i + S_i$ for its transmission, then none of the nodes from $H2(u)$ can schedule the same slot for itself.

Proof: The theorem can be proven by an example: Let node u be in $H2(v)$. According to the TF rule, node *v* cannot assign the slot that has been already reserved by node u; therefore, node *v* schedules S_v within the time frame of 2^v. Then without loss of generality, it is assumed that $2^i \leq 2^j$, thus $S_v < S_u$. Similarly, by the proposed algorithms $S_v \neq S_u$. Thus, both of the nodes, *u* and *v*, assign one slot for their transmission in the TF of 2^i and 2^j. Note that the TF of node u 2^i and the TF of node v 2^j are different from each other, so it is clear that node u and v will be assigned different time slots for their transmission to avoid conflict and collision.

3.10 SLOT ASSIGHMENT BY PROPOSED ALGORITHMS

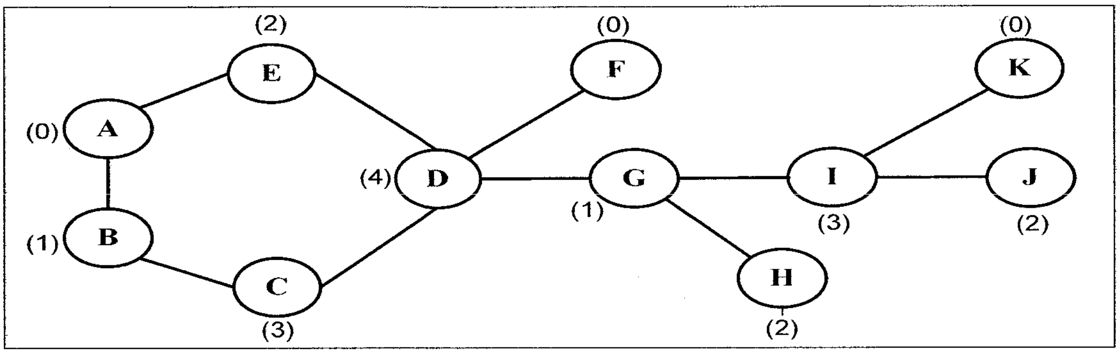


Figure 3.7 Topology description of scheduling in IDSA and DSSA

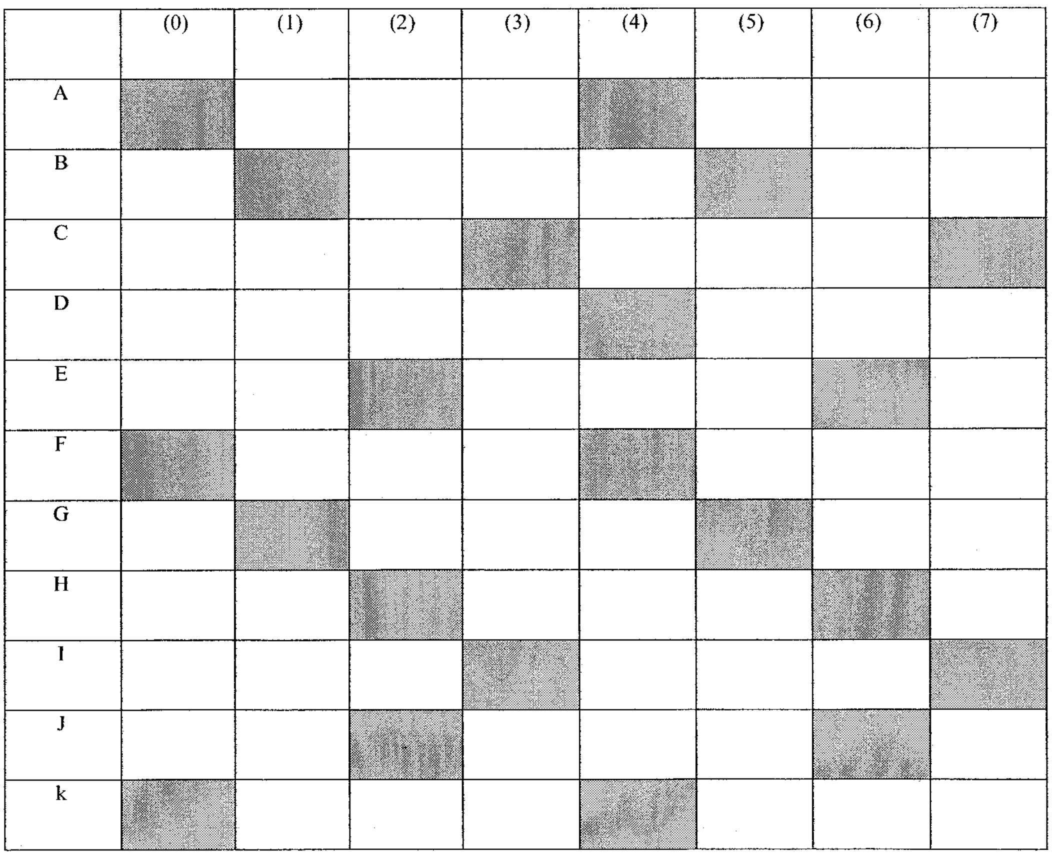


Figure 3.8 Slot assignments through the IDSA and DSSA algorithms

According to the TF rule, each node reserves its own slot within a frame based on its, $H2$ neighbouring nodes. The TF rule allows the propose algorithms to adapt to the changes in topology by resizing the time frame based on the $H2$ neighbouring nodes. Figure 3.7, shows the slot assignment achieved for topology in Figure 3.8 through the TF rule and it can be seen that none of the nodes have assigned the same slot as that of the nodes in $N2$. Here, the TF is based on local information rather than global information; if global information was used, then the frame size would be 5 and each node could use its slot only once in the whole frame. The adaptation of TF rule have been able to breakdown the whole frame into small portions and the nodes with less numbers of $H2$ can reuse their slots. This reuse of slots in a frame reduces the delay and increases the concurrency in the channel usage.

3.11 SUMMARY

This chapter has presented a set of novel heuristic scheduling algorithms. These algorithms are topology independent and provide a contention-free slot reservation. The algorithms have been designed to frequently handle topology changes and update the schedule based on local information to minimize the network degradation parameters as much as possible such as message overhead, number of rounds, execution time and energy consumed through the scheduling process. The performance of a WSN is affected by conflicting nodes (two-hop) thus, both of the proposed algorithms, DSSA and IDSA, reserve conflict-free slots based on local information in a distributive manner. The DSSA and IDSA are insensitive to the network topology and sizes; therefore, both are scalable, fully distributed, self stabilizing, locally configurable and can be used for large dynamic WSNs.

CHAPTER 4

RESULTS AND DICSUSSION

The methodology and design of the proposed algorithms introduced in the previous chapter, is more performance optimal than the existing technique. This chapter presents a more comprehensive qualitative evaluation of the existing technique through actual results of the matrices. For better performance evaluation the proposed algorithms and the existing one were simulated in the network simulator (NS-2). Furthermore, the subsequent comparison metrics followed by the results give a clear image of the relative performance of the algorithms obtained through the simulation based on the parameters introduced. Finally, the chapter ends by concluding the performance metrics of DRAND and the proposed algorithms DSSA and IDSA based on which this research has been carried out.

4.1 PERFORMANCE EVALUTION

In this section, the simulation scenario is presented that has been introduced for the performance evaluation. The simulation scenario enabled the prediction of the significant perspective concerning the validation of the algorithms. Furthermore, the simulation environment enabled the improvement of the credibility and accuracy of the algorithms assuring the repeatability and verification according to the real world scenario. Hence, the simulated result verification and validation allow the foundation for the practical use of the algorithms. Firstly, the performance of the proposed DSSA scheduling algorithm was compared to DRAND in terms of all of the important parameters at the MAC layer. Then, the second proposed algorithm, IDSA, was compared to the DRAND technique. Finally, all of the three techniques, DSSA, IDSA and DRAND, were compared.

4.1.1 Simulation Tool

There are many simulators available which support wireless sensors networks such as NS-2 [74], EmStar [75], OMNet++ [76], OPNET [77] and so many others. For this present work, NS-2 is used because of its extensive use and specialised feature for WSNs. NS-2 has different simulation environments based on C++. It provides a rich environment for simulation of WSNs at different layers especially on the MAC layer.

4.1.2 Network simulator (NS-2)

NS-2 is an open-source event-driven simulator. It has gained tremendous attention specifically for research in communication networks in the academic, industrial, and government sectors. In order to explore network performance, researchers can simply modify the tool command language (TCL) script to configure a network, and can easily observe the performance of the network through the results generated by NS-2. NS-2 for the last few decades, due to its widely supportive nature, has become the preferable network simulator among the researchers (See Appendix C).

4.1.3 Simulation Setup

The provision of the simulation development environment has to support both the modeling of the communication networks and distributed systems. The analysis towards the behavior and performance of the simulation is feasible by performing discrete event simulations and comparing them to the existing techniques. The simulation environment could include model design, simulation, data collection and data analysis (See Appendix C).

4.1.4 DSSA

In this subsection, the performance of DSSA has been analysed to verify the main objective, which is to achieve the optimal and efficient distributive scheduling.

Furthermore, for the comparative study and analyses, the DSSA technique was compared with the standard and current technique known as DRAND in terms of the number of rounds, message overhead, execution time, and energy consumption through the scheduling process.

From the DSSA proposed methodology and the results in the following sub-sections, it has been evidenced that DSSA outperformed DRAND. DSSA provided a distributive scheduling technique as well as obviated the weaknesses of traditional algorithms. In proportion to the slot assignment opportunities, the DSSA also focused on trimming down the network degradation parameters as much as possible. DSSA minimised the chances of unsuccessful cycles because each unsuccessful cycle would result in additional computation. In DSSA, each node decided its own time slot based on the local information gathered by its N2 neighbouring nodes. The DSSA algorithm ran in rounds where each node reserved a conflict-free slot in a heuristic manner. In the DSSA simulation scenario, at the beginning, all of the nodes were in the IDLE state and each node competed to schedule a conflict-free slot. In the DSSA approach, if a node successfully passed the coin toss and lottery phase, then the whole cycle was successful. The nodes that passed through the coin toss and lottery phase got either a GRANT message immediately after the time of the REQUEST or was granted a WAIT message to ensure that the GRANT message would be given on the basis of priority and wait. Finally, the number of rounds, messages, execution time and energy dissipated were incremented after each round.

