Bahria University
Islamabad

Low Latency Asynchronous Hybrid Sender And Receiver Initiated Duty Cycle MAC Protocol For Wireless Sensor Networks

By

Numan Mushtaq

01-244122-015

Supervisor: Dr. Shagufta Henna

Submitted For The Degree Of MS(Telecommunication And Networks)

Bahria University, Islamabad Campus

Department Of Computer Science

May 2014

Copyright©May, 2014 by Numan Mushtaq

ALL RIGHTS RESERVED

Declaration

I, Numan Mushtaq, declare that this thesis submitted in partial fulfillment of the requirements for the conferral of the degree MS(Telecommunication And Networks), from the Bahria University, Islamabad, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Numan Mushtaq

22nd September 2014

Abstract

Low Latency Asynchronous Hybrid Sender And Receiver Initiated Duty Cycle MAC Protocol For Wireless Sensor Networks

MS(Telecommunication And Networks)

By

Numan Mushtaq

Energy conversation is one of the primary objective of duty cycle MAC protocols. A number of synchronous and asynchronous duty cycle MAC protocols are proposed in the recent years. Especially recently proposed Asynchronous duty cycle MAC protocols are highly energy efficient, however they suffer significant packet delivery latency on wireless links given dynamic traffic loads in diverse networks.In our thesis, we present an asynchronous hybrid protocol called Hybrid MAC (H-MAC), which uses both sender and receiver initiated mechanisms to combat the packet delivery latency .In H-MAC each node schedules its sleep and wake up time based on cross layer routing information in receiver initiated part. In sender initiated part each sender chooses its wake up time based on the receiver's wake up information. We have evaluated H-MAC in diverse networks and heavy traffic loads, experiment reveal that, H-MAC significantly reduces packet delivery latency and energy consumption compared to RI-MAC. Furthermore H-MAC increases number of packet delivered in less time.

Acknowledgements

I simply bow my heads before Almighty Allah for giving me faith in my abilities and enabling me to accomplish this work and granting me with His Special Mercy, Blessings and unlimited help throughout the phases of the research work.

Foremost, I would like to express my sincere gratitude to my supervisor Dr. Shagufta Henna for the continuous support of my research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my MS study.

Last but not the least, I would like to thank my family: my parents and my siblings, for supporting me spiritually throughout my life.

Contents

Abstract **iv**

List of Tables **viii**

List of Figures **x**

Abstract **xi**

1 Introduction **1**

1.1 Duty cycling in WSNs 2

1.2 Motivation 3

1.3 Contribution 4

1.4 Organization 5

2 Related Work **6**

2.1 Synchronous duty cycle MAC Protocols 7

2.1.1 S-MAC 7

2.1.2 DW-MAC 8

2.1.3 R-MAC 10

2.2 Asynchronous duty cycle MAC protocols 10

2.2.1 Sender initiated duty cycle MAC protocols 12

2.2.2 Receiver initiated duty cycle MAC protocols 15

2.3 Summary 17

3 Design Of H-MAC **19**

3.1 Overview 19

3.2 Routing enhanced wake-up mechanism 20

3.2.1 Collision avoidance in REW mechanism 21

3.3 On demand sender initiated wake up mechanism 22

3.4 Beacon frame 24
3.5 Comparison of H-MAC with other duty cycle MAC protocols 24
3.6 Summary 25

4 Simulation And Evaluation 26
4.1 Simulation 26
4.2 Evaluation 27
4.2.1 Clique Networks 27
4.2.2 Chain Network 30
4.2.3 Grid Network 36
4.2.4 Random Network 39
4.2.5 Simulation with High Data Rate 42
4.3 Summary 44

5 Conclusion And Future Work 45

Bibliography 47

A Topologies 54
A.1 Clique Topology 54
A.1.1 25flows.TCL 54
A.2 Chain topology 56
A.2.1 25nodes.TCL 56
A.3 Grid topology 56
A.3.1 7x7RCE.TCL 56
A.4 Random Network with RCE model 50 nodes 58
A.4.1 Rand-network1.TCL 58
A.4.2 Rand-network2.TCL 60
A.4.3 Rand-network3.TCL 61

List of Tables

1.1 Energy consumption parameters in Telosb 3

3.1 Comparison of H-MAC with other duty cycle MAC protocols 25

4.1 Simulation parameters 26

List of Figures

1.1 An example of multi-hop WSN architecture 1
1.2 Idle listening in WSNs 2

2.1 Duty Cycle MAC protocols categorization 6
2.2 Synchronous duty cycling in WSNs 7
2.3 Adaptive listening in S-MAC 8
2.4 On demand forwarding in DW-MAC 9
2.5 Packet forwarding on multi-path in R-MAC 10
2.6 Asynchronous duty cycle schedule- Transmitter 11
2.7 Asynchronous duty cycle schedule- Receiver 11
2.8 Long preamble in B-MAC 12
2.9 Short preamble in X-MAC 13
2.10 Data transmission in UPMA-X-MAC 14
2.11 Receiver initiated transmission in RI-MAC 15
2.12 Wake up mechanism in REA-MAC 16

3.1 REW mechanism in H-MAC 21
3.2 On demand sender initiated wake up mechanism 23

4.1 Clique network with independent flows 27
4.2 Average PDR vs number of flows 28
4.3 Average duty cycle vs number of flows 29
4.4 Average latency vs number of flows 30
4.5 Chain network with n number of nodes 31
4.6 Average PDR vs number of nodes 31
4.7 Average Duty Cycle vs number of nodes 32
4.8 Average latency vs number of nodes 33
4.9 Average Duty Cycle vs packet generation interval 34

4.10 Average Latency vs packet generation interval 35
4.11 Average PDR vs generation interval 36
4.12 RCE model 37
4.13 Average Duty Cycle vs sensing range 37
4.14 Average latency vs sensing range 38
4.15 Average PDR vs sensing range 39
4.16 Average duty cycle vs sensing range 40
4.17 Average latency vs sensing range 41
4.18 Average PDR vs sensing range 42
4.19 Average Duty cycle vs sensing range 43
4.20 Average latency vs sensing range 44

Abbreviations

WSN	**Wireless Sensor Network**
H-MAC	**Hybrid MAC**
RI-MAC	**Receiver initiated MAC**
CCA	**Clear Channel Assessment**
LPL	**Low Power Listening**
BW	**Back Off**
FCF	**Frame Control Field**
FCS	**Frame Control Sequence**
PDR	**Packet Delivery Ratio**
RTS	**Request To Send**
CTS	**Clear To Send**
RCE	**Random Correlated Event**

Chapter 1

Introduction

Wireless sensor network (WSN) is a special type of ad hoc networks, where each node serves as a node may act as a source or destination or a router. A WSN consists of number of sensors nodes distributed across a geographical area and may be used for extracting data from the environment around them [1]. Several nodes may forward data to a base station via a sink node as shown in Figure 1.1. Typically a sink node has unlimited or fair amount of battery power provided. Each sensor has a small capability of processing. Sensor nodes

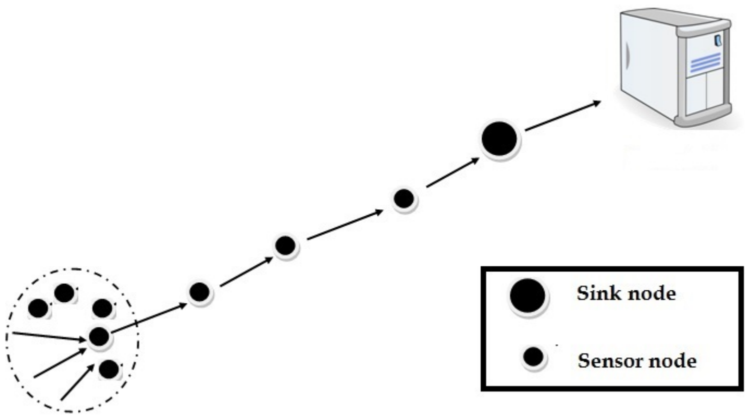

Figure 1.1: An example of multi-hop WSN architecture

can send and receive the data over a wireless links. Every sensor node consists of three basic parts, a radio transceiver, a small processing unit, and a battery. They are usually very low powered devices, working on a frequency range 300 to 1000 MHz on low supply voltage (2.1 V to 3.6 V). Their data rate is up to 76.8 kBaud [2]. Their data rate is up to 76.8 kBaud. WSNs are usually deployed to monitor the properties, such as pressure, temperature, vibration, movement and speed. Military applications were initial motivation for the deployment WSN [3], however with the passage of time they are being used in many industrial and civilian applications, such as environment monitoring, industrial processes, health monitoring and medical diagnosis. Sensor nodes usually rely on limited battery life to provide power source thus, energy efficiency in critically important for wireless sensor nodes [4].

1.1 Duty cycling in WSNs

Typically WSNs nodes have limited battery, which is not rechargeable. to design a WSN protocol, reducing energy consumption is the main goal. Idle listening is the main source of battery consumption in WSNs. In idle listening a node remains on high voltage and listens for packet transmission even when there is no communication on wireless channel [5]. An example is given below in Figure 1.2 to illustrate the concept of idle listening.

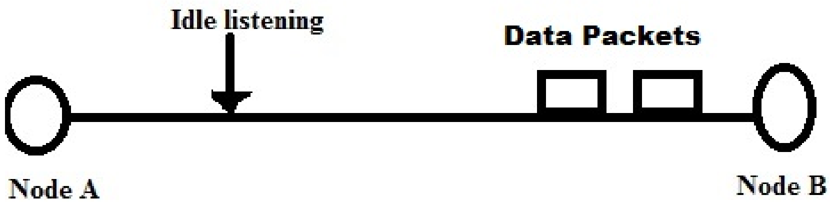

Figure 1.2: Idle listening in WSNs

Nodes A and B are wireless sensor nodes, Node A sends data 1 and data 2 to Node B over a wireless link. After finishing this communication both the nodes are still is listening mode. This listening when there is no communication between nodes is called idle listening, and it consumes the battery of sensor because still it is working on high voltages. Idle listening should be reduced in order to maximize the network life time.

Most existing radios, i.e., CC2420 [6] provide different modes in WSN, such as idle mode and sleep mode. In idle mode, a node does not communicate but still its radio is on and it consumes significant energy. In sleep mode a node turns its radio off, and it remains on very low power as shown in the Table 1.1. Thus, a better way to save energy is to put a sensor node on low radio, such as sleep state when it is not communicating with any other sensor node on the network. A typical energy consumption parameters for Telosb [7] are given in the Table 1.1.

Table 1.1: Energy consumption parameters in Telosb

Remarks	Power
Active mode	1.8 mA
Sleep mode	1 μA
Receiving	18.8 mA
Transmission	17.4 mA

Duty cycling is a significant and widely used mechanism to reduce the idle listening in WSNs. In duty cycling each node periodically switches between active and sleep mode to conserve the energy [8]. In active state a node can transmit and receive the packet data but in sleep state, a node turns its radio off in order to reduce the energy consumption [9]. MAC protocols based on duty cycling for WSNs can be classified into two main categories, synchronous and asynchronous. In synchronous duty cycle MAC protocols, such as S-MAC [5], T-MAC[10], R-MAC [11], SCP-MAC [12] and DW-MAC [13] neighboring nodes synchronize their selves in order to align their active and sleep periods. Sender and receiver can only communicate within a common wake up time. On the other hand asynchronous duty cycle MAC protocols, such as B-MAC[9], X-MAC [14], Wise-MAC [15], RI-MAC [16], REA-MAC [17] and PW-MAC [18] do not synchronize the neighboring nodes before sending or receiving the data. Each node is independent and has its own duty cycle schedule.

1.2 Motivation

Synchronous duty cycle approach requires synchronization among the nodes before starting the communication. Required synchronization adds extra overhead and complexity and nodes occupy the link for exchanging the control information. In dynamic and high traffic load, synchronous duty cycle MAC protocols experience low performance i.e., high energy consumption, longer latency, and low packet delivery ratio [16]. Asynchronous duty cycle MAC protocols are energy efficient, but they cause another challenge, called the time varying transmission latency. Each node has an independent and random wake up time in an operational cycle without considering the network topology and hop count to the destination resulting in significant packet delivery latency. An asynchronous duty cycle MAC protocol could be either sender initiated or receiver initiated. In sender initiated approach a sender transmits a preamble to notify that it has data to transmit. In receiver initiated approach when a sender has data to transmit it waits for the receiver to transmit a beacon packet. After the beacon transmission from the receiver announcing that it is in active state, the sender transmits the data packet. In the existing sender initiated asynchronous MAC protocols sender transmits long or short preamble to notify the packet transmission, in both cases preambles occupy the wireless link for long time which results in wastage of bandwidth. Furthermore sender node has to wait for the receiver to wake up in order to start communication, which results into longer packet delivery latency. In existing receiver initiated asynchronous MAC protocols, when data arrives at the sender, it waits silently for the receiver to send a beacon packet in order to notify that receiver is in active state and ready to start the transmission. On receiving the beacon sender starts the data transmission. This long wait of the sender node results into idle listening and it not only increases the energy consumption of the node but it also results in longer packet delivery latency. Motivated by the problem of packet delivery latency in diverse network topologies and random traffic loads, we present a new routing enhanced hybrid sender and receiver initiated MAC protocol called H-MAC.

H-MAC uses cross layer hop information to decide when a node wakes up in an operational cycle to forward the data packet like a pipeline on a multi-hop path. Furthermore H-MAC provides a hybrid receiver and sender initiated mechanism to reduce the time varying latencies in packet delivery. Receiver nodes can wake up multiple times in an operational cycle on the demand of sender nodes to minimize the packet delivery latency cased by idle listening of the sender nodes.

1.3 Contribution

In this thesis work, we present a new asynchronous hybrid sender and receiver initiated duty cycle MAC protocol for WSNs. Usually we observe dynamic bursty traffic in WSNs where the sensor nodes are resource constrained consisting of micro-controllers with limited processing power, low powered short range radios, and a limited powered battery. A duty cycle MAC protocol must be energy efficient and robust in dynamic traffic load and under any type of network. In our simulation we consider high traffic loads to evaluate the performance of H-MAC in diverse networks. Through the detailed simulations, results show that H-MAC outperforms RI-MAC under a wide range of traffic loads including high loads and bursty traffic. H-MAC experiences significantly low latency, and conserves energy compared to RI-MAC. Furthermore H-MAC improves packet delivery ratio compared to RI-MAC under diverse networks and high traffic loads. We simulated H-MAC in grid, chain, clique, and dynamic networks for performance evaluation.

1.4 Organization

The rest of the dissertation is organized as follows: We first summarize the related work in Chapter 2. Classification of duty cycle MAC protocols, and some duty cycle MAC protocols are explained in Chapter 2.

In Chapter 3 we describe the design of H-MAC protocol in detail. Overview of H-MAC is given in Section 3.1. Routing enhanced wake up mechanism is described in Section 3.2. On demand wake up mechanism is given in detail in Section 3.3.

Simulation environment and evaluation for H-MAC is given in Chapter 4. Simulation results for clique, chain, grid, and random networks are given in Chapter 4. Conclusion and future work is given in Chapter 5.

Chapter 2

Related Work

many synchronous and asynchronous protocols are proposed in the literature and they are energy efficient. Existing duty cycle MAC protocols are optimized for light traffic in WSN. Different approaches are used to improve the performance of these protocols. Duty cycle MAC protocols can be mainly categorized into two categories Synchronous and Asynchronous duty cycle MAC protocols. Duty cycle MAC protocols can be categorized as shown in the Figure 2.1. This chapter will be organized as shown in given in the Figure 2.1.

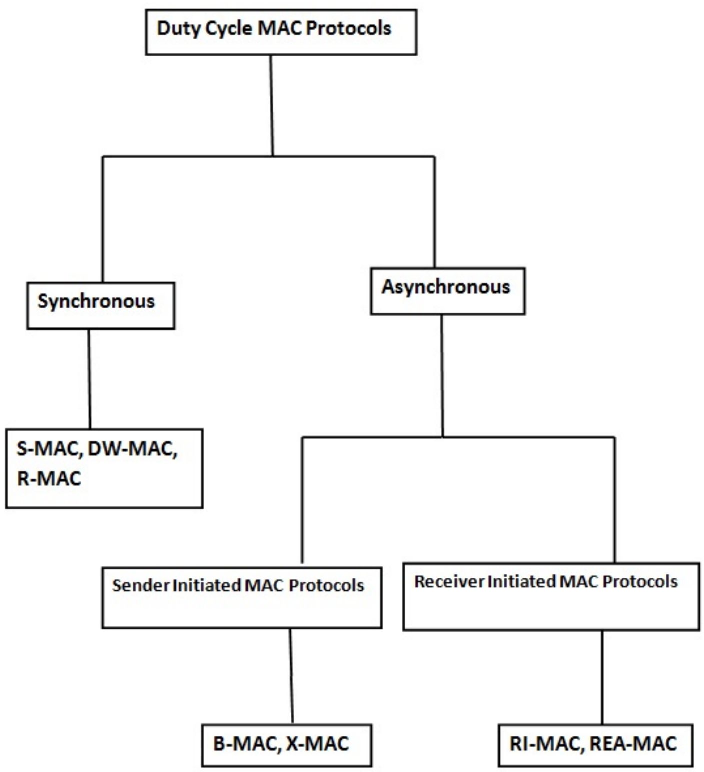

Figure 2.1: Duty Cycle MAC protocols categorization

2.1 Synchronous duty cycle MAC Protocols

A node can receive and and transmit the data when it is active. When in sleep mode a node completely turns its radio off in order to conserve the energy[19]. To transmit the data packet synchronous duty cycle MAC protocols use synchronization between the transmitter and receiver to align their active and sleep stats[20]. Figure 2.2 illustrates the

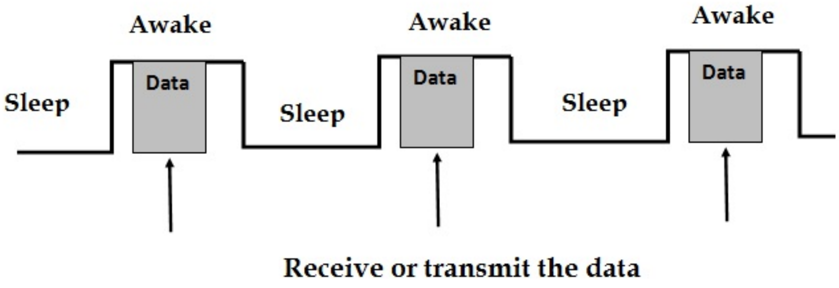

Figure 2.2: Synchronous duty cycling in WSNs

concept of synchronous duty cycling in WSN. Synchronous duty cycle protocols use clock synchronization in the network[21],[22]. All the nodes are aware of each other duty cycle schedules[23]. Nodes exchange the data in common active time[24]. In this section some synchronous duty cycle MAC protocols are described.

