DEVELOPMENT OF A MULTICHANNEL BASED LOW LATENCY ASYNCHRONOUS MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS

By
Md. Mustafa Kamal

MASTER OF SCIENCE
IN
INFORMATION AND COMMUNICATION TECHNOLOGY

INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

The thesis titled "**DEVELOPMENT OF A MULTICHANNEL BASED LOW LATENCY ASYNCHRONOUS MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS**" submitted by Md. Mustafa Kamal, Roll No. 0412312012 P and Session April, 2012, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Information and Communication Technology on April 30, 2017.

BOARD OF EXAMINERS

1. **Dr. Mohammad Shah Alam** Associate Professor IICT, BUET, Dhaka	**Chairman** **(Supervisor)**
2. **Dr. Md. Saiful Islam** Professor and Director IICT, BUET, Dhaka	**Member** **(Ex-Officio)**
3. **Dr. Md. Rubaiyat Hossain Mondal** Associate Professor IICT, BUET, Dhaka	**Member**
4. **Dr. Md. Shariful Islam** Professor Institute of Information Technology University of Dhaka	**Member** **(External)**

Declaration

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

Signature of the Candidate

Md. Mustafa Kamal
0412312012 P
IICT, BUET

Dedication

THIS THESIS IS DEDICATED

TO

MY PARENTS

Contents

List of Tables

List of Figures

List of Abbreviations

X-MAC/CA	X-MAC/ Collision Avoidance
ZeroMAC	Zero MAC
MCAS-MAC	Multichannel Asynchronous Scheduled MAC Protocol
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access/ Collision Avoidance
CTS	Clear To Send
RTS	Request To Send
FDMA	Frequency Division Multiple Access
TDMA	Time Division Multiple Access
BS	Base Station
WSN	Wireless Sensor Network
ID	Identification Number
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISA	Instrumentation Systems and Automation society
ISM	Industrial, Scientific and Medical
S-MAC	Sensor-MAC

List of Symbols

λ	Data arrival probability
μ	Data transmission probability
τ	Length of a slot
π_0	Probability of empty queue
A_i	Probability of i packets arrive
P_s	Probability of success
P_c	Probability of collision
S_P	Duration of sender preamble
S_d	Duration of send data
t_d	Data transmission period
E_s	Expected energy to send a packet
E_r	Expected energy to receive a packet
R_a	Duration of receiver ACK period
R_d	Duration of receiver data period
f_n	Number of frequencies used in multichannel
H_p	Duration of sender hello message

Acknowledgment

All praises are for the almighty Allah for giving me the strength, without which I could not afford to attempt this research work.

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

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

Abstract

In Wireless Sensor Network (WSN), nodes have to share the medium for data transmission. Therefore, Medium Access Control (MAC) protocol has been developed to provide collision free access through this common channel. Standard MAC protocols for WSNs can be categorized into synchronous, asynchronous and hybrid. Synchronous MAC protocols, specify the duration of wake-up and sleep time to reduce the unnecessary communication time and energy wasted in idle listening. One major drawback of these protocols is that they are highly dependent on clock synchronization. On the other hand, asynchronous MAC protocols are found to be more energy efficient compared to synchronous MAC protocols as they remove the system overhead required for synchronization. However, asynchronous MAC protocols inherently introduce high latency, especially in multi hop communication scenarios. Another drawback is that nodes may severely suffer from 'hidden node problem' due to the lack of synchronization. In this thesis, a multi-radio multi-channel based X-MAC protocol (MX-MAC) with the objective of eliminating hidden node problem of existing asynchronous MAC protocol, as well as reducing average packet delay is proposed. For multi-radio WSNs, each node is equipped with two or more radio interfaces where one radio is dedicated to control channel and the rests to data channel. Both the separate interfaces can be remain active and can send control information and data simultaneously. To address the hidden terminal problem, multi channel enables a receiver to send a preamble with next receiver ID on the control channel to notify its neighbors that an ongoing reception is in progress and alert intended next hop neighbor to wake up on time as well as to forward received data to next hop without sending any additional preamble. In consequence of this process, packet collisions are reduced and the network throughput is increased. In order to carry out the performance analysis, analytical models of the existing X-MAC/CA, ZeroMAC, MCAS-MAC as well as the proposed MX-MAC protocols are developed. Based on these models, different performance metrics of the proposed MX-MAC protocol are compared with those of the existing asynchronous MAC protocols. Analytical results show that the proposed protocol can significantly improve the throughput as well as can reduce average packet delay.

Chapter 1

Introduction

Wireless Sensor Network (WSN), which consists of several energy constraint sensor nodes, always thrives for energy saving to extend the lifetime of the deployed network. Designing an energy efficient medium access control (MAC) protocol is therefore one of the most challenging criteria for WSN. So far, many energy efficient MAC protocols have been proposed among which asynchronous MAC protocols, such as B-MAC [2] , Wise-MAC [3] and X-MAC [4] are found to be more energy efficient compared to synchronous MAC protocols . The major advantage of such protocols is that it completely removes the protocol overhead needed for synchronization. Again, in multi hop communication system asynchronous MAC protocols spontaneously introduce high latency. Furthermore, the major pitfall involves their severe suffering from hidden node problem owing to the lack of synchronization.

1.1 Motivation

Energy saving and optimized MAC protocol design is an essential aspect of extending life time of wireless sensor networks. WSN consists of a large number of battery-powered sensors are distributed within an area of interest in order to track, measure and monitor various important events. Duty cycling, a primary mechanism for achieving low energy consumption in energy-constrained WSNs, cycles each sensor node periodically between an awake state and a sleep state. Key parameters that characterize the duty cycle include sleep time, wake time and the energy consumed during its operation. The major challenges of designing a communication protocol include achieving high throughput, low delay, and energy efficiency [9]. One of the leading sources of energy consumption in WSN is idle listening, i.e., listening to an idle channel to receive possible traffic, which is due to nodes that periodically listen to the channel. Overhearing is another cause of energy exhaustion, which occurs due to receiving packets by a node that are destined to other nodes not for this particular node. Furthermore, since all packets experiencing collision have to be discarded, the re-transmissions of these packets are required which ultimately increase the energy consumption. Although some packets could be recovered by a capture effect, a number of requirements have to be achieved for its success. Besides this, energy is also wasted as a result of control packet overhead and, therefore, minimal number of control packets should be used to make a data transmission. Moreover, energy is consumed by over emitting, which is caused by the transmission of a message when the destination node is not ready. Hence, a lot of research has been carried out over the last decade with an aim to provide an efficient MAC protocol which can minimize interference and packet collisions as well as reduce all the above mentioned energy consumption efficiently. Traditional low cost radios for wireless sensor networks operate with one frequency channel at any given time. The recent development in radio hardware for WSNs make it possible to use transceivers that support two channels for sending control information and data simultaneously [10]. Dedicated control channel protocol uses separate control and data channel. The efficiency of this protocol is limited by the contention in control channel and the number of usable data channels. OFDMA-based multi-channel MAC protocol can trim down this limitation of dedicated control channel method [11]. Another problem of the dedicated control channel protocol is Control Channel Saturation (CCS). A strategy named TXOP (Transmission Opportunity) is used to solve CCS problem for Wireless mesh networks (WMNs) [12]. In WSN, nodes usually have to share a common channel. Therefore, the MAC protocol task is to provide fair access to channels by avoiding possible collisions. Standard MAC protocols developed for WSNs can be roughly categorized into synchronized and asynchronous approaches [13], along with hybrid combinations. Synchronous MAC protocols specify the period of wake-up and sleep for communication to reduce the unnecessary time and energy wasted in idle listening. On the other hand, these protocols increase network latency due to clock synchronization. S-MAC [14] and T-MAC [15] are the popular duty-cycled MAC protocol designed for wireless sensor networks to save energy and extend the network lifetime [13]. Asynchronous MAC protocols are usually characterized by their simplicity. A lot of energy efficient MAC protocols have been proposed. Among them asynchronous MAC protocols, such as B-MAC, Wise-MAC and X-MAC are found to be more energy efficient compared to synchronous MAC protocols. Asynchronous duty-cycled MAC protocols remove the energy overhead for synchronization, and they are easier to implement as they do not require clock synchronization. Asynchronous MAC protocols inherently introduce high latency, especially in multi hop communication scenarios. Another major drawback is that nodes severely suffer from 'hidden node problem' due to the lack of synchronization. Hidden nodes in a wireless network refer to nodes that are out of range of other nodes or a collection of nodes [16]. The hidden node problem occurs when a node is detectable from a wireless access point (AP), but not from other nodes communicating with that AP. Asynchronous protocols like X-MAC have more promising applications in wireless sensor network. X-MAC protocol, an enhanced version of B-MAC, solves the overhearing problem by dividing the one long preamble into a series of short preamble packets, each containing the ID of the target node. Despite all these benefits provided by the XMAC protocol, the hidden terminal problem, one of the major challenges in wireless MAC protocols, has still remained unresolved in this efficient MAC protocol. This research work aims to eliminate the hidden node problem of X-MAC protocol, a popular asynchronous MAC protocol for WSNs, as well as to reduce the average packet delay, thereby improving the lifetime and Quality of Service (QoS) of sensor networks.

1.2 Main Contributions

This research work makes the following contributions:

- A multichannel based low latency asynchronous MAC protocol (MX-MAC) has been developed to eliminate the hidden node problem and minimize the latency of existing asynchronous X-MAC protocol of WSN.

- Markov chain analytical models have been developed for ZeroMAC [6], MCAS-MAC and the MX-MAC to investigate the performance of proposed MAC protocol.

- Finally, analytical results have been compared with the Analytical results of X-

MAC [4], X-MAC/CA [5], ZeroMAC [6], and MCAS-MAC [7]. From analytical results it is evident, in the proposed model, energy savings is increased and latency is decreased with respect to X-MAC, X-MAC/CA, ZeroMAC, and MCAS-MAC.

1.3 Research Objectives

- To propose a new Multichannel based X-MAC protocol (MX-MAC) with the objective of eliminating hidden node problem of existing asynchronous X-MAC protocol, as well as reducing average packet delay.
- To carry out performance analysis proposed Markov model of MX-MAC with a finite queue capacity.
- Finally, to show the efficacy of the proposed MAC protocol, performance comparison between the proposed protocol and other related existing protocols will be investigated.

1.4 Organization of Thesis

The rest of the thesis is prepared as follows: Chapter 2 discusses fundamental issues of MAC design and related works. Different medium access related mechanisms X-MAC, X-MAC/CA, ZeroMAC and MCAS-MAC which are the exclusive background of this thesis are also described in this chapter. Multichannel asynchronous MAC protocol is introduced in Chapter 3. In this chapter, proposed multichannel asynchronous MX-MAC protocol to reduce the impact of collisions of X-MAC protocol, as well as to reduce the average packet delay, thereby improving the lifetime and QoS of sensor networks is illustrated. Chapter 4 depicts the mathematical modeling for WSN MAC protocols. In this chapter modeled and analyzed the throughput, latency and energy consumption of MX-MAC and related MAC protocols X-MAC, X-MAC/CA, ZeroMAC and MCAS-MAC. In Chapter 5, results and performance is evaluated in terms of latency, energy consumption and throughput. The concluding chapter or chapter 6 will be dedicated to an exploration of future work that will include the reduction of collisions in control channel.

1.5 Summary

In this research work, a multi-radio multi-channel asynchronous MX-MAC protocol has been proposed to reduce the impact of collisions of X-MAC protocol, as well as to reduce the average packet delay, thereby improving the lifetime and QoS of sensor networks. The key intention of the protocol is to extend the life span of the WSN by reducing latency and energy loss. From the analytical results it is evident that the proposed algorithm has an enhanced throughput compared to X-MAC, X-MAC/CA, ZeroMAC and MCAS-MAC. Furthermore, it is shown that the average packet delay can be effectively reduced by using “Next Receiver ID” information in the receiver’s preamble packets. The results obtained in this research are pragmatic and more beneficial in comparison with the existing protocols, especially regarding MX-MAC protocol’s ability of reducing energy consumption and average packet delay in WSN effectively.

Chapter 2

Design Issues of MAC Protocol and Related Works

2.1 Overview of Wireless Sensor Networks

Wireless Sensor Networks (WSN) consists of a large number of small, battery-powered, inexpensive and wireless sensors which send their data wirelessly [10]. More specifically, a WSN consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc [9]. Typically, data packets are generated by each node and are sent to a Base Station (BS) or a sink. Here the data is aggregated and forwarded to the user. Figure 2.1 shows the basic architecture of a WSN using sensors to monitor the physical conditions. From the above a WSN has peculiarities such as limited processing power, limited memory, low power, low rate, limited range radio and battery driven sensor nodes.

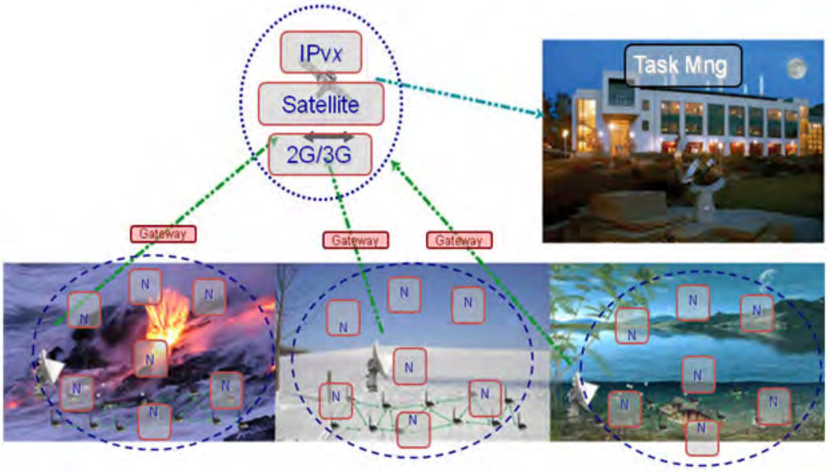


Figure 2.1: Basic architecture of a wireless sensor network

2.1.1 Wireless Sensor Network Applications

WSNs are widely used in various kinds of applications, such as military surveillance applications, health applications, and environmental applications [10]. For example, in military applications WSNs can be an integral part of military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting systems. Besides, these can also be used for target-detection, for monitoring forces and equipment, for detecting nuclear, biological or chemical attacks and for the surveillance of battlefields. Another example is using WSN for health applications, which are providing interfaces for the disabled, integrated patient monitoring, diagnostics and hospital drugs administration. WSN are also used in environmental monitoring where WSN can be used for tracking the movements of the small animals, detecting forest fires, flood detection and environmental monitoring in marine and soil environments. One of the applications is the use of WSN in homes such as home automation. In recent years WSNs are widely used in more fields such as wireless factory, smart (intelligent) buildings and implantable medical sensors which are used for medical applications [13].

2.1.2 Classification of WSN MAC Protocols

Standard MAC protocols developed for WSNs can be roughly categorized into synchronized and asynchronous approaches [17], along with hybrid combinations.

2.1.2.1 Synchronous Protocol

Synchronous MAC protocols specify the period of wake-up and sleep for communication to reduce the unnecessary time and energy wasted in idle listening. On the other hand, these protocols increase network latency due to clock synchronization. S-MAC [14] and T-MAC are the popular duty-cycled MAC protocol designed for wireless sensor networks to save energy and extend the network lifetime [18].

Advantages of Synchronous MAC Protocol:

- Nodes are communicated in common active schedule which reduce the delay for the sender to meet the receiver's active period.
- Less energy waste for idle listening.

Disadvantages of Synchronous MAC Protocol:

- Dependence on clock synchronization is obviously one of the salient defects of synchronous MAC protocol.

2.1.2.2 Asynchronous Protocol

The Asynchronous MAC protocols are usually characterized by their simplicity. A lot of energy efficient MAC protocols [19] have been proposed. Among them asynchronous MAC protocols, such as B-MAC, Wise-MAC and X-MAC are found to be more energy efficient compared to synchronous MAC protocols. Asynchronous duty-cycled MAC protocols remove the energy overhead for synchronization, and they are easier to implement as they do not require clock synchronization. Asynchronous MAC protocols inherently introduce high latency, especially in multi hop communication scenarios [20]. Another major drawback is that nodes severely suffer from 'hidden node problem' due to the lack of synchronization. Asynchronous protocols like X-MAC [4] have more promising applications in wireless sensor network.

Advantages of Asynchronous MAC protocol:

- Asynchronous MAC protocols are found to be more energy efficient compared to synchronous MAC protocols [19] as they remove the system overhead required for synchronization.

Disadvantages of Asynchronous MAC Protocol:

- Due to the lack of synchronization nodes may severely suffer from 'hidden node problem' [21]. It is conspicuously another remarkable shortcoming of asynchronous MAC protocols.

2.1.2.3 Hybrid Protocol

Hybrid protocol designed for wireless sensor networks which combines the strengths of synchronous and asynchronous while offsetting their weaknesses. For example, SCP-MAC [22] is hybrid MAC protocol where a dual-radio framework for MAC protocol implementation in wireless sensor networks.

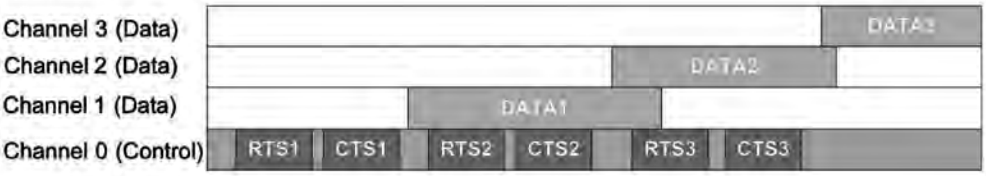


Figure 2.2: Operations of dedicated control channel (adopted from [1]).

2.2 Multi-radio Multi-channel WSNs

In multi-radio multi-channel wireless sensor network, each node is equipped with one or more radio interfaces. The radio interfaces assigned with different channels and can be remain active simultaneously [23]. Thus, the network throughput can be significantly improved as compared with a single-radio system. In dedicated control channel MAC protocol every device has two radios. One radio is tuned to a channel dedicated to control message and the other radios can be tuned to any other channels. Figure 2.2 illustrates the operations of dedicated control channel. In the figure, channel 0 is for control information and channels 1, 2, and 3 are for data transmission. [1].The RTS and CTS control informations are sent by the control channel. The control information contains a Network Allocation Vector (NAV) field, as in 802.11. This NAV is used to inform other devices about the duration of data exchange between sender and receiver. During the data transmission, the data channel is shown busy by the NAV.

2.3 Challenging Issues of Asynchronous MAC Protocol

2.3.1 Hidden Terminal Problem

Hidden terminal problem refers to the collision of packet at the receiving node due to the simultaneous transmission of those node that are not within the direct transmission range of the sender, but are within the transmission range of the receiver. Collision occurs when both node transmit packet at the same time without knowing about the transmission of each other. In the figure 2.3, when communication in between node A and B is in progress, node C sends RTS to node B, this RTS cannot hear by the node A, this RTS frame is collided with the ongoing transmission between node A and B.

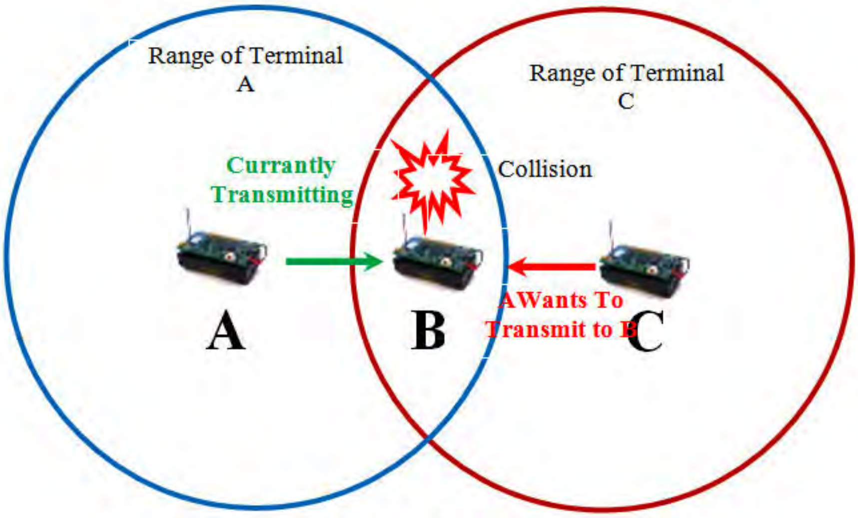


Figure 2.3: Hidden terminal problem

2.3.2 Latency

Latency is a time interval between the stimulation and response. In WSN latency includes sampling time, processing time, propagation time, scheduling, MAC protocols, use of directional antennas, predictions, sleep/wakeup cycles, use of dual-frequency radios and more. Energy conservation in wireless sensor networks has been a primary design objective. At the same time latency is another important perimeter in WSN, specially in asynchronous MAC protocol design [10]. Latency can be reduced by using the technique i.e. CSMA/CA [24] and MACA(Multiple Access Collision Avoidance) [25].

2.4 Related Research Works

2.4.1 B-MAC

B-MAC [2] is an energy efficient MAC protocol. B-MAC is a carrier sense medium access (CSMA) protocol for wireless sensor networks. B-MAC is the first LPL [26] protocol in which a node exploits original physical layer information . To reach low power, B-MAC uses an adaptive preamble sampling scheme to reduce duty cycle and minimize idle listening. B-MAC protocol contains a small core of media access functionality. B-MAC uses CCA and packet backoffs for channel arbitration, link layer acknowledgments for reliability, and LPL for low power communication which is shown in figure 2.4. The use of layer 1 requires B-MAC's nodes to remain awake

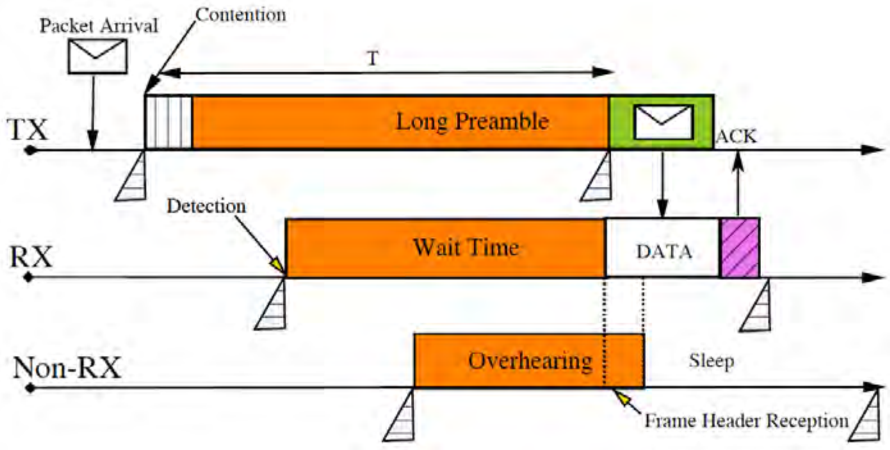


Figure 2.4: Sequence of operations perform in B-MAC (adopted from [2]).

throughout the duration of long-preamble transmissions before obtaining the useful data packet since there is no mean to learn the link layer address of the target. EA-ALPL [27] and SEESAW [28] optimize B-MAC by setting node's listening mode according to its current and past forwarding loads. Since B-MAC uses long preamble, it causes extra latency at each hop. Furthermore, the use of periodic layer 1 receive checks obligates B-MAC transmitters to send packets using long bit-stream preambles equivalent to one wake-up interval, without target ID. These preambles wake-up every node within vicinity of the transmitter, whether or not it is the target. This results in the well-known overhearing problem.