4.1.5 Number of rounds

Number of rounds refers to the cycles that a node utilizes to acquire its slot. The DSSA and DRAND both run in rounds and each node decides on its time slot during the rounds. Each node up to two-hop neighbour assigns a unique time slot for itself to carry out further operations within its own slot only. From Figure 4.2, it can be observed that DSSA technique results in lesser number of rounds to achieve scheduling task as compared to DRAND. This advantage comes from the adaptation

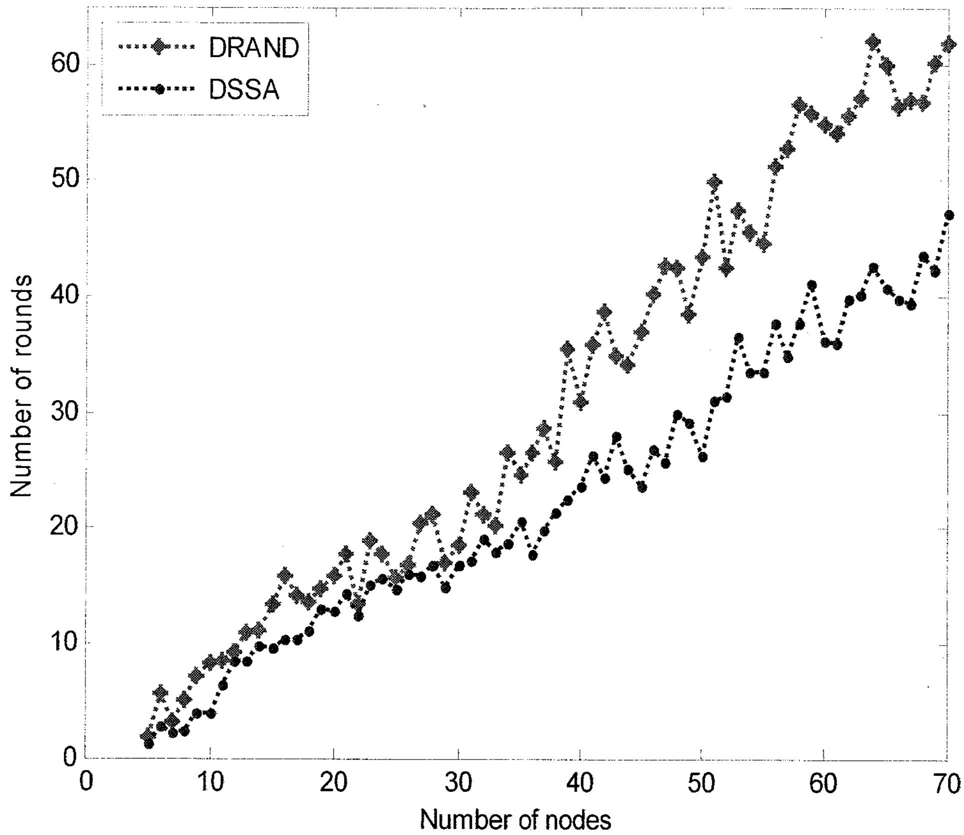


Figure 4.2 Number of rounds to acquire a time slot

of an improved scheduling technique that results in less number of rounds to accomplish the scheduling task.

4.1.6 Control packet overhead

The total number of scheduling control packets that are exchanged during slot reservation are known as control overhead. In initialization or the setup phase, excessive numbers of control messages are exchanged by handshaking and scheduling among the neighbors. These control messages also consume network resources and most of the energy is consumed through unnecessary messages transmission. In order to conserve energy, the scheduling algorithm should have low communication overhead. Figure 4.3 show the average number of messages exchanged in the DSSA and DRAND techniques during the scheduling process. The proposed technique temporary stores the information of the node that has REQUEST

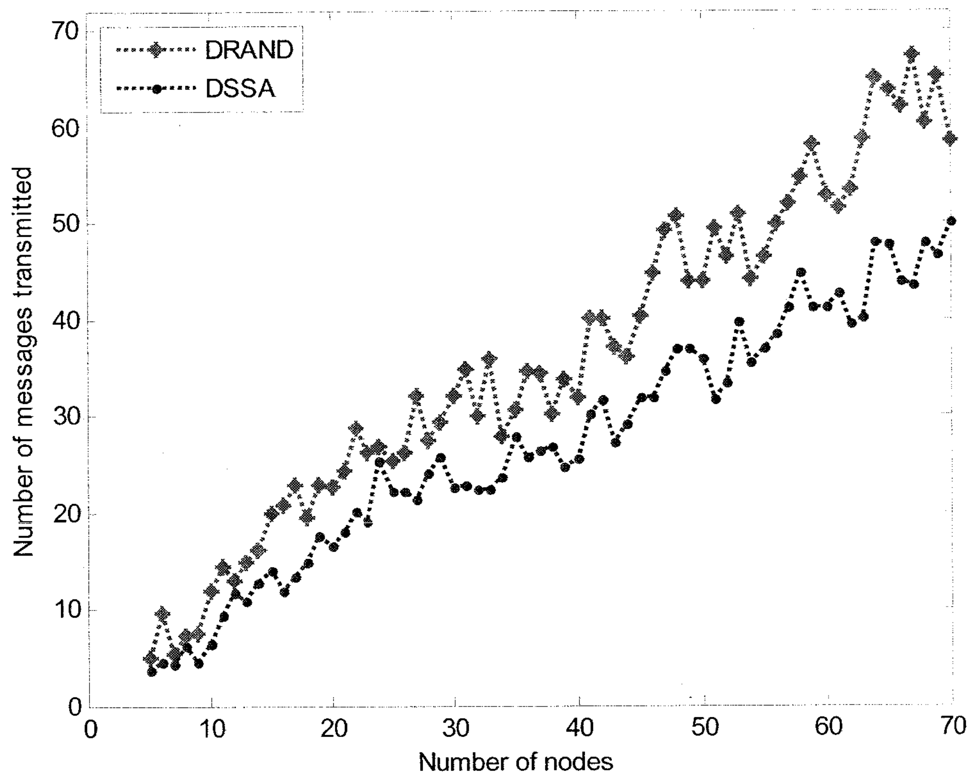


Figure 4.3 Messages exchanged during slot reservation

for the GRANT to a node that's state was not IDLE or RELEASE. Latter on GRANT message is sent to the node base on priority. It has been found that for higher network sizes, the average number of message-transmissions of DSSA is much lower than DRAND. This is achieved by avoiding unnecessary cycle, which results in retransmissions of extra control message. After each unsuccessful cycle, each node has to go through a number of primitive states and message exchange, which results in a higher communication message overhead.

4.1.7 Energy consumption

Collision, idle listening, overhead, and overhearing are the major source of energy wastage and has direct impact on network life time. Network energy resources are consumed during set-up and transmission phase. The energy consumption during transmission phase can be reduced by avoiding collision and idle listening among all

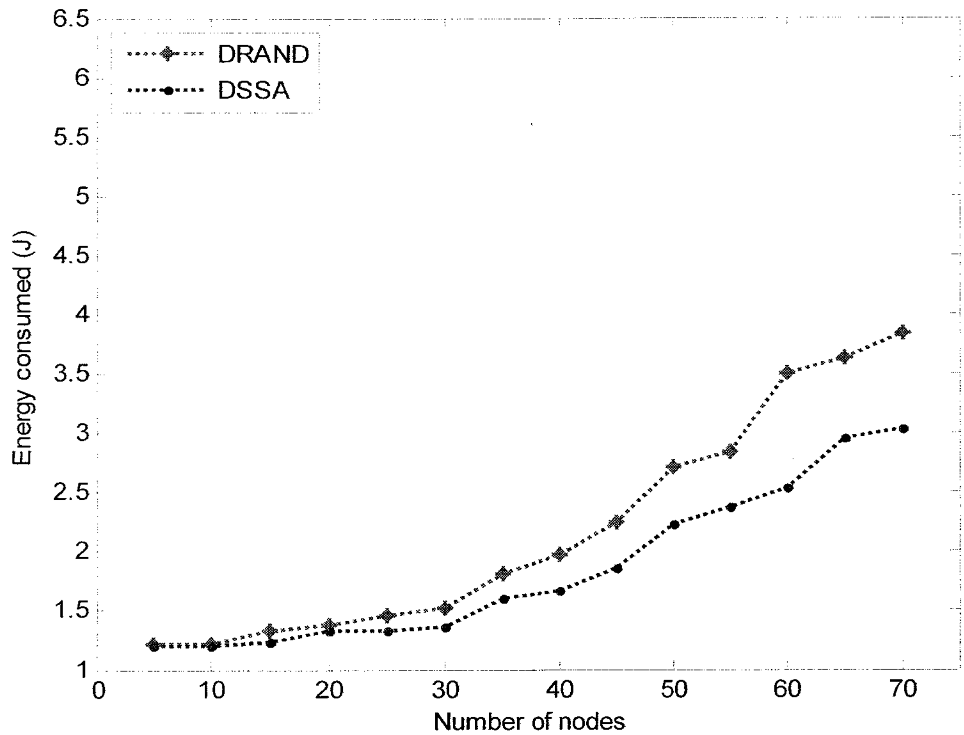


Figure 4.4 Average energy consumption per node during slot assignment

the conflicting nodes which can be achieved during set-up phase. However, the ultimate goal of the proposed technique is to assign conflict-free slots among all the neighboring nodes while consuming minimum energy. In WSNs, most of the energy is consumed by radio transceivers rather than calculation or code execution as both the techniques, DSSA and DRAND, run in rounds and have to pass through many primitive operations i.e. (idle listening, receiving and transmitting). From Figure 4.4, it has been found that DSSA has much lower energy consumption as compared to DRAND due to lower communication overhead by avoiding unsuccessful cycles. After every unsuccessful cycle each node repeatedly exchanges control messages with its neighbours to reserve a slot which results in extra energy consumption.