2.1.1 S-MAC

Ye et al.[5] proposed S-MAC it was one of the original synchronous duty cycle MAC protocols. In S-MAC neighboring nodes form clusters to periodically sleep. Nodes in the same clusters has same duty cycle schedule. In this protocol they used different sleep and active schedules in different clusters to minimize energy consumption, however they mentioned that per-node fairness and latencies are less important. In the original asynchronous protocols a long preamble was used to notify that there is a packet transmission in the radio link. We will explain in the later part of the chapter, how these long preambles introduces latency in packet delivery. This approach was energy efficient but it adds latency on per hop which results in significant network latency. S-MAC has also very low throughput because less packets are delivers in unit time. The developers of S-MAC later introduces a modification in the protocol called adaptive listening[25] to improve its packet delivery latency on multi-hop path. If a node overhears communication between other node e.i., RTS or CTS it waits until the overheated communication ends. After the communication ends the sender node can send the data immediately to neighbor node node, if it is its next hop rather than waiting for the Acknowledgment[26]. Figure 2.3 illustrates the concept

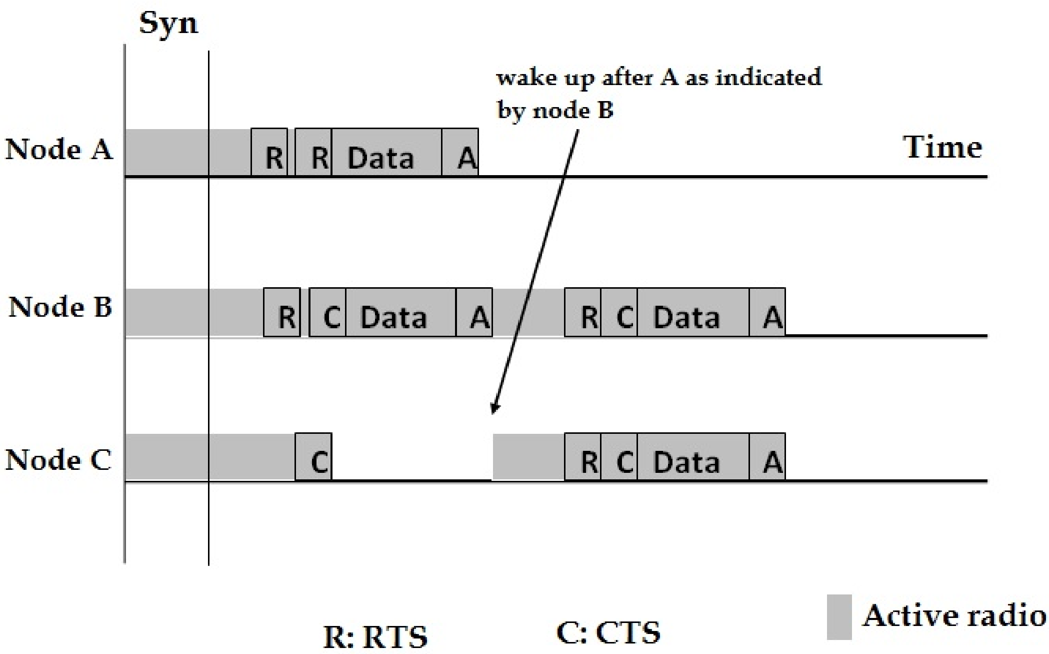

Figure 2.3: Adaptive listening in S-MAC

of adaptive listening in S-MAC. Node C listens for RTS,CTS from Node A and Node B. When Node B transmits CTS to Node B it mentions the total transmission time. Node C intermediately goes to sleep after hearing CTS from Node B. Node C adaptively wakes up after ack from node B, and transmits the data packet after exchanging RTS CTS with the Node B. Adaptive listing in S-MAC reduces packet delivery latency compared to simple S-MAC. Using adaptive listening S-MAC can transmit the data packet up to two hops in one operation cycle but generally it can not go beyond that in one operations cycle. Because when Node B transmits the data packet to Node C possibility some other Node C in is adaptively listening to some other node and it goes to sleep in that time, and fails to receive the CTS, the node might have missed hearing an RTS or CTS of another data transmission in the neighborhood, and the nodes starts transmission of packets may cause collision at other node. Adaptive listening in S-MAC has low packet delivery latency[25]. But listening for the overhearing data makes this protocol less energy efficient as illustrated in the Figure 2.3.

2.1.2 DW-MAC

Sun et al. [13] proposed a synchronous low latency, energy efficient Demand-Wakeup MAC Protocol (DW-MAC) for wireless sensor networks. Because WSNs are usually prone to burty nature of traffic, DW-MAC considers bursty and high traffic loads due to broadcast traffic. Number of sensor node report the data to sink node at the same time which forms broadcast type of traffic. Nodes wake up to receive data packet on demand to make it sure that nodes are not listening for the packets when there is no data packet to receive. Each time to forward a data packet a node has to send a wake up message fist to receiving node. Receiver wakes up after receiving the wake up message from the sender and then receives the data packet. Figure 2.4 illustrates the concept of on demand forwarding in

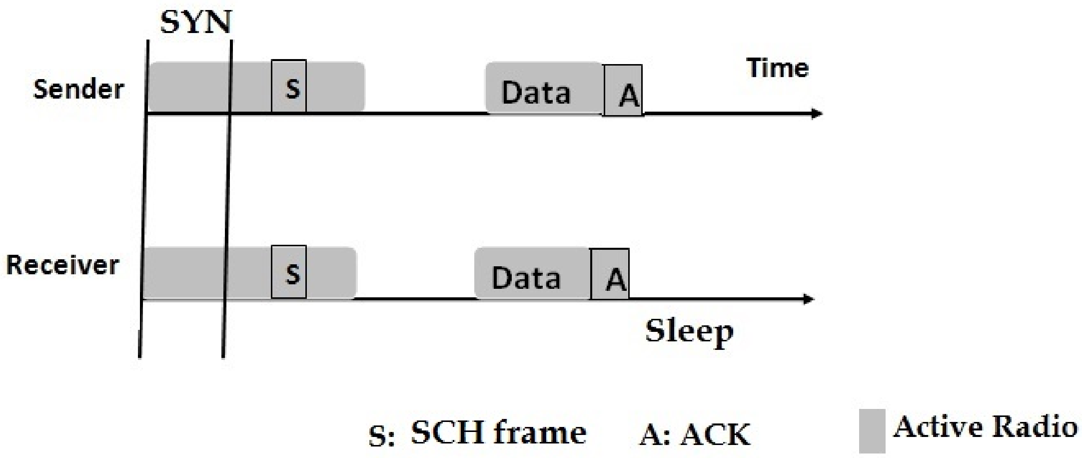

Figure 2.4: On demand forwarding in DW-MAC

DW-MAC. sleep time T_{sleep}, active time T_{active} and time needed to data transmission T_{data} are divided into different frames. Using these time frames both sender and receiver can predict, when to start the data transmission. After synchronization i.e., RTS CTS both sender and receiver wait for a SCH time frame and then start the data transmission based on T_{sleep}, T_{active} and T_{data}. SCH includes some cross-layer information[27]. For broadcast packet, SCH includes the network layer address of the sender and receiver. SCH frame help to identify the receiver the incoming packet has already been received on not. For unicast packets SCH includes the network layer address of last hop[28]. This approach significantly reduces the packet delivery latency at each hop, however DW-MAC introduces overhead for the purpose of synchronization and to send the wake up messages. When sender sends wake up message to the receiver, and another node C also want to communicate to the receiver node and transmits the wake up message. In this case collision can occur on receiver node.

2.1.3 R-MAC

Cross layer routing information is used in R-MAC[11] in order to avoid packet delivery latency. R-MAC is an synchronous protocol which uses hop information to relay the packet to the destination node. All the node along the path from source to destination sleep and wake up intelligently at scheduled time, so that each node can immediately forward the data packet along the path to the sink node as shown in the Figure 2.5.
In R-MAC cross layer routing information to make sensor nodes work in the lazy mode, keep their radios off as long as they can. after synchronization nodes can forward data according to cross layer routing information. To estimate the link multivariate Bernoulli link model[29] is used, which characterizes bursty links in low duty cycle WSNs.

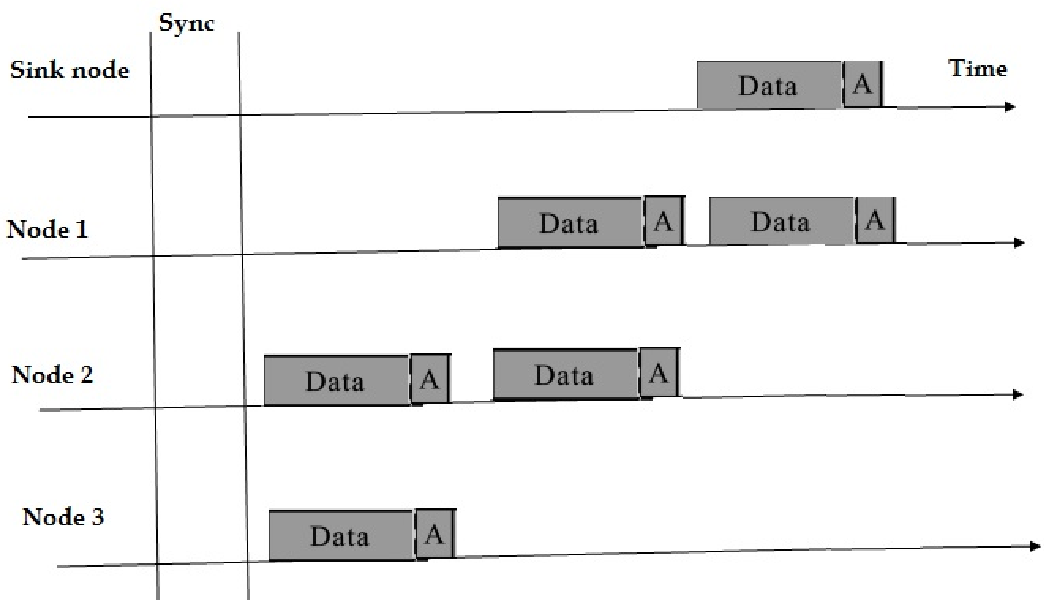

Figure 2.5: Packet forwarding on multi-path in R-MAC

Since R-MAC is a synchronous duty cycle MAC protocol, Each node needs to synchronize before data transmission as indicated in the Figure 2.5. Using cross layer routing information data packets can be forwarded on multi-path link. each node adaptively wakes up to receive the data packet. R-MAC reduces the packet delivery latency at each hop, however synchronization between the node adds extra overhead in the data transmission.

2.2 Asynchronous duty cycle MAC protocols

In asynchronous duty cycling each node has its own independent duty cycle schedule, and nodes do not rely on clock synchronous[30],[31]. Any node can transmit the data to any other awaking neighbor when ever it wants according to the MAC protocol in use[32].

Figure 2.6 shows how a transmitter transmits data packets using asynchronous duty cycle schedule. Figure 2.7 show how a receiver receives data packets during asynchronous duty cycle schedule. Nodes go to sleep and wake up periodically. since there is no clock

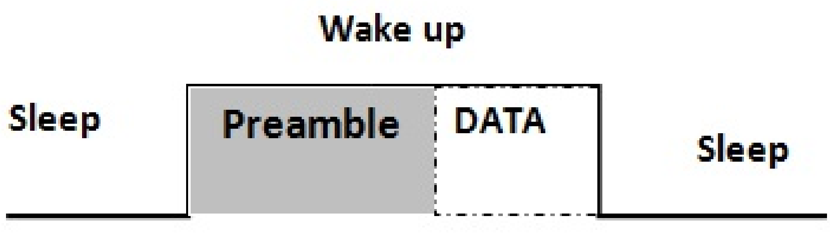

Figure 2.6: Asynchronous duty cycle schedule- Transmitter

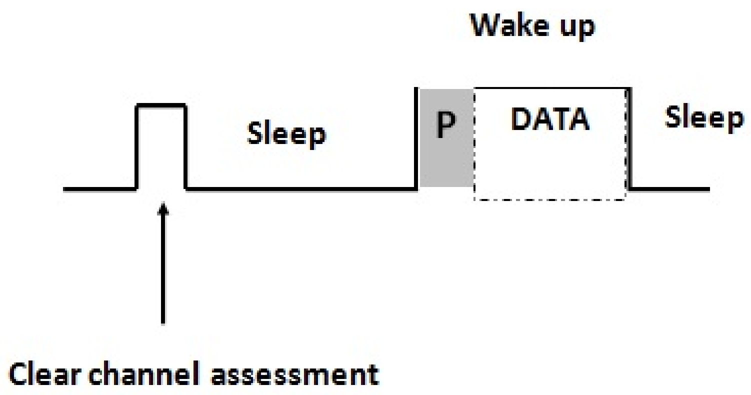

Figure 2.7: Asynchronous duty cycle schedule- Receiver

synchronization between transmitter and receiver, however there should be a mechanism to let the transmitter know that the receiver is in active state and ready to receive the data. Usually transmitter leaves a preamble to notify the receiver that there is a data transmission[33]. Preambles are packets containing the address of source and destination and other information depending on the MAC protocol in use.

There are two ways to start a data transmission in the asynchronous duty cycling either sender or receiver initiated. In sender initiated transmission sender transmits long preamble preamble in order to notify the receiver, there is a data transmission[34]. When receiver receives the long preamble it notifies the sender by transmitting an acknowledgment that it is ready to receive tha data, and then sender transmits the data packet. Short or strobed preambles are also used to notify the receiver about the data transmission[35], we will explain the mechanism of long and short preamble in the section 2.3. In receiver initiated approach receiver broadcasts a beacon packet to announce that it is in active state[36]. when a sender wants to transmit a data packet it waits silently until it receives beacon from receiver. After beacon transmitter sender the data packet[37].

2.2.1 Sender initiated duty cycle MAC protocols

B-MAC

Polastre et al.[9] have proposed an asynchronous duty cycle MAC protocol for wireless sensor networks called B-MAC. In B-MAC, each node has independent duty cycle schedule. Each node wakes up periodically and checks if there is any communication on the link, by using clear channel assessment(CCA). CCA is described in the section 2.2. If there is communication on the link the node remains active for possible incoming packets, and receives the data packet after acknowledgment from both the sender and receiver. If a node want to transmit the data packet, it has to leave a preamble to notify the receiver, that it has data transmission pending. The preamble lasts for longer than receiver's sleeping interval to make it sure that receiver does not miss the data packet as shown in the Figure 2.8. This long preamble increases the energy consumption of the nodes and it occupies the link for a long time, which results in significant energy consumption and less packets delivered to the destination. This energy consumption reduces the network lifetime and decreases QOS. Figure 2.8 illustrates the concept of long preamble used in B-MAC. When

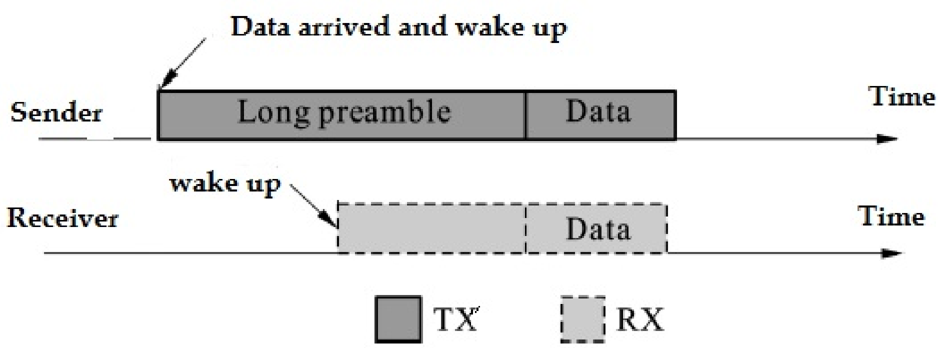

Figure 2.8: Long preamble in B-MAC

data arrives at the sender, it leaves long preamble to notify the receiver about the data packet. when receiver wakes up and receives the long preamble. After long preamble receiver can receive the data. The size of the preamble is always longer than the sleep period of the receiver. These longer duty cycle duration result into significant energy consumption. Developers of B-MAC used periodic channel sampling, and they called it low-power listening (LPL). They introduced LPL to enable low power communication without explicit synchronization. The receiver only wakes up for short time interval, however sender wake up time increases the energy consumption and reduces the network lifetime.

In B-MAC each node hearing for the preamble transmission must stay active until the end of the preamble to find out, which node is subjected to receive the data[38]. Nodes which are not subjected to receive the data overhear for the preamble, this overhearing not just only causes extra overhead but it causes significant energy consumption and packet delivery latency[39].