2.4.2 WiseMAC

WiseMAC (Wireless Sensor MAC) [3] is a infrastructure based wireless sensor networks specially design for multi-hop communication. WiseMAC is based on short asynchronous duty cycle preamble sampling scheme. This method consists in periodically sampling the medium to check for activity. By sampling the medium, it means listening to the radio channel for a short duration. All sensor nodes in a network sample the medium with the same constant period. If the medium is found busy, a sensor node continues to listen until a data packet is received or until the medium becomes idle again. The idea is to learn the direct neighbors' sampling period in order to use a wake-up preamble of minimized size. When the receiver's wakeup pattern is still unknown, the duration of the preamble is equal to the full basic cycle duration. The own schedule offset is then added to the frame and transmitted to the

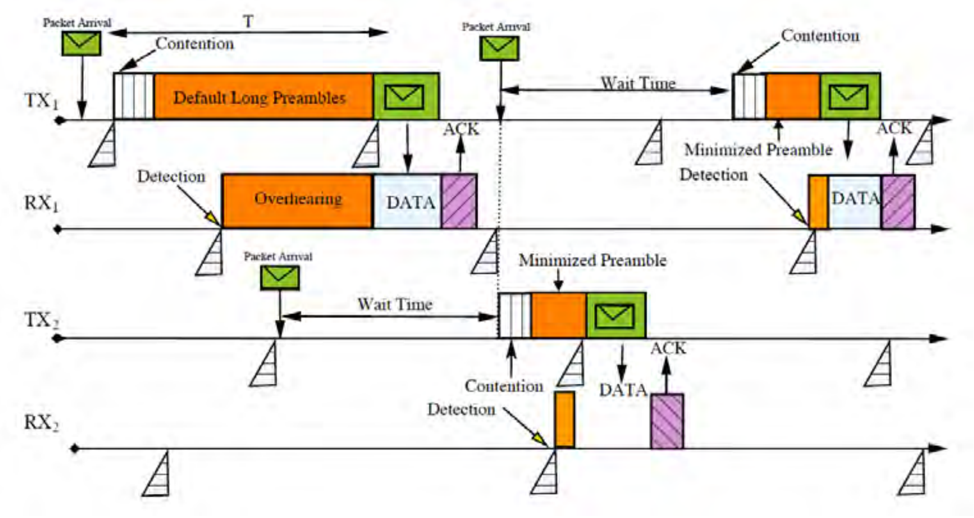


Figure 2.5: WiseMAC - principle of operation (adopted from [3]).

receiver. Each receiver adds its own schedule to the successful received frame's acknowledgement. Received schedule offsets of all neighboring nodes are subsequently kept in a table and are dynamically updated whenever frames and schedules are exchanged or possibly overheard. Based on the schedule offset table, a node can determine the wake-up time of all its neighbors. So it wakes-up at the exact moment to send data, which leaves the channel free for as long as possible, improving thus transmission delay of its neighbors. The principle operation of WiseMAC shown in figure 2.5.

2.4.3 X-MAC

X-MAC [4] is contention-based protocol which is shown by visual representation in the top of segment of figure 2.6. In this protocol when a node has data to send, it first transmits an extended preamble, and then sends the data packet. All other nodes maintain their own unsynchronized sleep schedules. When the receiver awakens, it samples the medium. If a preamble is detected, the receiver remains awake for the remainder of the long preamble, and then determines if it is the target [29]. After receiving the full preamble, if the receiver is not the target, it goes back to sleep. X-MAC, solve the overhearing problem by dividing the one long preamble into a series of short preamble packets, each containing the ID of the target node, as indicated in Figure 2.6. The stream of short preamble packets effectively constitutes a single long preamble. When a node wakes up and receives a short preamble packet, it looks

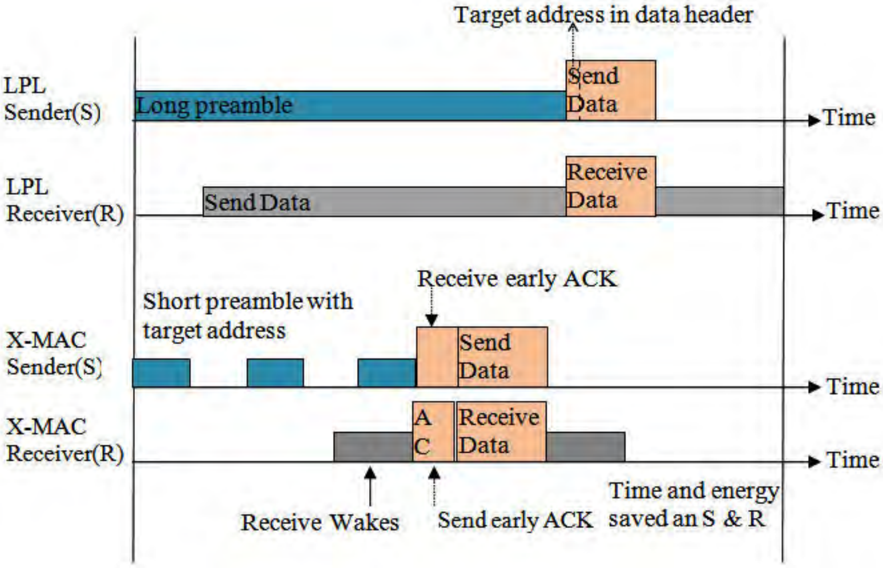


Figure 2.6: A visual representation of X-MAC (adopted from [4]).

at the target node ID that is included in the packet. If the node is not the intended recipient, the node returns to sleep immediately and continues its duty cycling as if the medium had been idle [30]. If the node is the intended recipient, it remains awake for the subsequent data packet. As seen in the figure 2.6, a node can quickly return to sleep, thus avoiding the overhearing problem.

2.4.4 X-MAC/CA

This section illustrates the behavior of X-MAC/CA [5] protocol of a sender shown in Figure 2.7. Note that the receiver part of X-MAC/CA protocol is omitted since it is the same as the X-MAC. As depicted in Figure 2.7, at first when a sensor node wakes up and has some data to send in its queue, the sensor node performs preamble backoff in which it randomly draws an integer number i from a range (1, Wmax) and waits for i backoff slots without polling the channel state unlike IEEE 802.11. At the expiration of this waiting interval, it runs one CCA (Clear Channel Assessment) operation to make sure that the channel is free just before the transmission [31]. If the channel is idle, then it begins transmitting its short preambles in sequence until the early acknowledgement comes back. Once the early acknowledgement arrives, it immediately delivers one data frame and goes to sleep. When the channel is already busy and the polled short preamble carries the same destination address as what the sender wants to deliver at, then it waits for the ongoing transmission without

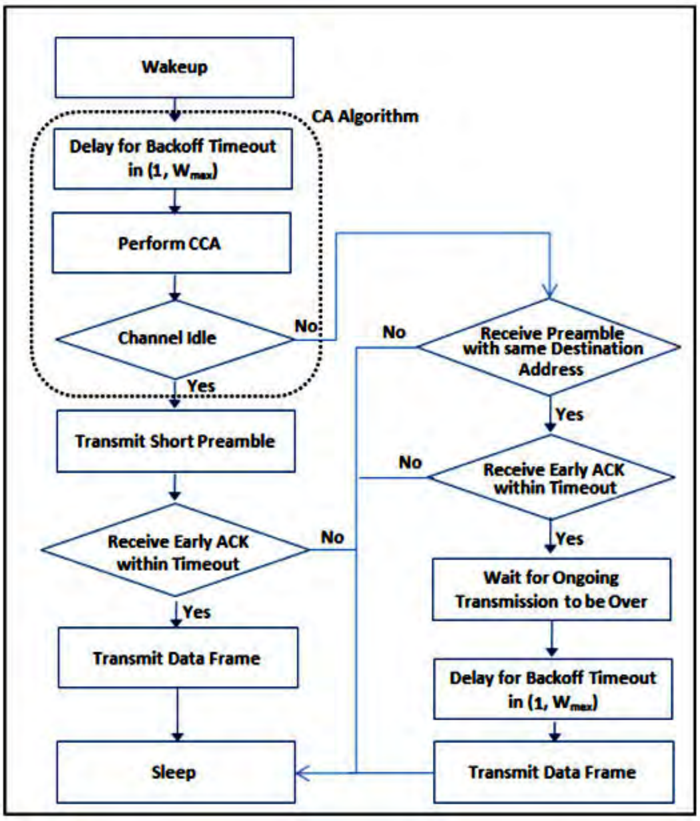


Figure 2.7: Flow chart of X-MAC/CA sender (adopted from [5]).

going to sleep. Once the transmission is over, it runs data backoff whose operation is the same as preamble backoff in that it decides the appropriate transmission time by selecting a random waiting time [32]. After this random timeout elapses, it sends its data frame without introducing short preambles. It is due to that the receiver is supposed to wait for the second transmission for a while without sleeping immediately after the first transmission.

2.4.5 ZeroMAC

Figure 2.8 shows the communication process of ZeroMAC [6]. The only difference between communication processes of 802.11 DCF [33] and ZeroMAC is a wakeup signal. ZeroMAC still uses RTS/CTS exchange to avoid hidden terminal problem in ZeroMAC. Both a sender and a receiver need to broadcast a wakeup signal before transmitting their first control packet [34]. If the receiver skips the wakeup signal, the overhearing nodes of a receiver would wake up when the receiver sends CTS. In this case, they need to keep awake until they receive an ACK packet, which would

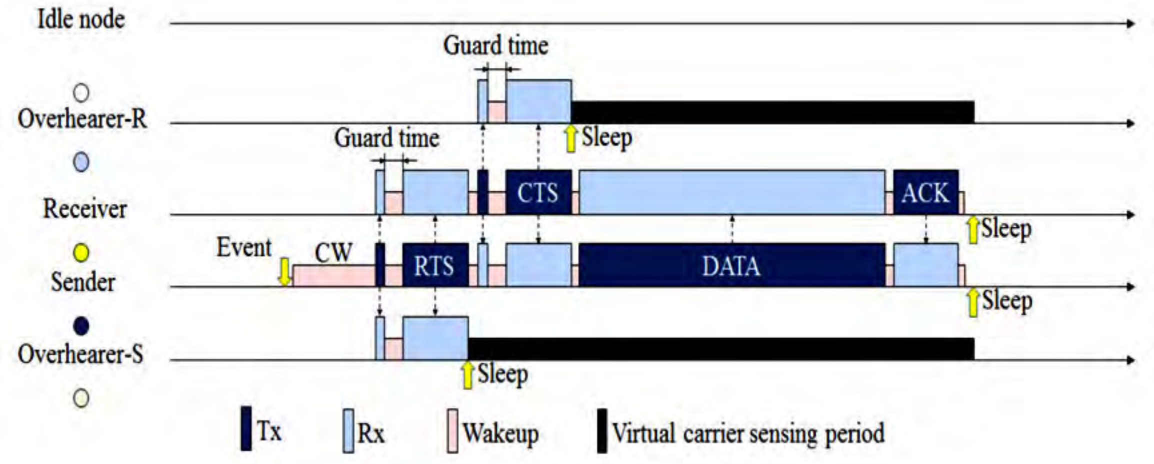


Figure 2.8: The communication process of ZeroMAC (adopted from [6]).

increase unnecessary idle listening. Note that the sender transmits a wakeup signal right before transmitting an RTS since only senders need to monitor a channel during a contention window. After transmitting a wakeup signal the transmitter has to wait until its neighbors turn on their RF module [35]. This waiting time define as guard time. Instead of a wakeup signal, a sender can also wake up its neighbors by extending a preamble included in the PHY header of control packets. In this case the preamble length need to be increase in order to cover the guard time. This overhead is comparable to the size of control packets used for WSNs [36]. Therefore, ZeroMAC use a separate wakeup signal which consumes less energy than the preamble extension.

2.4.6 MCAS-MAC

MCAS-MAC [7] is a multichannel asynchronous scheduled MAC protocol, which inherits the basic asynchronous scheduling operation from AS-MAC and adds back-to-back packet transmissions and multichannel support for high traffic dense WSN. Like AS-MAC [37], MCAS-MAC asynchronously schedules the wakeup time of neighboring nodes, but each node wakes up on its own home channel, which is decided during the initialization phase. Using Hello packets, each node learns neighbors' schedules and home channels [38]. For data packet transmissions, a sender predicts when and on what channel the intended receiver wakes up. After switching to the intended receiver's home channel immediately before the receiver's wakeup time, the

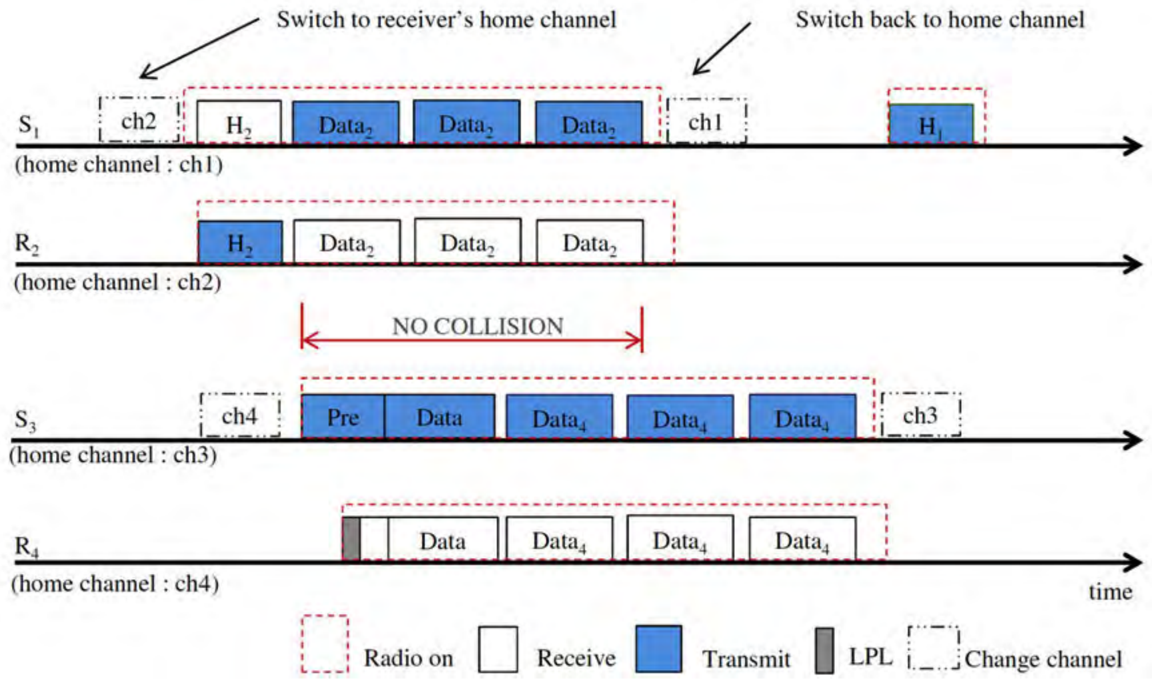


Figure 2.9: Overview of MCAS-MAC operation (adopted from [7]).

sender performs a backoff and attempts to transmit data packets in the queue [39]. In addition to extending the operation of AS-MAC to multiple channels, MCAS-MAC further allow a node to send multiple queued packets at a wakeup time in case of high traffic load by introducing an additional dwell time after each data packet reception [39]. With the use of multiple orthogonal channels and additional dwell time, MCAS-MAC allows for nearly collision free back to back packet transmission, resulting in improved packet delivery ratio and delivery latency without an increase in energy consumption [7]. MCAS-MAC extends the basic operations of AS-MAC with multichannel support. Due to the power consumption and the cost constraints, many sensor nodes are equipped with a single transceiver. Therefore, MCAS-MAC is designed for nodes with a single half-duplex transceiver. As shown in figure 2.9, every node in MCAS-MAC has its own home channel that is chosen during the initialization phase described in Section 2.3.6. Every node sends a Hello message every Hello interval. The Hello message contains a Src (the MAC address of the source node transmitting the message), Wakeup Interval, Hello Interval and Home Channel (the source node's home channel). Each node receiving the Hello message will save all the information in the Hello message with the local clock time when the start symbol of the Hello is detected into its neighbor table [7].

2.5 Summary

Fundamental issues of MAC design and related works are described in this chapter. Issues related to MAC protocol design and different existence MAC protocols are also described in this chapter. A detailed analysis of various MAC protocols used in WSNs, including their operating principle as well as their benefits and shortcomings, is elucidated in Chapter 2.

Chapter 3

A Low Latency Multichannel Asynchronous MAC Protocol

3.1 Introduction

Wireless Sensor Networks (WSN) consist of a large number of battery-powered sensors are distributed within an area of interest in order to track, measure and monitors various important events. Energy saving is an essential aspect of extending life time of wireless sensor networks [10]. WSNs constitute a special class of wireless data communication networks. A node (called a sensor node) in a wireless sensor network is a low cost, resource constrained device. Sensor nodes are typically deployed in large number (hence the requirement to be low cost), and are often positioned randomly. Sensor nodes are generally battery powered. In many applications they are placed in inaccessible locations, making battery replacement impractical. As a consequence, energy efficiency is an important requirement in a medium access control protocol for most wireless sensor networks. Many MAC protocols for wireless sensor networks have been proposed in the recent years. Most of these protocols have energy conservation as an objective. The pattern of energy used in the sensor nodes, however, depends on the nature of the application. As the range of applications which use WSNs is large and diverse, the proposed protocols display much diversity. Most of these protocols use either a contention based mechanism or a time schedule or a combination of the two for accessing the shared medium. A protocol proposed by Michael Buettner, named X-MAC, is a robust MAC protocol for wireless sensor networks. Owing to its success in significant reduction in energy consumption and its robustness, X-MAC has been used in many WSNs. Although these protocols effectively reduce energy consumption, it still suffer from hidden node problem and high latency issue.

3.2 Proposed MX-MAC Protocol

To achieve the higher throughput, a Multichannel asynchronous X-MAC (MX-MAC) protocol for Wireless Sensor Network based on X-MAC protocol have been designed. MX-MAC protocol utilizes a chain of short preamble as like as X-MAC protocol in addition the proposed protocol use two separate channel control and data channel [40] to avoid hidden terminal problem and to efficiently manage the limited resources in terms of node energy consumption and packet delay. The key design is to abolish the hidden node problem of X-MAC scheme by transmitting preamble packets in the control channel by the receiver node whenever it receives data from the sender node.

3.2.1 MX-MAC Short Preamble Packet Format

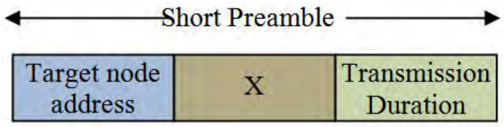
The proposed protocol use two types of short preamble:

1. Sender short preamble and

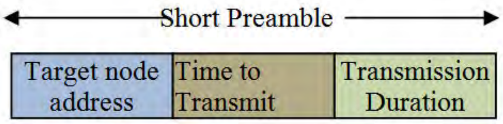
2. Receiver short preamble.

3.2.1.1 Sender short preamble

The sender short preamble as like as X-MAC protocol, where preamble contain ‘Target node address’ with additionally transmission duration field. The figure 3.1(a) shows the sender node short preamble packet format.



(a) Sender short preamble packet format



(b) Receiver short preamble packet format

Figure 3.1: MX-MAC short preamble packet format.

3.2.1.2 Receiver short preamble

The receiver short preamble embeds 'Time to Transmit' and 'Transmission Period' information in addition to 'Target node address' information in the preamble packets. The function of the 'Time to Transmit' field is to make the neighbor nodes aware that a data transmission is going on and the data packet being received at data channel. After that it will be forwarded to the neighbor node with same ID as mentioned in 'Target node address'. 'Transmission Period' field defines the required period of time for the next hop data transmission. Based on this information they can go to sleep mode to conserve energy very effectively. The figure 3.1(b) shows the receiver node short preamble packet format.

3.2.2 Multichannel Communication Process of MX-MAC Protocol

MX-MAC is an asynchronous protocol for this reason all the nodes maintain their own unsynchronized sleep schedules. When receiver send sender preamble with next-hope receiver address the neighbor of the receiver never try to send any data thus solve the hidden terminal problem. With help of the next-hope receiver address the neighbor of the receiver can send early ACK and can get ready channel for next-hop

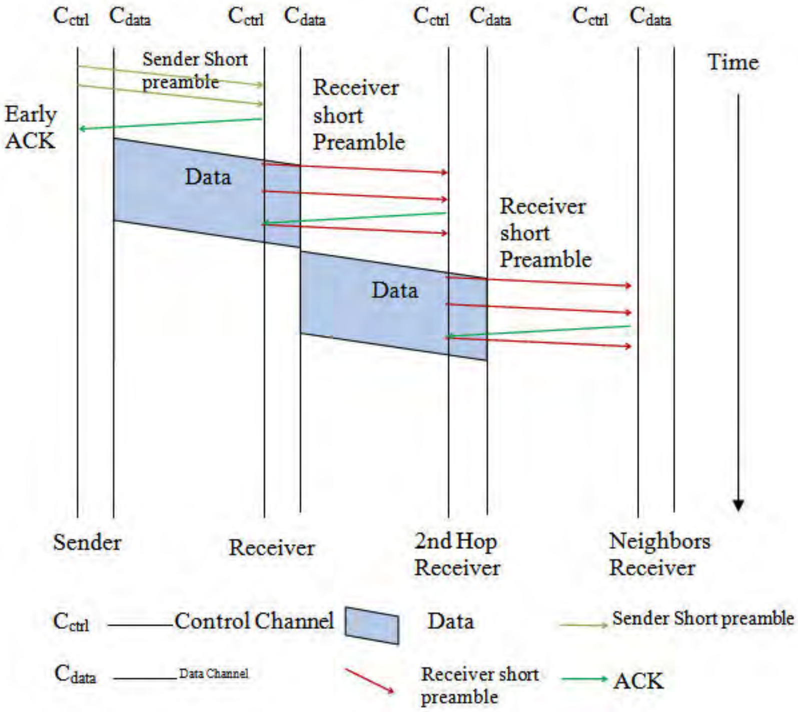


Figure 3.2: Multichannel communication process of MX-MAC protocol.

transmission which saves lot of time and energy. The multichannel communication process of MX-MAC Protocol shows in figure 3.2.