4.1.8 Run time

Run time refers to the time that a node requires to schedule a collision-free time slot. Figure 4.5 shows the average run time utilized for acquiring a time slot with an

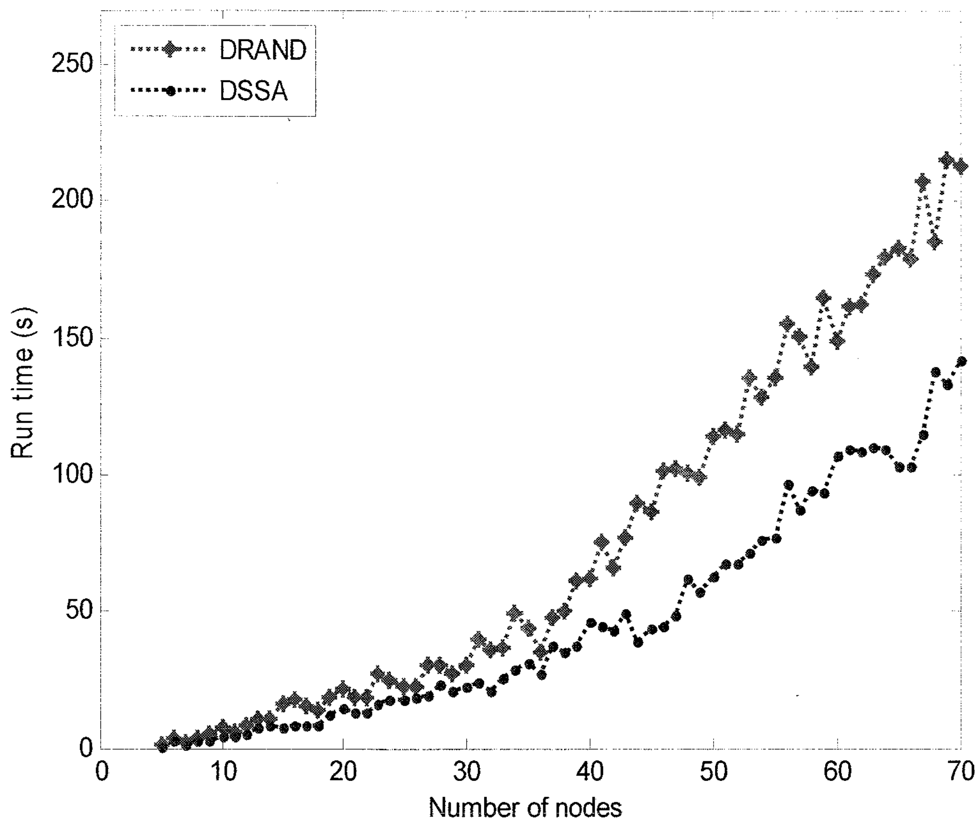


Figure 4.5 Run time during slot assignment

increasing number of nodes. It can be observed that the scheduling duration of both DSSA and DRAND are nearly the same up to a 20 nodes in a network. As the network becomes much denser, the scheduling process becomes more complicated and the run time of DSSA outperforms DRAND. Thus, this illustrates that DSSA avoids unsuccessful cycles because after each unsuccessful cycle extra time is required to carry out scheduling task.

4.2 IDSA

Although, the design of the ZMAC has introduced DRAND scheduling scheme, but still there are some drawbacks in the DRAND scheduling technique. To overcome the drawbacks of DRAND, Self distributed scheduling MAC algorithm has been introduced, it is a new and independent scheduling algorithm where each node is capable of scheduling its own slot. The IDSA is much more improved technique than DRAND and DSSA. According to IDSA, at the beginning, all nodes are in the IDLE state. Each node contends to reserve a conflict free slot for itself. Thus, each node has to toss the coin and then go through lottery process. If a node wins the lottery, it will move to the PROPOSE state. In the PROPOSE state, the node schedules the minimum numbered unassigned slot for itself based on its $H1(u)$ and $H2(u)$ neighboring record. Here, in the IDSA, the node looks up its own record rather than each time sending a REQUEST message to all its 1-hop neighboring nodes as in DRAND. IDSA, not only provides better performance than traditional transmission scheduling algorithms designed for general workloads, but also has the following prominent features: The IDSA can easily adapt its transmission schedule in response to the topology changes (addition/removal of nodes) without rescheduling a whole network transmission schedule with minimum message overhead. The IDSA transmissions are executed dynamically by a node in each schedule slot, as a result, it may adjust to workload more effectively and efficiently than traditional TDMA MAC protocol algorithms. The IDSA has low runtime, message overhead, energy consumption and limited memory requirements making it suitable for resource constrained devices.

4.2.1 Number of rounds

Figure 4.6 shows the number of rounds required for reservation of slots for different topology size. In the simulation, we found that the DRAND scheduling algorithm results in more numbers of rounds as compared to the IDSA scheme. It all comes at the cost of unsuccessful cycles because in DRAND unsuccessful cycle is due to following events: 1) If none of the unscheduled nodes gets head during coin toss. 2) If none of the unscheduled node wins a lottery. 3) If REJECT message is send by any of the 1-hop neighbouring nodes. While in the IDSA an unsuccessful cycle occurs only when none of the unscheduled nodes gets head or does not wins the lottery during cycle. This advantage comes from the adaptation of an improved scheduling technique, which removes extra control messages and number of rounds. Each node in the IDSA has to look up its own record and reserve slot for itself; furthermore, there is no need of exchanging REQUEST, GRANT, RELEASE, FAIL and REJECT messages. This results in less rounds.

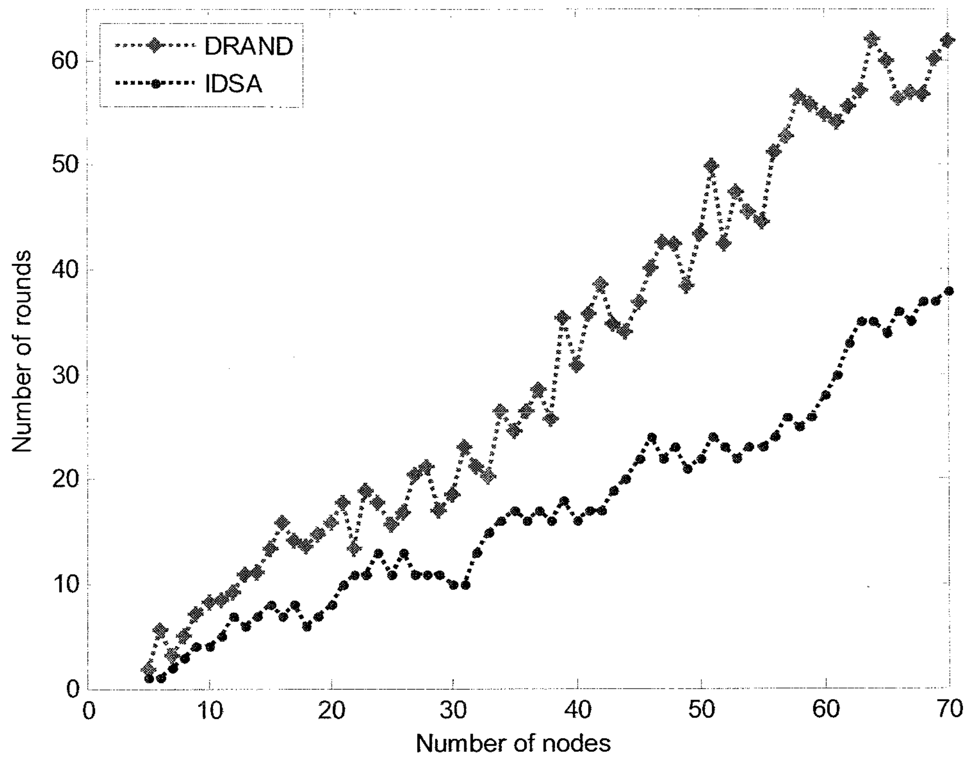


Figure 4.6 Number of rounds to acquire a slot

4.2.2 Messages overhead

Control packets are exchanged to establish a channel and reserve a conflict-free slot. Control packets consume resources such as energy, bandwidth and time. Since control packets are only used for network management, control packets are considered overhead. Figure 4.7 shows the comparative message overhead costs in the network while reserving a slot. It is found that DRAND has much greater message overhead than the IDSA. Message overhead in DRAND is greater because of the unsuccessful cycle due to which nodes extensively exchange control messages. After each unsuccessful cycle, nodes have to go through a number of primitive states which results in excessive amount of message overhead.