X-MAC

X-MAC proposed by Micheal et al.[14] solves the problem of overhearing in B-MAC by introducing short strobed preamble[40] or preamble sampling[41] instead of using long preamble as shown in Figure 2. sender imbeds the target address in strobed preambles. This short preamble notifies all the neighbor nodes, that only intended receiver has to listen to the data transmission. All nodes which are not subjected to receive the data go to sleep immediately after receiving the short preamble. When receiver wakes up and finds out that the data packet is subjected to it, it sends an early acknowledgment, so that sender stop the preamble transmission and start the data transmission. After acknowledgment from the receiver, the sender transmits the data packet. X-MAC also reduces per hop packet delivery latency caused by long preamble in wireless sensor networks. Figure 2.9

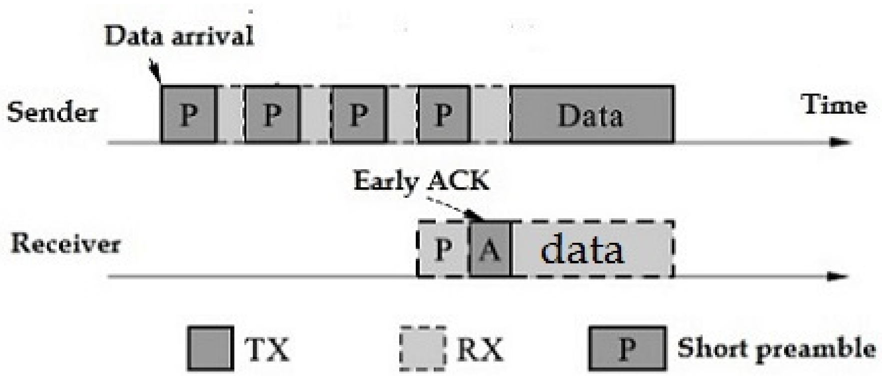

Figure 2.9: Short preamble in X-MAC

illustrates the concept of short preamble transmission in X-MAC. Seder transmits short preambles imbedded with target address until receiver notifies that is has receives the preamble. When receiver receives the short preamble and finds its own address imbedded in the preamble packet, It sends an early acknowledgment to the sender. Sender, on receiving acknowledgment stops sending preambles and starts data transmission. After completing

the data transmission both sender and receiver can go to sleep state again.
X-MAC significantly reduces the energy consumption and per hop packet delivery latency compared to B-MAC, however these short preambles occupy the wireless link for long time which causes lower delivery of packets in one operational cycle. These short preambles cause extra overhead at the wireless link. Seder nodes wait for the receiver and continuously sends the preamble, which causes idle listening at sender node, this idle listening minimizes the network lifetime.

UPMA-X-MAC

A variation in X-MAC is proposed called UPMA-X-MAC[42]. In this protocol the preamble part is removed and sender keeps sending data bytes to the receiver instead of preamble packets. Each node keeps listening for the channel using CCA check for incoming data. When data arrives at the sender it keeps sending the data packets on the wireless link. Target address is imbedded with each data packet header same as in X-MAC. When receiver finds out that it is subjected to receive the data packet in receives the data and sends an acknowledgment to the sender and sender stops sending data as shown in Figure 2.10. In UPMA-X-MAC receiver do not need to send an early acknowledgment to the sender. Figure 2.10 shows that strobed preambles used in X-MAC are replaced by a

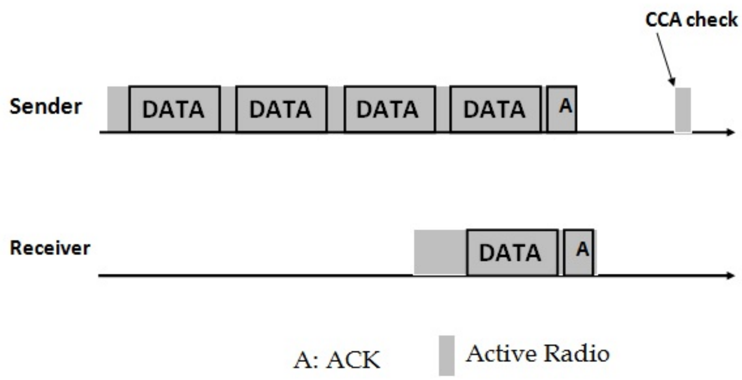

Figure 2.10: Data transmission in UPMA-X-MAC

chain of data frame transmissions in UPMA-X-MAC. Sender keeps sending chain of data frames until receiver sends an acknowledgment, indicating that it has received the data and now the sender can stop the data transmission. UPMA-X-MAC has reduced the time consumed in the transmission of preambles and an early acknowledgment, however data frame has more bytes in it compared to short preamble transmitting data frame continuously increases the overhead at wireless channel. Furthermore the idle listening of the sender is still unattended in UPMA-X-MAC.

2.2.2 Receiver initiated duty cycle MAC protocols

RI-MAC

RI-MAC used the idea of receiver initiated transmission [43][44] in WSNs. Each node wakes up periodically based on its own independent duty cycle schedule and transmits a beacon packet announcing that it is in active state[45]. When a sender wants to transmit a data packet to a receiver it waits silently and starts data transmission upon receiving beacon from the receiver.After completion of first data packet receiver transmits another beacon, this beacon serves as an acknowledgment. If sender has another data packet it transmits the data if not then both the sender and receiver go back to sleep state in order to conserve the energy. Beacon has dual role in RI-MAC. It serves as request for the data transmission initiation, and it also serves as acknowledgment after completion of data transmission. Figure 2.11 illustrates the concept of receiver initiated transmission in RI-

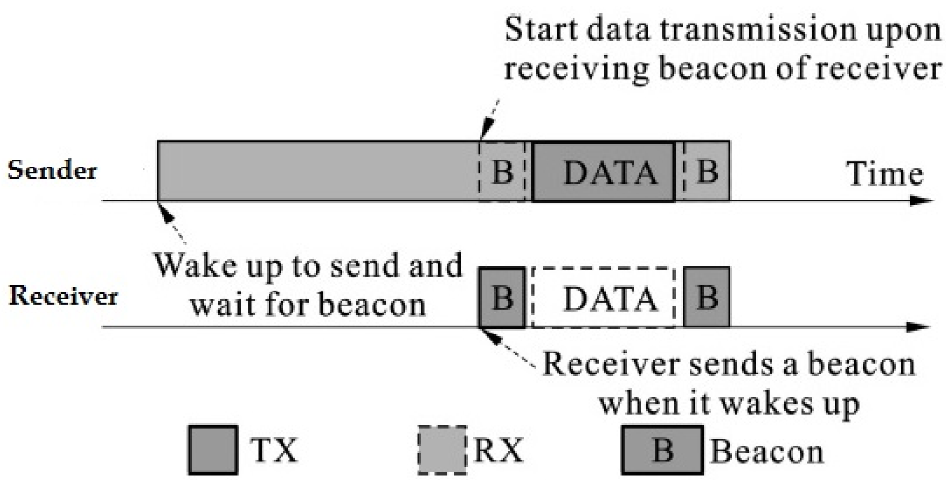

Figure 2.11: Receiver initiated transmission in RI-MAC

MAC. Data arrives at the sender node and it waits silently for the beacon packet from the receiver. Receiver wakes up based on its own duty cycle schedule and transmits a beacon. After receiving beacon from the receiver, sender node starts data transmission. after completing of the data transmission both node exchange beacon as an acknowledgment. Both nodes go to sleep again after completing data transmission.

RI-MAC is energy efficient, it minimizes the time in which sender synchronizes itself with the receiver for a communication session. It also reduces the time, in which sender occupies the wireless link for for preamble transmission, such as in B-MAC, X-MAC and UPMA-

X-MAC, however Once after arriving the data frame at the sender, the sender wakes up immediately and wait until the receiver sends a beacon. This idle wait adds extra packet delivery latency. It also waists the significant amount of energy due to idle listening of the sender node.

REA-MAC

Typically in asynchronous duty cycle MAC protocols each node chooses its wake up schedule randomly, which results in packet delivery latency increased significantly on a multi-hop path. To address the issue of packet idle listening of the sender node and packet delivery latency on multi-hop path, wei et al.[17] proposed a routing enhanced asynchronous duty cycle protocol called REA-MAC. REA-MAC uses cross layer routing information in order to decide the duty cycle schedule[46].

REA-MAC addresses the problem of idle listening in RI-MAC by using sender on demand wake up mechanism. Using cross layer information data packets can be forwarded continuously on a multi-path. Next hop of each hop wakes up after its previous hop. Using this mechanism sender nodes does not wait for the receiver node. If a sender node wakes up after the completion of wake up time of sender in an operational cycle. In this case, when data packet arrives at sender, the sender node does not wake up immediately but wakes up based on the relationship between its own wake up time and the wake up time of receiver in next operation cycle as shown in the Figure 2.12.

Sender can get the information of receiver wake up time in next operation cycle using cross layer routing information. In the worst case, when data arrives on sender, when

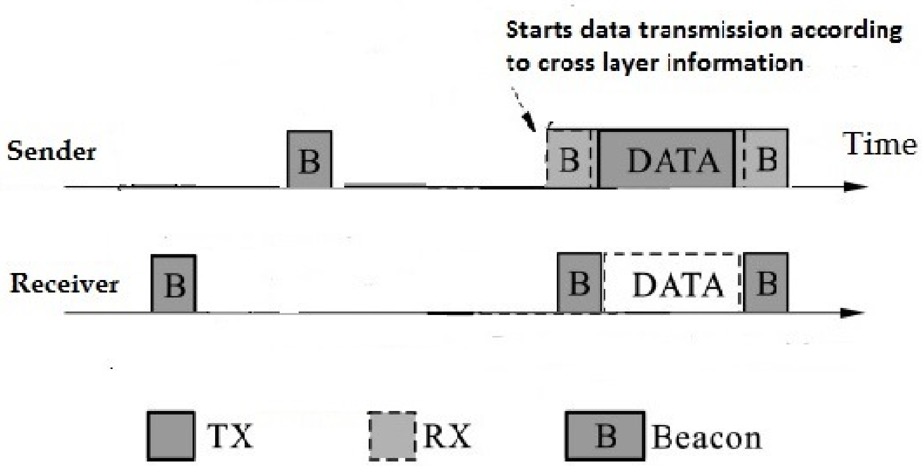

Figure 2.12: Wake up mechanism in REA-MAC

the receiver has already completed its wake up time in one operational cycle as shown in Figure 2.12.In this case sender does not wake up immediately, it wakes up based on the relationship between its own wake up time and the receiver's wake up time in the next operational cycle. REA-MAC significantly reduces the packet deliver latency at each hop compared to RI-MAC as shown in the paper [17]. Although it is not consuming energy because it is in sleep mode, but it introduces packet delivery latency of one extra cycle due to idle listening of the sender. When sender finds out that receiver has already completed its wake up time in this operation cycle it does not wake up immediately and waits for the next cycle. This idle wait of the sender nodes results into packet delivery latency.

Proposes synchronous duty cycle MAC protocols in the literature case extra overhead at the wireless link due to clock synchronization. This control information on the link decrease the QOS of the actual data packets.

These recently proposed asynchronous duty cycle MAC protocols are energy efficient. In these asynchronous duty cycle protocols each nodes chooses its wake up schedules almost randomly in an operational cycle, Which results into packet delivery latency on multi-ho path[47]. Asynchronous duty cycle Protocols could be either sender or receiver initiated as described in the sections 2.2.1 and 2.2.2. Both sender and receiver initiated MAC protocols have their own pros and cons on different scenarios. These protocols usually suffer longer packet delivery latency under dynamic and heavy traffic loads. Another technique could be adopted to combat these problem. A hybrid duty cycle MAC protocol is used to combat these problem.

2.3 Summary

Existing duty cycle MAC protocols conserve energy at the expense of increase in packet delivery latency, which is not acceptable for time critical application in WSNs. Application, such as medical, military and disaster recovery require quick event notification[48],citeanas. Existing synchronous duty cycle MAC protocols cause extra overhead on wireless medium resulting in low performance in heavy traffic loads[49]. When sensor nodes join or leave the cluster or group synchronous duty cycle MAC protocols, such as S-MAC and X-MAC require to re-synchronize with the network over and over again, resulting in higher energy consumption[50]. Implementation of long preambles and overhearing in asynchronous duty cycle MAC protocols, such as B-MAC results in higher transmission energy. However, the problem of long of long preambles is reduced by introducing short preambles such as X-MAC[51]. Short preambles occupy the wireless medium resulting in lower throughput and idle listening. Beacon packet are introduced in RI-MAC to combat the problem of preambles and overhearing[52]. Most of existing asynchronous duty cycle MAC protocols are perform well under light traffic load, however they perform poor under dynamic and heavy traffic loads on multi-hop path in WSNs. We can observe through literature review that asynchronous duty cycle MAC protocols are more scalable compared to synchronous duty cycle MAC protocols[48].

Chapter 3

Design Of H-MAC

In this Chapter we will describe the design of H-MAC in detail. After an overview of H-MAC we will explain, routing enhanced wake up mechanism in the Section 3.2, on-demand sender initiated wake up mechanism in the Section 3.3. After the detailed design we will give the comparison of H-MAC with other duty cycle MAC protocols in the Section 3.4.

3.1 Overview

Hybrid MAC (H-MAC) is an asynchronous hybrid sender and receiver initiated MAC protocol, which takes the advantages of both sender and receiver initiated duty cycle MAC protocols . Each node has independent duty cycle schedule similar to [16, 17] synchronous duty cycle MAC protocols. In the receiver initiated part, each node wakes up independently based on its wake up schedule in each operational cycle and transmits a beacon immediately. This beacon transmission notifies all neighbors in the transmission range that the receiver is in active state and ready to receive the data packet. After beacon transmission the sender node transmits the data packet immediately. If there are multiple packets in the queue the receiver keeps listening to the channel. If there is no communication on the link after beacon transmission, the node goes again in the sleep state in order to conserve the energy.

Firstly, similar to REA-MAC, in H-MAC each node does not wake up randomly, but schedules its wake up time based on cross layer routing information. Each node wakes up according to hop distance information and forwards the packet in a sequence on the multi-hop path. Secondly, H-MAC provides on- demand sender initiated wake up mechanism, which makes it a hybrid sender and receiver initiated MAC protocol. If a data packet arrives at the sender at the time when receiver has already completed its wake up time in this operational cycle, receiver does not wait for next cycle and wakes up in this operation cycle on the demand of sender using in order to reduces the transmission delays.

3.2 Routing enhanced wake-up mechanism

R-MAC [11] uses the routing enhanced schedule in a synchronous duty cycle MAC protocol where wake up time for each node is maintained and exchanged according to the number of hops to the sink node on a multi-hop path. REA-MAC [17] uses this approach in an asynchronous MAC protocol where each node schedules its wake up time according to hop distance to the sink node in order to forward the packets in a pipeline on a multi-hop path. H-MAC uses cross layer routing information in order to forward the data packets continuously on a multi-hop path in less time where each node wakes up based on hop count to the sink node.
Typically in a WSN, all nodes gather the data and report it back to a sink node on a multi-hop path. In asynchronous duty cycle MAC protocols each node wakes up independently based on its own duty cycle schedule. H-MAC introduces Routing Enhanced Wake up (REW) mechanism [53, 54]. Each node schedules its wake up time after its previous node to forward the packet on multi-hop path
After initialization each node chooses its wake up time from 0 to T, where T is time period of one operational cycle. Typically an operational cycle is the time in which a node wakes up only once and after waking up first time a node decides its next wake up time based on the hop count to the sink node. Each node can get its routing information from its network layer header. In H-MAC Time To Live (TTL) window is used to extract the hop count information. TTL count reduces each time when a node forwards a packet to its next hop. Nodes further away from the sink node will have greater value of TTL compared to the nodes near to sink node. A node with higher value of TTL will wake up earlier compared to a node with smaller value of TTL. This mechanism ensures that each node decides when to wake up on multi-hop path based on the hop distance to the sink node. The nodes away from the sink node wake up early compared to nodes nearer to sink as shown in the Figure 3.1. Next hop of each hop wakes up in a sequence making it continuous forwarding on the multi-hop path.

As shown in the Figure 3.1, node 3 is on three hop distance from the sink node and therefore it has the higher value of TTL for a packet compared to node 1 and node 2. According to cross layer hop information node 3 wakes up earlier compared to node 2 and

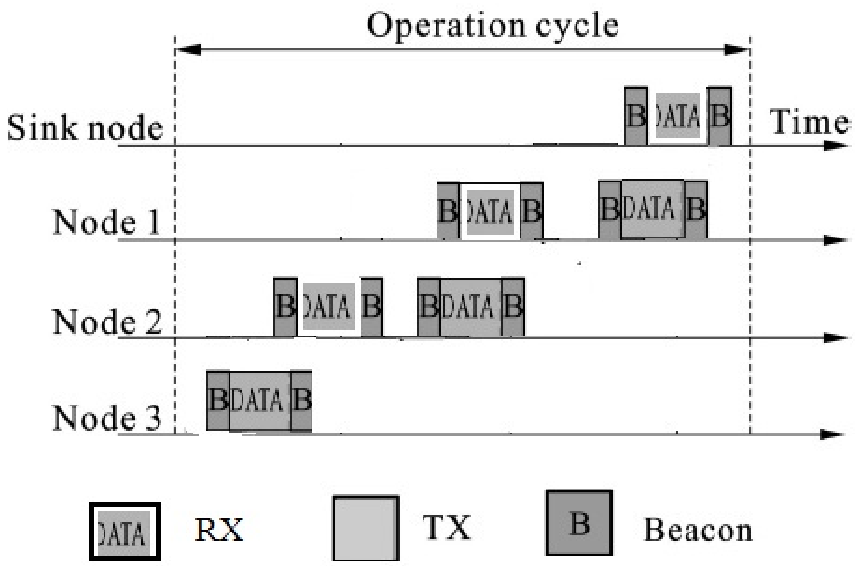

Figure 3.1: REW mechanism in H-MAC

node 1. Node 2 is two hop away from the sink. Node 2 wakes up after node 3, and receives the data packet. Node 1 will have the smaller value of TTL compared to node 2 and node 3. Node 1 wakes up after node 2 and node 3. Packet reaches the destination when sink node wakes.

To make it sure that a node wakes up for enough time, that it does not go to sleep state without receiving the complete data packet. Total wake up time is calculated as follows

$$T_{wake} = 2T_{bacon} + T_{DATA} + 2T_{SIFS}$$

Where T_{wake} is total wake up time, $2T_{bacon}$ is equal to two beacon transmission time, T_{DATA} denoted the total time needed to transmit the data packet on the wireless link and $2T_{SIFS}$ is equal to two Short Interframe Spacing (SIFS). SIFS is the time interval between a data packet and its acknowledgment. Each time a node wakes up and calculates its next wake up time according to its hop distance to the sink node, its total wake up time is computed according to T_{wake}. REW mechanism is explained in Algorithm 1. Ip-ttl is the TTL information from Network layer header. Max-hop is used to save the value of maximum TTL value on a multi-hop path. A node with maximum value of TTL will wake up first compared to other nodes on a multi-hop path.