3.2.3 Flow Chart of MX-MAC Sender

In this protocol, when a node wants to send data, it first performs Clear Channel Assessment (CCA) in the control channel as well as in the data channel to keep away from any possible collision with the ongoing transmission. If both data and control

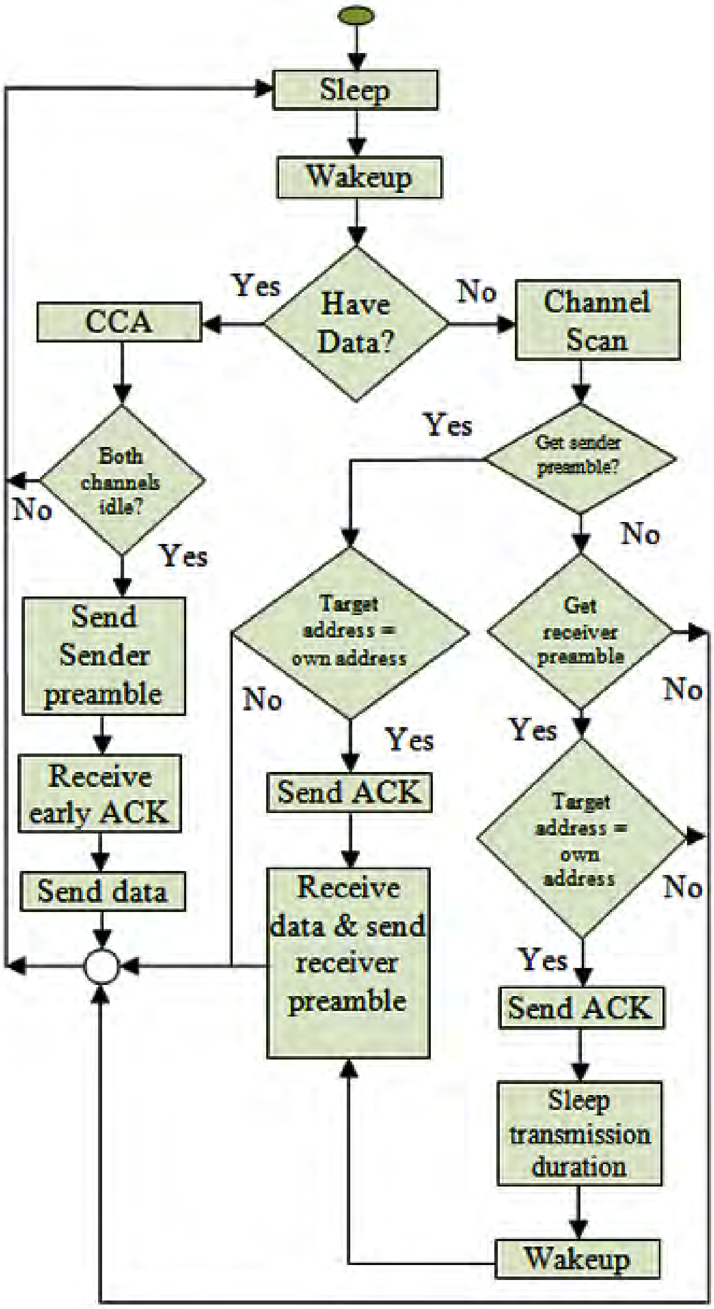


Figure 3.3: Flow chart of MX-MAC sender.

channel are idle then sender send short preamble by control channel with pauses into the series of short preamble packets, creating a strobed preamble, which enables the target receiver to shorten the strobed preamble via an early acknowledgement, thereby achieving additional energy saving at both the sender and receiver, as well as a reduction in per-hop latency. If sender received early acknowledge (ACK), it instantly ends sending short preamble and switch to data channel for sending data packets. If the destination node gets any preamble packet with its own address in the 'Target address' field, it sends an ACK packet by control channel and turns on its data channel to received data from sender node. The whole receiving period the receiver switched on both its data and control channel and received data by data channel and sends preamble packet with its next hop receiver address by control channel at a time. In this way the receiver preamble can restricts all the neighbor nodes of the receiver from transmitting any data for the receiver which eliminates the hidden terminal problem. The sender node details shown by flowchart in the following Figure 3.3.

3.2.4 Graphical Representation of MX-MAC Protocol

When a neighbor node of receiver awakes, it first scans the medium. If sensor node detects any preamble then the node decides is it the target node or not. If target node then send early ACK and immediately it goes to sleep. In this way, neighbor node of the receiver node can easily sense from the receiver's preamble that a data transmission is going on with the receiver end, even the node are not within the range of the sender node. Neighbor of the receiver node with same address as the 'Target address' in preamble field can understand that the data packet currently being received by the receiver will be forwarded to me after completing the 'Transmission Period' of time. So, it may go to sleep mode accordingly and will wake up just before the data packet is forwarded to it by the receiver node. Therefore receiver node will not require sending preamble packets before it forwards the received data packet to its next hop receiver. This significantly reduces the number of collisions due to the hidden node problem that dominantly exists in X-MAC protocol. Figure 3.4 shows the graphical representation of MX-MAC protocol. Here sender wants to send data, first the sender node send short preamble with desired destination

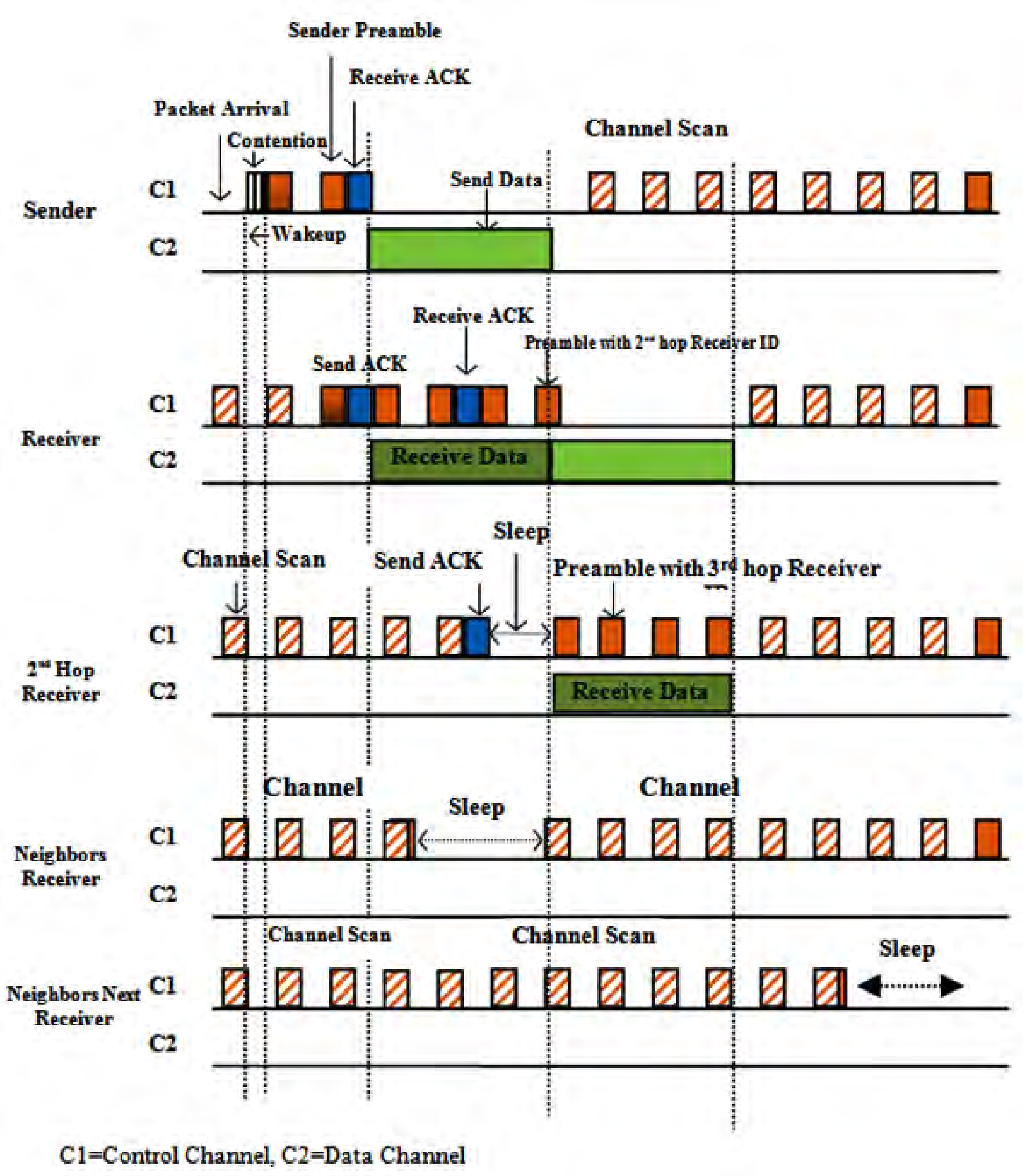


Figure 3.4: Graphical representation of MX-MAC protocol.

address, and receiver send early ACK. After receiving ACK send start sending data, at this time receiver start sending receiver short preamble by control channel until end of receiving data.

3.3 Summary

In this chapter, a multichannel asynchronous MX-MAC protocol have been proposed to reduce the impact of collisions of X-MAC protocol, a popular asynchronous MAC protocol for WSNs, as well as to reduce the average packet delay, thereby improving the lifetime and QoS of sensor networks. Furthermore, it is shown that the average packet delay can be effectively reduced by using “Next Receiver ID” information in the receiver’s preamble packets.

Chapter 4

Analytical Modeling of Proposed MAC Protocol

4.1 Analytical Modeling of MAC Protocols

A lot of methods have been prepared to calculate the performance of various wireless sensor networks MAC protocols. Most of the performance evaluations are obtained from analytical model. However, simulations are usually time consuming and require a large number of runs to obtain statistically significant results. Latency, throughput and available resources, are difficult to draw general or quantitative conclusions on the performance of a MAC protocol. Therefore, analytical models are needed to provide insight into the performance of MAC protocols. Analytical models have been proposed to evaluate the performance of a specific MAC protocol [41]. For example, Bianchi proposed a Markov model to analyze the saturation throughput of IEEE 802.11. Unlike these previous approaches, our model can be used to obtain throughput, delay and energy consumption for asynchronous duty-cycled MAC protocols.

4.1.1 Markov Model of a WSN

Markov-chains are certain discrete space stochastic processes which are amenable for analysis and hence are very popular for analysis, traffic characterization and modeling of queuing and telecommunications networks and systems. They can be classified into two groups: (a) Discrete- time Markov-chains and (b) Continues time Markov-chains. A discrete-time Markov-chain is a discrete-time stochastic process X_n, n = 0, 1, 2, . . . with the Markov property; namely, that at any point in time n, the future evolution of the process is dependent only on the state of the process at time n, and is independent of the past evolution of the process. The state of the process can be a scalar or a vector. A continuous-time Markov-chain is a continuous-time stochastic process X_t. At any point in time t, X_t describes the state of the process which is discrete and considered only continuous-time Markov-chain where X_t takes values that are nonnegative integer. The time between changes in the state of the process is exponentially distributed.

4.1.2 Markov Queuing Model for Duty- Cycled Nodes

The Markov model has a finite number of states, each of which represents a different status of a node, i.e., a different queue length, at the wake-up instant of a cycle. A node may change status cycle by cycle, corresponding to the transition from one state to another in the Markov model.

4.1.3 Protocol-Specific Performance Analysis

Markov queuing model holds for any duty-cycled nodes with a fixed cycle length. It provides a relationship, between the stationary probability π_0 of the empty-queue state and the probability p for each node to win the contention in a cycle. However, another relationship between π_0 and p is needed together to solve for both π_0 and p, and finally using these values to obtain the throughput, delay, and energy consumption.

4.1.3.1 System Model

A certain number of nodes create a fully-connected network. The nodes are homogeneous in initial energy, power and communication capabilities. Every node has a finite queue to buffer the incoming DATA packets. DATA packet arrivals at different nodes are independent, and they arrive at the nodes with the same distribution. A node randomly selects one of its neighbors as the destination to transmit DATA packets for a certain time [42].

4.1.3.2 Throughput Analysis

Throughput is defined as the amount of data successfully delivered within a unit time. Since the protocols work in a duty-cycled fashion, the throughput can be calculated within a cycle time. Therefore, the throughput of the system can be calculated as follows [42].

$$THR = \frac{N.\big(1 - \pi_0\big).P_s.S}{T} \tag{4.1.3.1}$$

Where N nodes in the network, a DATA packet has size S , a cycle has T slots, the length of a slot is τ , and the probability for each node to successfully deliver a DATA packet in a cycle is p_s , the stationary probability π_0 of the empty-queue state.

4.1.3.3 Delay Analysis

The delay of a DATA packet can be divided into two parts. The first part is the queuing delay D_Q, which is defined as the time interval from when a DATA packet joins the queue at the tail to the DATA packet becoming the head of the queue. The second part is the contending delay D_C, which is defined as the time interval from when the DATA packet is at the head of the queue to when the DATA packet is transmitted and hence removed from the queue. Therefore,

$$D = D_Q + D_C \tag{4.1.3.2}$$

The queuing delay D_Q of a DATA packet is the time that the DATA packet must wait in the queue until all the DATA packets in front of it finish contending for the media. Specifically, a newly joined DATA packet has to wait for a contending delay D_C for each of the DATA packets that are in front of it but behind the head of the queue. However, the newly joined DATA packet may arrive at the queue when the DATA packet at the head of the queue (if the queue is not empty) has already started contending for the media. Hence, for the DATA packet at head of the queue, the newly joined DATA packet has to wait on average for a half of the contending delay D_C of a DATA packet. The queuing delay D_Q of a DATA packet can be calculated as described in [42].

$$D_Q = \sum_{i=0}^{Q-1} \Big(max\Big(0, i - 0.5\Big).\pi_i / \Big(1 - \pi_Q\Big)\Big) \tag{4.1.3.3}$$

The contending delay D_C of a DATA packet can be calculated according to the stationary probability π of our proposed Markov model. A node with a DATA packet to send contends for the media once in a cycle, until the node finally wins the contention. For each contention, the node has a probability of p to win, and a probability of $1 - p$ to lose. Given a cycle length of T

$$D_C = T \sum_{i=0}^{\infty} \Big(i + 1\Big).p.\Big(1 - p\Big)^i \tag{4.1.3.4}$$

Plugging (4.1.3.3) and (4.1.3.4) into (4.1.3.2), the delay of a DATA packet can be obtained.

4.1.3.4 Energy Consumption Analysis

Considering the duty-cycle MAC protocols, the energy consumption per second P of a node can be obtained by calculating the energy consumption E of a node in a cycle divided by the cycle length T, i.e. [42],

$$P = E/T \tag{4.1.3.5}$$

The energy consumption per second P of a node can also be used to estimate the lifetime of the network. Assuming each node in the network has an initial energy E_{init}, the lifetime of the network can be represented as

$$L = E_{init}/P \tag{4.1.3.6}$$

4.2 Analytical Modeling of Existing Protocols

4.2.1 Performance Analysis of X-MAC Protocol

Performance of X-MAC Protocol will examine in terms of throughput, delay and energy consumption. A certain number of nodes create a fully-connected network. The nodes are homogeneous in initial energy, power and communication capabilities. Every node has a finite queue to buffer the incoming DATA packets. DATA packet arrivals at different nodes are independent, and they arrive at the nodes with the same distribution.

4.2.1.1 Throughput Analysis of X-MAC

Throughput is defined as the amount of data successfully deliver. The existing X-MAC protocol works in a duty-cycled fashion, so, the throughput will be calculated within a cycle time.

$$THR_{X-MAC} = \frac{N.(1-\pi_0).P_s.S}{T} \tag{4.2.1.1}$$

N, S and T are known variable, only π_0 and P_s are need to be calculated. π_0 and P_s are determined by Markov model and the media access rule of X-MAC. The Markov model has a state space 0,1. . . .Q and a transition matrix. The stationary probability of the empty queue state can be obtain from Markov model which is shown in Figure 4.1. This Markov model has Q+1 states, each of which, from left to right, corresponds to 0 packets in the queue,1 packet in the queue, to Q packets in the queue (full queue). Specifically, when the queue is not empty, a node will attempt to access the media to transmit a DATA packet. A DATA packet is removed from the queue either when it is transmitted successfully, or when it encounters a collision as no retransmission is allowed. A DATA packet is dropped when the queue overflows. Hence, the transition probabilities from one state to another can be described as follows.

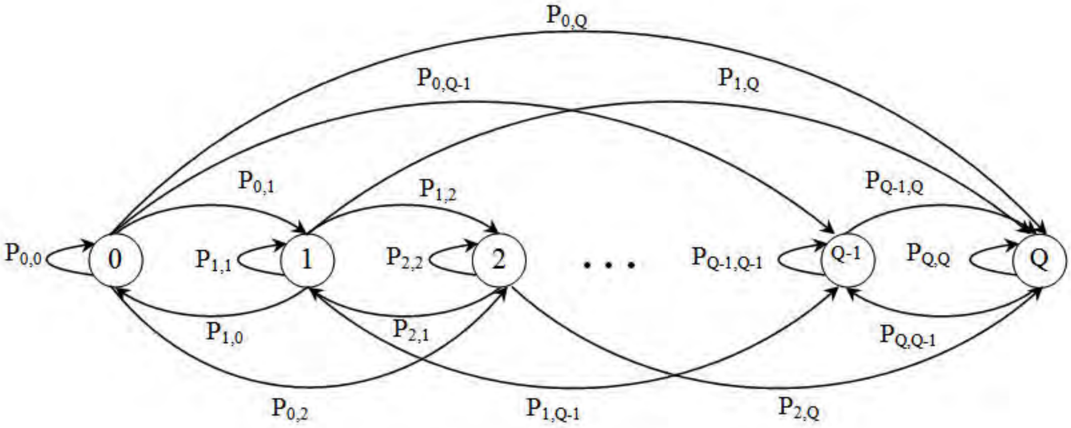


Figure 4.1: Markov model for X-MAC protocol (adopted from [8]).

$$P_{0,i} = A_i, i = 0...Q-1 \tag{4.2.1.2}$$

$$P_{0,Q} = A_{\geq Q} \tag{4.2.1.3}$$

$$P_{i,j} = p.A_0, i = 1...Q \tag{4.2.1.4}$$

$$P_{i,j} = p.A_{j-i+1} + (1-p).A_{j-i}, i = 1...Q-1, j = i...Q-1 \tag{4.2.1.5}$$

$$P_{i,Q} = p.A_{\geq Q-i+1} + (1-p).A_{Q-i}, i = 1...Q \tag{4.2.1.6}$$

$$P_{i,j} = 0, i = 2...Q, j = i...i-2 \tag{4.2.1.7}$$

where, A_i is the probability that i DATA packets arrive at a node in a cycle, $A_{\geq i}$ is the probability that no less than i DATA packets arrive at a node in a cycle, and p is the probability for a node to transmit a DATA packet in a cycle. Specifically, Equations (4.2.1.2) and (4.2.1.3) describe the fact that all the transitions from the empty-queue state to a non-empty-queue state depend only on new packet arrivals. Equations (4.2.1.4) and (4.2.1.7) describe the fact that a node can only transmit one DATA packet per cycle with a probability p, and the probability of having one packet less in the queue equals the probability of winning the contention times the probability of no packet arrivals in a cycle. Moreover, (4.2.1.5) and (4.2.1.6) describe the fact that the probability of having a non-decreasing queue can be divided into two parts depending on whether the oldest DATA packet in the queue wins the contention (first term) or not (second term). Finally, (4.2.1.3) and (4.2.1.6) show that packets are dropped when the queue overflows.

A. Stationary probability (empty queue state π_0) calculation for X-MAC

The Existing Markov model with state space S = 0,1, . . .,Q and transition matrix P has a unique stationary distribution $\pi = \pi_0, ...\pi_Q$ since the Markov model is irreducible and aperiodic. Therefore, $\pi_{i \geq 0}$ for any $s_i \in S$

$$\sum_{s_i \in S} \pi_i = 1 \tag{4.2.1.8}$$

$$\pi = P.\pi \tag{4.2.1.9}$$

In X-MAC protocol, assume packet arrival information (A_k, and $A_{\geq k}$) is known, the probability p for each node to win the contention becomes the only variable in the transition matrix P. Since π_0 is the unique solution for (4.2.1.8), for any $s_i \in S$ can be represented as a function of p. Specifically, let function f(.) describe the relationship between π_0 and p, i.e.,

$$\pi_0 = f(p) \tag{4.2.1.10}$$

According to (4.2.1.10), for a given probability p for each node to win the contention, the stationary probability π_0 of the empty-queue state can be obtained using the existing Markov model of X-MAC.