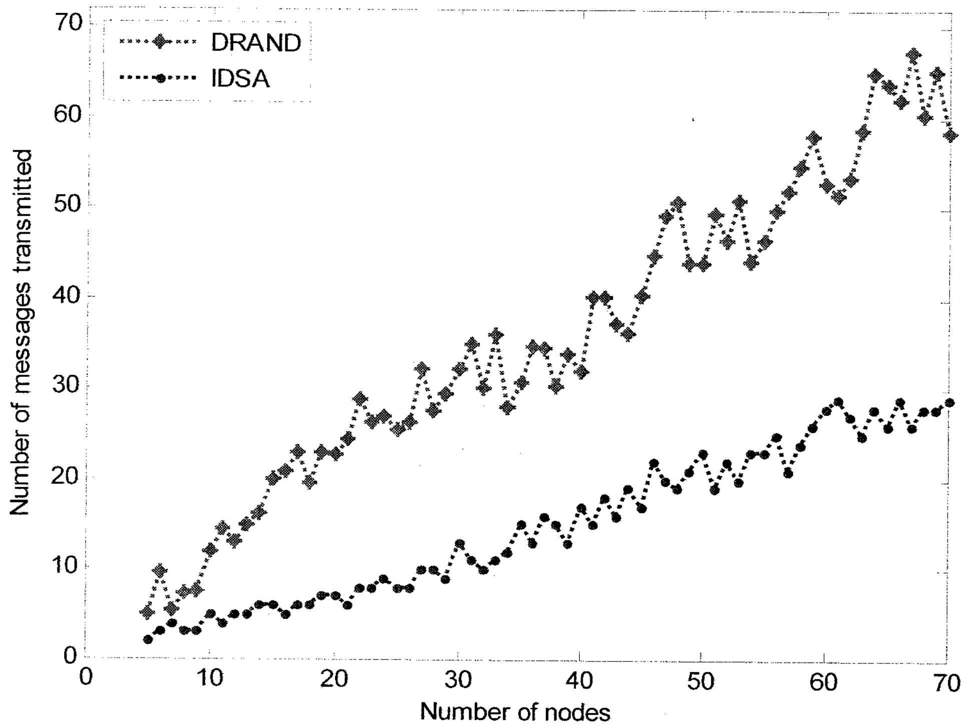


Figure 4.7 Messages exchanged during slot reservation

4.2.3 Run time

Run time refers to the time that a node requires to schedule a collision free time slot. Figure 4.8 shows the average run time utilized for acquiring a time slot with an increasing number of nodes. It can be observed that the scheduling duration of both DSSA and DRAND are nearly the same up to a 20 neighborhood size. As the network becomes much denser, the scheduling process becomes more complicated and the run time of DSSA outperforms DRAND. Thus, this illustrates that DSSA avoids unsuccessful cycles because after each unsuccessful cycle extra time is required to carry out scheduling task.

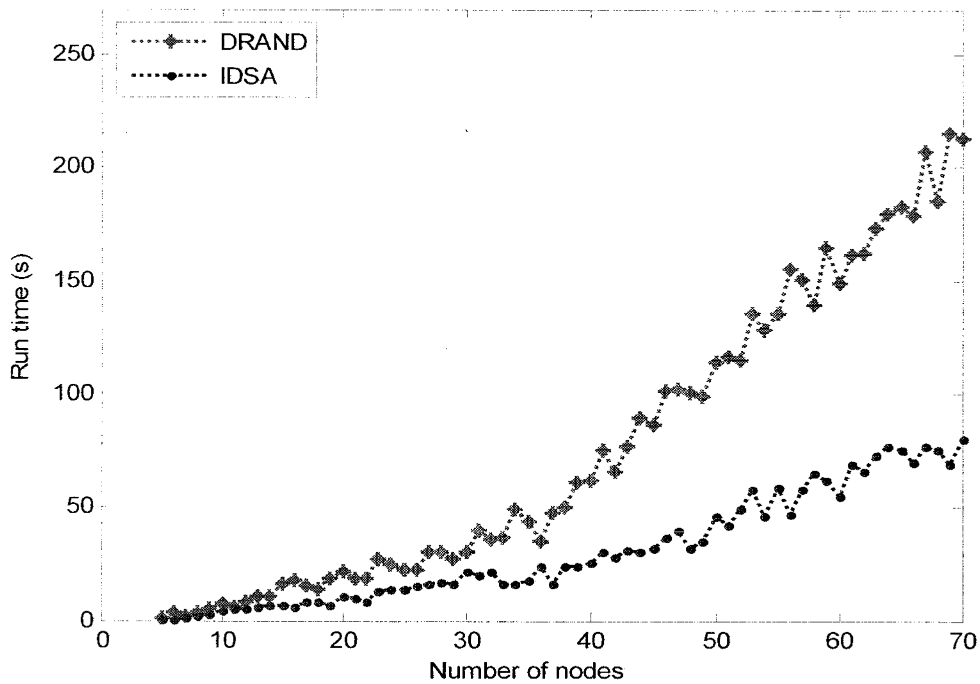


Figure 4.8 Execution time during scheduling process

4.2.4 Energy consumption

This section shows energy consumed by nodes during slot reservation. As DRAND and the IDSA both run in rounds, thus each node within the network has to record all the primitive operations (idle listening, receive a byte and transmit a byte, and sleep mode). Hence, the energy consumed by all the nodes in a network will be the sum all the operations executed in each state. Figure 4.9 shows the relation of energy consumed by nodes during slot scheduling with the increasing amount of neighborhood size. It is found that the IDSA energy consumption is much less than DRAND; this is due to avoiding unnecessary message overhead in the IDSA, which consumes extra energy. As previously noted, it is shown that DRAND has more overhead. Thus, each node in DRAND has to pass through and record more primitive operations. Thus, it is evident that energy consumption in DRAND will be much more than in the IDSA.

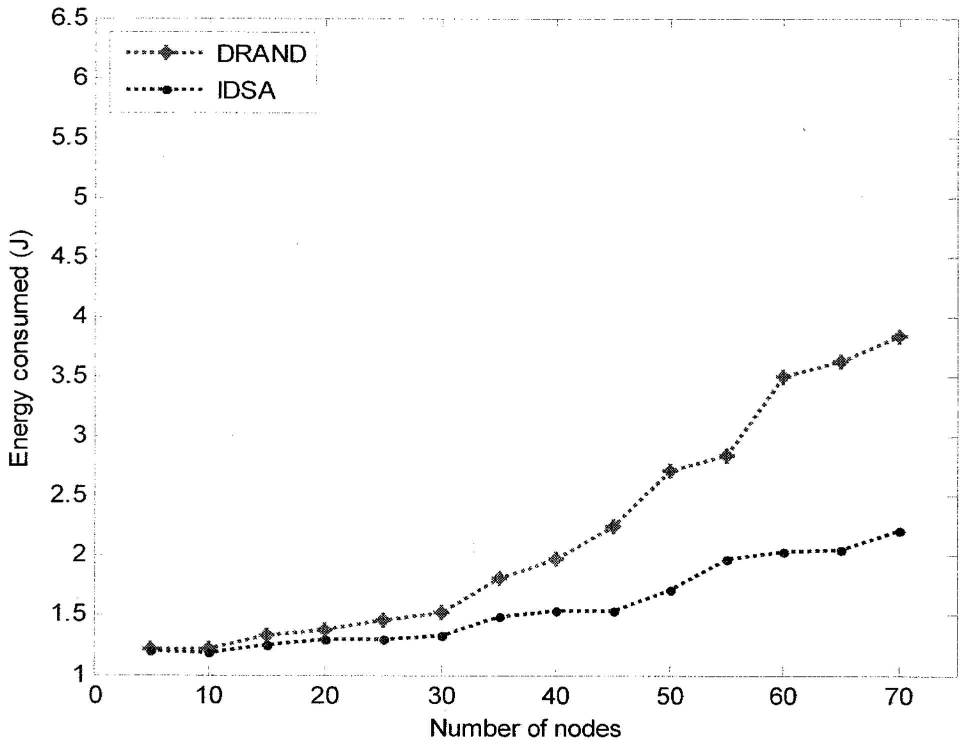
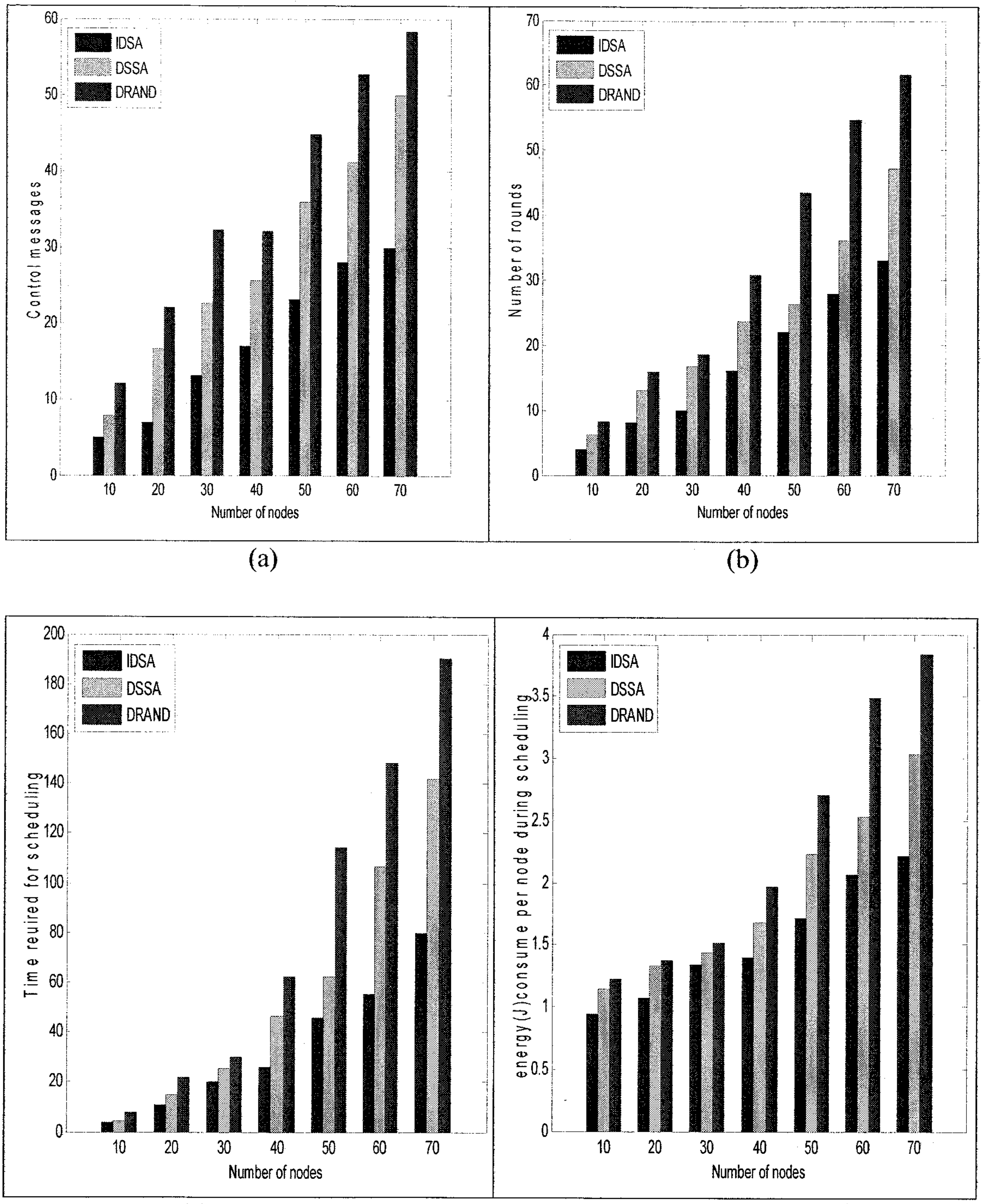