3.2.1 Collision avoidance in REW mechanism

A node further away from sink node wakes up early compared to a node near to sink node on the multi-hop path. Nodes having the same distance from the sink node wake up at

Algorithm 1 Routing enhanced wake-up mechanism

if wake up = 1^{st} **then** wake up time = [0 - T]
else
loop:
if max-hop <= ip-ttl **then** max-hop = ip-ttl
if radio is sleep **then** wake up

the same time, because their hop count distance will be same. To ovoid the collision of the nodes waking up in the same time, each node selects a random number from 0 to T to offset its wake up starting time. Random offset makes each node having same hop distance from the sink node to have different wake up time, therefore avoiding collision.

3.3 On demand sender initiated wake up mechanism

In this technique, we have used multiple wake-up provisioning in one operational cycle [33]. multiple wake up provisioning allows receiver nodes to wake up multiple times in one cycle on the demand of sender to avoid the delay in packet delivery.

In this technique sender uses wake up information of the receiver in order initiate the transmission, when the receiver has already completed its wake up time in current operational cycle. As shown in the Figure 3.2, data arrives at the sender, and the sender realizes that receiver has already completed its wake up time in this operational cycle . In this case sender transmits wake up beacon to receiver and starts the data transmission.

Without using this mechanism, there can be two other ways to transmit the data packet successfully to the receiver. In the first method, sender has to wait silently for the receiver to transmit the beacon in next operational cycle as in RI-MAC [16]. In the second method sender does not wake up immediately but wakes up in the next cycle based on the relationship between its own wake up time and the wake up time of receiver as in REA-MAC [17]. Both RI-MAC and REA-MAC are receiver initiated MAC protocols, and they add latency due to the idle waiting of the sender. Figure 3.2 explains the concept of on-demand wake up mechanism. Data arrives at the sender and it realizes on the basis of wake up information of neighbor that receiver has already completed its wake up time in the current operational cycle. Sender transmits a wake up beacon to the receiver and receiver wakes up to receive the data packet. After data transmission the beacon serves as acknowledgment of the data packet completion. Both the sender and receiver go to sleep after completing the data transmission.

In this mechanism we use multiple wake up provisioning in operational cycle in order to

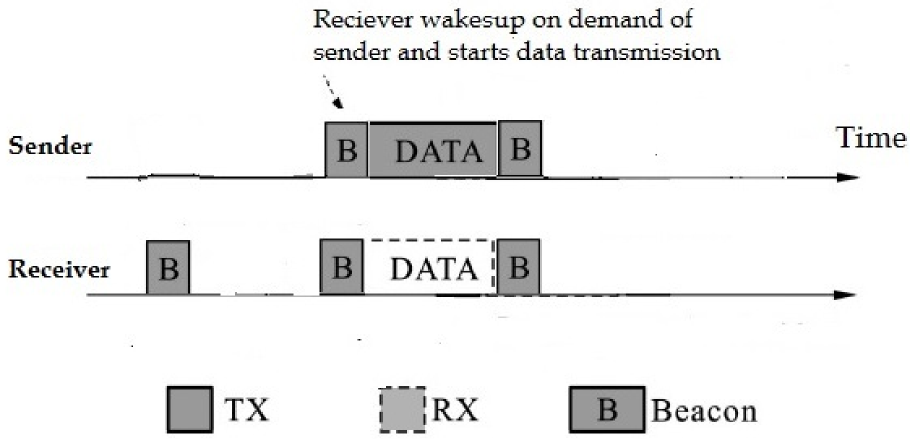

Figure 3.2: On demand sender initiated wake up mechanism

minimize the packet delivery latency caused by idle listening of the sender node. We give an algorithm for on demand sender initiated mechanism. If wake up time of the sender node is less than the wake up time of the receiver node , which means that sender node wakes up before the completion of the wake up time of receiver node in the current operational cycle. In this case REW mechanism will be used, REW mechanism is described in detail in the Section 3.2. If the wake up time of sender node is greater than the wake up time of receiver, which means that sender node wakes up after the completion of the wake up time of receiver node in current operational cycle. In this case receiver will wake up on the demand of sender in order to minimize the packet delivery latency caused by idle listening of the sender node. On demand sender initiated mechanism is explained in algorithm 2. Time-now in the algorithm is the current time when sender nodes wakes up in an operational cycle. $T_{receiver}$ is the wake up time of the receiver An overall algorithm is

Algorithm 2 On demand sender initiated mechanism

if Time-now $>T_{receiver}$ **then** generate wake up beacon

given algorithm 3 for H-MAC when we use both REW mechanism and on demand sender initiated mechanism in different situations.

Algorithm 3 H-MAC

if Time-now $<T_{receiver}$ **then** use routing enhanced wake up mechanism
if Time-now $>T_{receiver}$ **then** Use on demand sender initiated mechanism

3.4 Beacon frame

Beacon frame in H-MAC contains source address, destination address, Back Off Window (BW), Frame Control Field (FCF), Frame Control Sequence (FCS) and frame length for 802.15.4 IEEE standard. When a nodes receives the beacon, it can determine that which fields are present in the beacon by looking at the size of beacon. A beacon in H-MAC serves as an acknowledgment for the completion of the data, and the initiation of the new data transmission. When a node wakes up and transmits a beacon and observes that the medium is busy, it backs-off and retries beacon later. For the acknowledgment of the received data receiver transmits the beacon to the sender with the destination field as source field. For the initiation of new data transmission sender transmits the beacon to the receiver as the request of the transmission of the data packet as destination field with the address of the receiver.

3.5 Comparison of H-MAC with other duty cycle MAC protocols

Different duty cycle MAC protocols use different approaches to achieve some common goals, such as conserving energy, reducing packet delivering latency, and increasing throughput of the network [55]. The main purpose of a duty cycle protocol in to minimize energy consumption because sensor nodes has limited battery life. We have to reduces the energy consumption of the sensor nodes in order to maximize the network lifetime [56]. Moving towards energy efficiency in WSNs these energy efficient protocols cause packet delivery latency. Some application need longer life of network due to unattended sensor nodes, and consider packet delivery latency is less important for these applications. On the other hand some application need critically robust communication, and for these energy efficiency is less important. Different protocols can be used in different application depending on the nature if the application. But a duty cycle MAC protocol should be suitable for different scenarios, and it should be deploy able in diverse networks and dynamic traffic loads.
A comparison of parameters is given between H-MAC and some other existing duty cycle protocols in the table 3.1.

Table 3.1: Comparison of H-MAC with other duty cycle MAC protocols

Parameters	X-MAC	RI-MAC	H-MAC
Special Frame	Preamble	Beacon	Beacon
Synchronous or Asynchronous	Synchronous	Asynchronous	Asynchronous
sender or receiver initiated		Receiver initiated	Hybrid
Retry Limit	0 To 5	0 To 5	5

3.6 Summary

A novel asynchronous hybrid sender and receiver initiated duty cycle MAC Protocol is presented in this chapter, called H-MAC. REW mechanism is used for receiver initiated part. In REW mechanism each node chooses its own wake up time from 0 to T initially, where T is the time of one operational cycle. When a node wakes up it chooses it next wake up time, bases on the hop distance from the sink node. The nodes further away from the sink node wake up earlier compared to the nodes nearer to sink node. Using REW mechanism, each node wakes up after it previous hop to receive the data packet, resulting in a sequential wake up of all the nodes from source to destination on multi-hop path. REW mechanism reduces the packet delivery latency caused by random wake ups of the nodes on multi-hop path. In sender initiated part receiver nodes can wake up multiple times to receive data packet in a an operational cycle. when a receiver node has pending data to receive and it has already completed it wake up time in an operational cycle. In this case receiver node can wake up multiple time on the demand of sender node. On demand sender initiated mechanism reduces the packet delivery latency caused by the idle listening of the sender node.

Chapter 4

Simulation And Evaluation

We evaluated both RI-MAC and H-MAC in NS-2.29. We have used simulations to compare the performance of H-MAC and RI-MAC in diverse network scenarios under dynamic traffic loads. We have used gnuplot tool to plot the results in the form of graphs. Sun et al. [16] have already compared RI-MAC with X-MAC[14] and X-MAC-UPMA[42] and results show that RI-MAC outperforms them.

4.1 Simulation

We have used single omnidirectional antenna in our simulations. we have used TwoRay-Ground propagation model and simple LL link layer type. We have used cbr traffic for all the experiments. We have used sedest emulator in NS-2.29 directory to emulate random network scenarios. We have used Cbrgen tool in NS-2.29 directory in ordor to emulate random cbr traffic events. Other simulation parameters are given in the Table 4.1.

Table 4.1: Simulation parameters

Bandwidth	250 Kbps
SIFS	192 μs
Tx Range	250 m
Size of ACK	5 B
CCA Check Delay	128 μs
Carrier Sensing Range	550 m
Backoff Window	0 to 255
Retry Limit	5
Beacon frame	6 B to 9 B

4.2 Evaluation

We have used four different types of network scenarios to evaluate the performance of H-MAC including clique network, chain network, grid network, and random network. In the clique network we have varied the number of flows in order to evaluate the performance of H-MAC. Traffic load increases with an increase in the number of flows in the clique network. In chain network we have varied the number of nodes to evaluate the results.If we increase the number of nodes in chain network a data packet will have to travel more hops in order to reach the destination on a multi-hop path. We also varied the the packet generation interval in the chain network. The traffic intensity increases with a reduction in the packet generation interval in the chain network. In the grid network we have varied the sensing range to evaluate the results using Random Co related Event RCE model[16]. With larger sensing range more nodes report the data to sink node, traffic intensity increases with an increase in the sensing range in the grid network. For the random network we also varied the sensing range using RCE model. Sink can be any where and an event can occur at any x,y location. Dynamic traffic load increase with an increase in the sensing range in the random network.

4.2.1 Clique Networks

In a clique network each node is with each others transmission range. We have varied the traffic load by varying the number of independent flows in the network. Each flow is independent such that each flow has different source and destination as shown in the Figure 4.1.

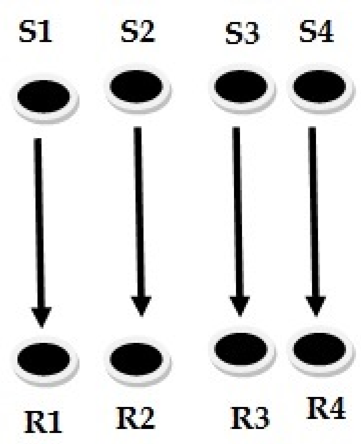

Figure 4.1: Clique network with independent flows

In each clique network the total number of nodes are twice the number of independent flows in the network. For each flow, the source node starts to generate the packet 10 seconds after the simulation starts. Source node generate new packets from any interval between 0.5 to 1.5 seconds. At the beginning of the simulation each node randomly chooses its initial wake up time from 0 to T_{cycle}, where T_{cycle} is total time period of an operational cycle. Each receiver counts the number of packets received after every 50 seconds. If a packet is not received or it is still in queue or being transmitted, it is as lost packet. Number of flows are varied from 5 to 25 and the number of nodes are from 10 to 50.

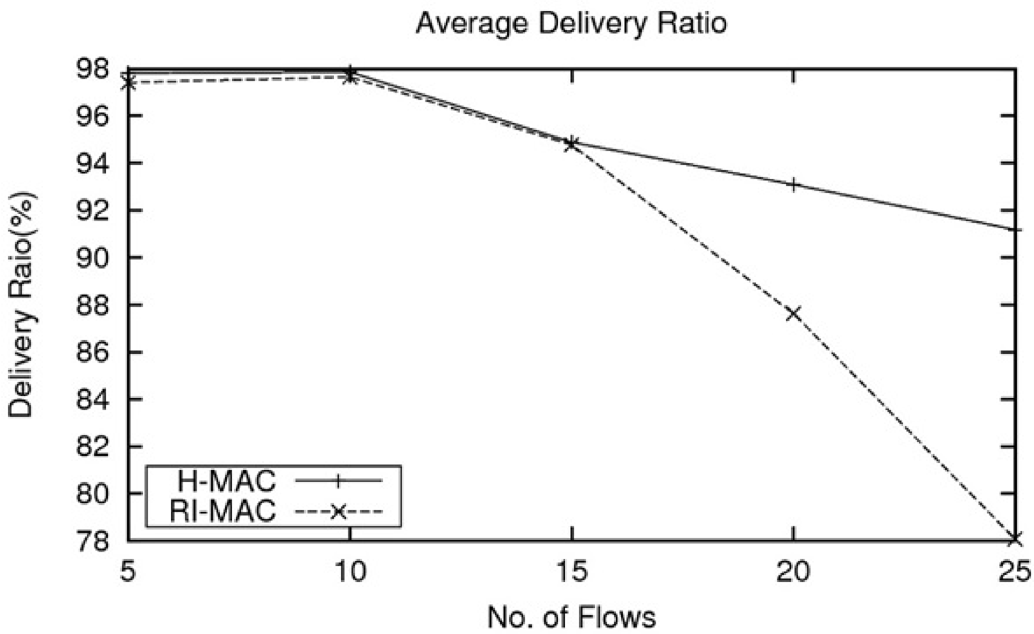

Figure 4.2: Average PDR vs number of flows

Figure 4.2 shows the Packet Delivery Ratio(PDR) achieved by RI-MAC and H-MAC with an increase in the number of flows in the clique network. PDR of H-MAC is more than 92% indicating that throughput achieved by H-MAC is more than 92% with an increase in the number of independent flows. When the number of flows are below or equal to 15 both H-MAC and RI-MAC perform same in terms of PDR. However, when the number of flows exceed 15, PDR of H-MAC is significantly better than RI-MAC. For 25 flows the PDR of H-MAC is 15% better than RI-MAC. This degradation in PDR of RI-MAC is due to an increase in the idle time of sender nodes.

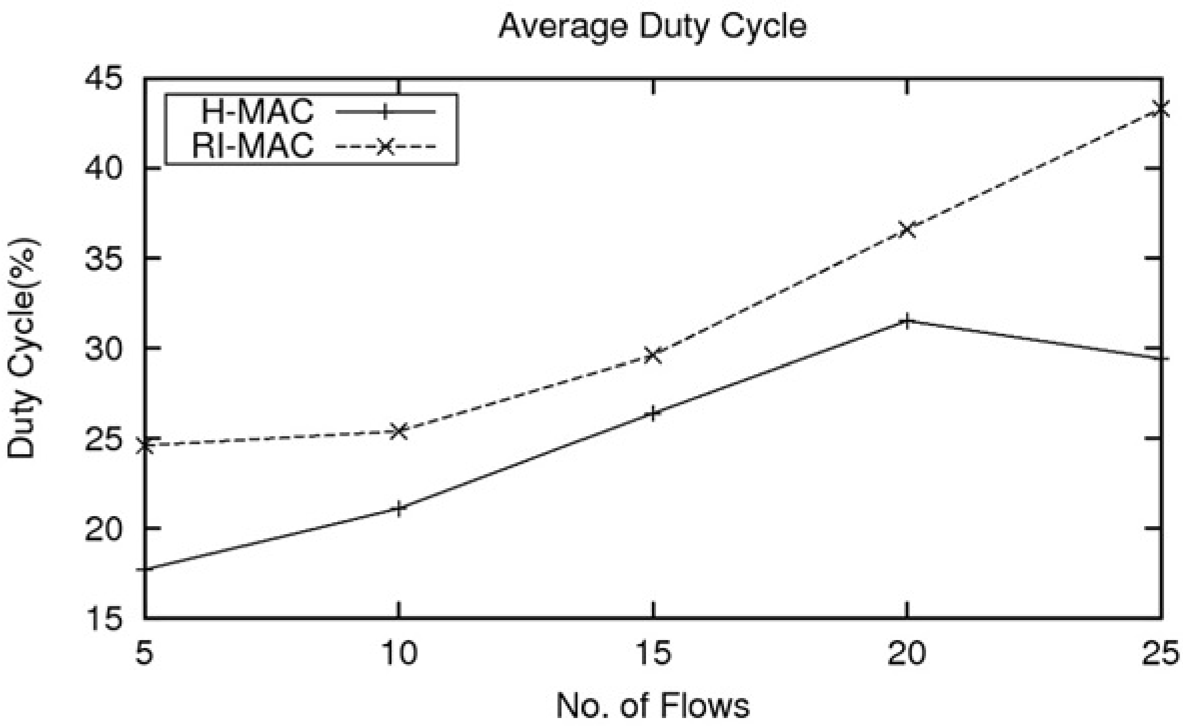

Figure 4.3: Average duty cycle vs number of flows

Figure 4.3 shows the average duty cycle with RI-MAC and H-MAC. Results show that H-MAC conserves much more energy compared to RI-MAC with an increase in number of flows in the clique network. Duty cycle with H-MAC remains up to 20% on an average with an increase in the number of flows. With RI-MAC, duty cycle remains up to 35% on an average which indicates that H-MAC consumes about 15% less energy compared to RI-MAC. RI-MAC consumes more energy due to senders idle listening when sender wakes up and waits silently for the receiver to send the beacon. On the other hand, with H-MAC wake up decision is based on the hop distance, and sender does not need to wait for the receiver to transmit the data beacon.