B. Probability Success (P_s) Calculation by Media Access Rules of X-MAC

Since the number of nodes in the network N, the X-MAC layer DATA packet size S, and the length of a cycle T are known, once π_0 is solved by (4.2.1.10), the only unknown variable in (4.2.1.1) is the probability p_s for each node to successfully transmit a DATA packet. When π_0 is known, p_s can be obtained according to the media access rules of the investigated protocol. Similar way obtain p, every node has a packet to send with a probability of $1 - \pi_0$ in a cycle. For a given π_0, the probability p_s for a node to successfully transmit a DATA packet is determined by the number of nodes in the network N and the manner in which nodes compete with each other. The relationship between π_0 and p_s can be described as:

$$p_s = h(\pi_0) \tag{4.2.1.11}$$

Plugging (4.2.1.11) into (4.2.1.1), the throughput of the network can be determined.

$$P = P_S + P_C \tag{4.2.1.12}$$

Data transmission probability P can be define as a addition of Probability of success P_s and collision P_c . The probability of Success depend of status of the queue and channel free probability.

$$P_s = P_{avs} + P_{free} \tag{4.2.1.13}$$

The average probability of success in data channel can be calculated by the following equation:

$$P_{avs} = \sum_{t=1}^{T} \frac{1}{T} \Bigg(\sum_{w=1}^{C_w} \Bigg(\sum_{T_a=0}^{N-1} \Bigg(\sum_{W_i=0}^{T_a} \Bigg(\sum_{S_d=0}^{W_i} P_r \Big(A|affect, \overline{empty} \Big) \\ P_r \Big(B|wake, \overline{empty} \Big) P_r \Big(C|data, \overline{empty} \Big) \Bigg) \Bigg) \Bigg) \Bigg) \tag{4.2.1.14}$$

$$P_r \Big(A|affect, \overline{empty} \Big) = \binom{N-1}{T_a} \left(\frac{W_m}{T} \right)^{T_a} \left(\frac{T - W_m}{T} \right)^{N-1-T_a} \tag{4.2.1.15}$$

$$P_r \Big(B|wake, \overline{empty} \Big) = \binom{T_a}{W_i} \left(\frac{1}{W_m} \right)^{W_i} \left(\frac{W_m - 1}{W_m} \right)^{T_a - W_i} \tag{4.2.1.16}$$

$$P_r \Big(C|data, \overline{empty} \Big) = \binom{W_i}{S_d} \Big(\pi_0 \Big)^{W_i - S_d} \Big(1 - \pi_0 \Big)^{S_d} \left(\frac{T-1}{W_m} \right)^{S_d} \tag{4.2.1.17}$$

$P_r \Big(A|affect, \overline{empty} \Big)$= Probability of Event A where number of T_a nodes wakeup at contention widow affected area.
$P_r \Big(B|wake, \overline{empty} \Big)$= Probability of Event B where number of nodes W_i wakeup at i^{th} slot of contention window.
$P_r \Big(C|data, \overline{empty} \Big)$= Probability of Event C where number of nodes S_d have some data but they do not hamper the transmission on this contention window. Which indicates that a node can successfully transmit a DATA packet if no other nodes wake up at the same time, or some nodes wake up at the same time, but they have no packets to send.

All the above equations has derived by using binomial probability formula. For example, in equation (4.2.1.15), N= Total number of trial, T_a=Number of specific events that wants to obtain, W_m/T = the probability of success and $(T - W_m)/T$= the probability of failure. Probability of collision P_c is depends on a nodes who hamper transmission when some others node transmitting data . Probability of collision:

$$P_c = P_{avc} + P_{free} \tag{4.2.1.18}$$

Average Probability of collision:

$$P_{avc} = \sum_{t=1}^{T} \frac{1}{T} \Bigg(\sum_{w=1}^{C_w} \Bigg(\sum_{T_a=0}^{N-1} \Bigg(\sum_{W_i=0}^{T_a} \Bigg(\sum_{S_d=0}^{W_i} P_r\Big(A|affect, \overline{empty}\Big) \\ P_r\Big(B|wake, \overline{empty}\Big) P_r\Big(C|coll, \overline{empty}\Big)\Bigg)\Bigg)\Bigg)\Bigg) \tag{4.2.1.19}$$

$$P_r\Big(A|affect, \overline{empty}\Big) = \binom{N-1}{T_a} \left(\frac{W_m}{T}\right)^{T_a} \left(\frac{T-W_m}{T}\right)^{N-1-T_a} \tag{4.2.1.20}$$

$$P_r\Big(B|wake, \overline{empty}\Big) = \binom{T_a}{W_i} \left(\frac{1}{W_m}\right)^{W_i} \left(\frac{W_m-1}{W_m}\right)^{T_a-W_i} \tag{4.2.1.21}$$

$$P_r\Big(C|data, \overline{empty}\Big) = \binom{W_i}{S_d} \Big(\pi_0\Big)^{W_i-S_d} \Big(1-\pi_0\Big)^{S_d} \left(\frac{T-t-1}{W_m}\right)^{S_d} \tag{4.2.1.22}$$

The average free channel probability $P_r(free)$ at given time is the ratio of duration of channel free and entire channel free and busy time.

$$P_r\Big(feee\Big) = \frac{E_{free}}{E_{free} - E_{busy}} \tag{4.2.1.23}$$

The probability of a free channel, $P_r(free)$, can be obtained if the following two parameters are known. (1) the average length of a free channel between two transmissions over the media, E_{free}, and (2) the average length of a busy channel between the two chunks of a free channel, E_{busy}.

The channel free probability:

$$P_{free} = \sum_{i=0}^{N-1} \sum_{j=1}^{N-i} \sum_{k=1}^{j} \binom{N}{i} \left(\frac{t}{T}\right)^{i} \left(\frac{T-W_m}{T}\right)^{N-1-i} \binom{N-i}{j} \left(\frac{1}{W_m}\right)^{j} \\ \left(\frac{W_m-1}{W_m}\right)^{N-i-j} \binom{j}{k} \Big(1-\pi_0\Big)^{k} \pi_0^{j-k} \left(\frac{T-t-1}{W_m}\right)^{j-k} \tag{4.2.1.24}$$

The expected channel free probability (E_{free}) depends on the probability of the event that at the time instant when a transmission ends and the interval of free channel begins. To calculate E_{free} considered the time instant of channel is free for n cycles and t time slots, therefore $n.T + t$ is the duration the of free channel and probability of this event is P_{free}. Equation (4.2.1.25) shows the calculation of E_{free}

where k out of j nodes which are competing for the media at the same time when there is no backoff mechanism. In this case there is high probability of collision. By using backoff mechanism the probability that one node send data which given by equation (4.2.1.24) can be used to find the probability P_{free} .

The average channel free probability:

$$E_{free} = \sum_{n=0}^{\infty} \sum_{t=1}^{T} \left(nT + t \right) P_{free} \tag{4.2.1.25}$$

Equation (4.2.1.26) illustrate the probability of the channel being busy between two free channel intervals due to successful transmission. To calculate the average duration of time interval for which the channel is busy between two free channel intervals E_{Busy}, first find the probability P_{busy}^{succ} of the event when the transmission is successful between two free channel intervals and the probability P_{busy}^{coll} of the event when there is collision between two free channel intervals.

$$E_{busy} = \sum_{n=0}^{\infty} \sum_{t=1}^{T} \left(\left(\frac{T}{2} + L_{DATA} \right) P_{busy}^{succ} + T.P_{busy}^{pre} \right) \tag{4.2.1.26}$$

The probability of busy channel between two free channel intervals due to collisions of short preambles is given by Equation (4.2.1.28). Here are k out of j nodes which are competing for the medium at the same time when there is no backoff mechanism will cause collision.

$$\begin{aligned} P_{busy}^{succ} = \sum_{i=0}^{N-1} \sum_{j=1}^{N-i} \sum_{k=1}^{j} \binom{N}{i} \left(\frac{t}{T} \right)^{i} \left(\frac{T - W_m}{T} \right)^{N-1-i} \binom{N-i}{j} \left(\frac{1}{W_m} \right)^{j} \\ \binom{j}{k} \left(1 - \pi_0 \right)^{k} \pi_0^{j-k} \left(\frac{T - t - 1}{W_m} \right)^{j-k} \end{aligned} \tag{4.2.1.27}$$

The average duration of busy channel between two free channel intervals P_{busy}^{coll}, define by the length of successful transmission time duration $(T/2 + L_{Data})$ multiplied by the probability of this event plus time duration of busy channel due to collisions of short preambles sending by control channel.

$$P_{busy}^{coll} = \sum_{i=0}^{N-2}\sum_{j=2}^{N-i}\sum_{k=2}^{j}\binom{N}{i}\left(\frac{t}{T}\right)^{i}\left(\frac{T-W_m}{T}\right)^{N-1-i}\binom{N-i}{j}\left(\frac{1}{W_m}\right)^{j} \binom{j}{k}\left(1-\pi_0\right)^{k}\pi_0^{j-k}\left(\frac{T-t-1}{W_m}\right)^{j-k} \qquad (4.2.1.28)$$

Plugging equations (4.2.1.25) and (4.2.1.26) into (4.2.1.23), obtain $P_r(free)$.Plugging equation (4.2.1.23) and (4.2.1.14) into (4.2.1.13), and (4.2.1.19) and (4.2.1.23) into (4.2.1.18), the probability for each node to successfully transmit a DATA packet p_s, and the probability for each node to encounter a collision p_c are solved as a function of the stationary probability of the empty queue state π_0. According to equation (4.2.1.10), the probability for each node to transmit a DATA packet p can also be obtained as a function of π_0. Let function h(.) describe the relationship between p and π_0

$$p = h(\pi_0) \qquad (4.2.1.29)$$

4.2.1.2 Delay Analysis of X-MAC

The end-to-end delay is defined as the latency from the time when a source node has a data packet ready to send to the time the first packet is received at the target node. There are many factors affecting latency of the network. Factors such as overhearing, over-emitting, collisions, and control packets overhead affect energy saving as well as forwarding delay. On the other hand, low duty-cycling and sleep mode of a radio cause end-to-end delay degradation.

The expected latency for a single hop data exchange is given by [4]:

Lat_{X-MAC} = (duration of preamble + ACK listen) × (expected number of iterations required) + duration to send a data packet

$$= \left(\frac{1}{\frac{R_l-S_p}{R_l+R_s}}\right) \times \left(S_p + S_{al}\right) + S_d \qquad (4.2.1.30)$$

$$= \frac{\left(R_l + R_s\right)}{\left(R_l - S_p\right)} \times \left(S_p + S_{al}\right) + S_d \qquad (4.2.1.31)$$

The total latency for X-MAC protocol is derived as

$$Lat_{X-MAC} = t_{contention} + ENIR \times (t_p + t_{al}) \tag{4.2.1.32}$$

$$ENIR = \frac{\Big(R_l + R_s\Big)}{\Big(R_l - S_p\Big)} \tag{4.2.1.33}$$

ENIR=Expected Number of Iteration Required

4.2.1.3 Energy Consumption Analysis of X-MAC

The average energy consumption can be calculated in terms of the duration of sender and receiver sleep, listing time and transmission periods. The total energy consumption for the X-MAC is:

$$E_{(X-MAC)} = E_s + E_r \tag{4.2.1.34}$$

E_s = Expected energy to send a packet and E_r =Expected energy to receive a packet. Expected energy (E_s) to send a packet:
E_s =(preamble energy + energy per ACK listen) × (expected preamble-listen iterations required) +(energy to send packet)

$$= (P_{Tx}S_p + P_{Rx}S_{al})\left(\frac{1}{\frac{R_l - S_p}{R_l + S_p}}\right) + S_d P_{Tx} \tag{4.2.1.35}$$

Table 4.1: Mathematical model variables for X-MAC

Symbol	**Meaning**
S_p	Duration of the sender's preamble
S_{al}	Duration of the Acknowledgement listen
R_l	Receiver listen period
R_s	Receiver sleep period
t_d	Data transmission period for a single packet
R_a	Duration of the receiver's ACK period
R_d	Duration of the receiver's data period

$$= \frac{((P_{Tx}S_p + P_{Rx}S_{al})(R_l + S_p))}{(R_l - S_p)} + S_d P_{Tx} \tag{4.2.1.36}$$

The expected number of preamble listen iterations required, giving:

$$E_S = \frac{(P_{Tx}S_p + P_{Rx}Sal)}{\left(1 - (1 - P_d R_{qpl})(1 - \frac{R_l - S_p}{R_l + R_s})\right)} + S_d P_{Tx} \tag{4.2.1.37}$$

The expected energy to receive a packet is given by: E_r =(listen cycle energy + sleep cycle energy) *(expected iterations for a preamble to arrive)+(energy to send an ACK)+(energy to receive packet)

$$\frac{(P_s R_s + P_{R_x} R_l)}{(1 - (1 - P_d(t)))^{(R_l + R_s)}} + P_{Tx} R_a + R_d P_{Rx} \tag{4.2.1.38}$$

4.2.2 Performance Analysis of X-MAC/CA protocol

Based on the general description of X-MAC/CA protocol, this section derives important equations necessary for its throughput, latency and energy consumption.

4.2.2.1 Throughput Analysis of X-MAC/CA

The throughput of X-MCA/CA protocol will be calculated by the following equation:

$$THR_{X-MAC/CA} = \frac{N.\big(1 - \pi_0\big).P_s.S}{T} \tag{4.2.2.1}$$

The probability of Success:

$$P_s = P_{avs}.P_{free} \tag{4.2.2.2}$$

The average probability of success in data channel can be calculated by the following equation:

$$\begin{aligned} P_{avs} = \sum_{t=1}^{T} \frac{1}{T} \Bigg(\sum_{w=1}^{C_w} \Bigg(\sum_{T_a=0}^{N-1} \Bigg(\sum_{W_i=0}^{T_a} \Bigg(\sum_{S_d=0}^{W_i} \\ P_r\Big(A|affect, \overline{empty}\Big).P_r\Big(B|wake, \overline{empty}\Big). \\ P_r\Big(C|data, \overline{empty}\Big)\Bigg)\Bigg)\Bigg)\Bigg) \end{aligned} \tag{4.2.2.3}$$

$$P_r\Big(A|affect, \overline{empty}\Big) = \binom{N-1}{T_a} \left(\frac{W_m}{T}\right)^{T_a} \left(\frac{T - W_m}{T}\right)^{N-1-T_a} \tag{4.2.2.4}$$

$$P_r\Big(B|wake, \overline{empty}\Big) = \binom{T_a}{W_i}\left(\frac{1}{W_m}\right)^{W_i}\left(\frac{W_m-1}{W_m}\right)^{T_a-W_i} \tag{4.2.2.5}$$

$$P_r\Big(C|data, \overline{empty}\Big) = \binom{W_i}{S_d}\Big(\pi_0\Big)^{W_i-S_d}\Big(1-\pi_0\Big)^{S_d}\left(\frac{t_i-1}{W_m}\right)^{S_d} \tag{4.2.2.6}$$

Equation (4.2.2.6) presents the probability that s_d nodes out of w_i have some data but their backoff timer expires after t so that they do not hamper the transmission at t while the other nodes (w_i–s_d) have no data to send. Probability of collision P_c is depends on a nodes who hamper transmission when some others node transmitting data and channel free probability.

Probability of collision:

$$P_c = P_{avc}.P_{free} \tag{4.2.2.7}$$

Average Probability of collision:

$$P_{avc} = \sum_{t=1}^{T}\frac{1}{T}\left(\sum_{w=1}^{C_w}\left(\sum_{T_a=0}^{N-1}\left(\sum_{W_i=0}^{T_a}\left(\sum_{S_d=0}^{W_i} P_r\Big(A|affect, \overline{empty}\Big).\right.\right.\right.\right.$$
$$\left.\left.\left.\left.P_r\Big(B|wake, \overline{empty}\Big).P_r\Big(C|coll, \overline{empty}\Big)\right)\right)\right)\right) \tag{4.2.2.8}$$

$$P_r\Big(A|affect, \overline{empty}\Big) = \binom{N-1}{T_a}\left(\frac{W_m}{T}\right)^{T_a}\left(\frac{T-W_m}{T}\right)^{N-1-T_a} \tag{4.2.2.9}$$

$$P_r\Big(B|wake, \overline{empty}\Big) = \binom{T_a}{W_i}\left(\frac{1}{W_m}\right)^{W_i}\left(\frac{W_m-1}{W_m}\right)^{T_a-W_i} \tag{4.2.2.10}$$

$$P_r\Big(C|coll, \overline{empty}\Big) = \binom{W_i}{S_d}\Big(\pi_0\Big)^{W_i-S_d}\Big(1-\pi_0\Big)^{S_d}\left(\frac{W_0-t_i-1}{W_m}\right)^{S_d} \tag{4.2.2.11}$$

The average free channel probability at given the is the ratio of duration of channel free and entire channel time.

$$P_r\Big(feee\Big) = \frac{E_{free}}{E_{free}-E_{busy}} \tag{4.2.2.12}$$

The probability of a free channel, $P_r(free)$, can be obtained if the following two parameters are known. (1) the average length of a free channel between two transmissions over the media, E_{free}, and (2) the average length of a busy channel between the two chunks of a free channel, E_{busy}.

4.2.2.2 Delay Analysis of X-MAC/CA

The expected latency for a single hop data exchange is given by:

$Lat_{X-MAC/CA}$ = (backoff time + duration of preamble + ACK listen) × (expected number of iterations required) + duration to send a data packet

$$= \left(\frac{1}{\frac{R_l - S_p}{R_l + R_s}}\right) \times \left(t_i + S_p + S_{al}\right) + S_d \tag{4.2.2.13}$$

$$= \frac{\left(R_l + R_s\right)}{\left(R_l - S_p\right)} \times \left(t_i S_p + S_{al}\right) + S_d \tag{4.2.2.14}$$

For n packets with h hop count, the total latency for the X-MAC/CA protocol is derived as:

$$Lat_{X-MAC/CA} = t_{contention} + ENIR \times (t_i + t_p + t_{al}) + t_d \times n \times h \tag{4.2.2.15}$$

$$ENIR = \frac{\left(R_l + R_s\right)}{\left(R_l - S_p\right)} \tag{4.2.2.16}$$

Table 4.2: Delay analysis variables for X-MAC/CA

Symbol	**Meaning**
t_i	Backoff time
S_p	Duration of the sender's preamble
S_{al}	Duration of the Acknowledgement listen
R_l	Receiver listen period
R_s	Receiver sleep period
t_d	Data transmission period for a single packet
R_a	Duration of the receiver's ACK period
R_d	Duration of the receiver's data period

ENIR=Expected Number of Iteration Required

4.2.2.3 Energy Consumption Analysis of X-MAC/CA

The total energy consumption for the X-MAC/CA is:

$$E_{(X-MAC/CA)} = E_s + E_r \tag{4.2.2.17}$$

E_s = Expected energy to send a packet and E_r =Expected energy to receive a packet

Expected energy (Es) to send a packet:

E_s = (preamble energy + energy per ACK listen) × (expected preamble-listen iterations required) +(energy to send packet)

$$= (P_{Tx}S_p + P_{Rx}S_{al})\left(\frac{1}{\frac{R_l - S_p}{R_l + S_p}}\right) + S_d P_{Tx} \tag{4.2.2.18}$$

$$= \frac{((P_{Tx}S_p + P_{Rx}S_{al})(R_l + S_p))}{(R_l - S_p)} + S_d P_{Tx} \tag{4.2.2.19}$$

Table 4.3: Energy consumption analysis variables for X-MAC/CA

Symbol	**Meaning**
P_{Tx}	Power required to transmit
PRx	Power required to receive
P_s	Power required to sleep
S_p	Duration of the sender's preamble
S_{al}	Duration of the sender's acknowledgment
S_d	Duration of the sender's data transmission periods
R_l	Receiver listen periods

The expected number of preamble listen iterations required, giving:

$$E_S = \frac{(P_{Tx}S_p + P_{Rx}Sal)}{\left(1 - (1 - P_d R_{qpl})(1 - \frac{R_l - S_p}{R_l + R_s})\right)} + S_d P_{Tx} \tag{4.2.2.20}$$

The expected energy to receive a packet is given by: E_r =(listen cycle energy + sleep cycle energy) *(expected iterations for a preamble to arrive)+(energy to send an ACK)+(energy to receive packet)

$$\frac{(P_s R_s + P_{R_x} R_l)}{(1 - (1 - P_d(t)))^{(R_l + R_s)}} + P_{Tx} R_a + R_d P_{Rx} \tag{4.2.2.21}$$

4.2.3 Performance Analysis of ZeroMAC Protocol

4.2.3.1 Throughput Analysis of ZeroMAC

The throughput of ZeroMAC protocol :

$$THR_{ZeroMAC} = \frac{N.\big(1 - \pi_0\big).P_s.S}{T} \tag{4.2.3.1}$$

The stationary probability π_0 of the empty-queue state can be obtained using the proposed Markov model. Probability success (Ps) calculation by Media Access Rule of ZeroMAC. The probability of Success depend of status of the queue and channel free probability.

$$P_s = P_{avs}.P_{free} \tag{4.2.3.2}$$

The average probability of success in data channel can be calculated by the following equation:

$$\begin{aligned} P_{avs} = \sum_{t=1}^{T} \frac{1}{T} \Bigg(\sum_{w=1}^{C_w} \Bigg(\sum_{T_a=0}^{N-1} \Bigg(\sum_{W_i=0}^{T_a} \Bigg(\sum_{S_d=0}^{W_i} P_r\Big(A|affect, \overline{empty}\Big) \\ P_r\Big(B|wake, \overline{empty}\Big) P_r\Big(C|data, \overline{empty}\Big)\Bigg)\Bigg)\Bigg)\Bigg) \end{aligned} \tag{4.2.3.3}$$

$$P_r\Big(A|affect, \overline{empty}\Big) = \binom{N-1}{T_a} \left(\frac{W_m}{T}\right)^{T_a} \left(\frac{T - W_m}{T}\right)^{N-1-T_a} \tag{4.2.3.4}$$

$$P_r\Big(B|wake, \overline{empty}\Big) = \binom{T_a}{W_i} \left(\frac{1}{W_m}\right)^{W_i} \left(\frac{W_m - 1}{W_m}\right)^{T_a - W_i} \tag{4.2.3.5}$$

$$P_r\Big(C|data,\overline{empty}\Big) = \binom{W_i}{S_d}\Big(\pi_0\Big)^{W_i-S_d}\Big(1-\pi_0\Big)^{S_d}\Big(\frac{t_{gt}-1}{W_m}\Big)^{S_d} \tag{4.2.3.6}$$

Equation (4.2.3.6) presents the probability that s_d nodes out of w_i have some data but their guard time t_{gt} timer expires after t so that they do not hamper the transmission at t while the other nodes (w_i–s_d) have no data to send. Probability of collision P_c is depends on a nodes who hamper transmission when some others node transmitting data and channel free probability.

$$P_c = P_{avc}.P_{free} \tag{4.2.3.7}$$

Average Probability of collision:

$$P_{avc} = \sum_{t=1}^{T}\frac{1}{T}\Bigg(\sum_{w=1}^{C_w}\Bigg(\sum_{T_a=0}^{N-1}\Bigg(\sum_{W_i=0}^{T_a}\Bigg(\sum_{S_d=0}^{W_i}P_r\Big(A|affect,\overline{empty}\Big) P_r\Big(B|wake,\overline{empty}\Big)P_r\Big(C|coll,\overline{empty}\Big)\Bigg)\Bigg)\Bigg)\Bigg) \tag{4.2.3.8}$$

$$P_r\Big(A|affect,\overline{empty}\Big) = \binom{N-1}{T_a}\Big(\frac{W_m}{T}\Big)^{T_a}\Big(\frac{T-W_m}{T}\Big)^{N-1-T_a} \tag{4.2.3.9}$$

$$P_r\Big(B|wake,\overline{empty}\Big) = \binom{T_a}{W_i}\Big(\frac{1}{W_m}\Big)^{W_i}\Big(\frac{W_m-1}{W_m}\Big)^{T_a-W_i} \tag{4.2.3.10}$$

$$P_r\Big(C|coll,\overline{empty}\Big) = \binom{W_i}{S_d}\Big(\pi_0\Big)^{W_i-S_d}\Big(1-\pi_0\Big)^{S_d}\Big(\frac{W_0-t_{gt}-1}{W_m}\Big)^{S_d} \tag{4.2.3.11}$$

The average free channel probability $P_r(free)$ at given the is the ratio of duration of channel free and entire channel time.

$$P_r\Big(feee\Big) = \frac{E_{free}}{E_{free}-E_{busy}} \tag{4.2.3.12}$$

The probability of a free channel, $P_r(free)$, can be obtained if the following two parameters are known. (1) the average length of a free channel between two transmissions over the media, E_{free}, and (2) the average length of a busy channel between the two chunks of a free channel, E_{busy}.