Figure 4.9 Energy consumed during the scheduling process

4.3 COMPARATIVE ANALYSIS OF PROPOSE AND CURRENT ALGORITHM

The results reported in Figure 4.10 were obtained after 15 repetitions of trials. From Figure 4.10(a), it can be observed that DSSA and IDSA both results in lesser number of rounds to achieve scheduling task as compared to DRAND. This advantage comes from the adaptation of an improved scheduling technique that results in less number of rounds to accomplish the scheduling task. Figure 4.10(b) show the average number of messages exchanged in the DSSA and IDSA as compared to DRAND techniques during the scheduling process. The DSSA technique temporary stores the information of the node that has REQUEST for the GRANT to a node that's state was not IDLE or RELEASE. Latter on GRANT message is sent to the node base on priority. While in case of IDSA there is no concept of REQUEST, GRANT, RELEASE, FAIL, and REJECT messages. In IDSA each node has to just exchange only two message the PROPOSE and ACCEPT message for their scheduling.



(a) (b) (c) (d)

Figure 4.10: control messages, number of rounds, run time, and energy of DSSA and IDSA as compared to DRAND algorithm.

Figure 4.10 (c) shows the average run time utilized for acquiring a time slot with an increasing number of nodes. It can be observed that as the network becomes much denser, the scheduling process becomes more complicated and the run time of DSSA and IDSA outperforms DRAND. Thus, this illustrates that DSSA and IDSA avoids unsuccessful cycles and also trim down the number of messages those requires more time to accomplish scheduling task. Finally, Figure 4.10(d) shows the relation of energy consumed in DSSA and IDSA technique as compared to DRAND during slot scheduling. It is found that the DSSA and IDSA energy consumption is much less than DRAND; this is due to avoiding unnecessary message overhead, which consumes extra energy.

It is clear from the cumulative results of three techniques from Figure 4.10 that both the proposed algorithms outperform DRAND. Moreover, Table 4.2 gives reflection of percentage improvement of the propose algorithms over DRAND with variable network size. The simulation was carried out in NS-2 for multi-hop scenario and it can be clearly visualized that proposed techniques performs distributive scheduling with less number of control messages, number of rounds, run time, and energy as compared to DRAND algorithm.

Table 4.2 Percentage improvement of DSSA and IDSA over DRAND

No. of nodes / % improvement	**30** **DSSA**	**30** **IDSA**	**45** **DSSA**	**45** **IDSA**	**60** **DSSA**	**60** **IDSA**
Rounds	8%	12%	19%	24%	25%	34%
Message overhead	11%	16%	14%	19%	17%	22%
Run time	15%	21%	18%	26%	21%	29%
Energy consumption	3%	6%	7%	8%	9%	11%

4.4 DSSA CONTRIBUTIONS

This subsection summarized the performance of distributive and self-sustainable scheduling algorithm (DSSA). For comparative analysis the simulation results from DSSA technique were compared against the previously in practice and established works of DRAND. DRAND technique give rise to huge amount of control overhead, run time, number of rounds, and energy consumption due to unsuccessful rounds and exchange of many states. In contrary, DSSA reduces the chances of unsuccessful rounds and also reduces the number of states. The proposed technique effectively utilizes the network resources by avoiding unsuccessful cycles and provides conflicting free schedule among all the nodes up to two hop neighbors.

4.5 IDSA CONTRIBUTIONS

In this subsection a new and improved distributive scheduling algorithm IDSA is introduced. IDSA is a scheduling scheme for WSNs and it allows the nodes in a network to schedule their slot based on local information rather than each time dependent on neighboring information as in DRAND. IDSA is self distributive scheduling algorithm and novel heuristic scheduling technique that can provide effective collision free broadcasting, lower energy consumption, minimum message overhead and enhanced channel utilization. In contrast to earlier traditional scheduling algorithms of medium access control (MAC), which are generally designed for sequential slot assignments, this thesis presents an improved algorithm for distributed scheduling. The IDSA has several unique features. First, it optimizes energy through collision free transmission by scheduling conflict-free slots. Second, it can adapt the changes in topology explicitly without reconstructing the global transmission schedule with minimum message overhead. Furthermore, the IDSA also provides improved performance in terms of message overhead, slot assignment per round and energy consumption. Simulation results show that the IDSA significantly outperforms a representative distributed random slot assignment algorithm (DRAND).

4.6 SUMMARY

This chapter presents the performance results of two proposed scheduling techniques compared with DRAND. To analyze the reliability of proposed techniques DSSA and IDSA were simulated under different network scenarios. The first technique enhances DRAND by minimizing the chances of unsuccessful cycle and reduces the number of states as well as number of scheduling parameters. While the second technique IDSA is independent of REQUEST, GRANT, and RELEASE message, but still it enable the nodes in a network to schedule the conflict-free slot based on their restored information. This chapter not only provides the comparison results but also show the percentage improvement of the techniques. From the simulation results and comparative matrices it is proved that DSSA and IDSA outperform the existing algorithm in all the comparative parameters.

CHAPTER 5

CONCLUSION

5.1 CONCLUSION REMARKS

Energy efficient, scalable, and dynamic topology independent wireless sensor networks (WSNs) pose great challenges for the design of dynamic bandwidth allocation to maximize the spatial reuse of time slot with minimum frame length. Most of the existing scheduling techniques are either centralized or topology dependent. Therefore, it is complex and inefficient approach to manage several nodes by only one centralized controller with limited memory and battery power. Moreover, these techniques cannot efficiently adapt to the dynamic wireless environment. The scheduling algorithm is a fundamental design problem to allocate resources among different entities in distributive WSNs. Thus, to design a large and scalable network, the scheduling algorithms should be computationally distributed and simple. In this research, the main focus is on investigating various perspectives of the MAC scheduling algorithm and have presented two distributed and topology independent MAC scheduling algorithms.

The major contribution of this thesis is improved distributive and self-sustainable MAC scheduling algorithms to resolve challenging issues related to scheduling. To reduce the complexity and variety of scheduling problems, this research has proposed two distributive scheduling algorithms DSSA and IDSA. The DSSA and IDSA both does not require any synchronization and can effectively adapts dynamic topology changes without incurring global communication overhead. The DSSA technique enhances DRAND through minimizing the chances of unsuccessful cycle. The entire network scheduling through the DSSA is simple and local because every node collaborates up to its two-hop neighbouring nodes, only.

While according to IDSA algorithm, sensor nodes do not collaborate with their H1 and H2 for scheduling their schedule; instead each node map a time slot of its one and two hop neighboring nodes in its queue and generates its own transmission schedule based on its own record. The scheduling in the IDSA is quite simple because each node maintains a record of its neighbouring node schedule in the H1 and H2 queue. Consequently, each node successfully schedules a unique time slot for itself in a heuristic manner based on its local information. Both the proposed algorithms, guarantees conflict-free scheduling because all the conflicting nodes are assigned different time slot for their transmission.

It is found that DSSA and IDSA not only outperform the existing DRAND technique but also obviates its weakness by avoiding unsuccessful cycles. Moreover, it can be easily illustrated through simulation results and performance metrics that DSSA and IDSA achieves better performance than DRAND in terms of number of rounds, message complexity, run time and, energy consumption. In addition, both are distributive technique and are robust against any dynamic change without incurring extra message overhead. Furthermore, these algorithms utilize minimal resources to provide optimal collision free scheduling by reducing all the network degradation parameters. Thus, both of the proposed algorithms were built upon the following principles and design decisions, which have been achieved:

- Adapting dynamically to topological changes.
- Assigning collision free schedules among all of the conflicting nodes.
- Reducing communication overhead by reducing unsuccessful cycles.
- Computing that is simple and optimal in term of following parameters such as:
    - Number of rounds,
    - Communication overhead,
    - Run time, and
    - Energy consumption.

5.2 FUTURE DIRECTION

The IDSA and DSSA both scheduling algorithms were designed and implemented to introduce new distributive scheduling TDMA techniques in distributive and self-organizing WSNs. One of the weak aspects of traditional TDMA techniques is that they could not efficiently react to the network changes. Therefore, that results in more message overhead and more chances of packet losses at routing time. However, this problem was taken in account and up to large extend the problem was resolved by the introduction of one and two way neighboring node discovery protocol. But in a very lossy link connectivity and frequently changing network there could be unexpected heavy chances of packet losses that results in delay and moreover for high sensitive operation like military and health care the delay or packet losses could not be accepted. For highly frequent changing network over lossy link when the data loss is more than any predefined threshold IDSA and DSSA both deal with this problem and the best solution is to periodically re-run both the algorithm. Instead of re-running the better option should be to introduce neighbouring discovery algorithm that can tackle these changes in a more efficient way than the one and two way neighboring protocol.