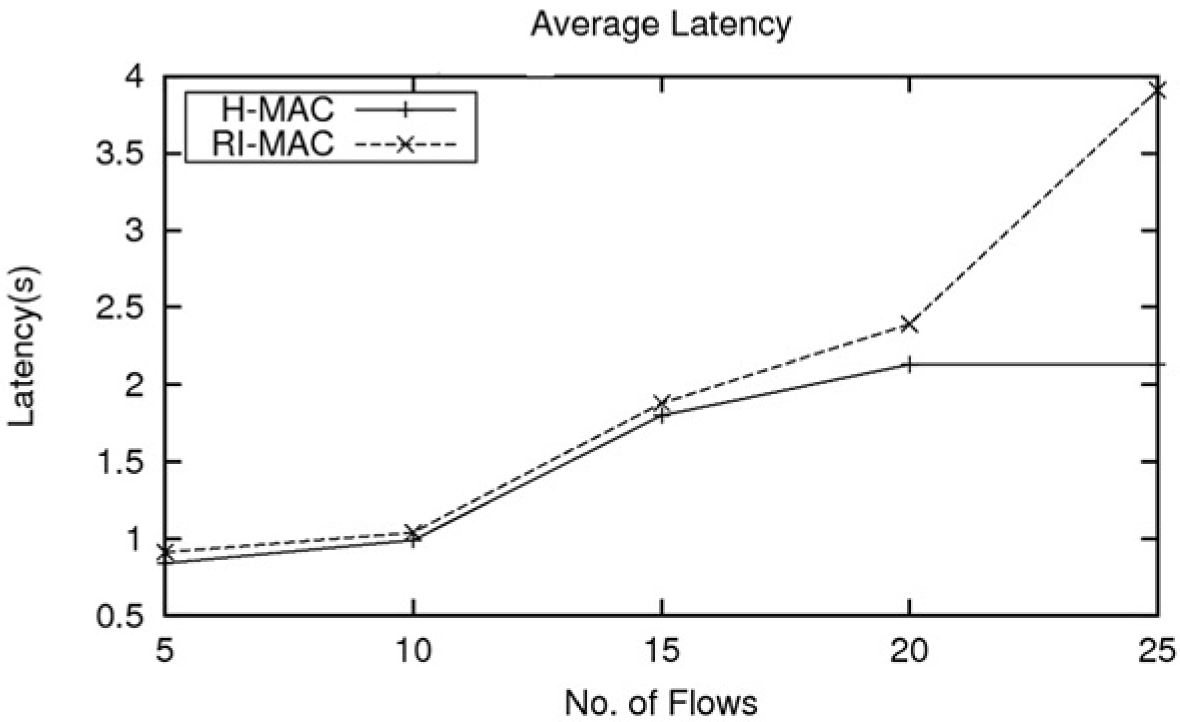

Figure 4.4: Average latency vs number of flows

Figure 4.4 shows that despite of higher duty cycle RI-MAC has longer latency compared to H-MAC with an increase in the number of flows. Results show that H-MAC experiences about 1.4s latency on an average with an increase in the number of flows in the clique network. On the other hand, RI-MAC experiences latency of about 2.4s on an average. This longer latency is due to idle listening of the sender which results in higher duty cycle and higher end to end delay. H-MAC experiences lower latency due to reduced Idle listening of the sender. Even in the case when receiver has already completed its wake up time in one operational cycle, it wakes up on the demand of the sender to complete the data transmission in current operational cycle.

4.2.2 Chain Network

In a chain network of number of nodes are connected in the form of a chain as shown in the Figure 12, Where each node is connected directly to its next hop. In a chain network packets are delivered to the destination through all intermediate nodes. In our chain network each node is 200m apart from each other as shown in the Figure 4.5.

We vary the number of nodes for calculating the average duty cycle, average latency and average packet delivery ratio. After the start of simulation source node generates new packets after an interval from 0.5 to 1.5 seconds. At the beginning of the simulation each

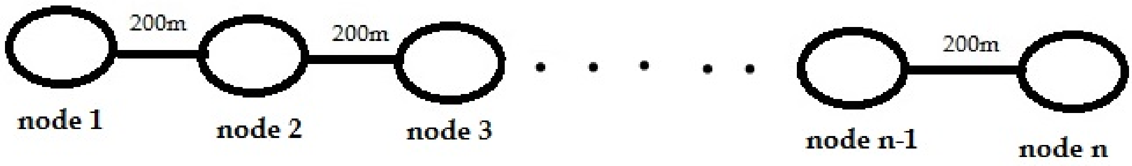

Figure 4.5: Chain network with n number of nodes

node randomly chooses its first wake up time from 0 to T_{cycle}, where T_{cycle} is total time period of an operational cycle. Each receiver counts the number of packets received after every 50 seconds. If a packet is not received or it is still in queue or still being transmitted, it is not counted as received packet, it is counted as lost packet.
The results in our chain network are calculated by varying the number of nodes from 5 to 20 and results are discussed below in detail.

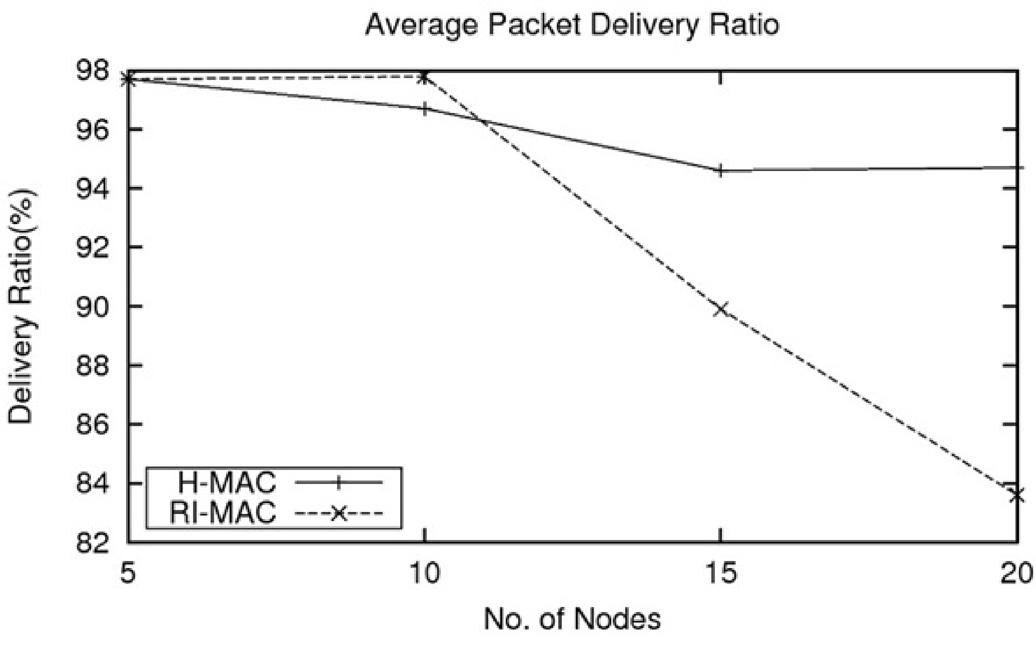

Figure 4.6: Average PDR vs number of nodes

Fig 4.6 shows the average packet delivery ratio achieved by RI-MAC and H-MAC with varying number of nodes from 5 to 20. Results show that average delivery ratio with H-MAC remains 97% on average indicating higher throughput with an increase in the number of nodes in the chain network. RI-MAC have sharp decline in average delivery ratio after increasing the number of nodes up to 10. H-MAC has steady Packet delivery ratio with an increase in the number of nodes compared to RI-MAC. RI-MAC has lower packet deliver ration due to end to end delay in packet delivery. H-MAC achieves higher Packet delivery ratio due to less end to end delay.

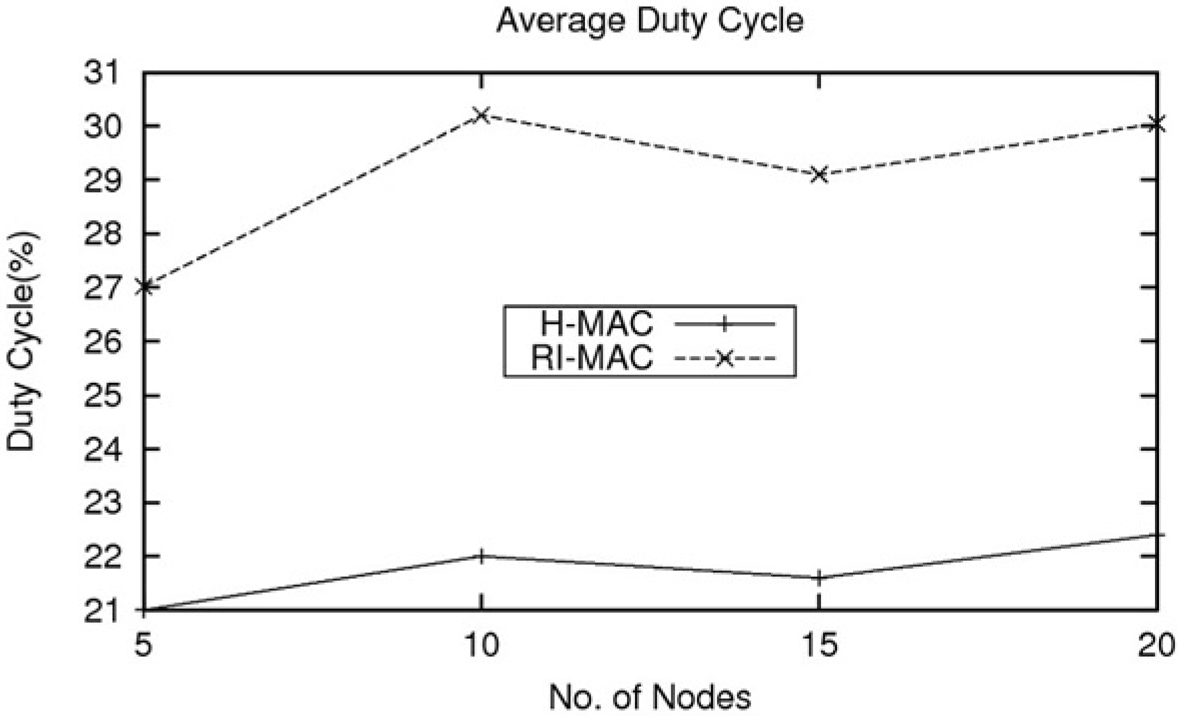

Figure 4.7: Average Duty Cycle vs number of nodes

Figure 4.7 shows the average duty cycle with RI-MAC and H-MAC in a chain topology. Results indicate that H-MAC conserves more energy compared to RI-MAC with an increase in the number of nodes in chain topology. Duty cycle with H-MAC remains 21.5% on average compared to RI-MAC, which experiences about 28.5% duty cycle on average. H-MAC consumes less energy compared to RI-MAC with an increase in the number of nodes in a chain topology, thus minimizing the network lifetime. RI-MAC consumes more energy due to idle listening , where sender and receiver wake up asynchronously and sender has to wait for the preemptive beacon from the receiver. H-MAC consumes less energy by reducing the idle listening of the sender.

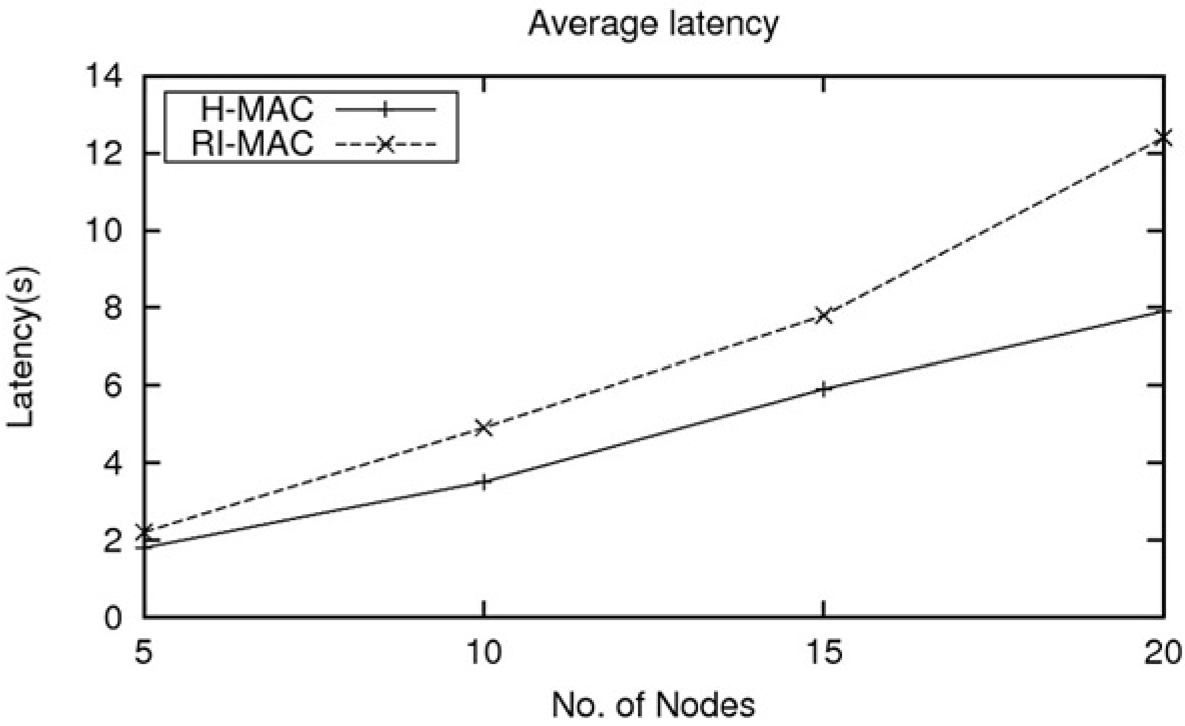

Figure 4.8: Average latency vs number of nodes

Figure 4.8 shows the average latency with RI-MAC and H-MAC. Results clearly show that despite of high duty cycle, RI-MAC experiences longer latency compared to H-MAC. H-MAC experiences latency of 4s on average, and RI-MAC experiences latency of 6.1s on average. In a chain topology a packet has to pass from all the intermediate nodes in order to reach to the destination which adds the latency. This longer latency due to idle listening of sender increases duty cycle and end to end delay. H-MAC experiences lower latency because Idle listening of sender is reduced, and even in the worst case sender does not need to wait for the receiver to preempt.

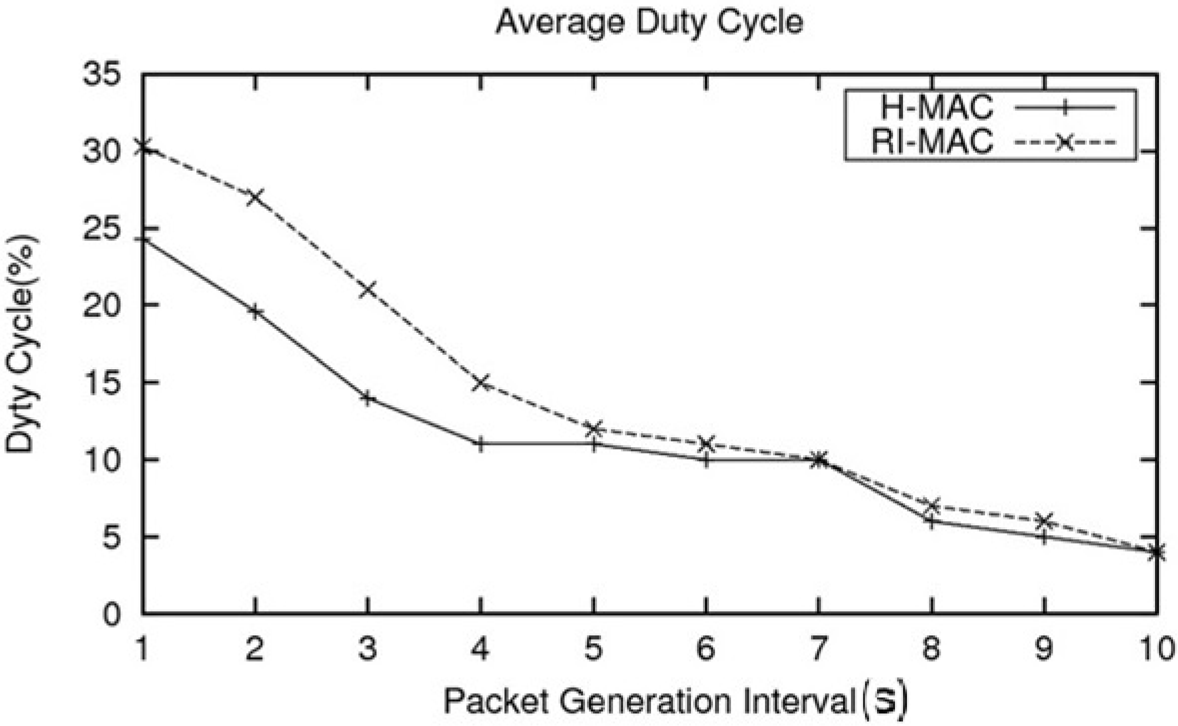

Figure 4.9: Average Duty Cycle vs packet generation interval

Figure 4.9 illustrates the impact of different packet generation intervals on average duty cycle in chain network. We have varied 10 nodes for our experiment. During high data rate the duty cycle of RI-MAC is much higher than H-MAC. H-MAC consumes less energy due to its reduced wake up time in an operational cycle, on the other hand RI-MAC consumes more energy due to idle listening of the source nodes.

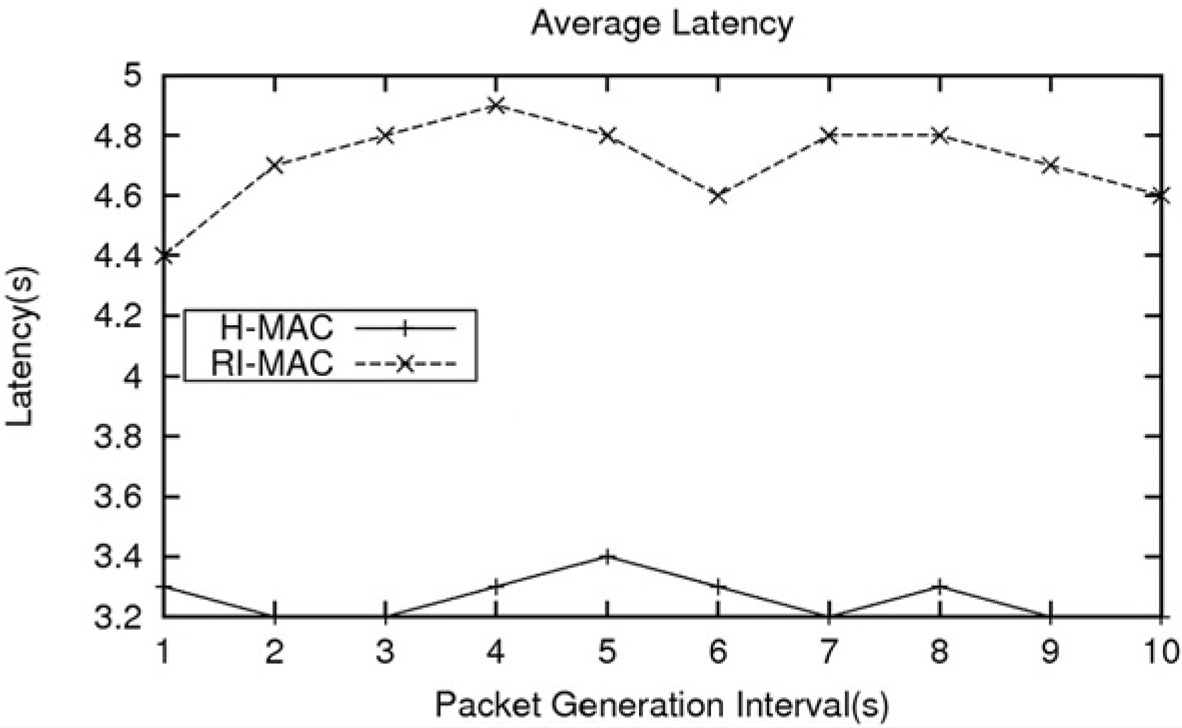

Figure 4.10: Average Latency vs packet generation interval

Figure 4.10 exhibits the impact of diverse packet generation intervals on the average latency in a chain network. As the figure shows, RI-MAC experiences longer latency compared to H-MAC. H-MAC experiences latency of 3.1s on average on the other hand RI-MAC experiences latency of 4.7s on average. In the worst case when packet generation interval is 3s latency of H-MAC is 1.6s higher than RI-MAC. RI-MAC has longer latency because the sender has to wait for the beacon from the receiver, where all nodes have different wake up and sleet schedule. H-MAC has low latency because wake up decision is based on the hop count information.