4.2.3.2 Delay Analysis of ZeroMAC

The expected latency for a single hop data exchange is given by:

$Lat_{ZeroMAC}$ = (duration of guard time + Receiver Active time + RTS sending time

+ CTS receiving time + ACK listen) × (expected number of iterations required) + duration to send a data packet

$$= \left(\frac{1}{\frac{R_l - S_{RTS}}{R_l + R_s}}\right) \times \left(S_{GT} + R_{act} + S_{RTS} + R_{CTS} + S_{al}\right) + S_d \tag{4.2.3.13}$$

$$= \frac{\left(R_l + R_s\right)}{\left(R_l - S_{RTS}\right)} \times \left(S_{GT} + R_{act} + S_{RTS} + R_{CTS} + Sp + Sal\right) + S_d \tag{4.2.3.14}$$

Therefore, the total latency for the ZeroMAC protocol is derived as

$$\begin{aligned} Lat_{ZeroMAC} = t_{contention} + ENIR \times (S_{GT} + R_{act} S_{RTS} + R_{CTS} + S_{al}) + \\ t_d \times n \times h \end{aligned} \tag{4.2.3.15}$$

$$ENIR = \frac{\left(R_l + R_s\right)}{\left(R_l - S_{RTS}\right)} \tag{4.2.3.16}$$

Table 4.4: Delay analysis variables for ZeroMAC

Symbol	**Meaning**
S_{GT}	Duration of the sender Guard time
S_{RTS}	Duration of the sender's RTS
R_{CTS}	Duration of the receiver CTS
S_{al}	Duration of the Acknowledgement listen
S_d	Duration of the sender's data transmission periods
R_{act}	Receiver Activation Time
R_l	Receiver listen period
R_s	Receiver sleep period
t_d	Data transmission period for a single packet
R_a	Duration of the receiver's ACK period
R_d	Duration of the receiver's data period

Table 4.5: Energy consumption analysis variables for ZeroMAC

Symbol	**Meaning**
$P_d(t)$	Probability of packet arival ration
P_{Tx}	Power required to transmit
P_{Rx}	Power required to receive
P_s	Power required to sleep
S_{RTS}	Duration of the sender's RTS
R_{CTS}	Duration of the receiving CTS
S_{al}	Duration of the sender's acknowledgment
S_d	Duration of the sender's data transmission periods
R_l	Receiver listen periods

ENIR=Expected Number of Iteration Required

4.2.3.3 Energy Consumption Analysis of ZeroMAC

The total energy consumption for the ZeroMAC is:

$$E_{(ZeroMAC)} = E_s + E_r \tag{4.2.3.17}$$

E_s = Expected energy to send a packet and E_r =Expected energy to receive a packet Expected energy (Es) to send a packet:

E_s =(RTS sending energy + CTS receiving + energy per ACK listen) × (expected preamble-listen iterations required) +(energy to send packet)

$$= (P_{Tx}S_{RTS} + P_{Rx}R_{CTS} + P_{Rx}S_{al})\left(\frac{1}{\frac{R_l - S_p}{R_l + S_{RTS}}}\right) + S_d P_{Tx} \tag{4.2.3.18}$$

$$= \frac{((P_{Tx}S_{RTS} + P_{Rx}R_{CTS} + P_{Rx}S_{al})(R_l + S_p))}{(R_l - S_{RTS})} + S_d P_{Tx} \tag{4.2.3.19}$$

The expected energy to receive a packet is given by:

E_r =(listen cycle energy + sleep cycle energy+CTS sending energy) ×(expected iterations for a preamble to arrive)+(energy to send an ACK)+(energy to receive packet)

$$\frac{(P_s R_s + P_{Rx} R_l + P_{Rx} R_{CTS})}{(1-(1-P_d(t)))^{(R_l+R_s)}} + P_{Tx} R_a + R_d P_{Rx} \qquad (4.2.3.20)$$

4.2.4 Performance Analysis of MCAS-MAC protocol

In MCAS-MAC protocol Each node stores its neighbors' scheduling information in its neighbor table. Based on the table, a sender wakes up at the wake-up time of the receiver to send its packet [43].Since different receivers wake up at different times, senders that intend to send their packets to different receivers also wake up to send their packets at different times, thus reducing contention and eliminating overhearing (only sender intending to send to the same receiver are simultaneously awake).Hello message, the receiver sets a timeout after the maximum contention window size for the random backoff for data transmissions. Upon a timeout (i.e., no incoming packet is detected), the receiver immediately goes back to sleep mode. The Markov model has a finite number of states, each of which represents a different status of a node, i.e., a different queue length, at the wake-up instant of a cycle. A node may change status cycle by cycle, corresponding to the transition from one state to another in the Markov model. Figure 4.2 shows the proposed Markov model with a queue capacity Q. This Markov model has Q+1 states, each of which, from left to right, corresponds to 0 packets in the queue, 1 packet in the queue, to Q packets in the queue (full queue). The Markov model also has (Q x Cn) states, each of which, from left to right, corresponds to number of channel to send data parallaly. Specifically, when the queue is not empty, a node will attempt to access the media to transmit a DATA packet. A DATA packet is removed from the queue either when it is transmitted successfully, or when it encounters a collision as no retransmission is allowed. A DATA packet is dropped when the queue overflows. Hence, the transition probabilities from one state to another can be described as follows.

$$P_{0,i} = A_i, i = 0...Q-1 \tag{4.2.4.1}$$

$$P_{0,Q} = A_{\geq Q} \tag{4.2.4.2}$$

$$P^{(n)}_{i,j-1} = p^{(n)}.A_0 i = 1...Q \tag{4.2.4.3}$$

$$P_{i,j} = p.A_{j-i+1} + (1-p).A_{j-i} i = 1...Q-1, j = i...Q-1 \tag{4.2.4.4}$$

$$P_{i,Q} = p.A_{\geq Q-i+1} + (1-p).A_{Q-i}, i = 1...Q \tag{4.2.4.5}$$

$$P^{(n)}_{i,j} = p^{(n)} B^{(n)} i = 2...Q, j = i...i-2 \tag{4.2.4.6}$$

where, A_i is the probability that i DATA packets arrive at a node in a cycle, $A_{\geq i}$ is the probability that no less than i DATA packets arrive at a node in a cycle, and p is the probability for a node to transmit a DATA packet in a cycle and $B^{(n)}$ is the probability that n channels are paralleli use to transmit data. Specifically, Equations (4.2.4.1) and (4.2.4.2) describe the fact that all the transitions from the empty-queue state to a non-empty-queue state depend only on new packet arrivals. Equations (4.2.4.3) and (4.2.4.6) describe the fact that n number of nodes can transmit DATA packet paralleli per cycle with a probability $P^{(n)}$, and the probability of having one packet less in the queue equals the probability of winning the contention times the

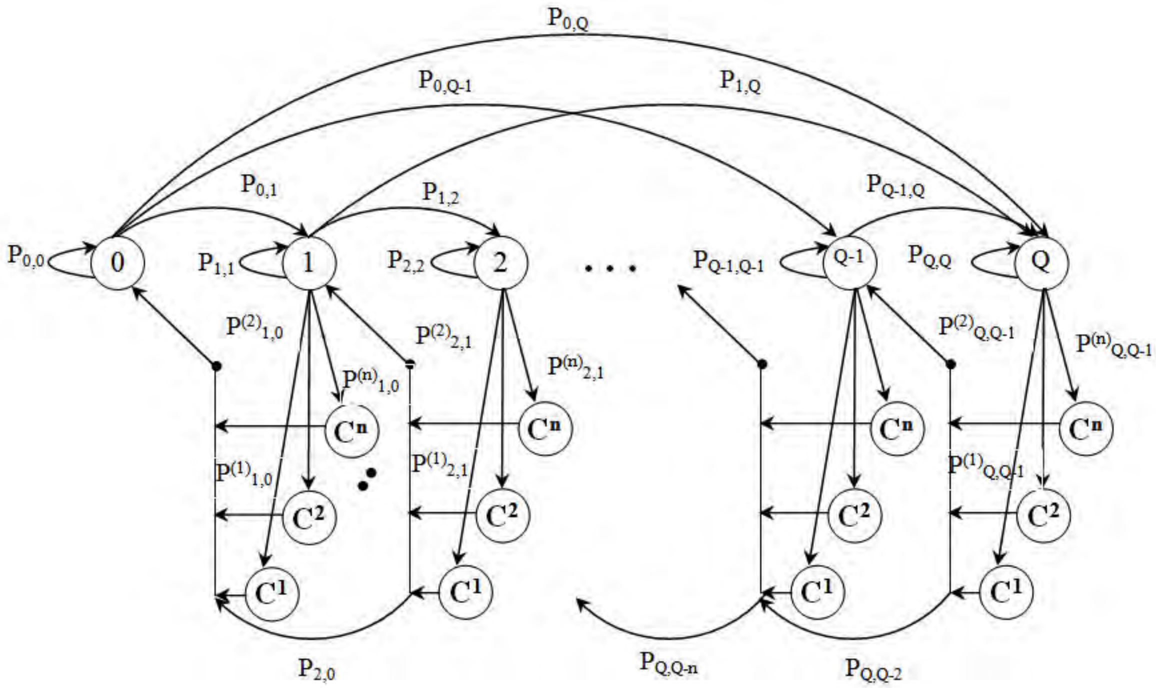


Figure 4.2: Markov model for MCAS-MAC protocol

probability of no packet arrivals in a cycle. Moreover, (4.2.4.4) and (4.2.4.5) describe the fact that the probability of having a non-decreasing queue can be divided into two parts depending on whether the oldest DATA packet in the queue wins the contention (first term) or not (second term). Finally, (4.2.4.2) and (4.2.4.5) show that packets are dropped when the queue overflows.

4.2.4.1 Throughput Analysis of MCAS-MAC

The MCAS-MAC protocol works in a duty-cycled fashion, so, the throughput will be calculated within a cycle time.

$$THR_{MCAS-MAC} = \frac{N.\big(1-\pi_0\big).P_s.S}{T} \tag{4.2.4.7}$$

A. Stationary probability (empty queue state π_0) calculation

Markov model for MCAS-MAC has been proposed, where (1) the data packet arrivals at each node in our model can be from distributions and (2) the service rate at each node (the packet transmission rate at each node in a cycle) is unknown, and it depends on the contention in the network. The proposed Markov model with state space S = 0,1, . . .,Q and transition matrix P has a unique stationary distribution $\pi = \pi_0, ...\pi_Q$ since the Markov model is irreducible and aperiodic. Therefore, $\pi_{i \geq 0}$ for any $s_i \in S$

$$\sum_{s_i \in S} \pi_i = 1 \tag{4.2.4.8}$$

$$\pi = P.\pi \tag{4.2.4.9}$$

Assuming packet arrival information (A_k, and $A_{\geq k}$) is known, the probability p for each node to win the contention becomes the only variable in the transition matrix P. Since π_0 is the unique solution for (4.2.4.9), for any $s_i \in S$ can be represented as a function of p. Specifically, let function f(.) describe the relationship between π_0 and p, i.e.,

$$\pi_0 = f(p) \tag{4.2.4.10}$$

According to (4.2.4.10), for a given probability p for each node to win the contention, the stationary probability π_0 of the empty-queue state can be obtained using the proposed Markov model.

B. Probability success (P_s) calculation by Media Access Rule of MCAS-MAC

Since the number of nodes in the network N, the MCAS-MAC layer DATA packet size S, and the length of a cycle T are known, once π_0 is solved by (4.2.4.10) and (4.2.4.11), the only unknown variable in (4.2.4.7) is the probability p_s for each node to successfully transmit a DATA packet. When π_0 is known, p_s can be obtained according to the media access rules of the investigated protocol. Similar to the way obtain p, every node has a packet to send with a probability of $1-\pi_0$ in a cycle. For a given π_0, the probability p_s for a node to successfully transmit a DATA packet is determined by the number of nodes in the network N and the manner in which nodes compete with each other. The relationship between π_0 and p_s can be described as:

$$\pi_s = h(\pi_0) \tag{4.2.4.11}$$

Plugging (4.2.4.10) into (4.2.4.7), the throughput of the network can be determined.

$$P = P_S + P_C \tag{4.2.4.12}$$

Data transmission probability P can be define as a addition of Probability of success P_s and collision P_c . The probability of Success depend of status of the queue and channel free probability.

$$P_s = P_{avs} + P_{free} \tag{4.2.4.13}$$

The average probability of success in data channel can be calculated by the following equation:

$$\begin{aligned} P_{avs} = \sum_{t=1}^{T} \frac{1}{T} \Bigg(\sum_{w=1}^{C_w} \Bigg(\sum_{T_a=0}^{N-1} \Bigg(\sum_{W_i=0}^{T_a} \Bigg(\sum_{S_d=0}^{W_i} P_r \Big(A | affect, \overline{empty} \Big) \\ P_r \Big(B | wake, \overline{empty} \Big) P_r \Big(C | data, \overline{empty} \Big) \Bigg) \Bigg) \Bigg) \Bigg) \end{aligned} \tag{4.2.4.14}$$

$$P_r\Big(A|affect, \overline{empty}\Big) = \binom{N-1}{T_a}\left(\frac{W_m}{T}\right)^{T_a}\left(\frac{T-W_m}{T}\right)^{N-1-T_a} \tag{4.2.4.15}$$

$$P_r\Big(B|wake, \overline{empty}\Big) = \binom{T_a}{W_i}\left(\frac{1}{W_m}\right)^{W_i}\left(\frac{W_m-1}{W_m}\right)^{T_a-W_i} \tag{4.2.4.16}$$

$$P_r\Big(C|data, \overline{empty}\Big) = \binom{W_i}{S_d}\Big(\pi_0\Big)^{W_i-S_d}\Big(1-\pi_0\Big)^{S_d}\left(\frac{(t_i-1)\times C_n}{W_m}\right)^{S_d} \tag{4.2.4.17}$$

$P_r\Big(A|affect, \overline{empty}\Big)$= Probability of Event A where number of T_a nodes wakeup at contention widow affected area. $P_r\Big(B|wake, \overline{empty}\Big)$= Probability of Event B where number of nodes W_i wakeup at i^{th} slot of contention window. $P_r\Big(C|data, \overline{empty}\Big)$= Probability of Event C where number of nodes S_d have some data but they do not hamper the transmission on this contention window.

Equation (4.2.4.17) presents the probability that S_d nodes out of Wi nodes have some data but their backoff timer expires after ti with receiver home channel C_n so that they do not hamper the transmission at t while the other nodes $(W_i - S_d)$ have no data to send.

Probability of collision P_c is depends on a nodes who hamper transmission when some others node transmitting data and channel free probability.

$$P_c = P_{avc} + P_{free} \tag{4.2.4.18}$$

Average Probability of collision:

$$P_{avsc} = \sum_{t=1}^{T}\frac{1}{T}\left(\sum_{w=1}^{C_w}\left(\sum_{T_a=0}^{N-1}\left(\sum_{W_i=0}^{T_a}\left(\sum_{S_d=0}^{W_i} P_r\Big(A|affect, \overline{empty}\Big)\right.\right.\right.\right.$$
$$\left.\left.\left.\left. P_r\Big(B|wake, \overline{empty}\Big)P_r\Big(C|coll, \overline{empty}\Big)\right)\right)\right)\right) \tag{4.2.4.19}$$

$$P_r\Big(A|affect, \overline{empty}\Big) = \binom{N-1}{T_a}\left(\frac{W_m}{T}\right)^{T_a}\left(\frac{T-W_m}{T}\right)^{N-1-T_a} \tag{4.2.4.20}$$

$$P_r\Big(B|wake, \overline{empty}\Big) = \binom{T_a}{W_i}\left(\frac{1}{W_m}\right)^{W_i}\left(\frac{W_m-1}{W_m}\right)^{T_a-W_i} \tag{4.2.4.21}$$

$$P_r\Big(C|coll,\overline{empty}\Big) = \binom{W_i}{S_d}\Big(\pi_0\Big)^{W_i-S_d} \Big(1-\pi_0\Big)^{S_d}\Bigg(\frac{(W_m-t_i-1)/C_n}{W_m}\Bigg)^{S_d} \tag{4.2.4.22}$$

The average free channel probability $Pr_{(free)}$ at given the is the ratio of duration of channel free and entire channel free and busy time.

$$P_r\Big(feee\Big) = \frac{E_{free}}{E_{free}-E_{busy}} \tag{4.2.4.23}$$

The probability of a free channel, $P_r(free)$, can be obtained if the following two parameters are known. (1) the average length of a free channel between two transmissions over the media, E_{free}, and (2) the average length of a busy channel between the two chunks of a free channel, E_{busy}.

The channel free probability:

$$P_{free} = \sum_{i=0}^{N-1}\sum_{j=1}^{N-i}\sum_{k=1}^{j}\binom{N}{i}\Bigg(\frac{t}{T}\Bigg)^{i}\Bigg(\frac{T-W_m}{T}\Bigg)^{N-1-i}\binom{N-i}{j}\Bigg(\frac{1}{W_m}\Bigg)^{j} \Bigg(\frac{W_m-1}{W_m}\Bigg)^{N-i-j}\binom{j}{k}\Bigg(1-\pi_0\Bigg)^{k}\pi_0^{j-k}\Bigg(\frac{(W-t+1)\times C_n}{W_m}\Bigg)^{j-k} \tag{4.2.4.24}$$

The expected channel free probability (E_{free}) depends on the probability of the event that at the time instant when a transmission ends and the interval of free channel begins. To calculate E_{free} considered the time instant of channel is free for n cycles and t time slots, therefore $n.T+t$ is the duration the of free channel and probability of this event is P_{free}. Equation (4.2.4.25) shows the calculation of E_{free} where k out of j nodes which are competing for the media at the same time when there is no backoff mechanism. In this case there is high probability of collision. By using backoff mechanism the probability that one node send data which given by equation (4.2.4.24) can be used to find the probability P_{free} .

The average channel free probability:

$$E_{free} = \sum_{n=0}^{\infty}\sum_{t=1}^{T}\Bigg(nT+t\Bigg)P_{free} \tag{4.2.4.25}$$

Equation (4.2.4.25) illustrate the probability of the channel being busy between two free channel intervals due to successful transmission. To calculate the average duration of time interval for which the channel is busy between two free channel intervals E_{Busy}, first find the probability P_{busy}^{succ} of the event when the transmission is successful between two free channel intervals and the probability P_{busy}^{coll} of the event when there is collision between two free channel intervals.

$$E_{busy} = \sum_{n=0}^{\infty} \sum_{t=1}^{T} \left(\left(\frac{T}{2} nT + L_{DATA} \right) P_{busy}^{succ} + T P_{busy}^{pre} \right) \tag{4.2.4.26}$$

The probability of busy channel between two free channel intervals due to collisions of short preambles is given by Equation (4.2.4.28). In equation (4.2.4.28) k out of j nodes which are competing for the medium at the same time when there is no backoff mechanism will cause collision.

$$\begin{aligned} P_{busy}^{succ} = \sum_{i=0}^{N-1} \sum_{j=1}^{N-i} \sum_{k=1}^{j} \binom{N}{i} \left(\frac{t}{T} \right)^{i} \left(\frac{T - W_m}{T} \right)^{N-1-i} \binom{N-i}{j} \left(\frac{1}{W_m} \right)^{j} \\ \binom{j}{k} \left(1 - \pi_0 \right)^{k} \pi_0^{j-k} \left(\frac{(T - t + 1) \times C_n}{W_m} \right)^{j-k} \end{aligned} \tag{4.2.4.27}$$

Thus using the same procedure as in Equation (4.2.4.27), but here are at least $C_n \times 2$ nodes wakeup at the same time use P_{busy}^{coll}. The average duration of busy channel between two free channel intervals P_{busy}^{coll}, define by the length of successful transmission time duration $(T/2 + t_{Data})$ multiplied by the probability of this event plus time duration of busy channel due to collisions of short preambles sending by control channel.

$$\begin{aligned} P_{busy}^{coll} = \sum_{i=0}^{N-2} \sum_{j=2}^{N-i} \sum_{k=2}^{j} \binom{N}{i} \left(\frac{t}{T} \right)^{i} \left(\frac{T - W_m}{T} \right)^{N-1-i} \binom{N-i}{j} \left(\frac{1}{W_m} \right)^{j} \\ \binom{j}{k} \left(1 - \pi_0 \right)^{k} \pi_0^{j-k} \left(\frac{(T - t + 1)/C_n}{W_m} \right)^{j-k} \end{aligned} \tag{4.2.4.28}$$

Plugging equations (4.2.4.27) and (4.2.4.28) into (4.2.4.26), obtain E_{busy}. Then plugging equation (4.2.4.24) into (4.2.4.25), obtain E_{free} . Plugging equation (4.2.4.23) and (4.2.4.14) into (4.2.4.13) the probability for each node to successfully transmit a DATA packet P_s, and Plugging equation (4.2.4.23) and (4.2.4.19) into (4.2.4.18) the probability for each node to encounter a collision P_c are solved as a function of the stationary probability of the empty queue state π_0.