The main purpose of IDSA and DSSA is to enhance exiting scheduling algorithms those should also lead to energy efficiency at routing time. Moreover, the proposed algorithms should also overcome the defects of CSMA schemes like SMAC, BMAC. Based on the work carried out in this thesis, we will implement an entire protocol both on MAC layer and also at network layer to accomplish energy efficiency during setup phase and communication phase.

REFERENCES

[1] A. A. Ahmed, H. Shi and Y. Shang, "A survey on network protocols for wireless sensor networks," in *Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on*, 2003, pp. 301-305.

[2] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE,* vol. 40, pp. 102-114, 2002.

[3] J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey," *Computer Networks,* vol. 52, pp. 2292-2330, 2008.

[4] F. Khoshnoud and C. W. de Silva, "Recent advances in MEMS sensor technology-mechanical applications," *Instrumentation & Measurement Magazine, IEEE,* vol. 15, pp. 14-24, 2012.

[5] M. A. Sheikh, M. Drieberg and Noohul. Basheer. Z. Ali, "Fair scheduling algorithm for wireless sensor networks," in *National Postgraduate Conference (NPC), 2011,* 2011, pp. 1-5.

[6] M. A. Sheikh, M. Drieberg and Noohul. Basheer. Z. Ali, "An improved distributed scheduling algorithm for wireless sensor networks," in *Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on*, 2012, pp. 274-279.

[7] P. Sharma and P. S. Mandal, "Reconstruction of aggregation tree in spite of faulty nodes in wireless sensor networks," in *Wireless Communication and Sensor Networks (WCSN), 2010 Sixth International Conference on*, 2010, pp. 1-5.

[8] Y. Bo and L. Jianzhong, "Making Aggregation Scheduling Usable in Wireless Sensor Networks: An Opportunistic Approach," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*, 2011, pp. 666-671.

[9] M. Brandl, J. Grabner, K. Kellner, F. Seifert, J. Nicolics, S. Grabner and G. Grabner, "A Low-Cost Wireless Sensor System and Its Application in Dental Retainers," *Sensors Journal, IEEE,* vol. 9, pp. 255-262, 2009.

[10] L. Galluccio, T. Melodia, S. Palazzo and G. E. Santagati, "Challenges and implications of using ultrasonic communications in intra-body area networks," in *Wireless On-demand Network Systems and Services (WONS), 2012 9th Annual Conference on*, 2012, pp. 182-189.

[11] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang and X. Gao, "A wireless LAN-based indoor positioning technology," *IBM Journal of Research and Development,* vol. 48, pp. 617-626, 2004.

[12] H. Medeiros, J. Park and A. Kak, "Distributed Object Tracking Using a Cluster-Based Kalman Filter in Wireless Camera Networks," *Selected Topics in Signal Processing, IEEE Journal of,* vol. 2, pp. 448-463, 2008.

[13] D. Zhuxin, U. C. Wejinya and W. J. Li, "An Optical-Tracking Calibration Method for MEMS-Based Digital Writing Instrument," *Sensors Journal, IEEE,* vol. 10, pp. 1543-1551, 2010.

[14] X. Mingdong and H. Leung, "A Joint Fusion, Power Allocation and Delay Optimization Approach for Wireless Sensor Networks," *Sensors Journal, IEEE,* vol. 11, pp. 737-744, 2011.

[15] L. Krishnamachari, D. Estrin and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, 2002, pp. 575-578.

[16] S. Fangting and M. Shayman, "Prolonging Network Lifetime via Partially Controlled Node Deployment and Adaptive Data Propagation in WSN," in *Information Sciences and Systems, 2007. CISS '07. 41st Annual Conference on*, 2007, pp. 226-231.

[17] W. Hui, N. Agoulmine, M. Maode and J. Yanliang, "Network lifetime optimization in wireless sensor networks," *Selected Areas in Communications, IEEE Journal on,* vol. 28, pp. 1127-1137, 2010.

[18] T. Bin, X. Yang, L. Shoushan, Y. Xiou, L. Dong, G. Zhe and Y. Yixian, "A novel key management method for wireless sensor networks," in *Broadband Network and Multimedia Technology (IC-BNMT), 2010 3rd IEEE International Conference on*, 2010, pp. 1106-1110.

[19] Y. C. Tay, K. Jamieson and H. Balakrishnan, "Collision-minimizing CSMA and its applications to wireless sensor networks," *Selected Areas in Communications, IEEE Journal on,* vol. 22, pp. 1048-1057, 2004.

[20] H. Keong, K. Thotahewa and M. Yuce, "Transmit-Only Ultra Wide Band (UWB) Body Sensors and Collision Analysis," *Sensors Journal, IEEE,* vol. PP, pp. 1-1, 2013.

[21] E. Lattanzi and A. Bogliolo, "Ultra-low-power sensor nodes featuring a virtual runtime environment," in *Communications (ICC), 2012 IEEE International Conference on*, 2012, pp. 5731-5735.

[22] B. Otal, L. Alonso and C. Verikoukis, "Highly reliable energy-saving mac for wireless body sensor networks in healthcare systems," *Selected Areas in Communications, IEEE Journal on,* vol. 27, pp. 553-565, 2009.

[23] T. Torfs, T. Sterken, S. Brebels, J. Santana, R. van den Hoven, V. Spiering, N. Bertsch, D. Trapani and D. Zonta, "Low Power Wireless Sensor Network for Building Monitoring," *Sensors Journal, IEEE,* vol. 13, pp. 909-915, 2013.

[24] M. Kuroda, Q. Shuye and O. Tochikubo, "Low-power secure body area network for vital sensors toward IEEE802.15.6," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, 2009, pp. 2442-2445.

[25] R. Bista, K. Yong-ki and C. Jae-Woo, "A New Approach for Energy-Balanced Data Aggregation in Wireless Sensor Networks," in *Computer and Information Technology, 2009. CIT '09. Ninth IEEE International Conference on*, 2009, pp. 9-15.

[26] I. Rhee, A. Warrier, M. Jeongki and X. Lisong, "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks," *Mobile Computing, IEEE Transactions on,* vol. 8, pp. 1384-1396, 2009.

[27] H. M. Rathod and B. V. Buddhadev, "Comparative study of congestion control techniques for wireless sensor network," in *Engineering (NUiCONE), 2011 Nirma University International Conference on*, 2011, pp. 1-5.

[28] Y. Qicai, X. Jianping, Z. Yan and Z. Lei, "Performance Research and Simulation Analysis of the MAC Layer Protocols in Wireless Sensor Networks," in *Communications and Networking in China, 2006. ChinaCom '06. First International Conference on*, 2006, pp. 1-3.

[29] K. Adere and G. R. Murthy, "Solving the hidden and exposed terminal problems using directional-antenna based MAC protocol for wireless sensor networks," in *Wireless And Optical Communications Networks (WOCN), 2010 Seventh International Conference On*, 2010, pp. 1-5.

[30] H. Caishi, L. Chin-Tau and A. K. S, "On Fairness Enhancement for CSMA/CA Wireless Networks," *Systems Journal, IEEE,* vol. 4, pp. 511-523, 2010.

[31] Y. Wei, J. P. M. G. Linnartz and I. G. M. M. Niemegeers, "Adaptive CCA for IEEE 802.15.4 Wireless Sensor Networks to Mitigate Interference," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, 2010, pp. 1-5.

[32] L. Cheng, W. Pu, C. Hsiao-Hwa and M. Guizani, "A Cluster Based On-demand Multi-Channel MAC Protocol for Wireless Multimedia Sensor Networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, 2008, pp. 2371-2376.

[33] B. Yahya and J. Ben-Othman, "An Energy Efficient Hybrid Medium Access Control Scheme for Wireless Sensor Networks with Quality of Service Guarantees," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 2008, pp. 1-5.

[34] S. Gobriel, D. Mosse and R. Cleric, "TDMA-ASAP: Sensor Network TDMA Scheduling with Adaptive Slot-Stealing and Parallelism," in *Distributed Computing Systems, 2009. ICDCS '09. 29th IEEE International Conference on*, 2009, pp. 458-465.

[35] J. Degesys, I. Rose, A. Patel and R. Nagpal, "DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, 2007, pp. 11-20.

[36] W. Huahui and L. Tongtong, "Hybrid ALOHA: A Novel MAC Protocol," *Signal Processing, IEEE Transactions on,* vol. 55, pp. 5821-5832, 2007.

[37] S. A. Chowdhury, M. T. Islam, F. T. Jaigirdar, M. R. U. Faruqui and S. Al Noor, "Performance study and simulation analysis of CSMA and IEEE 802.11 in wireless sensor networks and limitations of IEEE 802.11," in *Computers and Information Technology, 2009. ICCIT '09. 12th International Conference on*, 2009, pp. 431-436.

[38] Y. Wei, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002, pp. 1567-1576 vol.3.

[39] J. Mizuguchi, M. Murata and W. Uemura, "A broadcasting method based on RTS/CTS for an ad-hoc network," in *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on*, 2011, pp. 104-106.

[40] T. v. D. a. K. Langendoen, "An adaptive energy-efficient macprotocol for wireless sensor networks," p. 13, In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems, Nov.2003.

[41] J. H. J. Polastre, and D. Culler, "Versatile low power media access for wireless sensor networks," *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), Baltimore,MD, November 2004.*, p. 13, 2004.

[42] O. G. Yanjun Sun, David B. Johnson, "RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks," *Proceeding SenSys '08 Proceedings of the 6th ACM conference on Embedded network sensor systems* pp. 1-14 2008 2008

[43] T. Lei, S. Yanjun, O. Gurewitz and D. B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 1305-1313.

[44] M. Junchao, L. Wei, W. Yanwei, L. Xiang-Yang and C. Guihai, "Energy Efficient TDMA Sleep Scheduling in Wireless Sensor Networks," in *INFOCOM 2009, IEEE*, 2009, pp. 630-638.