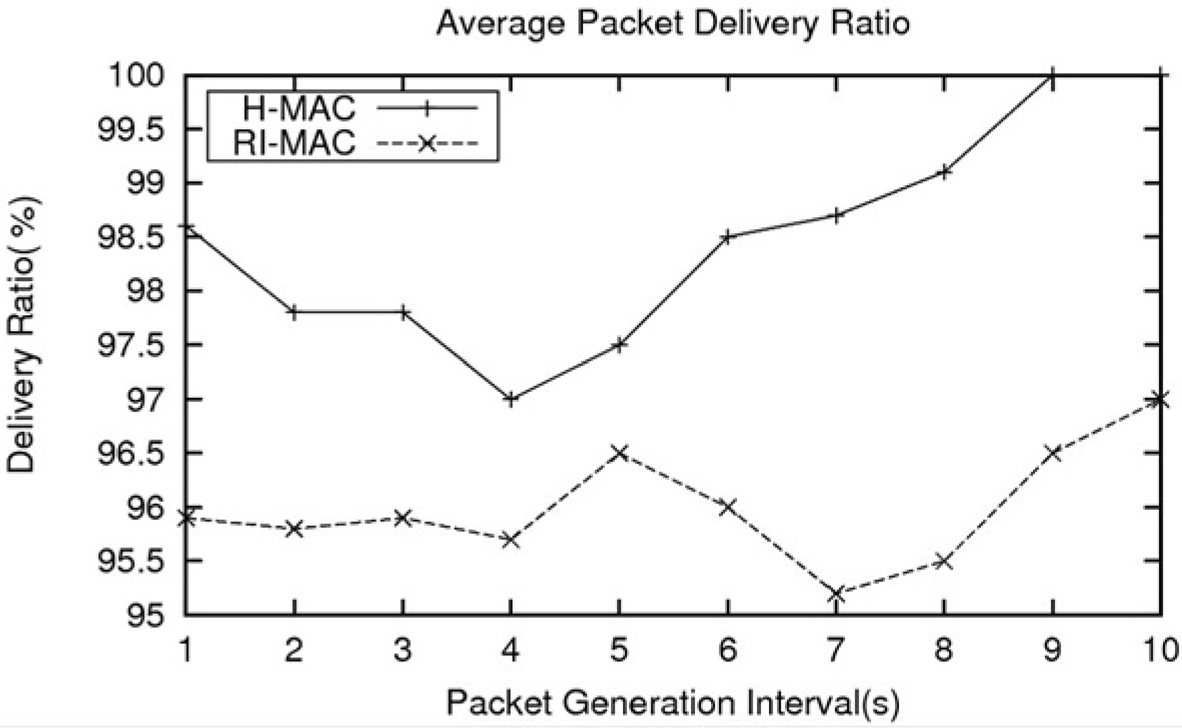

Figure 4.11: Average PDR vs generation interval

Figure 4.11 exhibits the impact of different packet generation intervals on the average packet delivery ratio in chain network. It is apparent from Figure 4.11 that with an increase in the packet generation interval, the number of transmitted packets reduced, resulting in a lower delivery ratio. Figure 4.11 shows that H-MAC has high packet delivery ratio with an increase in packet generation intervals. In the worst case when packet generation interval is 10s H-MAC has 100% PDR on the other hand RI-MAC has 96.5% PDR. RI-MAC experiences low packet delivery ratio because it has higher end to end delay in packet delivery due to idle listening of sender nodes, and packet does not reach to intended destination.

4.2.3 Grid Network

In our grid network simulation there are total 49 nodes placed in a 7x7 grid. The distance between each node is 200m and the sink node is placed in the center.

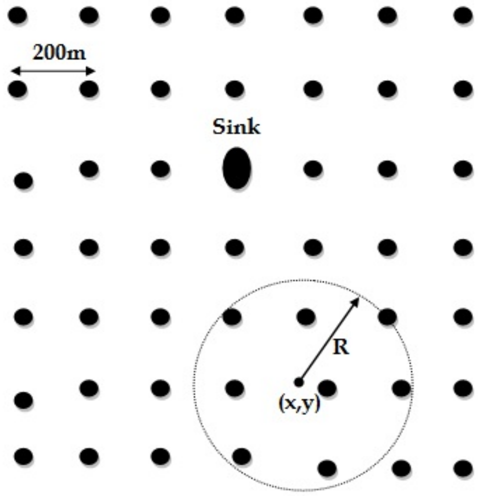

Figure 4.12: RCE model

In the simulation we have used Random Correlated Event (RCE) model [16]. This model is based on correlated event workload [57], where some specific nodes in a geographical area sense the the event and report it back to the sink node. RCE model selects a random point at x,y for each event. When different groups of nodes report the data to sink, it forms a random correlated event. R is the sensing range for a random event at x,y location. Only the nodes that are in the sensing range which is shown as a circle in the Figure 4.12 with radius R will sense the data and report it to the sink node. We have varied the sensing range R to simulate the results in the grid network. We have varied the sensing range from 200m to 1000m. A packet is generated after every 60s. Each node which senses the event sends one packet to the sink node. The length of the path from each node to sink node is 1 to 6 hop Which indicates that every packet has to travel 1 to 6 hop path to reach the destination.

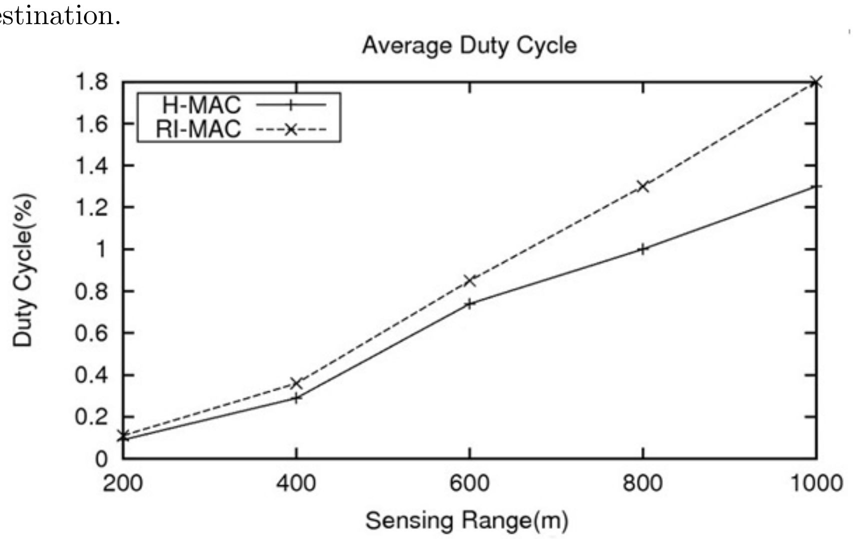

Figure 4.13: Average Duty Cycle vs sensing range

Figure 4.13 exhibits the impact of different sensing ranges on average duty cycle in a grid network with RCE model. Figure 4.13 shows that as the sensing range increases, H-MAC outperforms RI-MAC. An increase in sensing range results in an increase in the traffic load because large sensing range the more number of nodes report the data back to sink node. H-MAC experiences 0.6% duty cycle on average, on the other hand RI-MAC experiences 1% duty cycle on average. Results indicate that duty cycle of RI-MAC increase significantly beyond 600m sensing range, whereas the duty cycle of H-MAC increases only 0.4s. Results indicate that H-MAC consumes less energy under random and high traffic load compared to RI-MAC. RI-MAC has higher duty cycle due to idle listening. H-MAC has lower duty cycle due to reduced idle listening.

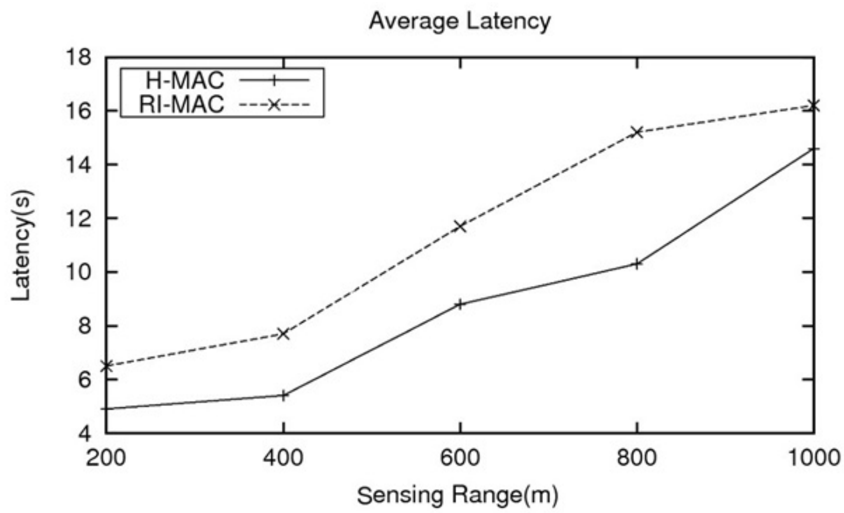

Figure 4.14: Average latency vs sensing range

Figure 4.14 shows the impact of different sensing ranges on average latency in a grid network with RCE model. Results show that H-MAC experiences significantly less latency compared to RI-MAC with an increase in the sensing range or traffic load. In the worst case H-MAC experiences 6s less latency compared to RI-MAC on 800m sensing range. H-MAC performs better than RI-MAC due to reduced idle time, where it transmits more packets is in one operation cycle. On the other hand, RI-MAC has longer latency due to sender idle time and they transmit less number of packets in an operational cycle.

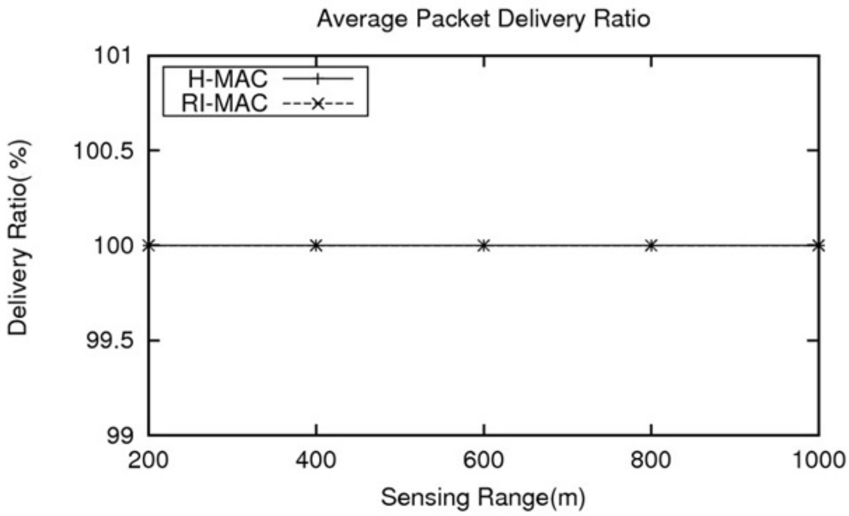

Figure 4.15: Average PDR vs sensing range

Figure 4.15 shows that PDR for both RI-MAC and H-MAC always remains on 100% in grid network. There is only 1 to 6 hopes distances from each node to destination, therefore there is very low probability of packet drop.

4.2.4 Random Network

We have compared RI-MAC, X-MAC and H-MAC in 3 different random networks. Each network with 50 nodes and 1000m X 100m area.We have followed RCE model in order to generate the traffic in the random network. We have detected an event occurred at random x and y location. A sink node is randomly selected from the random network, and all the other nodes report the data to the sink node. An event is generated every 30 second. Sensing range is varied from 200m to 1000m.

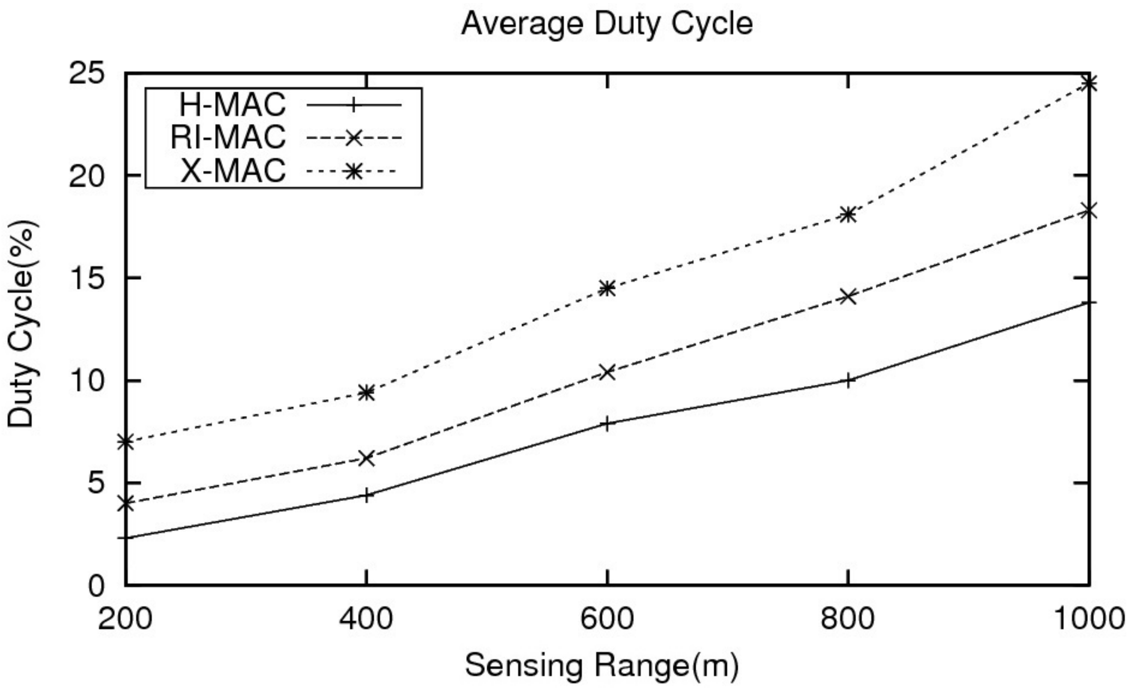

Figure 4.16: Average duty cycle vs sensing range

Figure 4.16 exhibits the impact of diverse sensing ranges on average duty cycle in random networks with RCE model in 3 runs. Results clearly show the H-MAC conserves more energy compared to RI-MAC and X-MAC. H-MAC experiences 6.3% on an average, on the other hand RI-MAC experiences 10.8% and X-MAC experiences 15% duty cycle on an average. On extreme when sensing range is 1000m, H-MAC experiences 9% duty cycle and RI-MAC has 18.2% and X-MAC has 25% duty cycle. This higher duty cycle in the case of RI-MAC is due to idle listening of the nodes. H-MAC has less duty cycle because the idle listening is reduced and a packet is delivered to destination by awaking the nodes on the basis of hope information.

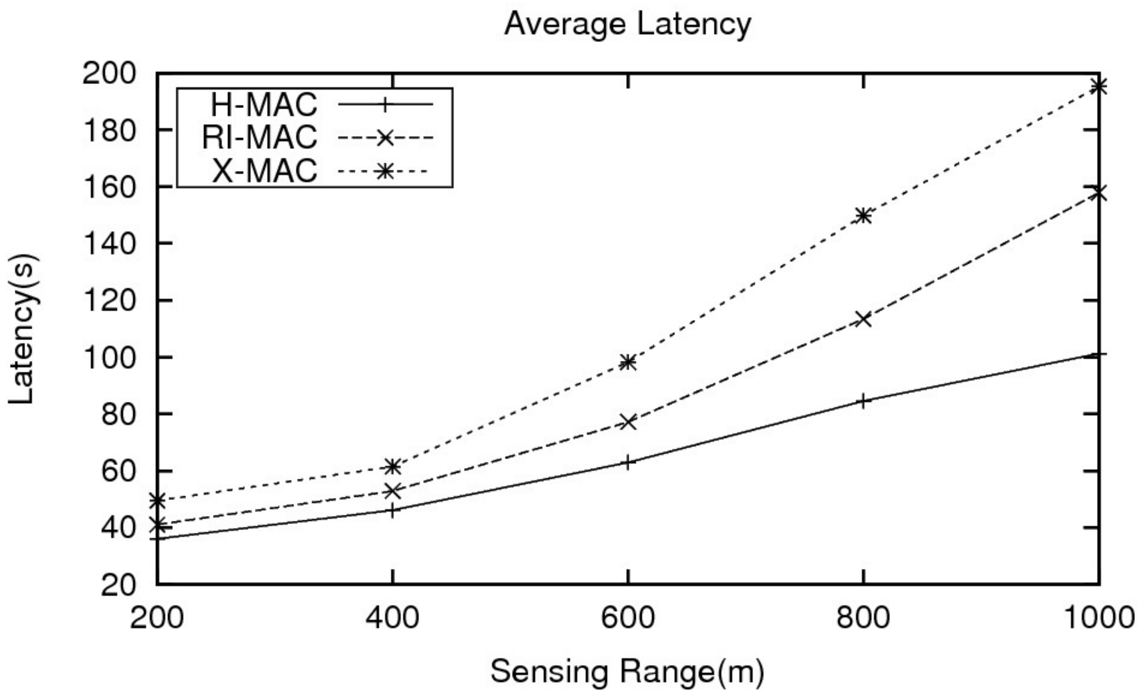

Figure 4.17: Average latency vs sensing range

Figure 4.17 exhibits the impact of different sensing ranges on average latency in 3 random networks using RCE model. Results show that H-MAC experiences lower latency compared to RI-MAC and X-MAC. H-MAC experiences latency of 60s on an average on the other hand RI-MAC experiences the latency of 80s and X-MAC experiences latency of 113s on an average. On extreme when sensing range is 1000m, RI-MAC experiences longer latency with an increase in the traffic intensity. H-MAC experiences lower latency when traffic intensity is increased for 1000m sensing range. RI-MAC experiences longer latency because sender nodes wait silently for the receiver to wake up in order to transmit the data packet. In H-MAC senders does not need to wait for the receiver because the wake up decision is based on hop information.