4.2.4.2 Delay Analysis of MCAS-MAC

The expected latency for a single hop data exchange is given by:

$Lat_{MCAS-MAC}$ = (compensate clock drift time + Hello message time + channel switching time+ random backoff time) × (expected number of iterations required) + duration to send a data packet

$$= \left(\frac{1}{\frac{R_l - S_p}{R_l + R_s} \times f_n} \right) \times \left(H_p + T_{cdt} + T_{cst} + T_{backoff} \right) + S_d \tag{4.2.4.29}$$

$$= \frac{\left(R_l + R_s \right)}{\left(R_l - S_p \right) \times f_n} \times \left(H_p + T_{cdt} + T_{cst} + T_{backoff} \right) + S_d \tag{4.2.4.30}$$

The total latency for the MCAS-MAC protocol is derived as

$$\begin{aligned} Lat_{MCAS-MAC} = t_{contention} + ENIR \times (H_p + T_{cdt} + T_{cst} + T_{backoff}) \\ + t_d \times n \times h \end{aligned} \tag{4.2.4.31}$$

Table 4.6: Delay analysis variables for MCAS-MAC

Symbol	**Meaning**
H_p	Duration of the sender's Hello message
T_{cdt}	Compensate clock drift time
T_{cst}	Channel switching time
$T_{backoff}$	Random backoff time
f_n	Number of frequencies used in multichannel
R_l	Receiver listen period
R_s	Receiver sleep period
t_d	Data transmission period for a single packet
R_a	Duration of the receiver's ACK period
R_d	Duration of the receiver's data period

$$ENIR = \frac{\left(R_l + R_s\right)}{\left(R_l - S_p\right) \times f_n} \tag{4.2.4.32}$$

ENIR=Expected Number of Iteration Required

4.2.4.3 Energy Consumption Analysis of MCAS-MAC

The total energy consumption for the MCAS-MAC is:

$$E_{(MCAS-MAC)} = E_s + E_r \tag{4.2.4.33}$$

E_s = Expected energy to send a packet and Er=Expected energy to receive a packet

Expected energy (Es) to send a packet:

E_s =(Hello message energy + energy per ACK listen) ×

(expected preamble - listen iterations required) +(energy to send packet)

$$= (P_{Tx}S_h)\left(\frac{1}{\frac{R_l-S_H}{R_l+S_H} \times f_n}\right) + S_dP_{Tx} \tag{4.2.4.34}$$

Table 4.7: Energy consumption analysis variables for MCAS-MAC

Symbol	**Meaning**
$P_d(t)$	Probability of packet arival ration
P_{Tx}	Power required to transmit
P_{Rx}	Power required to receive
P_s	Power required to sleep
f_n	Number of frequencies used in multichannel
S_H	Duration of the sender's Hello message
S_d	Duration of the sender's data transmission periods
R_l	Receiver listen periods

The expected number of preamble listen iterations required, giving:

$$E_S = \frac{(P_{Tx}S_h)(R_l + S_H)}{\left((R_l - S_H) \times f_n\right)} + S_d P_{Tx} \tag{4.2.4.35}$$

The expected energy to receive a packet is given by: E_r =(listen cycle energy + sleep cycle energy) × (expected iterations for a preamble to arrive) +(energy to receive packet)

$$\frac{(P_s R_s + P_{Rx} R_l)}{(1 - (1 - P_d(t)))^{(R_l + R_s)}} + P_{Tx} R_a + R_d P_{Rx} \tag{4.2.4.36}$$

4.3 Analytical Modeling of Proposed Protocol

4.3.1 Performance Analysis of Proposed MX-MAC Protocol

The proposed Markov model assumes independent packet arrivals at each node, finite queue capacity at each node, ideal channel, one transmission opportunity or one DATA packet reception per cycle at each node, no retransmissions, and every node has a constant probability of transmitting a DATA packet regardless of any node's queue length [4]. Figure 4.3 shows the proposed Markov model for MX-MAC. The Markov model has Q+1 states, first digit of the state indicates channels (c=control, d=data) and second digit from left to right corresponding to 0 to Q data packets in the queue, Q indicates the queue full. In Figure 4.3 the probability $P_{ij}(i < j)$ indicates that j new data packets arrive at a sender node i. Probability $P_{ij}(i > j)$ defines the probability of the transmission of only one data packet. Probability $P_{ij}(i = j)$ implies non-decreasing queue. $R_{ij}(i = j)$ means i new data packets arrive at sender node and j data packets is being received and forward to next hop without sending any additional preamble. $R_{ij}(i < j)$ reveals i+1 new data packets arrived at receiver node. $R_{ij}(i > j)$ exhibits i-1 data packets transmitted to final destination (sink). $S_{ij}(i \leq j)$ defines i data packets transmitted to sink and j new data packets arrive at the any sensor node.

This Markov model has two systems. First subsystem for state change its status cycle by cycle depending on data packet arrived at a node without next hope packet arrival space. Second subsystem change its states depending on data packet arrived at a node with sufficient queue space for post arrival packet. The transition probabilities from one state to another can be represented as follows:

$$P_{0,i} = A_i, i = 0...Q-1 \tag{4.3.1.1}$$

$$P_{0,Q} = A_{\leq Q} \tag{4.3.1.2}$$

$$P_{i,j} = \mu A_{j-i+1} + (1-\mu)A_{j-i}, i = 1...Q-1, j = i...Q-1 \tag{4.3.1.3}$$

$$P_{i,Q} = \mu A_{\geq Q-i+1} + (1-\mu)A_{\geq Q-i}, i = 1...Q \tag{4.3.1.4}$$

$$P_{i,j} = 0, i = 2...Q, j = i...Q-2 \tag{4.3.1.5}$$

$$R_{i,j} = \lambda_i i = j, i = 1...Q-2 \tag{4.3.1.6}$$

$$R_{i,j} = \mu A_i + \lambda_i, i > j, i = 0...Q-1, j = i+1...Q \tag{4.3.1.7}$$

$$S_{i,j} = \mu A_i, i \geq j, i = 0...Q-1, j = i-1...Q \tag{4.3.1.8}$$

where, λ_1 is the probability that DATA packets arrive at a node in a cycle without next hope packet arrival space, and λ_2 is the probability with next hope packet arrival space Specifically,

- Equations (4.3.1.1) and (4.3.1.2) show that transitions from the empty queue state to any other states depend only on the new packet arrivals. A_i is the probability that i DATA packets arrive at a node in a cycle. Equations (4.3.1.2) shows that packets are dropped when the queue overflows.

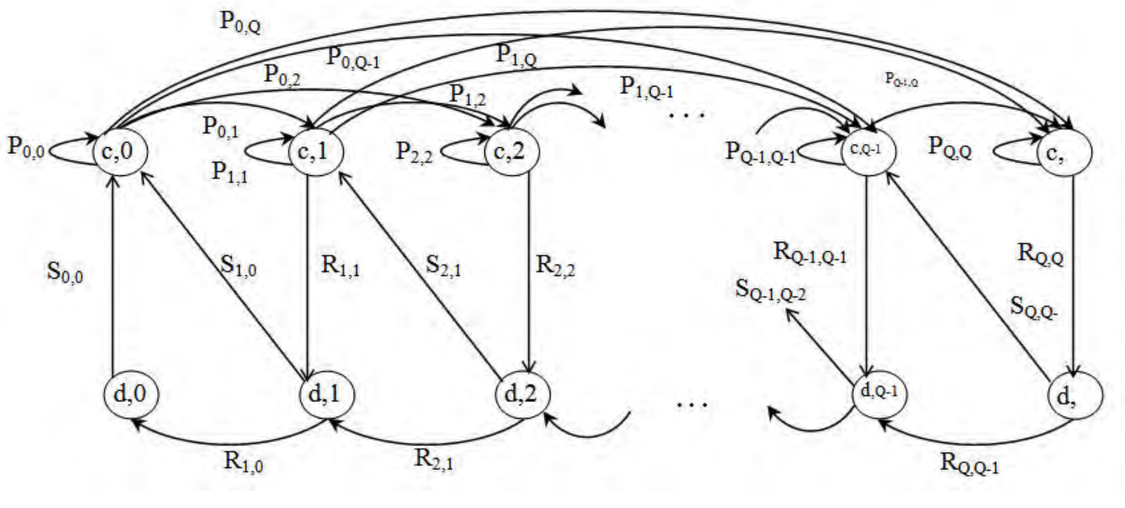


Figure 4.3: Markov model for MX-MAC

- Equations (4.3.1.3) and (4.3.1.4) show that a transition with a non-decreasing queue has two possible cases, depending on whether the oldest DATA packet in the queue wins the media (first term) or not (second term). Here only one DATA packet can be transmitted during a cycle with probability μ.
- Equations (4.3.1.6) show that i DATA packet in the queue wins the media. λ_i is the probability of i new data packets arrive at sender node and i data packets is being received and receiver send a preamble with next receiver ID on the control channel to notify its neighbors that an ongoing reception is in progress and alert intended next hop neighbor to wake up on time as well as to forward received data to next hop without sending any additional preamble. As a result, packet collisions are reduced with an increase of network throughput. .
- Equations (4.3.1.7) show that only one DATA packet can be transmitted during a cycle with probability μ.
- Equations (4.3.1.8) show that only one DATA packet can be transmitted during a cycle with probability μ and there is no next receiver node or it is the sink node.

Throughput is defined as the amount of data

4.3.1.1 Throughput Analysis of proposed MX-MAC

Throughput is defined as the amount of data successfully delive. Our proposed protocol MX-MAC works in a duty-cycled fashion, so, the throughput will be calculated

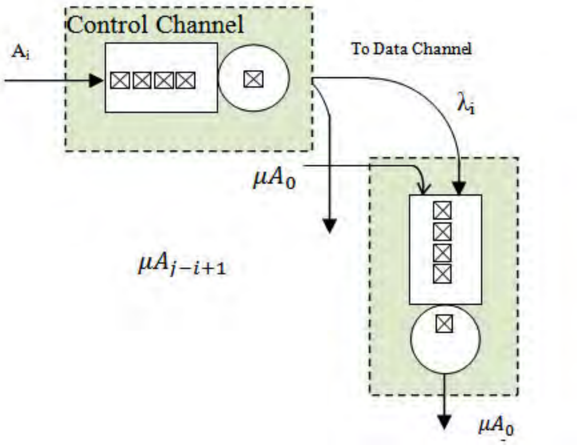


Figure 4.4: Model details (control channel to data channel transition)

within a cycle time.

$$THR_{MX-MAC} = \frac{N.\big(1-\pi_0\big).P_s.S}{T} \tag{4.3.1.9}$$

Here, there are N are the number of nodes in the network, S is the data packet size, T indicates number of slots in a cycle, τ refers the length of a slot, and Ps specifies the probability of successfully data transmission in a cycle and π_0 signifies the probability that a node has no data packet to send in a cycle. N, F and T are known variable, only π_0 and Ps are need to be calculated. π_0 and Ps are determined by our proposed Markov model and the media access rule of MX-MAC. The Markov model has a state space 0,1. . . .Q and a transition matrix. The stationary probability of the empty queue state can be obtain from Markov model which is shown in Figure 4.3.

A. Stationary probability (empty queue state π_0) calculation

Table 4.8: Imperative parameters for performance analysis of MX-MAC

Symbol	**Meaning**
N	Total Number of nodes in the network
T	Length of a cycle
Q	Queue Size in Markov model
F	Frame Size
A_i	Data Arrival rate in subsystem 1
λ_i	Data Arrival rate in subsystem 2
W_m	Maximum Contention Window size
π_0	Stationary Probability of empty Queue State
P_s	Probability of success
P_c	Probability of collision
W_{succ}	Success Window size

We propose MX-MAC Markov model where (1) the data packet arrivals at each node in our model can be from distributions and (2) the service rate at each node (the packet transmission rate at each node in a cycle) is unknown, and it depends on the contention in the network. The proposed Markov model with state space S = 0,1, . . .,Q and transition matrix P has a unique stationary distribution $\pi = \pi_0, ...\pi_Q$ since the Markov model is irreducible and aperiodic. Therefore, $\pi_{i \geq 0}$ for any $s_i \in S$

$$\sum_{s_i \in S} \pi_i = 1 \tag{4.3.1.10}$$

$$\pi = P.\pi \tag{4.3.1.11}$$

Assuming packet arrival information (A_k, and $A_{\geq k}$) is known, the probability p for each node to win the contention becomes the only variable in the transition matrix P. Since π_0 is the unique solution for (4.3.1.11), for any $s_i \in S$ can be represented as a function of p. Specifically, let function f(.) describe the relationship between π_0 and p, i.e.,

$$\pi_0 = f(p) \tag{4.3.1.12}$$

According to (4.3.1.12), for a given probability p for each node to win the contention, the stationary probability π_0 of the empty-queue state can be obtained using the proposed Markov model.

B. Probability success P_s calculation by Media Access Rule of MX-MAC

Since the number of nodes in the network N, the MX-MAC layer DATA packet size S, and the length of a cycle T are known, once π_0 is solved by (4.3.1.12) and (4.3.1.13), the only unknown variable in (4.3.1.9) is the probability p_s for each node to successfully transmit a DATA packet. When π_0 is known, p_s can be obtained according to the media access rules of the investigated protocol. Similar to the way obtain p, every node has a packet to send with a probability of $1 - \pi_0$ in a cycle. For a given π_0, the probability p_s for a node to successfully transmit a DATA packet is determined by the number of nodes in the network N and the manner in which nodes compete with each other. The relationship between π_0 and p_s can be described as:

[4]

$$p_s = h(\pi_0) \tag{4.3.1.13}$$

Plugging (4.3.1.12) into (4.3.1.9), the throughput of the network can be determined.

$$P = P_S + P_C \tag{4.3.1.14}$$

Data transmission probability P can be define as a addition of Probability of success P_s and collision P_c . The probability of Success depend of status of the queue and channel free probability.

$$P_s = P_{avs} + P_{free} \tag{4.3.1.15}$$

Average Probability of Success:

$$P_{avs} = P_{avsc}.P_{avsd} \tag{4.3.1.16}$$

The average probability of success P_{avs} in MX-MAC protocol depanding on the average probability of success in control channel P_{avsc} and P_{avsc}. The average probability of success in data channel can be calculated by the following equation:

$$\begin{aligned} P_{avsd} = \sum_{t=1}^{T} \frac{1}{T} \Bigg(\sum_{w=1}^{C_w} \Bigg(\sum_{T_a=0}^{N-1} \Bigg(\sum_{W_i=0}^{T_a} \Bigg(\sum_{S_d=0}^{W_i} P_r\Big(A|affect, \overline{empty}\Big) \\ P_r\Big(B|wake, \overline{empty}\Big) P_r\Big(C|data, \overline{empty}\Big)\Bigg)\Bigg)\Bigg)\Bigg) \end{aligned} \tag{4.3.1.17}$$

$$P_r\Big(A|affect, \overline{empty}\Big) = \binom{N-1}{T_a} \left(\frac{W_m}{T}\right)^{T_a} \left(\frac{T - W_m}{T}\right)^{N-1-T_a} \tag{4.3.1.18}$$

$$P_r\Big(B|wake, \overline{empty}\Big) = \binom{T_a}{W_i} \left(\frac{1}{W_m}\right)^{W_i} \left(\frac{W_m - 1}{W_m}\right)^{T_a - W_i} \tag{4.3.1.19}$$

$$P_r\Big(C|data, \overline{empty}\Big) = \binom{W_i}{S_d} \Big(\pi_0\Big)^{W_i - S_d} \Big(1 - \pi_0\Big)^{S_d} \left(\frac{W_{succ}}{W_m}\right)^{S_d} \tag{4.3.1.20}$$

$P_r\Big(A|affect, \overline{empty}\Big)$= Probability of Event A where number of nodes T_a wakeup at contention window affected area.

$P_r\Big(B|wake, \overline{empty}\Big)$= Probability of Event B where number of nodes W_i wakeup at $i^t h$ slot of contention window.

$P_r\Big(C|data, \overline{empty}\Big)$= Probability of Event C where number of nodes S_d have some data but they do not hamper the transmission on this contention window.

The average probability of success in data channel depends on value of W_{succ} . In this case $W_{succ} = W_0$, whice indicates the probability of success is approximate 100% and there is no collision will occure during data transmission period in data channel in MX-MAC protocol. The average probability of success in data channel can be calculated by the following equation:

$$P_{avsc} = \sum_{t=1}^{T} \frac{1}{T} \Bigg(\sum_{w=1}^{C_w} \Bigg(\sum_{T_a=0}^{N-1} \Bigg(\sum_{W_i=0}^{T_a} \Bigg(\sum_{S_d=0}^{W_i} P_r\Big(D|affect, \overline{empty}\Big) P_r\Big(E|wake, \overline{empty}\Big) P_r\Big(F|data, \overline{empty}\Big)\Bigg)\Bigg)\Bigg)\Bigg) \tag{4.3.1.21}$$

$$P_r\Big(D|affect, \overline{empty}\Big) = \binom{N-1}{T_a} \left(\frac{W_m}{T}\right)^{T_a} \left(\frac{T - W_m}{T}\right)^{N-1-T_a} \tag{4.3.1.22}$$

$$P_r\Big(E|wake, \overline{empty}\Big) = \binom{T_a}{W_i} \left(\frac{1}{W_m}\right)^{W_i} \left(\frac{W_m - 1}{W_m}\right)^{T_a - W_i} \tag{4.3.1.23}$$

$$P_r\Big(F|data, \overline{empty}\Big) = \binom{W_i}{S_d} \Big(\pi_0\Big)^{W_i - S_d} \Big(1 - \pi_0\Big)^{S_d} \left(\frac{t_t}{W_m}\right)^{S_d} \tag{4.3.1.24}$$

$P_r\Big(D|affect, \overline{empty}\Big)$= Probability of Event D where number of nodes T_a wakeup at contention window affected area.

$P_r\Big(E|wake, \overline{empty}\Big)$= Probability of Event E where number of nodes W_i wakeup at $i^t h$ slot of contention window.

$P_r\Big(F|data, \overline{empty}\Big)$= Probability of Event F where number of node S_d out of W_i nodes have Some data and will send preamble but their backoff timer expires after t so that they do not hamper the transmission at t on this contention window while the other nodes (W_i–S_d) have no data to send. All the above equations has derived by using binomial probability formula. For example, in equation (4.3.1.22) , N= total number of trial, T_a=number of specific events want to obtain, W_m/T =the probability of success and $(T - W_m)/T$ =the probability of failure. All the above equations has derived by using binomial probability formula. For example, in equation 10, N= Total number of tral, Ta=Number of specific events that wants to obtain, W_m/T=The probability of success and $(T - W_m)/T$ =The probability of failure. Probability of collision P_c is depends on a nodes who hamper transmission when some others node transmitting data and channel free probability. The probability of collision in MX-MAC very low(approximately 0) because the region of contention window W_{coll} define the in terms of receiver preamble. In receiver preamble section no other node try to send any data, so collision nullified.

$$P_c = P_{avc}.P_{free} \tag{4.3.1.25}$$

Average Probability of Success:

$$\begin{aligned} P_{avc} = \sum_{t=1}^{T} \frac{1}{T} \Bigg(\sum_{w=1}^{C_w} \Bigg(\sum_{T_a=0}^{N-1} \Bigg(\sum_{W_i=0}^{T_a} \Bigg(\sum_{S_d=0}^{W_i} P_r\Big(A|affect, \overline{empty}\Big) \\ P_r\Big(B|wake, \overline{empty}\Big) P_r\Big(C|coll, \overline{empty}\Big)\Bigg)\Bigg)\Bigg)\Bigg) \end{aligned} \tag{4.3.1.26}$$

$$P_r\Big(A|affect, \overline{empty}\Big) = \binom{N-1}{T_a} \left(\frac{W_m}{T}\right)^{T_a} \left(\frac{T-W_m}{T}\right)^{N-1-T_a} \tag{4.3.1.27}$$

$$P_r\Big(B|wake, \overline{empty}\Big) = \binom{T_a}{W_i} \left(\frac{1}{W_m}\right)^{W_i} \left(\frac{W_m-1}{W_m}\right)^{T_a-W_i} \tag{4.3.1.28}$$

$$P_r\Big(C|coll, \overline{empty}\Big) = \binom{W_i}{S_d} \Big(\pi_0\Big)^{W_i-S_d} \Big(1-\pi_0\Big)^{S_d} \left(\frac{W_{coll}}{W_m}\right)^{S_d} \tag{4.3.1.29}$$

The probability of collision in MX-MAC very low (approximately 0) because the region of contention window W_{coll} define the in terms of receiver preamble. As defined introduction section multi channel enables a receiver to send a preamble with next receiver ID on the control channel to notify its neighbors that an ongoing reception is in progress. if any neighbor node of the receiver have some data to send, it refrain from sending any preamble. As a result, collision has nullified and collision windew size (W_{coll}) reduced to approximately 0. The average free channel probability $P_r(free)$ at given the is the ratio of duration of channel free and entire channel free and busy time. The average free channel probability at given the is the ratio of duration of channel free and entire channel time.

$$P_r\Big(feee\Big) = \frac{E_{free}}{E_{free} - E_{busy}} \tag{4.3.1.30}$$

The probability of a free channel, $P_r(free)$, can be obtained if the following two parameters are known. (1) the average length of a free channel between two transmissions over the media, E_{free}, and (2) the average length of a busy channel between the two chunks of a free channel, E_{busy}. The channel free probability:

$$P_{free} = \sum_{i=0}^{N-1}\sum_{j=1}^{N-i}\sum_{k=1}^{j}\binom{N}{i}\left(\frac{t}{T}\right)^{i}\left(\frac{T-W_m}{T}\right)^{N-1-i}\binom{N-i}{j}\left(\frac{1}{W_m}\right)^{j} \left(\frac{W_m-1}{W_m}\right)^{N-i-j}\binom{j}{k}\left(1-\pi_0\right)^{k}\pi_0^{j-k}\left(\frac{t}{W_m}\right)^{j-k} \tag{4.3.1.31}$$

The expected channel free probability (E_{free}) depends on the probability of the event that at the time instant when a transmission ends and the interval of free channel begins. To calculate E_{free} considered the time instant of channel is free for n cycles and t time slots, therefore $n.T + t$ is the duration the of free channel and probability of this event is P_{free}. Equation (4.3.1.30) shows the calculation of E_{free} where k out of j nodes which are competing for the media at the same time when there is no backoff mechanism. In this case there is high probability of collision. By using backoff mechanism the probability that one node send data which given by equation (4.3.1.30) can be used to find the probability P_{free} . The average channel free probability:

$$E_{free} = \sum_{n=0}^{\infty}\sum_{t=1}^{T}\left(nT + t\right)P_{free} \tag{4.3.1.32}$$

Equation (4.3.1.32) illustrate the probability of the channel being busy between two free channel intervals due to successful transmission. To calculate the average duration of time interval for which the channel is busy between two free channel intervals E_{Busy}, first find the probability P_{busy}^{succ} of the event when the transmission is successful between two free channel intervals and the probability P_{busy}^{coll} of the event when there is collision between two free channel intervals.

$$E_{busy} = \sum_{n=0}^{\infty}\sum_{t=1}^{T}\left(\left(\frac{T}{2}nT + L_{DATA}\right)P_{busy}^{succ} + TP_{busy}^{pre}\right) \tag{4.3.1.33}$$

The probability of busy channel between two free channel intervals due to collisions of short preambles is given by Equation (4.3.1.35). Here are k out of j nodes which are competing for the medium at the same time when there is no backoff mechanism will cause collision. Thus using the same procedure as in Equation (4.3.1.35), but here are at least two k nodes wakeup at the same time so here use P_{busy}^{pre}.

$$P_{busy}^{succ} = \sum_{i=0}^{N-1}\sum_{j=1}^{N-i}\sum_{k=1}^{j}\binom{N}{i}\left(\frac{t}{T}\right)^{i}\left(\frac{T-W_m}{T}\right)^{N-1-i}\binom{N-i}{j}\left(\frac{1}{W_m}\right)^{j} \binom{j}{k}\left(1-\pi_0\right)^{k}\pi_0^{j-k}\left(\frac{t}{W_m}\right)^{j-k} \tag{4.3.1.34}$$

The average duration of busy channel between two free channel intervals P_{busy}^{coll}, define by the length of successful transmission time duration $(T/2+tData)$ multiplied by the probability of this event plus time duration of busy channel due to collisions of short preambles sending by control channel.

$$P_{busy}^{coll} = \sum_{i=0}^{N-2}\sum_{j=2}^{N-i}\sum_{k=2}^{j}\binom{N}{i}\left(\frac{t}{T}\right)^{i}\left(\frac{T-W_m}{T}\right)^{N-1-i}\binom{N-i}{j}\left(\frac{1}{W_m}\right)^{j} \binom{j}{k}\left(1-\pi_0\right)^{k}\pi_0^{j-k}\left(\frac{t}{W_m}\right)^{j-k} \tag{4.3.1.35}$$

Plugging equations (4.3.1.32) and (4.3.1.33) into (4.3.1.30), obtain $P_r(free)$. Then plugging equation (4.3.1.32) into (4.3.1.30), obtain $P_r(free|empty)$. Plugging equation (4.3.1.16) into (4.3.1.15), and (4.3.1.30) and (4.3.1.26) into (4.3.1.25), the probability for each node to successfully transmit a DATA packet P_s , and the probability for each node to encounter a collision P_f are solved as a function of the stationary probability of the empty queue state π_0. According to equation (4.3.1.36), the probability for each node to transmit a DATA packet p can also be obtained as a function of π_0 . Let function $h(.)$ describe the relationship between p and π_0

$$p = h(\pi_0) \tag{4.3.1.36}$$

Figure 4.4 show the graphical representation of contention window. Here, one cycle T is divided into two intervals $T_{affected}$ and $T_{unaffected}$ depending on whether or not awaken nodes can cause collisions and one time slot τ of cycle T is set to one tick of the backoff window while sets one time slot small enough as to synchronize transmissions. The contention window size define by a integer ranging from 1 to W_{max}. The equations-18 has derived by using binomial probability formula. In this equation, N= Total number of trail, T_a =Number of specific events that want to obtain, W_m/T = The probability of success and $(T-W_m)/T$ =The probability of failure according to the binomial probability formula.