[45] W. Yanwei, L. Xiang-Yang, L. Yunhao and L. Wei, "Energy-Efficient Wake-Up Scheduling for Data Collection and Aggregation," *Parallel and Distributed Systems, IEEE Transactions on,* vol. 21, pp. 275-287, 2010.

[46] C. You-Lin and T. Shiao-Li, "Energy-Efficient Sleep-Mode Operations for Broadband Wireless Access Systems," in *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, 2006, pp. 1-5.

[47] Z. Cheng, M. Perillo, B. Tavli, W. Heinzelman, S. Tilak and N. Abu-Ghazaleh, "Protocols for local data delivery in wireless microsensor networks," in *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on*, 2002, pp. I-623-6 vol.1.

[48] S. Mishra and A. Nasipuri, "An adaptive low power reservation based MAC protocol for wireless sensor networks," in *Performance, Computing, and Communications, 2004 IEEE International Conference on*, 2004, pp. 731-736.

[49] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, p. 10 pp. vol.2.

[50] K. O. Venkatesh Rajendran, J.J. GarciaLunaAceves, "EnergyEfficient, CollisionFree Medium Access Control for Wireless Sensor Networks," 2003.

[51] V. Zibakalam, "A New TDMA Scheduling Algorithm for Data Collection over Tree-Based Routing in Wireless Sensor Networks," *Hindawi Publishing Corporation,* vol. Volume 2012 (2012), Article ID 864694, 7 pages, p. 7 pages, 2012.

[52] Z. Zusheng, Y. Fengqi and Z. Baoli, "A Depth-Based TDMA Scheduling for Clustering Sensor Networks," in *Frontier of Computer Science and Technology, 2009. FCST '09. Fourth International Conference on*, 2009, pp. 261-266.

[53] W. Wei, W. Honggang, P. Dongming and S. Hamid, "An Energy Efficient Pre-Schedule Scheme for Hybrid CSMA/TDMA MAC in Wireless Sensor Networks," in *Communication systems, 2006. ICCS 2006. 10th IEEE Singapore International Conference on*, 2006, pp. 1-5.

[54] R. Injong, A. Warrier, M. Aia, M. Jeongki and M. L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *Networking, IEEE/ACM Transactions on,* vol. 16, pp. 511-524, 2008.

[55] CC1101 radio transceiver data sheet available on line at this link: http://www.usconverters.com/downloads/cc1101.pdf.

[56] Marco Zuniga and Bhaskar Krishnamachari, "An Analysis of Unreliability and Asymmetry in Low-Power Wireless Links," *ACM Transactions onSensor Networks, 2007,* 2007.

[57] P. D. Kannan Srinivasan, Arsalan Tavakoli, and Philip Levis,, "Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks," *Technical Report SING-06-00. Stanford, 2006.,* 2006.

[58] C. F. Ranveer Chanadra, and H. Karin, "Improving Wireless Simulation Through Noise Modeling," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, 2007, pp. 21-30.

[59] C. F. Ranveer Chanadra, and H. Karin, "Adaptive topology discovery in hybrid wireless networks," *In Proceedings of Informatics,1st International Conference on Ad Hoc Networks and Wireless, Toronto,,* vol. 16, pp. 1-16, 1st 20-22, 2002 2002.

[60] A. Booranawong and W. Teerapabkajorndet, "Impact of Radio Propagation on the Performance of Directed Diffusion Routing in Mobile Wireless Sensor Networks," in *Proc. International Conference on Embedded Systems and Intelligent Technology (ICESIT)*, 2009.

[61] H. Lee and H. Kwon, "Improving Estimation of the Nearest Neighbor in Mobile Wireless Sensor Networks," ed: Citeseer, 2009.

[62] L. Gang, D. De, X. Mingsen, S. Wen-Zhan and C. Jiannong, "TelosW: Enabling ultra-low power wake-on sensor network," in *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*, 2010, pp. 211-218.

[63] I. Al Ajarmeh, K. El-Zayyat and J. Yu, "A Hybrid Routing Protocol for Mobile Ad Hoc and Wireless Sensor Networks," in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, 2008, pp. 1-5.

[64] K. Nishide, H. Kubo, R. Shinkuma and T. Takahashi, "Detecting Hidden and Exposed Terminal Problems in Densely Deployed Wireless Networks," *Wireless Communications, IEEE Transactions on,* vol. 11, pp. 3841-3849, 2012.

[65] A. Tsertou and D. I. Laurenson, "Revisiting the Hidden Terminal Problem in a CSMA/CA Wireless Network," *Mobile Computing, IEEE Transactions on,* vol. 7, pp. 817-831, 2008.

[66] Z. Xizheng and W. Yaonan, "New mixed broadcast scheduling approach using neural networks and graph coloring in wireless sensor network," *Systems Engineering and Electronics, Journal of,* vol. 20, pp. 185-191, 2009.

[67] S. Gandham, M. Dawande and R. Prakash, "Link scheduling in sensor networks: distributed edge coloring revisited," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE,* 2005, pp. 2492-2501 vol. 4.

[68] A. Kabbani, T. Salonidis and E. W. Knightly, "Distributed Low-Complexity Maximum-Throughput Scheduling for Wireless Backhaul Networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE,* 2007, pp. 2063-2071.

[69] X.-l. Zhou and B. Gong, "Intra-cluster Nodes Scheduling Algorithm Satisfying Expected Coverage Degree of Application in Distributed Clustering WSNs," in *Computer Science and Software Engineering, 2008 International Conference on,* 2008, pp. 332-335.

[70] L. Yingshu, G. Longjiang and S. K. Prasad, "An Energy-Efficient Distributed Algorithm for Minimum-Latency Aggregation Scheduling in Wireless Sensor Networks," in *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on,* 2010, pp. 827-836.

[71] T. Yuan, E. Ekici and F. Ozguner, "Energy-constrained task mapping and scheduling in wireless sensor networks," in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on,* 2005, pp. 8 pp.-218.

[72] Y. Gong, "Vehicle scheduling problems of military logistics distribution based on improved Genetic Algorithm," in *Computer Application and System Modeling (ICCASM), 2010 International Conference on,* 2010, pp. V11-382-V11-386.

[73] I. Slama, B. Jouaber and D. Zeghlache, "A Free Collision and Distributed Slot Assignment Algorithm for Wireless Sensor Networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE,* 2008, pp. 1-6.

[74] P. Sung, A. Savvides and M. B. Srivastava, "Simulating networks of wireless sensors," in *Simulation Conference, 2001. Proceedings of the Winter,* 2001, pp. 1330-1338 vol.2.

[75] A. K. Vlado Handziski, Holger Karl, J. Elson, , "EmStar: An Environment for developing Wireless Embedded Systems Software . Proceedings of the USENIX Technical Conference, 2004.," 2004.

[76] X. Xiaodong, S. Weiren and H. He, "Comparison of OMNET++ and other simulator for WSN simulation," in *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*, 2008, pp. 1439-1443.

[77] H. H. Refai and J. J. Sluss, Jr., "Work in progress- incorporation of OPNET simulation into a telecommunications systems laboratory course," in *Frontiers in Education, 2004. FIE 2004. 34th Annual*, 2004, p. S2C/3 Vol. 3.

PUBLICATIONS

1) Sheikh, Muhammad Aman, Zain Ali, Noohul Basheer and, Azlan Awang (2013). **"Queue Based Distributed Scheduling for Wireless Sensor Networks",** *Annual Postgraduate Conference* (APC 2013) Malaysia (Submitted).
2) Sheikh, Muhammad Aman, Zain Ali, Noohul Basheer and, Azlan Awang (2013). **"Distributive *and Self-Sustainable Scheduling Algorithm for Wireless Sensor Networks",*** In IEEE The 22nd Wireless and Optical Communication Conference (WOCC 2013) 2013, Chongqing, China.16-18May 2013.
3) Sheikh, Muhammad Aman and Drieberg , Micheal and Zain Ali, Noohul Basheer (2012). ***"An Improved Distributed Scheduling Algorithm for Wireless Sensor Networks",*** in *Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on Intelligent and Advanced Systems*, 2012, pp. 274-279.
4) Sheikh, Muhammad Aman and Drieberg , Micheal and Zain Ali, Noohul Basheer (2011). ***"Fair Scheduling Algorithm for Wireless Sensor Networks",*** In: National Postgraduate Conference (NPC) 2011, 19-20 September, 2011, Universiti Teknologi PETRONAS.

APPENDIX A

DEFINITION OF TERMINOLOGIES

Running time: The time required by the nodes in a network to allocate collision free time slots.

Message overhead: The total number of scheduling control packets that are exchanged during slot reservation are known as control overhead. In initialization or the setup phase, excessive numbers of control messages are exchanged by handshaking and scheduling among the neighbors. These control messages also consume network resources and most of the energy is consumed through unnecessary messages transmission. In order to conserve energy, the scheduling algorithm should have low communication overhead.

Energy consumption: Amount of battery power utilized to set up the scheduling process. The energy consumption rate for sensors in a wireless sensor network varies greatly based on the protocols the sensors use for communications.

Number of Rounds: Number of rounds refers to the cycles that a node utilizes to acquire its slot. Both the algorithms run in round and all the sensor nodes computes to schedule their slot per round.

Euclidian distance: Each node finds between them that is know as Euclidian distance.

APPENDIX B

PROPAGATION MODEL AND CALCULATIONS

B.1 TWO-RAY GROUND REFLECTION MODEL

Received signal strength (RSS) of a signal is approximated by propagation model based on the distance between receiver and transmitter, transmission power, and antenna configurations. A success or failure of a packet's reception depends upon RSS. TwoRayGround is one of the famous deterministic propagation models and the RSS determine is always same between fixed point transceivers'. In TwoRayGround propagation model consider line of sight as well as the reflected ray. In TwoRayGround propagation model a signal from transmitted is delivered to a receiver through multi paths, depending on how many it is reflected, diffracted or scattered on the designated ray-paths. Figure B.1 illustrates that in two-ray model the single ground reflection is dominated over multi-path components [61][62].