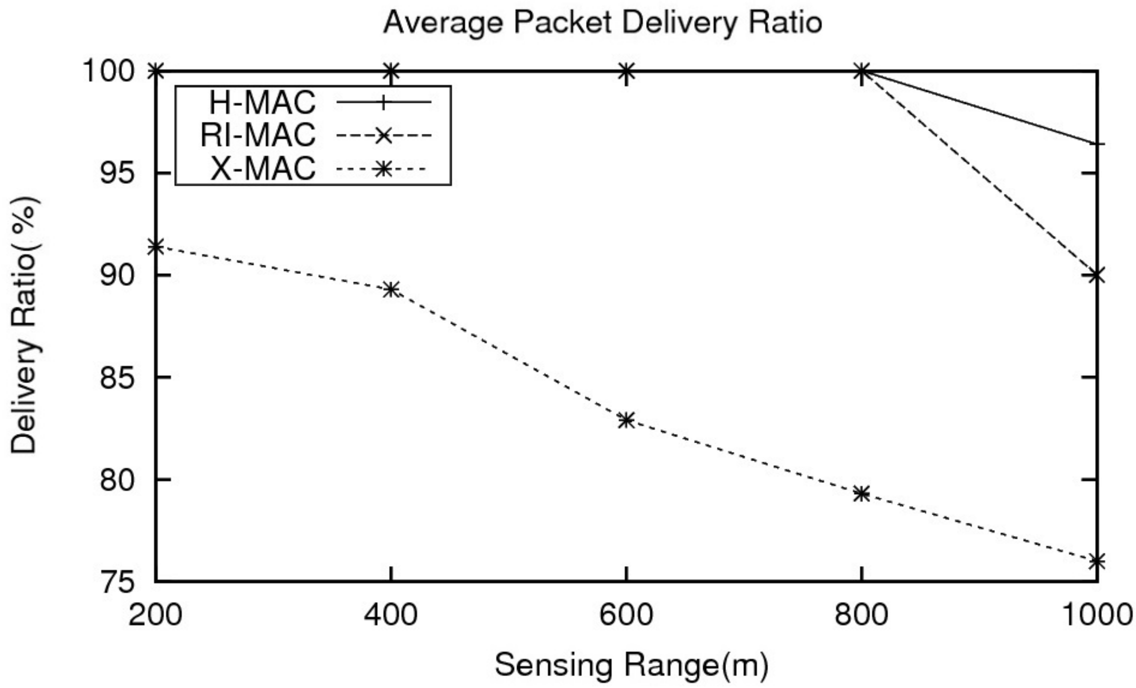

Figure 4.18: Average PDR vs sensing range

Figure 4.18 shows the impact of diverse sensing ranges on PDR in 3 random networks using RCE model. Results show that for different sensing ranges from 200m to 800m both RI-MAC and H-MAC has 100% PDR. X-MAC has much lower PDR compared to H-MAC and RI-MAC . For 1000m sensing range, RI-MAC has 90% of PDR on an average. H-MAC has 97% PDR for 1000m sensing range and X-MAC has 76% PDR. Results indicate that on low traffic intensity both RI-MAC and H-MAC has 100% delivery ratio, however, for increased traffic intensity, H-MAC has high PDR compared to RI-MAC. RI-MAC has low PDR due to traffic congestion in bursty and high traffic. In H-MAC the traffic is not congested because when an event is generated, it is reported back to sink node in the form of a pipeline. the event is reported back to sink node continuously on multi-hop path.

4.2.5 Simulation with High Data Rate

We have simulated grid network with high data rate. In previous simulation a packet is generated every 60s but in this simulation a packet is generated after every 30s in order to eveluate the performance of H-MAC in high and intense data rate. RCE model is used in this simulation and all other parameters are same as given in grid network.

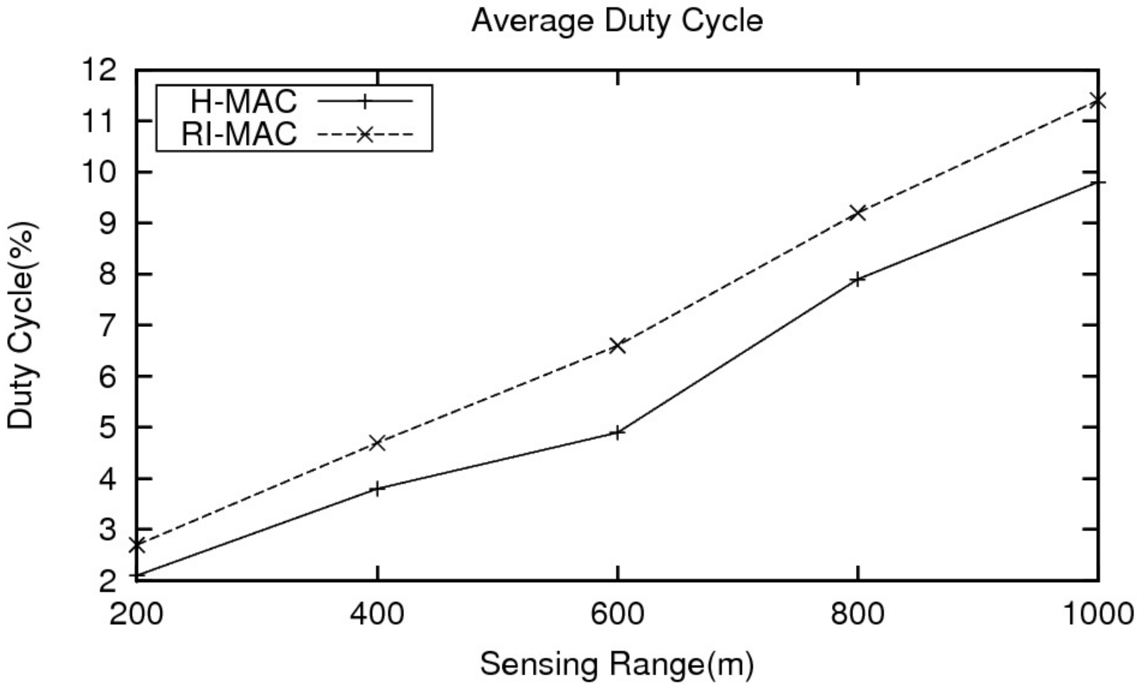

Figure 4.19: Average Duty cycle vs sensing range

Figure 4.19 exhibits the impact of different sensing ranges on average duty cycle in a grid network with RCE model with high data rate. Figure 4.19 shows that as the sensing range increases, H-MAC outperforms RI-MAC. An increase in sensing range results in an increase in the traffic load because large sensing range the more number of nodes report the data back to sink node. H-MAC experiences 4.74% duty cycle on average, on the other hand RI-MAC experiences 6.82% duty cycle on average. Results indicate that duty cycle of RI-MAC increase significantly beyond 800m sensing range, whereas the duty cycle of H-MAC increases only 2.9s. Results indicate that H-MAC consumes less energy under random and high traffic load compared to RI-MAC. RI-MAC has higher duty cycle due to idle listening. H-MAC has lower duty cycle due to reduced idle listening.

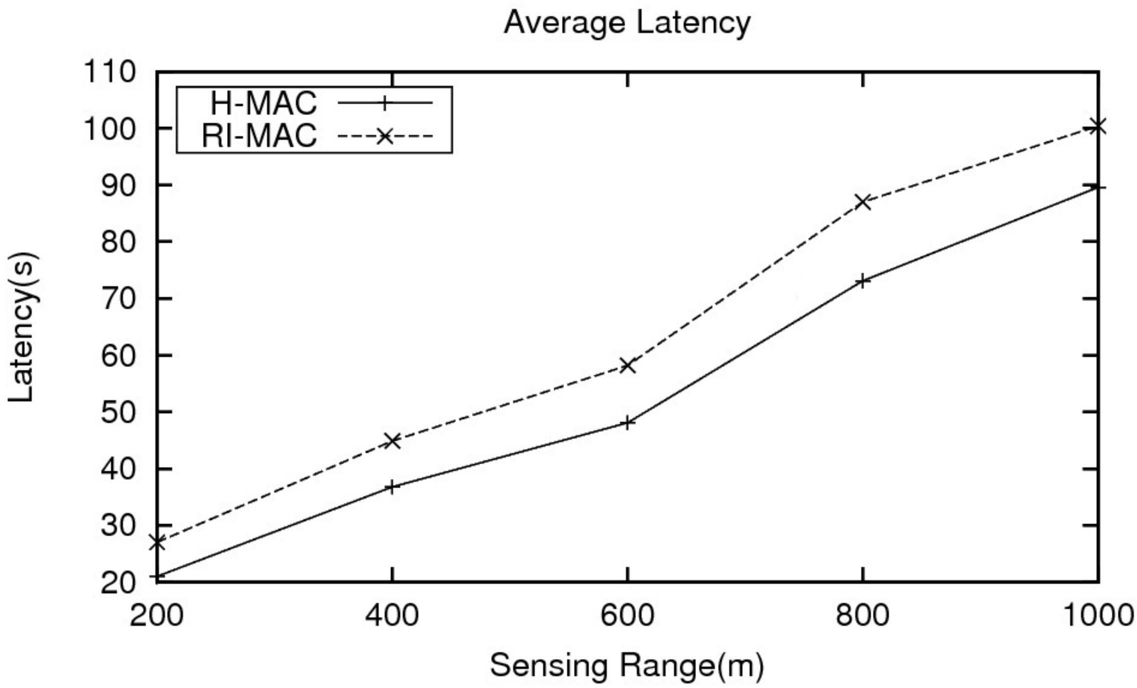

Figure 4.20: Average latency vs sensing range

Figure 4.20 shows the impact of different sensing ranges on average latency in a grid network with RCE model with high data rate where each node generates a packet ever 30s. Results show that H-MAC experiences significantly less latency compared to RI-MAC with an increase in the sensing range or traffic load. H-MAC experiences 11s less latency compared to RI-MAC on an average. H-MAC performs better than RI-MAC due to reduced idle time, where it transmits more packets is in one operation cycle. On the other hand, RI-MAC has longer latency due to sender idle listening.

4.3 Summary

Simulation environment and evaluation results are presented in this chapter. Simulation parameters used in the experiments are given in Table 4.1. simulations are presented in diverse network scenarios including clique, chain, grid and random networks. NS-2 simulator is used for simulation of results and gnuplot is used to plot the results in the form graphs. Performance of H-MAC is evaluated in dynamic and wide range of traffic. Evaluation results show that H-MAC outperforms RI-MAC in all network scenarios. Especially H-MAC has low packet delivery latency on multi-hop path in chain networks. Results reveal that, H-MAC experiences low packet delivery latency and it consumes less energizer. Furthermore H-MAC has better PDR compared to RI-MAC in almost all network scenarios and under wide range of traffic loads. Results are quantified in the explanation of the graphs.

Chapter 5

Conclusion And Future Work

WSNs are typically used for emergency applications, such as medical, disaster recovery and battlefield, where a faster event notification is important [58, 59]. Majority of duty cycle MAC protocols conserve energy at the expense at the expense of increase in packet delivery latency, which is not acceptable for time critical applications.

Existing asynchronous duty cycle MAC protocols experience less latency on multi-hop path and they are scalable [60]. Asynchronous duty cycle MAC protocols randomly choose their wake up time during an operation cycle resulting in higher packet delivery latency on multi-hop path. Asynchronous duty cycle MAC protocols could be either sender initiated or receiver initiated. In sender initiated MAC protocols sender transmits either long or short preamble to the receiver to start the data transmission. Preamble transmissions result in extra overhead and causes idle listening of the sender node. In receiver initiated MAC protocols, if sender node want to start data transmission, it has to wait for the beacon from receiver. Sender node waiting for the beacon from receiver results in idle listening. This idle listening of the sender node not just only consumes energy but it also adds significant amount of packet delivery latency.

In this thesis we have presented a new asynchronous hybrid sender and receiver initiated MAC protocol, called H-MAC. H-MAC uses cross layer routing information in order to reduce the packet deliver latency on multi-hop path. Furthermore, H-MAC uses receiver's wake up information for sender initiated data transmission in order to reduce the latency caused by idle listening of the sender. H-MAC enhances number of transmissions in one operational cycle by using on demand sender initiated mechanism.

We have simulated H-MAC for performance evaluation in diverse networks including clique, chain, grid, and random networks. We have used dynamic traffic models i.e., RCE model and we used heavy traffic loads to evaluate the performance of H-MAC. Evaluation of H-MAC through ns-2 simulation show that H-MAC has low packet delivery latency and it conserves more energy, furthermore H-MAC has better packet delivery ratio compared to RI-MAC. Results show that H-MAC outperforms RI-MAC in diverse networks and under wide range of traffic loads.

For future work we can implement multiple channel selection [61, 62] with duty cycle MAC protocol, where we can choose an idle channel from the pool of available channels. When traffic rate increase in one channel we can switch from this channel to another available idle channel. Multiple channel selection not only increases the robustness in communication but it also reduces the probability of collision in one congested channel due to increase traffic rate.

Bibliography

[1] W. Su I. F. Akyildiz and E. Cayirci Y. Sankarasubramaniam. Wireless sensor networks: a survey. *The international journal of computer and telecommunications networking*, Volume 38:pages 393–422, 2002.

[2] ChipCon. *Single chip very low power RF transceiver applications*, 2002.

[3] Shouwen Lai. Duty cycled wireless sensor networks: wakeup scheduling, routing, and broadcasting. Master's thesis, Blacksburg, Virginia, 2010.

[4] Zurich Romer K and Mattern F. The design space of wireless sensor networks. *Wireless communications*, Volume 11:pages 54–61, 2004.

[5] Heidemann Ye and Estrin. An energy efficient MAC protocol for wireless sensor networks. In *21st annual joint conference of the computer and communications societies*, pages 1567–1576, 2002.

[6] T. I. (TI). Cc2420 data sheet. http://www.ti.com/lit/ds/symlink/cc2420.pdf.

[7] Crossbow. Telosb datasheet. http://www.xbow.com/Products/Product pdf files/Wireless pdf/TelosB Datasheet.pdf.

[8] Feng zhang and Demirkol. Energy efficient duty cycle assignment for receiver based convergecast in wireless sensor networks. In *Global telecommunications conference (GLOBECOM)*, pages 1 – 5, 2010.

[9] Hill Polastre and Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on embedded networked sensor systems*, pages 95–107, 2004.

[10] Van Dam and Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *proceedings of the 1st international conference on embedded networked sensor systems*, pages 171–180, 2003.

[11] Kumar Du and Johnson. RMAC: a routing enhanced duty cycle MAC protocol for wireless sensor networks. In *26th international conference on computer communications.*, pages 1478–1486, 2007.

[12] Fabio S Wei Y and John H. Ultralow duty cycle MAC with scheduled channel polling. In *Proceedings of the 4th international conference on embedded networked sensor systems*, pages 321–334, 2006.

[13] Du Sun and Johnson Gurewitz. DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks. In *proceedings of the 9th international symposium on mobile ad hoc networking and computing*, pages 53–62, 2008.

[14] Gary V Y Micheal B and Richard Eric A. XMAC: A short preamble MAC protocol for dutycycled wireless sensor. In *Proceedings of the 4th international conference on embedded networked sensor systems*, pages 307 – 320, 2006.

[15] El-Hoiydi and Decotignie. WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. *computers and communications, 2004*, Volume 1:pages 244–251, july 2004.

[16] Omer Gurewitz Yanjun Sun and David B Johnson. RI-MAC: A receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *proceedings of the 6th conference on embedded network sensor systems*, pages 1–14, 2008.

[17] Jian-nong Cao Hong-wei Tang and Kai Lu Cai-xia Sun. REA-MAC: A low latency routing enhanced asynchronous duty-cycle MAC protocol for wireless sensor networks. *Journal of central south university*, Volume 20:pages 678–687, 2013.

[18] SUN Yanjun Tang Lei and David B J Omer G. PWMAC: an energyefficient predictivewakeup MAC protocol for wireless sensor networks. In *Proceedings of the 30th annual joint conference of the computer and communications societies*, pages 1305–1313, 2011.

[19] A Falchi G Anastasi and M Conti A Passarella. Performance measurements of motes sensor networks. In *Proceedings of the 7th international symposium on modeling, analysis and simulation of wireless and mobile systems*, pages 174–181, 2004.

[20] Lewis Girod Jeremy Elson and Deborah Estrin. Network time synchronization using

reference broadcasts. In *Proceedings of the fifth symposium on operating systems design and implementation*, pages 147–163, 2002.

[21] G. Zhong Ye and Zhang S Lu. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *International conference on distributed computing systems*, pages 28–37, 2003.

[22] Krishna M Sivalingam Christine E Jones and Jyh Cheng Chen Prathima Agrawal. A survey of energy efficient network protocols for wireless networks. *Wireless networks*, Volume 7:pages 343–358, 2001.

[23] Jonathan Hui Joseph Polastre, Jerry Zhao Philip Levis, and David Culler. Networking: a unifying link abstraction for wireless sensor networks. In *Proceedings of the 3rd international conference on embedded networked sensor systems*, pages 76–89, 2005.