Figure 4.5 show the graphical representation of contention window affected area. This equation also derived by using binomial probability formula. In this equation, n_i = Total number of trail, i^{th} slot = Number of specific events that want to obtain, $1/W_m$ = The probability of success and $(W_m - 1)/W_m$ = The probability of failure

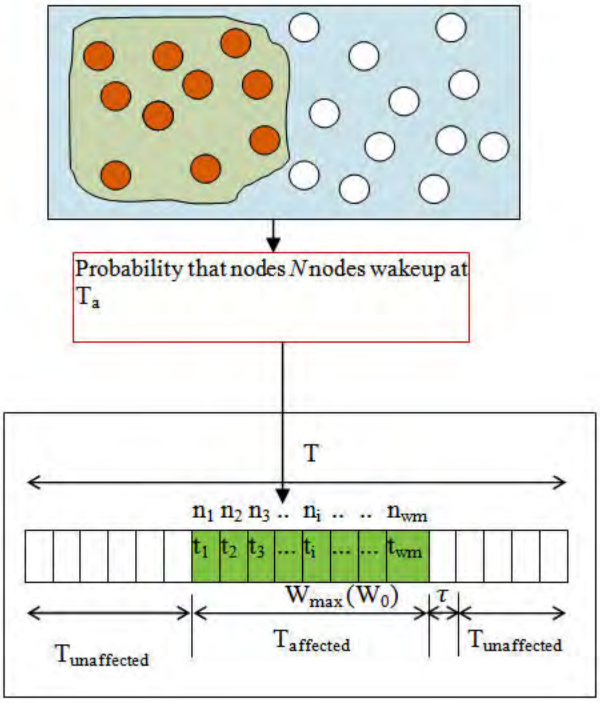


Figure 4.5: Graphical representation of contention window

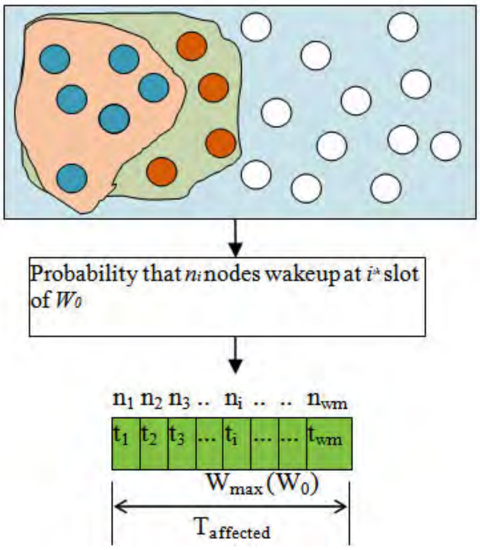


Figure 4.6: Graphical representation of contention window affected area

according to the binomial probability formula.

Figure 4.6 show the graphical representation of backoff time in contention window. This equation also derived by using binomial probability formula. In this equation, w_i = total number of trail, s_d = number of specific events that want to obtain, $(1-\pi_0)^{S_d}\left(\frac{W_{succ}}{W_m}\right)^{S_d}$ = the probability of success and $(\pi_0)^{(W_i-S_d)}$ = the probability of failure according to the binomial probability formula.

4.3.1.2 Energy Consumption in the Proposed MX-MAC

The average energy consumption can be calculated as average length of a successful transmission $T/2 + L_{data}$ times the probability of a successful transmission, plus receiver node transmits energy for preamble packets in addition to the sender node's preamble packets. However, receiver node does not need to transmit any preamble packet before it forwards the packet it received to the next hop receiver. For simplicity, it is assumed that the total energy required to exchange a data packet remains unchanged. Since MX-MAC eliminates the hidden node problem, no retransmission energy due to hidden node's collision is necessary. The only additional energy that is incurred in MX-MAC compared to X-MAC is its channel switching energy. The total energy consumption for the proposed MX-MAC is:

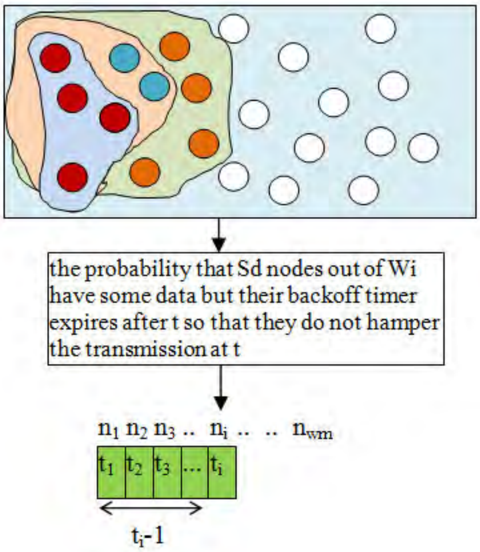


Figure 4.7: Graphical representation of backoff time in contention window

$$E_{(MX-MAC)} = E_s + E_r \tag{4.3.1.37}$$

E_s = Expected energy to send a packet and Er=Expected energy to receive a packet

Expected energy (Es) to send a packet:

E_s =(preamble energy + energy per ACK listen) × (expected preamble-listen iterations required) +(energy to send packet)

$$= (P_{Tx}S_p + P_{Rx}S_{al})\left(\frac{1}{\frac{R_l - S_p}{R_l + S_p}}\right) + S_dP_{Tx} \tag{4.3.1.38}$$

$$= \frac{((P_{Tx}S_p + P_{Rx}S_{al})(R_l + S_p))}{(R_l - S_p)} + S_dP_{Tx} \tag{4.3.1.39}$$

The expected number of preamble listen iterations required, giving:

$$E_S = \frac{(P_{Tx}S_p + P_{Rx}Sal)}{\left(1 - (1 - P_dR_{qpl})(1 - \frac{R_l - S_p}{R_l + R_s})\right)} + S_dP_{Tx} \tag{4.3.1.40}$$

The expected energy to receive a packet is given by: E_r =(listen cycle energy +

Table 4.9: Energy consumption analysis variables for MX-MAC

Symbol	**Meaning**
$P_d(t)$	Probability of packet arrival ration
P_{Tx}	Power required to transmit
P_{Rx}	Power required to receive
P_s	Power required to sleep
S_d	Duration of the sender's data transmission periods
R_l	Receiver listen periods

Table 4.10: Mathematical model variables for MX-MAC

Symbol	**Meaning**
S_p	Duration of the sender's preamble
S_{al}	Duration of the Acknowledgement listen
R_l	Receiver listen period
R_s	Receiver sleep period
t_d	Data transmission period for a single packet
R_a	Duration of the receiver's ACK period
R_d	Duration of the receiver's data period

sleep cycle energy) × (expected iterations for a preamble to arrive)+(energy to send an ACK)+(energy to receive packet)

$$\frac{(P_s R_s + P_(R_x) R_l)}{(1-(1-P_d(t)))^{(R_l+R_s)}} + P_{Tx} R_a + R_d P_{Rx} \tag{4.3.1.41}$$

4.3.1.3 Latency in the Proposed MX-MAC

There are many factors affecting latency of the network. Factors such as overhearing, over-emitting, collisions, and control packets overhead affect energy saving as well as forwarding delay [44]. On the other hand, low duty-cycling and sleep mode of a radio cause end-to-end delay degradation.

The expected latency for a single hop data exchange is given by [4]:

Lat = (duration of preamble + ACK listen) × (expected number of iterations required) + duration to send a data packet

$$= \left(\frac{1}{\frac{R_l - S_p}{R_l + R_s}}\right) \times \left(S_p + S_{al}\right) + S_d \tag{4.3.1.42}$$

$$= \frac{\left(R_l + R_s\right)}{\left(R_l - S_p\right)} \times \left(S_p + S_{al}\right) + S_d \tag{4.3.1.43}$$

For n packets with h hop count, the only sustained latency due to preamble packet will be only for once and the delay due to subsequent preambles will not be accounted for as the receiver node sends preambles during its packet reception and does not need to send any preamble before forwarding the received packet to the next hop receiver. Therefore, the total latency for the proposed scheme is derived as

$$Lat_{MX-MAC} = t_{contention} + ENIR \times (t_p + t_{al}) + t_d \times n \times h \tag{4.3.1.44}$$

$$ENIR = \frac{\left(R_l + R_s\right)}{\left(R_l - S_p\right)} \tag{4.3.1.45}$$

ENIR=Expected Number of Iteration Required

4.4 Summary

In this chapter Analytical Modeling of Existing protocols X-MAC, X-MAC/CA, ZeroMAC, MCAS-MAC and Proposed MAC Protocol , MX-MAC protocol described. The proposed Markov model and throughput analysis can be used to the estimate the throughput of MX-MAC under various conditions like traffic load, number of nodes, arrival rate.

Chapter 5

Results and Performance Analysis

5.1 Introduction

For performance evaluation purpose, different network configurations have been considered. Each configuration is characterized by two parameters: number of nodes and node density. The number of nodes ranges from 50 to 200, whereas the node density ranges from 10 to 20. The nodes are randomly deployed over the network area. The proposed MAC protocol exhibits very good performances as illustrated in the following Figures. This is due to the fact that it uses multichannel [20] instead of single channel and the novel concept of duty cycling.

5.2 Mathematical Modeling Parameters

In this section, the analytical model developed in the previous sections is used to compare the performance of MX-MAC with X-MAC, X-MAC/CA, ZeroMAC and MCAS-MAC with respect to energy consumption, latency under varying packet numbers, hop counts and collision due to hidden node problem. The table 5.1 shows the parameters used for throughput analysis.

5.3 Throughput Analysis

In this section, the results of the experiments are presented to evaluate the proposed protocol. The proposed algorithm MX-MAC is compared with four different algorithms, mentioned above. To evaluate the protocol several experiments has been carried out with Matlab tool . Analytical results it is evident that the proposed protocol performs better than X-MAC, X-MAC/CA, ZeroMAC and MCAS-MAC in most of the situations.

Table 5.1: Parameters used for throughput analysis

Symbol	**Meaning**	**Value**
N	Total Number of Nodes	200
T	Length of Cycle	100
F	Frame Size	50 Bytes
P_{TX}	Power required to transmit	86.2 mW
P_{RX}	Power required to receive	96.6 mW
P_S	Power required to sleep	0.0183 mW
P_{TX}	Channel switching power	200 mW
S_P	Duration of the sender's preamble	1.5 ms
R_a	Duration of the receiver's ACK period	1.84 ms
S_d	Data transmission period	3.8 ms
S_{al}	Duration of the Acknowledgement listen	15.25 ms
R_l	Receiver listen period	1.98 ms

5.3.1 Throughput Analysis with Respect to Number of Nodes

Figure 5.1 depicts throughput analysis with respect to number of nodes. This analysis shows how the MX-MAC and X-MAC, X-MAC/CA, ZeroMAC, MCAS-MAC throughput differs with the varying number of nodes. If number of nodes varies from 2 to 20, the throughput of the proposed MX-MAC performs better than the others related protocol for throughput analysis. Figure 5.1 is drawn with the help of the equation (4.3.1.9) and parameters used are described in Table 5.1.

5.3.2 Throughput Analysis with Respect to Data Arrival Rate

Figure 5.2 shows an average throughput as function of empty queue state (π_0). The MX-MAC protocol gives the highest value of average throughput as compared to the others protocol. From this figure it is observed that throughput is higher compared to other algorithms. In figure 5.2 the throughput obtained from the Markov model

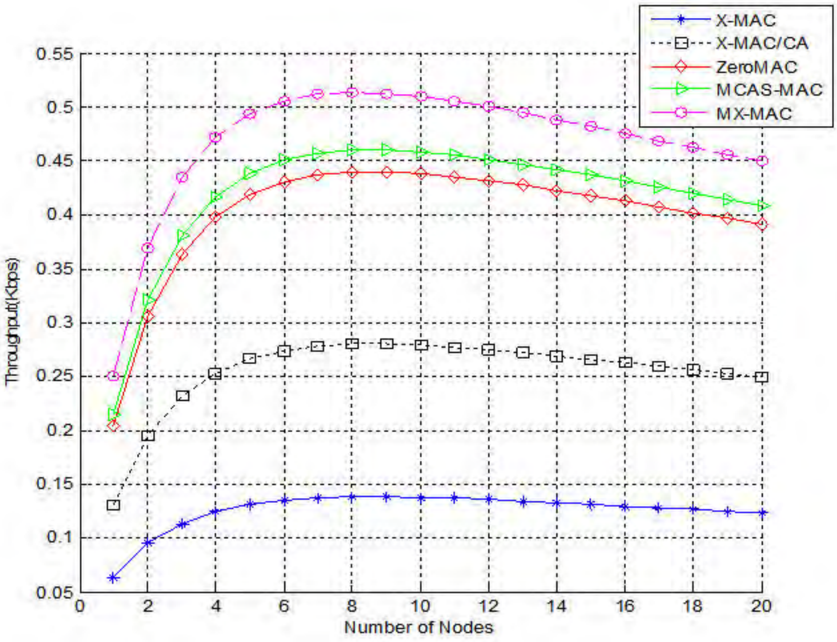


Figure 5.1: Throughput(Kbps) with respect to number of nodes

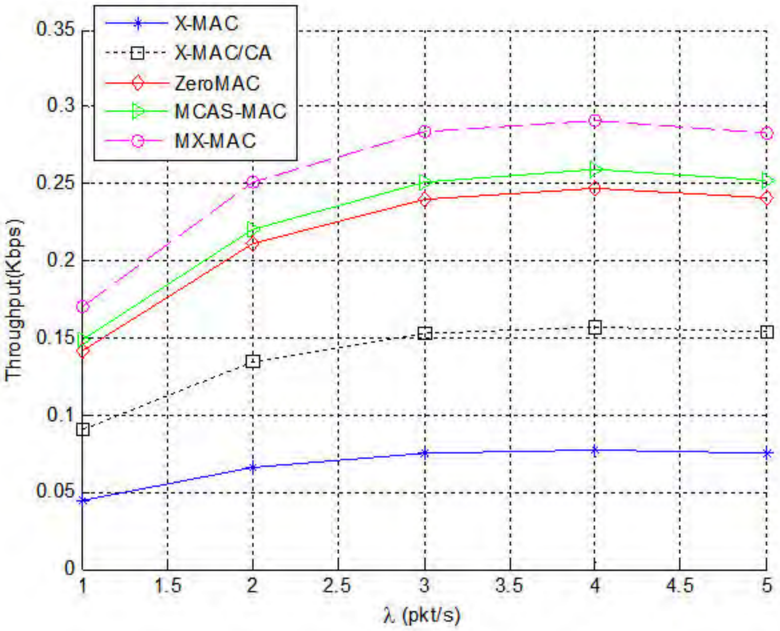


Figure 5.2: Throughput with respect to data arrival rate

for MX-MAC without retransmissions under different packet arrival rate.

5.4 Energy Consumption Analysis

In this section, the energy consumption of proposed scheme is compared with X-MAC, X-MAC/CA, ZeroMAC, MCAS-MAC by evaluating equation (4.3.1.37) and

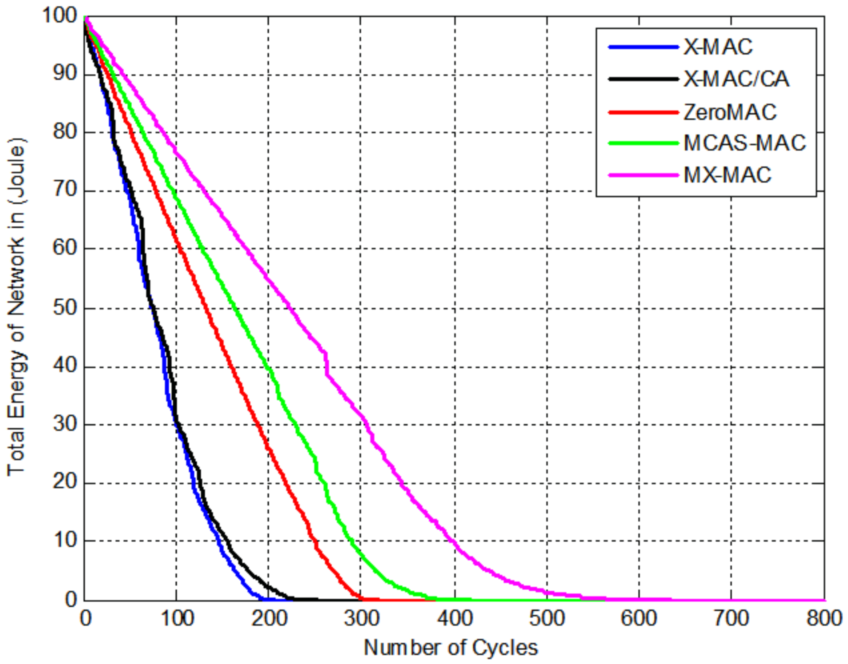


Figure 5.3: Total energy of network with respect to number cycles varying number of hops

(4.3.1.41).

5.4.1 Total Energy of Network with Respect to Number of Cycles Varying Number of Hop

The energy spending rate for sensors in a wireless sensor network varies based on the medium access rule of the sensors used for communications.The figure 5.3 presents total energy of the network as a function of the number of cycles. In this case number of hop varying from 0 to 20. The total energy of the network remains highest in case of MX-MAC protocol.

5.4.2 Total Energy of Network with Respect to Number of Cycles Varying Contention Window Size

To show the total energy losses with respect to number of cycles in the network the contention window size has been considered. The total energy measured in terms of mJ. As seen in figure 5.4, the total energy of the network decreases when number of cycles increases. In this case contention window size varying from 0 to 10. For MX-MAC, energy consumption remains relatively lowest as number no. cycles

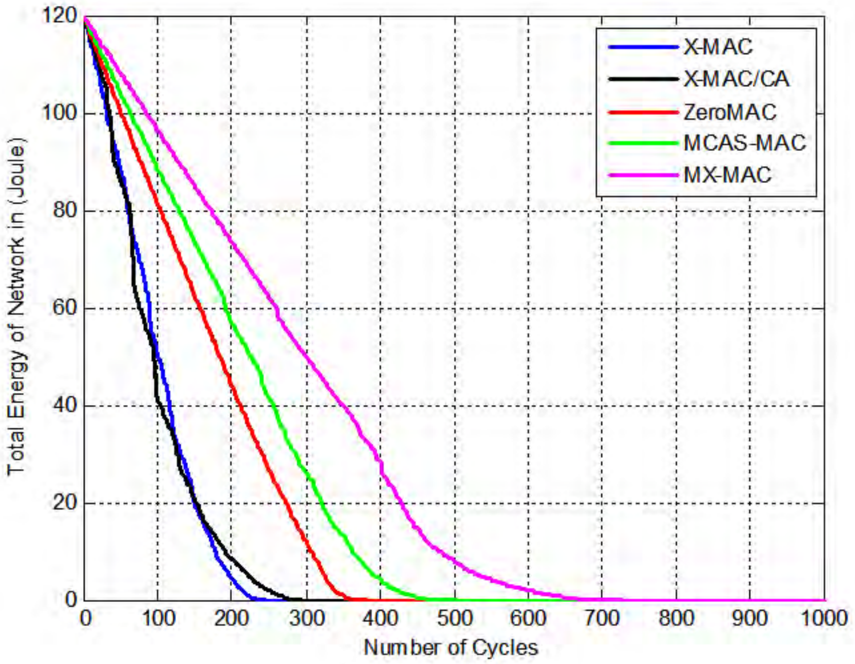


Figure 5.4: Total energy of network with respect to number of cycles varying contention window size

increases.