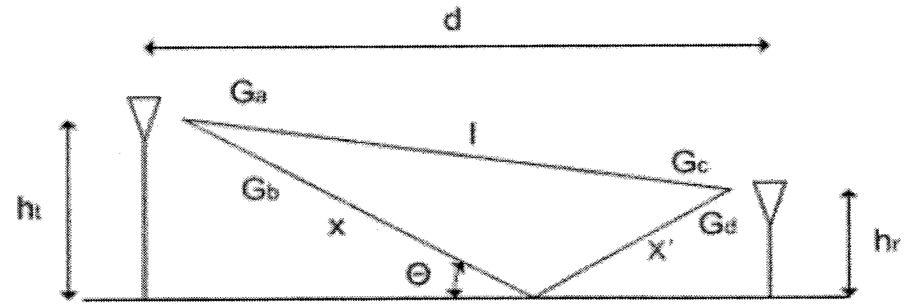


Figure B.1 Two-ray Ground propagation Model

Then, the received power, Pr, is calculated via the geometric view as follows.

$$P_r = P_t\left(\frac{\lambda}{4\pi}\right)^2\left|\frac{\sqrt{G_l}}{l}+\frac{R\sqrt{G_r}e^{-j\Delta\varphi}}{x-x^!}\right|^2 \qquad B.1$$

Pt is a transmitted power, where $G_l = G_aG_b$ is the product of receiver and transmitter antenna gains in the Line of sight, in corresponding to the reflective direction $G_r = G_c G_d$ is the product of receiver and transmitter antenna gains, and R is the ground reflection coefficient. $\Delta\Phi = 2\,\pi(x - x^! - l)/\lambda$ is the phase difference between the reflected and Line of sight path. From the geometry, the distance difference and the phase difference is given by:

$$x + x^! - l = \sqrt{(h_t + h_r)^2 + d^2} - \sqrt{(h_t + h_r)^2 + d^2} \qquad B.2$$

$$\Delta\varphi = \frac{2\pi(x + x^! - l)}{\lambda} \approx \frac{4\pi h_t + h_r}{\lambda d} \qquad B.3$$

where it is d is asymptotically large enough compared to $h_t + h_r$. If this assumption makes sense in the network model, then the parameters can be supposed like $x + x^! \approx l \approx d, \sigma \approx o, G_l \approx G_r, and\ R \approx -1.$

$$P_r \approx P_t \left(\frac{\lambda\sqrt{G_l}}{4\pi d}\right)^2 \left(\frac{4\pi h_t h_r}{\lambda d}\right)^2 \qquad B.4$$

$$P_r \approx P_t \left(\frac{\sqrt{G_l} h_t h_r}{d^2}\right)^2 \qquad B.5$$

B.2 TRANSMISSION RANGE

Transmission range is the minimum configured range that is required for connectivity among the senor nodes. Transmission range can be varied in order to observe the behavior of network at each node degree of node. We use TwoRayGround propagation model. Equation B.6 shows the path loss of our model from which different communication ranges can be computed against different transmission power.

$$L(dB) = 40\ log\ d - (10\ log\ G_t + 10\ log G_r + 20\ log h_t + 20\ log\ h_r) \qquad B.6$$

Where

G_t: Is the transmitter antenna gain

G_r: Is the receiver antenna gain

h_t: Is the height of transmitter antenna

h_r: Is the height of receiver antenna

L: Is the path loss

$$L(dB) = P_{tx(dB)} - P_{rx(dB)} \qquad B.7$$

$$P_{rx(dB)} = -60\ dBm = -90dB \qquad and \qquad P_{tx(dB)} = -30\ dB$$

Similarly from equation B.6 and B.7

$$60 = 40\ log\ d - (10\ log\ G_t + 10\ log G_r + 20\ log h_t + 20\ log\ h_r)$$

G_r *and* G_t *was* set *to* 1 *while* h_t *and* h_r $1.5m$

Therefore,

$$60 = 40\ log\ d - 7.04$$

$$d = 10^{\frac{67}{40}} \approx 40$$

We can further find distance based on different transmission power.

APPENDIX C

NS-2 SIMULATOR

C.1 NS-2 BASIC ARCHITECTURE

The NS-2 architecture is composed of two basic languages: C++, an object oriented language, and the other one is the Object-oriented Tool Command Language (OTcl). Functions, structures, classes or any parameters are defined in the C++ file while OTcl is an upper level implementation where users can feed a Tcl script argument as an input for the NS-2 executable command. Similarly, NS-2 has two classes of hierarchy: C++ compiled and Otcl interpreted; both of these are in one to one correspondence with each other. The C++ hierarchy facilitates uses for efficient and faster simulation execution by modifying the exiting algorithms or introducing their own algorithms. This is useful for the detailed operation of algorithms that helps to reduce event processing time. As mentioned earlier, OTcl is an upper level implementation; it provides linkage to the C++ objects. Thus, after defining all of the functions in C++ for the simulation execution, user moves towards the OTcl script. Any particular network topology is defined in OTcl. NS has the rich function of a library, therefore, the user can select the desired applications and specific protocols to simulate and conclude their algorithms behaviour under those conditions.

In this proposed approach, NS was selected because it offers a flexible high-level programming language using C/C++ for the modelled system and graphical editors that in turn enables the researchers to improve towards the desired models. Besides some built-in applications and modules in NS, new algorithms were introduced as per the methodology explained in chapter 3, and also modified some of the existence modules. In this way, it helped to delineate fresh applications and new algorithms for advanced and distributive WNs. Figure C.1 illustrates the simulation architecture adopted in the proposed algorithms.

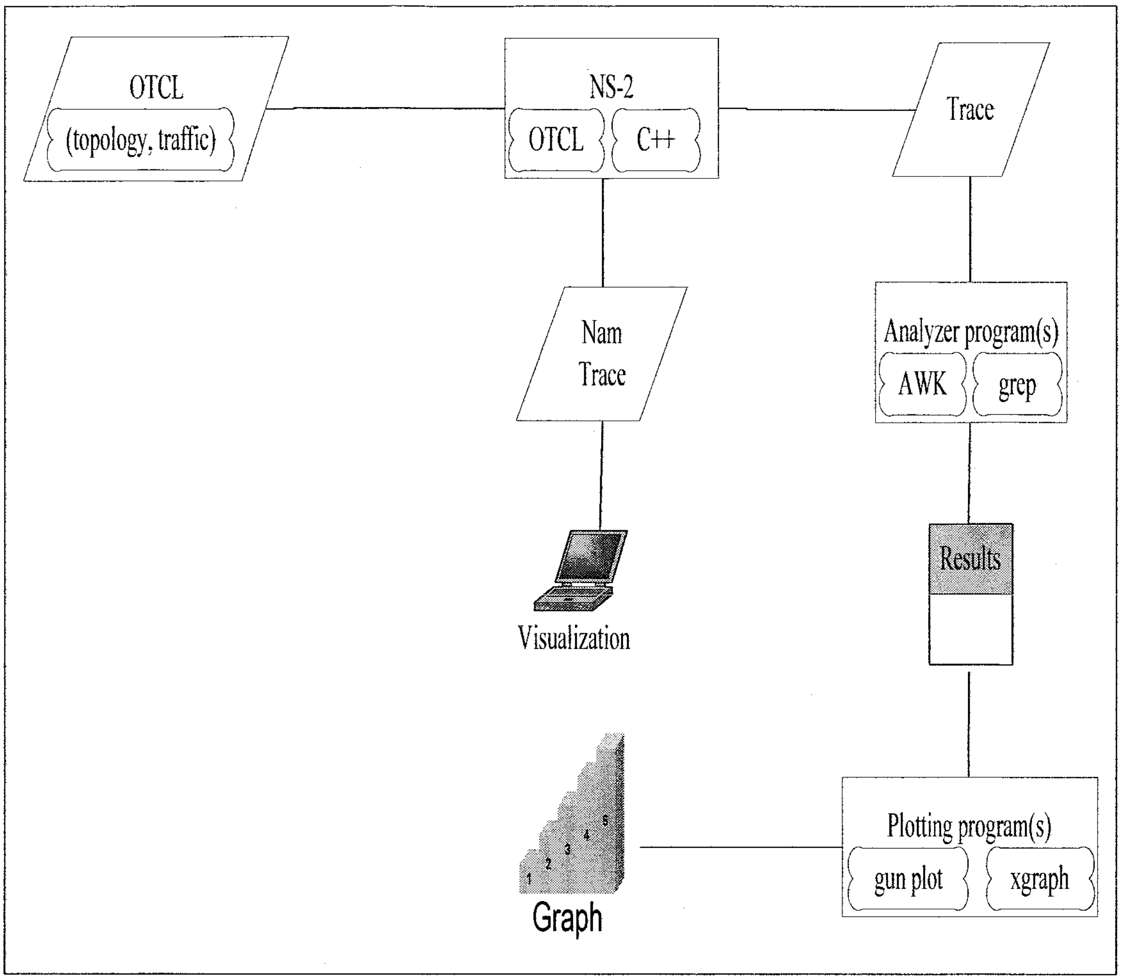


Figure C.1 Simulation architecture

NS is a discrete time event simulator, and the future event time depends on the event trace maintained in a scheduler. An event refers to the object in C++ that handles an object pointer and a scheduled time. The scheduler keeps all the data structures in a sequence according to the events to be executed by invoking the handler. After the simulation execution, the user is interested in the output; in NS, there are two types of outputs: either animation-based or text-based. In order to interpret these results graphically and interactively, tools such as gunplot, XGraph and NAM (Network AniMator) are used. To examine the behaviour of any particular portion of a network, users can extract a subset of a text-based data by using many functions such as "awk", "grep", etc. to transform the text-based information into a more conceivable presentation.