[24] Mike Woo Suresh Singh and C S Raghavendra. Power aware routing in mobile ad hoc networks. In *In the fourth annual international conference on mobile computing and networking*, pages 181–190, 1998.

[25] John Heidemann Wei Ye and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Transactions on networking*, Volume 12:pages 493–506, 2004.

[26] Tao Zheng. Pmac: an adaptive energy efficient mac protocol for wireless sensor networks. In *Parallel and distributed processing symposium*, pages 171–180, 2005.

[27] Philip Levis. Tinyos 2.0.2 documentation. http://www.tinyos.net/tinyos-2.1.0/doc/.

[28] Rong Xie Fei Tong and Young Chon Kim Lei Shu. A cross layer duty cycle MAC protocol supporting a pipeline feature for wireless sensor setworks. *Physical sensors*, Volume 11:pages 5183–5201, 2011.

[29] and A E Cerpa A Kamthe, A Carreira-Perpinn. M&M: multilevel markov model for wireless link simulations. In *Proceedings of SenSys*, pages 57–70, 2009.

[30] Jennifer C Hou Rong Zheng and Lui Sha. Resource management: Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th international symposium on Mobile ad hoc networking & computing*, pages 35–45, 2003.

[31] M Kwon Gerla and T. On demand routing in large ad hoc wireless networks with passive clustering,. In *Proceedings of wireless communications and networking (WCNC)*, 2002.

[32] Tao Xie Mohammed I Alghamdi and Xiao Qin. PARM: a power-aware message scheduling algorithm for real-time wireless networks. In *Proceedings of the 1st workshop on Wireless multimedia networking and performance modeling*, pages 86–92, 2005.

[33] Muhammad Mahbub Alam Md Obaidur Rahman, Choong Seon Hong Muhammad Mostafa Monowar, and Sungwon Lee. nW-MAC: multiple wake up provisioning in asynchronously scheduled duty cycle MAC protocol for wireless sensor networks. *Annales des telecommunications*, Volume 9:pages 567–582, 2011.

[34] J D Decotignie El hoiydi and J Hernandez. Low power MAC protocols for infrastructure wireless sensor networks. In *proceedings of the fifth european wireless conference*, pages 675–690, 2004.

[35] S Bock Mahlknecht. CSMA-MPS: a minimum preamble sampling MAC protocol for low power wireless sensor networks. In *Proceedings of international workshop on factory communication systems*, pages 73–80, 2004.

[36] Desheng Zhang Jinbao Li and Shouling Ji Longjiang Guo. ARM: an asynchronous receiver initiated multichannel MAC protocol with duty cycling for wsns. In *Performance computing and communications conference (IPCCC) 29th international*, pages 114–121, 2010.

[37] Shagufta Henna. SA-RI-MAC: sender assisted receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. *Social Informatics and telecommunications engineering*, Volume 81:pages 120–135, 2011.

[38] H W Tseng S H Yang and G H Chen E H K Wu. Utilization based duty cycle tuning MAC protocol for wireless sensor networks. In *Proceedings of the global telecommunications conference*, pages 3257–3262, 2005.

[39] Sang Hoon Lee and Lynn Choi. A+MAC: a streamlined variable duty-cycle MAC protocol for wireless sensor networks. *International journal of distributed sensor networks*, Volume 1:pages 1–8, 2013.

[40] Tai-Hoon Kim Tz Heng Hsu and Jyun Sian Wu Chao-Chun Chen. A dynamic traffic aware duty cycle adjustment MAC protocol for energy conserving in wireless sensor networks. *Journal of distributed sensor networks*, Volume 12:pages 1–10, 2012.

[41] El Hoiydi. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In *international conference on communications*, pages 3418–3423, 2002.

[42] Gregory Hackmann Kevin Klues and Chenyang Lu Octav Chipara. A component based architecture for power efficient media access control in wireless sensor networks. In *Proceedings of the 5th international conference on embedded networked sensor systems*, pages 59–72, 2007.

[43] Garcia Luna Aceves and Asimakis Tzamaloukas. Reversing the collision-avoidance handshake in wireless networks. In *Proceedings of the 5th annual international conference on mobile computing and networking*, pages 120–131, 1999.

[44] Chieh-Jan Mike Liang Razvan Musaloiu-E and Andreas Terzis. Koala: ultra low power data retrieval in wireless sensor networks. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 421–432, 2008.

[45] Ying Qiu Dongyu Yang and Zhigang Li Shining Li. RW-MAC: An asynchronous receiver initiated ultra low power MAC protocol for wireless sensor networks. In *IET International conference*, pages 393 – 398, 2010.

[46] Jeong Woo Jwa EunJin Lee and HeungSoo Kim. MFT-MAC: A duty cycle MAC protocol using multiframe transmission for wireless sensor networks. *International journal of distributed sensor networks*, Volume 1:pages 1–6, 2013.

[47] M Par Won and T Son. Asym-MAC: a MAC protocol for low power duty cycled wireless sensor networks with asymmetric links. *Communications letters*, volume 4:pages 1–4, 2014.

[48] M Riduan Ahmad and Eryk Dutkiewicz. *A survey of low duty cycle MAC protocols in wireless sensor networks*. Croatia : InTech, 2011.

[49] Ki-Il Kim Zheng Teng. A survey on real-time MAC protocols in wireless sensor networks. *Computer science & communications*, Volume.2:pages 104–112, 2010.

[50] T Watteyne and I Auge Blum. Proposition of a hard real time MAC protocol for wireless sensor networks. In *Proceedings of the 13th international symposium on modeling, analysis and simulation of computer and telecommunication system*, pages 533–536, 2005.

[51] K Karenos and V Kalogeraki. Real time traffic management in sensor metworks. In *Proceedings of real time system symposium*, pages 422–434, 2006.

[52] T H Hsu and J S Wu. An application specific duty cycle adjustment MAC protocol for energy conserving over wireless sensor networks. *Computer communications*, Volume 31:pages 4081–4088, 2008.

[53] Fulong Jiang and Hao Liu. An energy efficient asynchronous MAC for bulk data collection in wireless sensor networks. In *Cyber enabled distributed computing and knowledge discovery (CyberC)*, pages 316–319, 2013.

[54] Jaiyong Lee Jaehyun Kim and Seoggyu Kim. An enhanced cross layer protocol for energy effeciency in wireless sensor networks. In *Third international conference on sensor technologies and applications*, pages 657–664, 2009.

[55] Wei Ye Yuan Li and Heidemann. Energy and latency control in low duty cycle MAC protocols. In *Wireless communications and networking conference*, pages 676–682, 2005.

[56] Fabio Silva John Heidemann and Ramesh Chalermek Intanagonwiwat. Building effecient wireless sensor networks with low level naming. In *Proceedings of the symposium on operating systems principles*, pages 146–159, 2001.

[57] Kyle Jamieson Bret Hull and Hari Balakrishnan. Mitigating congestion in wireless sensor networks. In *proceedings of the second international conference on embedded networked sensor systems*, pages 134–147, 2004.

[58] W B Chandrakasan Heinzelman and A P Balakrishnan. An application specific protocol architecture for wireless microsensor network. *Journal on Wireless Communications*, Volume 1:pages 12–24, 2002.

[59] M Kuorilehto Kohvakka and M Hamalainen M Hannikainen. Performance analysis of 802.15.4 and zigbee for large scale wireless sensor network applications. In *Proceedings of international workshop on performance evaluation of wireless ad hoc, sensor, and ubiquitous networks,*, pages 1–6, 2006.

[60] Anna forster. Emerging communications for wireless sensor networks. Master's thesis, Janeza Trdine, Rijeka, Croatia, 2010.

[61] Jinbao Li and Shouling Ji. ARM: An asynchronous receiver initiated multichannel MAC protocol with duty cycling for WSNs. In *Performance computing and communications conference (IPCCC)*, pages 114–121, 2010.

[62] Desheng Zhang Jinbao Li and Shouling Ji Longjiang Guo. M-cube: A duty cycle based multi channel MAC protocol with multiple channel reservation for WSNs. In *Parallel and distributed systems (ICPADS)*, pages 107–114, 2010.

Appendix A

Topologies

A.1 Clique Topology

A.1.1 25flows.TCL

```
# flows=25 nodes=50

1 1 100 -1

2 1 101 15

3 2 100 15

4 2 101 16

5 3 100 16

6 3 101 15

7 4 100 -1

8 4 101 -1

9 8 102 -1

10 8 103 -1

11 8 107 -1

12 9 108 17

13 1 120 17

14 1 125 -1

15 3 130 17

16 3 135 17

17 8 140 11

18 1 145 -1

19 1 150 -1

20 2 155 -1
```

```
21 7 100 -1
22 9 102 -1
23 12 104 -1
24 15 107 -1
25 16 110 -1
26 20 112 11
27 20 116 11
28 22 118 11
29 24 120 15
30 25 125 15
31 4  137 7
32 4  140 7
33 7   141 8
34 9   142 8
35 12  143 8
36 20 103  -1
37 22 103 -1
38 24 104 -1
39 27 135 7
40 28 125 -1
41 10 110 -1
42 15 110 -1
43 3  134 -1
44 14 155 -1
45 17 159 -1
46 18 102  18
47 20 129 18
48 28 140 32
49 29 109 21
50 40 130 18
```

A.2 Chain topology

A.2.1 25nodes.TCL

```
1       200     500     -20
2       400     500     -1
3       600     500     -1
4       800     500     -1
5       1000     500     -1
6       1200     500     -1
7       1400     500     -1
8       1600     500     -1
9       1800     500     -1
10       2000     500     -1
11          2200     500     -1
12          2400     500     -1
13          2600     500     -1
14        2800       500     -1
15        3000       500     -1
16        3200       500     -1
17        3400       500     -1
18        3600       500     -1
19        3800       500     -1
20        4000       500     -25
21        4200       500     -1
22        4400       500     -1
23        4600       500     -1
24        4800       500     -1
25        5000       500     -1
```

A.3 Grid topology

A.3.1 7x7RCE.TCL

```
1  200  200  25
2  200  400  25
3  200  600  25
```

```
4 200 800 25
5 200 1000 25
6 200 1200 25
7 200 1400 25
8 400 200 25
9 400 400 25
10 400 600 25
11 400 800 25
12 400 1000 25
13 400 1200 25
14 400 1400 25
15 600 200 25
16 600 400 25
17 600 600 25
18 600 800 25
19 600 1000 25
20 600 1200 25
21 600 1400 25
22 800 200 25
23 800 400 25
24 800 600 25
25 800 800 -1
26 800 1000 25
27 800 1200 25
28 800 1400 25
29 1000 200 25
30 1000 400 25
31 1000 600 25
32 1000 800 25
33 1000 1000 25
34 1000 1200 25
35 1000 1400 25
36 1200 200 25
37 1200 400 25
```

```
38 1200 600 25
39 1200 800 25
40 1200 1000 25
41 1200 1200 25
42 1200 1400 25
43 1400 200 25
44 1400 400 25
45 1400 600 25
46 1400 800 25
47 1400 1000 25
48 1400 1200 25
49 1400 1400 25
```

A.4 Random Network with RCE model 50 nodes

A.4.1 Rand-network1.TCL

Node 6 randomly selected as sink node

```
#
# nodes: 50, pause: 0.00, max speed: 0.00, max x: 1000.00, max y: 1000.00
#
1 570.992715369557 462.676400373851 6
2 965.238082867470 774.406422413079 6
3 389.927221253715 265.254061942186 6
4 843.254166514815 427.473385565557 6
5 413.064383183930 324.265973979449 6
6 0.332720128169 970.803474291955 -1
7 560.087023651717 235.585161019516 6
8 456.244388338838 998.716153607927 6
9 87.766153797358 972.216039947452 6
10 202.800154728450 346.733612967793 6
11 254.044007007301 372.834599239192 6
12 367.496714331051 201.870929912933 6
13 328.435128860759 643.183019194302 6
14 506.461239034296 477.479710084335 6
```

15 353.148259095577 905.431249488541 6

16 692.607043791159 325.898048883955 6

17 634.741778258767 364.222200531098 6

18 701.224381070275 489.196977753418 6

19 920.154271971455 649.925623364870 6

20 524.341626340304 3.991269932637 6

21 395.502837203580 157.285442509542 6

22 672.990152142465 895.67262337075 6

23 6.612285593374 932.626144724488 6

24 105.802729960291 802.284518227722 6

25 356.872082974155 589.425081526958 6

26 388.846157090739 990.895571910376 6

27 386.096361630606 969.384266909195 6

28 72.477825934856 886.585700188257 6

29 711.967091096834 278.108011895434 6

30 646.133358216784 975.605895725551 6

31 910.798195620539 514.460850974512 6

32 445.979153915230 127.091298912417 6

33 270.955838113747 608.878198928820 6

34 784.566840666789 904.185832960218 6

35 407.091244979524 345.730228329051 6

36 226.689579931887 251.284933711231 6

37 275.184049093696 36.456106133496 6

38 765.804716918474 537.373251694636 6

39 258.351938970667 561.864723420670 6

40 670.107077896193 379.425116330484 6

41 23.981037314063 718.444614539966 6

42 209.500711778213 5.559028628347 6

43 273.980314374879 458.049278071674 6

44 269.269421698535 778.053931853552 6

45 142.541978892109 633.415786491354 6

46 395.558450202494 594.937195709659 6

47 623.552338615732 420.434674120138 6

48 385.721327790533 886.685101881274 6

```
49 956.665875619860 930.375443892109 6
50 352.740991015482 845.013620276664 6
```

A.4.2 Rand-network2.TCL

Node 9 randomly selected as sink node

```
#
# nodes: 50, pause: 0.00, max speed: 0.00, max x: 1000.00, max y: 1000.00
#
1 290.261929476280 38.682347872743 9
2 843.948644711943 20.157471809712 9
3 299.775818212277 880.618818143549 9
4 475.143589505429 324.262422380639 9
5 724.585508162572 955.033128533254 9
6 935.266143767014 648.620116504138 9
7 915.192033480840 452.630368794109 9
8 973.694359541039 302.538357192645 9
9 908.656234853085 714.170635805347 -1
10 254.641817874624 931.232802266353 9
11 829.091825627512 915.768011119158 9
12 905.670245266382 276.875482776691 9
13 959.297703237720 933.672402800028 9
14 998.591029715476 873.363910396512 9
15 928.875416332410 708.759316341471 9
16 656.674498363467 692.619233640097 9
17 333.873845740631 992.187830241180 9
18 307.753029980844 801.854476981268 9
19 287.017605663199 988.528927931100 9
20 915.201500142438 734.473964378840 9
21 958.473194940580 626.367562516698 9
22 488.400178865352 936.902525805804 9
23 43.625005072449 346.477453612562 9
24 368.902049197728 203.962060954447 9
25 353.930349838341 622.662069861244 9
26 496.890884673527 741.588664299446 9
```

```
27 119.796914029344 879.160398586039 9
28 629.065572713883 894.893000865761 9
29 688.315838847704 463.464372651789 9
30 763.872024343671 293.836793424108 9
31 769.858219225544 791.343065118268 9
32 485.406514016109 510.647703943477 9
33 562.159694481924 333.552584838806 9
34 236.305724399069 985.056628680736 9
35 188.159899352412 70.962804546641 9
36 995.300928601668 324.590952022681 9
37 63.997309494633 402.180026903154 9
38 490.230354705805 217.922581203258 9
39 831.046197530257 314.532275410070 9
40 989.324255329428 869.351061253115 9
41 357.809539983139 37.248759471751 9
42 226.742048785637 150.385758904796 9
43 911.373491996361 243.171226368196 9
44 673.032885415210 324.686471963019 9
45 911.493053331635 903.017994907625 9
46 409.310827762040 396.934745498567 9
47 941.493021517654 407.897891672971 9
48 30.668439198992 217.213198118924 9
49 356.998561941018 157.113587642002 9
50 545.806045301179 965.595977810203 9
```

A.4.3 Rand-network3.TCL

Node 25 randomly selected as sink node

```
#
# nodes: 50, pause: 0.00, max speed: 0.00, max x: 1000.00, max y: 1000.00
#
1 834.666499078887 667.878620307602 25
2 533.784569014607 516.866064283098 25
3 574.126822040039 134.596917311698 25
4 614.984306254596 598.983794587811 25
```

```
5 123.587565428161 474.429237814919 25
6 764.118263948848 22.348768228792 25
7 724.518235004459 839.308056648838 25
8 388.488563430873 513.519884974727 25
9 620.603784473051 916.165756192635 25
10 962.943118832113 565.579452933866 25
11 658.987079995990 599.128270665808 25
12 681.625052303544 133.024823542024 25
13 370.475177899664 110.829889320912 25
14 537.564017766462 95.327441307741 25
15 789.501601182002 874.359011851874 25
16 105.611964121295 185.371834216020 25
17 304.050080068972 939.427164010901 25
18 982.553587148699 211.394924896337 25
19 59.508479055428 990.878635342875 25
20 357.746002127223 976.416525918673 25
21 635.965572269829 712.085423319081 25
22 115.031711274431 191.163118407579 25
23 602.500185444960 426.424430845371 25
24 100.886676689710 920.217900631326 25
25 715.655915638532 238.143260482186 -1
26 701.294429569794 990.298365471433 25
27 771.165106306398 381.048568351684 25
28 0.040260611189 687.339166171510 25
29 338.298363463501 156.505980192042 25
30 954.191414516376 796.776859026772 25
31 568.003443336281 850.927621590194 25
32 280.820474589863 67.295695654467 25
33 172.083285821910 162.678953687945 25
34 68.140320054532 409.205238594369 25
35 135.0739025732900 981.201064095325 25
36 763.756089811508 328.648517457510 25
37 758.657423965806 669.241581857728 25
38 564.680197381759 30.693459181171 25
```

```
39 138.078696727839 692.945416581968 25
40 430.019618115407 228.397773463916 25
41 461.608867397216 641.349887568684 25
42 184.009606082458 824.171708297850 25
43 179.092253151161 110.477833538174 25
44 576.796922128126 288.573269039216 25
45 556.340610310167 917.723607478317 25
46 668.179092691571 320.756435095644 25
47 637.108403611590 209.451113260778 25
48 906.457853164345 924.076405867909 25
49 860.389237283022 242.651041706888 25
50 118.903843623586 199.383586522142 25
```