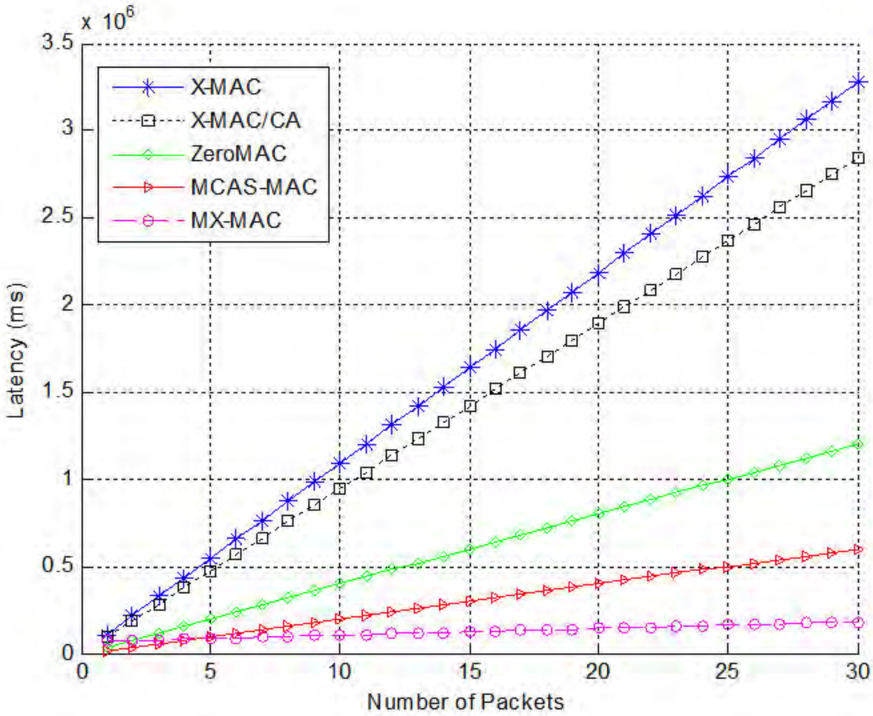


Figure 5.5: Latency(ms) with respect to number of packets

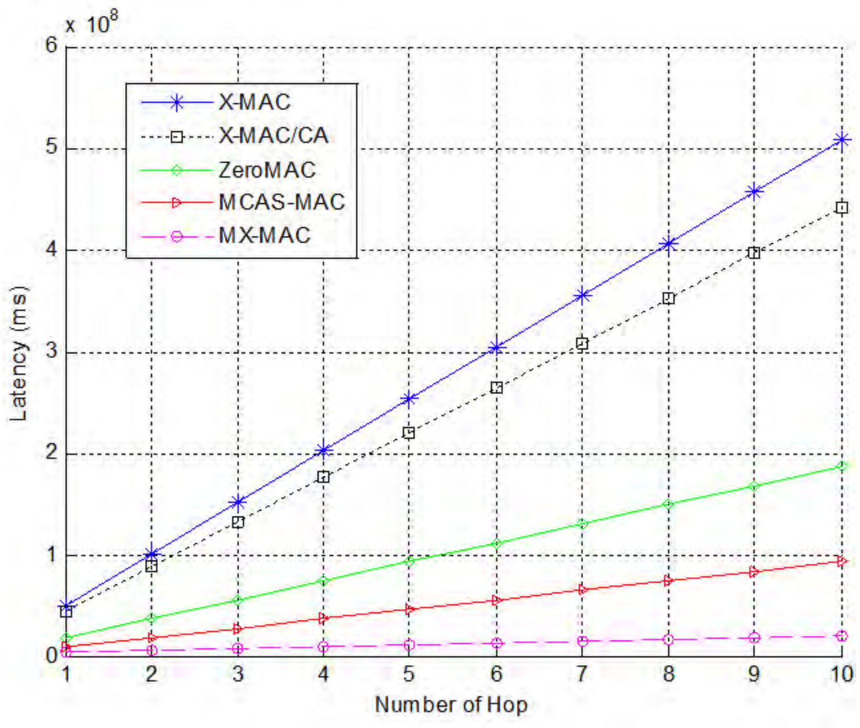


Figure 5.6: Latency(ms) with respect to number of hops

5.5 Latency

5.5.1 Latency with Respect to Number of Packets

Figure 5.5 shows the latency with respect to no. of packets. The packet generation is varied from 1 to 30. As shown in the figure MX-MAC achieves low latency with respect to X-MAC, X-MAC/CA, ZeroMAC and MCAS-MAC for all scenarios.

5.5.2 Latency with Respect to Number of Hops

Fig. 5.6 shows the average packet latency with respect to number of hop. As shown in the figure, MX-MAC outperforms to X-MAC, X-MAC/CA ZeroMAC and MCAS-MAC for all cases. MX-MAC does not suffer from the multi hop [45] delay since a receiver node can make ready the next hop receiver node. As the packet generation interval increases, MX-MAC shows better results .

5.6 Network Lifetime

Figure 5.7 to 5.9 illustrate the allotment of the total energy of the network in Joule with respect to time. This figure without a doubt shows that the proposed algorithm

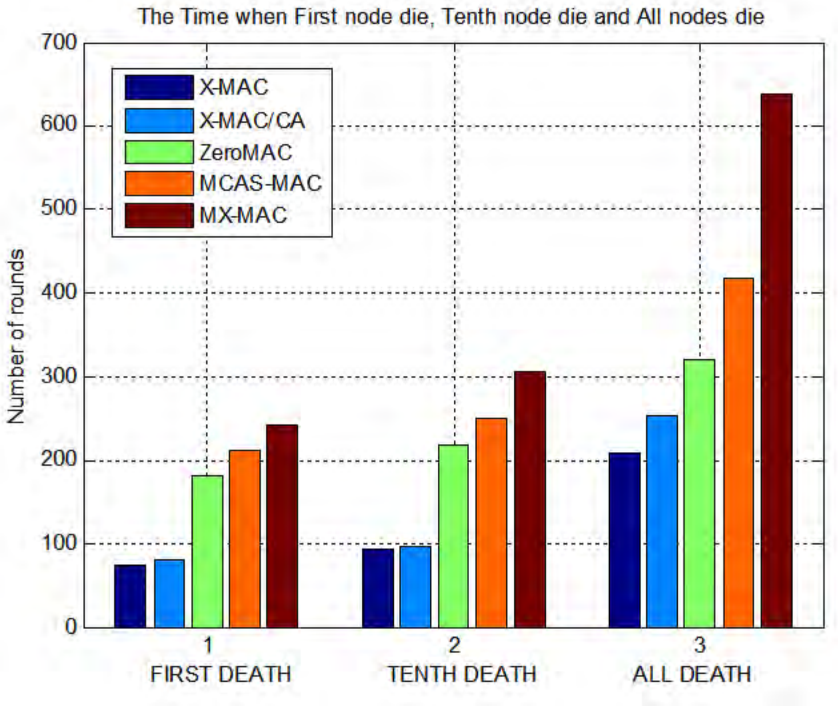


Figure 5.7: Total energy of the network with respect to number of cycles

is more energy efficient technique than the other algorithms, as the sensor node energy remains more than the other techniques.

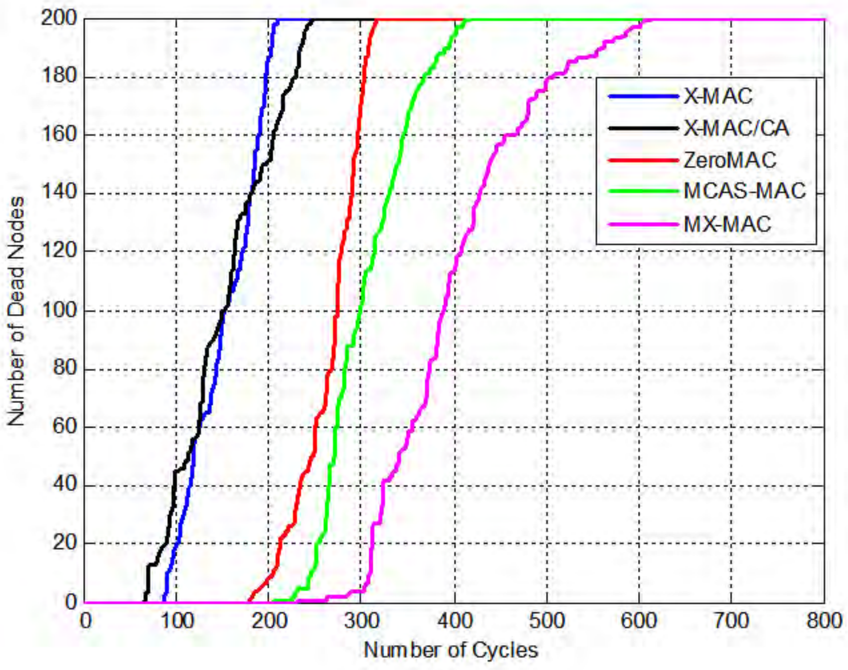


Figure 5.8: Number of dead nodes with respect to number of cycles

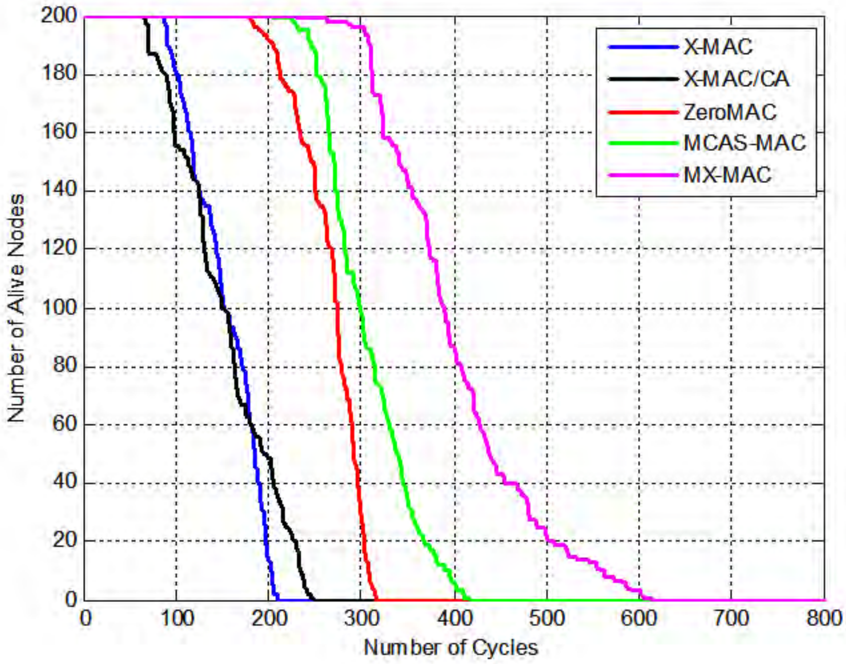


Figure 5.9: Number of alive nodes with respect to number of cycles

5.6.1 Total Energy with Respect Number of Cycles

Figure 5.7 illustrates the allotment of the total energy of the network in Joule with respect to time. This figure unambiguously shows that the proposed algorithm is more energy efficient technique than the other algorithms, since sensor node energy remains more compare to other techniques.

5.6.2 Number of Dead Node with Respect to Number of Cycles

Figure 5.8 illustrates the circulation of the number of dead sensor nodes with respect to number of cycles for each algorithm. This figure obviously shows that death rate is shorter with respect to other algorithm. Before the number of cycles 250, the death rate is almost zero and is better then other techniques until the number of cycles 350. This figure evidently shows that the lifetime of the proposed algorithm is more than the other algorithms, as more sensor node alive than the other technique.

5.6.3 Number of Alive Node with Respect to Number of Cycles

Figure 5.9 illustrates the distribution of the number of alive sensor nodes with respect to number of cycles for all algorithms. This figure perceptibly shows that alive rate is higher with respect to other algorithm. Before the number of cycles 250, the alive rate is maximum and it remains better with respect to other techniques until the time unit 350.

5.7 Summary

In this chapter, results and performance of proposed MX-MAC protocol have been described. Here the analytical results are evaluated. The proposed MX-MAC protocol shows better performance in terms of energy consumption and latency. This characteristic is more evidently observed for hidden node conditions.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

An energy efficient MAC protocol design is a challenging task in wireless sensor networks. This paper presents MX-MAC; a novel energy efficient multi channel X-MAC based medium access control protocol for wireless sensor networks to exclude the hidden node problem of X-MAC protocol, a popular asynchronous medium access control (MAC) protocol for wireless sensor networks as well as to reduce the average packet delay, thereby improving the lifetime and QoS of sensor networks. There are three novel contributions in this paper. Firstly it introduces the idea of using two channels in asynchronous MAC protocol which has been described in chapter four in details. One channel is used for passing control information and another channel is used for data transmission. In order to carry out the performance analysis of the proposed Multi Channel based X-MAC (MX-MAC) protocol, a simple analytical model has been developed in chapter four which includes equations for energy consumption as well as latency. Based on this model the performance of the proposed MX-MAC protocol is compared with some other popular asynchronous MAC protocols for WSNs in chapter four to show the efficiency of the proposed MAC protocol. Analytical results show that the proposed MX-MAC protocol can significantly reduce the energy consumption caused by the retransmission of data packets and it has lower energy consumption in case of high traffic than X-MAC. Furthermore, it is shown that the average packet delay can be effectively reduced by using "Next Receiver ID" information in the receiver's preamble packets. It is observed that MX-MAC significantly contributes to make asynchronous X-MAC protocol more energy efficient. The results obtained from this research are promising and show the suitability of this proposed MX-MAC protocol to reduce energy consumption as well as average packet delay in WSN effectively.

6.2 Future Work

Multi-radio Multi-channel has been applied in this research work to solve the hidden terminal problem in asynchronous MAC protocol. From the result of analytical model it is evident that in case of high traffic MX-MAC saves more energy. But it's performance is worse when the network traffic is low. Despite the low latency and low energy consumption by MX-MAC, there are many concerns left open for future work as well. One of the issues is the exploration of the factors that will help to reduce the collisions in control channel. The use of dual channel in the asynchronous MAC protocol for wireless sensor networks are explored. In future a simulation model of MX-MAC protocol will be developed. Comparisons of both analytical and simulation model of the MX-MAC will be also investigated. The proposed method still have control channel hidden terminal problem. In future the idea to reduce the collision in control channel will be addressed.

Bibliography

[1] J. Mo, H.-S. W. So, and J. Walrand, “Comparison of multichannel mac protocols,” *IEEE Transactions on mobile computing*, vol. 7, no. 1, pp. 50–65, 2008.

[2] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 95–107, ACM, 2004.

[3] A. El-Hoiydi, J.-D. Decotignie, and J. Hernandez, “Low power MAC protocols for infrastructure wireless sensor networks,” in *Proceedings of the fifth European wireless conference*, pp. 563–569, 2004.

[4] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks,” in *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 307–320, ACM, 2006.

[5] A. Ullah, J.-S. Ahn, and G. Kim, “X-MAC protocol with collision avoidance algorithm,” in *Ubiquitous and Future Networks (ICUFN), Fifth International Conference on*, pp. 228–233, IEEE, 2013.

[6] S. H. Lee, Y. S. Bae, and L. Choi, “On-demand radio wave sensor for wireless sensor networks: towards a zero idle listening and zero sleep delay MAC protocol,” in *Global Communications Conference (GLOBECOM),IEEE*, pp. 560–566, IEEE, 2012.

[7] J. B. Lim, B. Jang, and M. L. Sichitiu, “MCAS-MAC: A multichannel asynchronous scheduled MAC protocol for wireless sensor networks,” *Computer Communications*, vol. 56, pp. 98–107, 2015.

[8] O. Y. W. B. Heinzelman, "Modeling and throughput analysis for X-MAC with a finite queue capacity," in *IEEE Global Telecommunications Conference GLOBECOM*, pp. 1–5, IEEE Conference Publications, 2010.

[9] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.

[10] P. Nand *et al.*, "Contention based energy efficient wireless sensor network—a survey," in *Computing, Communication & Automation (ICCCA),International Conference on*, pp. 546–551, IEEE, 2015.

[11] J. Kim, J. Min, and Y. Kwon, "Multi-channel mac protocol to improve network throughput and end-to-end delay in OFDMA-based wireless network systems," in *International Conference on Hybrid Information Technology*, pp. 98–104, Springer, 2011.

[12] H. Lei, Z. Ren, C. Gao, and Y. Guo, "A new multi-channel mac protocol for 802.11-based wireless mesh networks," in *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, vol. 1, pp. 27–31, IEEE, 2012.

[13] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "The evolution of MAC protocols in wireless sensor networks: A survey," *IEEE communications surveys & tutorials*, vol. 15, no. 1, pp. 101–120, 2013.

[14] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking (ToN)*, vol. 12, no. 3, pp. 493–506, 2004.

[15] T. Van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 171–180, ACM, 2003.

[16] V. V. Kapadia, S. N. Patel, and R. H. Jhaveri, "Comparative study of hidden node problem and solution using different techniques and protocols," *arXiv preprint arXiv:1003.4070*, 2010.

[17] X. Han, L. Shu, Y. Chen, and H. Zhou, “WX-MAC: an energy efficient MAC protocol for wireless sensor networks,” in *Mobile Ad-Hoc and Sensor Systems (MASS),IEEE 10th International Conference on*, pp. 423–424, IEEE, 2013.

[18] O. Yang and W. Heinzelman, “Modeling and throughput analysis for SMAC with a finite queue capacity,” in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP),5th International Conference on*, pp. 409–414, IEEE, 2009.

[19] R. K. Panta, J. A. Pelletier, and G. Vesonder, “Efficient asynchronous low power listening for wireless sensor networks,” in *Reliable Distributed Systems (SRDS),IEEE 31st Symposium on*, pp. 291–300, IEEE, 2012.

[20] S. Morshed, M. Baratchi, P. K. Mandal, and G. Heijenk, “A multi-channel multiple access scheme using frequency offsets—modelling and analysis,” in *Wireless and Mobile Computing, Networking and Communications (WiMob),IEEE 12th International Conference on*, pp. 1–7, IEEE, 2016.

[21] A. Kakria and T. C. Aseri, “Survey of synchronous MAC protocols for wireless sensor networks,” in *Engineering and Computational Sciences (RAECS), Recent Advances in*, pp. 1–4, IEEE, 2014.

[22] B. Jang, J. B. Lim, and M. L. Sichitiu, “An asynchronous scheduled MAC protocol for wireless sensor networks,” *Computer Networks*, vol. 57, no. 1, pp. 85–98, 2013.

[23] H. Li, Y. Cheng, C. Zhou, and W. Zhuang, “Routing metrics for minimizing end-to-end delay in multiradio multichannel wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2293–2303, 2013.

[24] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, “MAC essentials for wireless sensor networks,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.

[25] P. Karn, “MACA-a new channel access method for packet radio,” in *ARRL/CRRL Amateur radio 9th computer networking conference*, vol. 140, pp. 134–140, 1990.

[26] A. El-Hoiydi and J.-D. Decotignie, “Low power downlink MAC protocols for infrastructure wireless sensor networks,” *Mobile Networks and Applications*, vol. 10, no. 5, pp. 675–690, 2005.

[27] R. Jurdak, P. Baldi, and C. V. Lopes, “Energy-aware adaptive low power listening for sensor networks,” in *Proceedings of INSS*, pp. 24–29, 2005.

[28] R. Braynard, A. Silberstein, and C. Ellis, “Extending network lifetime using an automatically tuned energy-aware mac protocol,” in *European Workshop on Wireless Sensor Networks*, pp. 244–259, Springer, 2006.

[29] L. M. Feeney and M. Nilsson, “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1548–1557, IEEE, 2001.

[30] E. Shih, P. Bahl, and M. J. Sinclair, “Wake on wireless: An event driven energy saving strategy for battery operated devices,” in *Proceedings of the 8th annual international conference on Mobile computing and networking*, pp. 160–171, ACM, 2002.

[31] W. Ye, F. Silva, and J. Heidemann, “Ultra-low duty cycle MAC with scheduled channel polling,” in *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 321–334, ACM, 2006.

[32] S.-H. Hong and H.-k. Kim, “A multi-hop reservation method for end-to-end latency performance improvement in asynchronous MAC-based wireless sensor networks,” *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, 2009.

[33] G. Bianchi, “Performance analysis of the ieee 802.11 distributed coordination function,” *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, 2000.

[34] A. Keshavarzian, H. Lee, and L. Venkatraman, “Wakeup scheduling in wireless sensor networks,” in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pp. 322–333, ACM, 2006.

[35] S. Liang, Y. Tang, and Q. Zhu, “Passive wake-up scheme for wireless sensor networks,” in *Innovative Computing, Information and Control. Second International Conference on*, pp. 507–507, IEEE, 2007.

[36] A. Irandoost, S. Taheri, and A. Movaghar, “PL-MAC: Prolonging network lifetime with a MAC layer approach in wireless sensor networks,” in *Sensor Technologies and Applications. Second International Conference on*, pp. 109–114, IEEE, 2008.

[37] B. Jang, J. B. Lim, and M. L. Sichitiu, “AS-MAC: An asynchronous scheduled MAC protocol for wireless sensor networks,” in *Mobile Ad Hoc and Sensor Systems. 5th IEEE International Conference on*, pp. 434–441, IEEE, 2008.

[38] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, “Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless,” in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pp. 1–14, ACM, 2010.

[39] O. D. Incel, S. Dulman, and P. Jansen, “Multi-channel support for dense wireless sensor networking,” in *European Conference on Smart Sensing and Context*, pp. 1–14, Springer, 2006.

[40] A. G. Ruzzelli, G. O’Hare, R. Jurdak, and R. Tynan, “Advantages of dual channel MAC for wireless sensor networks,” in *Communication System Software and Middleware. Comsware. First International Conference on*, pp. 1–3, IEEE, 2006.

[41] J. Zhang, F. Naït-Abdesselam, and B. Bensaou, “Performance analysis of an energy efficient mac protocol for sensor networks,” in *Parallel Architectures, Algorithms, and Networks. International Symposium on*, pp. 254–259, IEEE, 2008.

[42] O. Yang and W. Heinzelman, “Modeling and performance analysis for duty-cycled MAC protocols with applications to S-MAC and X-MAC,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 905–921, 2012.

[43] S.-H. Hong and S.-Y. Jung, “A multi-hop reservation method for end-to-end latency performance improvement in asynchronous mac-based wireless sensor networks,” *Journal of the Korea Academia-Industrial cooperation Society*, vol. 11, no. 7, pp. 2638–2647, 2010.

[44] C. Wang, Y. Sun, and H. Ma, “Analysis of data delivery delay in acoustic sensor networks,” in *Embedded and Ubiquitous Computing. IEEE/IFIP International Conference on*, vol. 1, pp. 283–287, IEEE, 2008.

[45] P. Murdiyat, K. S. Chung, and K. S. Chan, “A multi-channel mac for multi-hop wireless sensor networks minimizing hidden node collision,” in *Communications (APCC),22nd Asia-Pacific Conference on*, pp. 535–540, IEEE, 2016.