EFFICIENT MAC PROTOCOL DESIGN FOR WIRELESS SENSOR NETWORKS

BY

KIEN NGUYEN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Department of Informatics,
in the School of Multidisciplinary Sciences of the
Graduate University for Advanced Studies, 2012

Tokyo, Japan

Abstract

The past two decades have seen increasing interests in the field of wireless sensor networks (WSNs), which have potential applications covering all aspects of the human life. The major issue in WSNs is power conservation since wireless sensors are usually battery-powered. In a typical sensor node, the wireless interface consumes the largest share of the power budget. Hence, an energy efficient medium access control (MAC) protocol is vital. The MAC protocol always adopts the duty cycling mechanism to reduce idle listening, which is the most significant energy wastage. The mechanism, however, has negative effects on latency and throughput performance. Meanwhile, an increasing number of prospective applications not only imposes requirements on energy efficiency but also on other Quality of Services (QoS) parameters. Therefore, it is great of importance to design new efficient MAC protocols, which meet the energy efficiency and QoS requirements.

We first focus on designing energy efficient, low latency MAC protocols for low data rate WSNs. The traditional approach is letting a duty cycling MAC protocol forward packets via multiple hops in a cycle, i.e., the multi-hop MAC. However, the original multi-hop MAC protocol incurs a large control overhead, and a so-called long listening period problem. We propose a low latency, low control overhead MAC protocol (the LO-MAC), which overcomes the mentioned disadvantages by exploiting the physical properties of wireless channel. LO-MAC introduces a new traffic adaptive scheme based on carrier sensing characteristics. The scheme effectively controls the length of listening period following the traffic load. Moreover, LO-MAC takes full advantages of the broadcast nature and lets a packet containing different meanings during its transmission. Therefore, the number of transmitting packets and the

control overhead is significantly reduced.

Secondly, we introduce an approach in designing efficient MAC protocols for dynamic load environments. We propose MAC^2 protocol, a novel Multi-hop Adaptive MAC protocol with packet Concatenation. MAC^2 achieves a better performance than a state-of-the-art protocol in terms of energy efficiency, low latency and high throughput. The proposed protocol controls the adaptation to the traffic load by combining a signalling traffic adaptive scheme and a demand wakeup manner. The scheme and the manner are based on a synchronization process and a proportional mapping function, respectively. Besides that, the protocol has a concatenation scheme, which concatenates several queued packets into a bigger one before sending out of a node. The concatenation scheme reduces not only the control overhead but also the average latency. Additionally, MAC^2 is numerically optimized to achieve minimum latency and guarantee no data transmission collision.

Finally, we also target the dynamic load environments, but take an asynchronous approach of the efficient MAC design. We propose an Asynchronous MAC protocol with QoS awareness (the AQ-MAC), which is energy efficient and provides different QoS levels to relevant types of traffic. AQ-MAC achieves energy efficiency and collision avoidance by utilizing a receiver-initiated transmission and the concatenation scheme derived from MAC^2. Moreover, AQ-MAC adopts the differentiate service (DiffServ) model to provide QoS. Data packets are provided different transmission strategies depending on their levels of importance.

To My Family.

Acknowledgments

First and foremost, I would like to express my profound gratitude to my supervisor, Professor Yusheng Ji for opening my eye to research world and turning the direction of my life into a better way. Professor Ji's motivation, enthusiasm, patience, and immense knowledge make her a terrific advisor. I appreciate her professional skills in doing research and supervising students. Her research activities exhibit a role model for me during my studies at the Graduate University for Advanced Studies and in my future career.

In addition, I would like to send my thankful messages to the great dissertation members, Professor Yamada Shigeki, Professor Abe Shunji, Professor Kensuke Fukuda, Professor Koibuchi Michihiroi, and Professor Kenji Tei. Their expertise knowledge, five-year patience, numerous valuable guidances, fully-experienced comments and professional suggestions help me gradually gaining my research skills and my confidence from zero.

I also would like to acknowledge the Graduate University for Advanced Studies, National Institutes of Informatics (NII) for awarding me a MEXT scholarship, providing me with financial supports to complete five years of my doctor course. I especially thank the NII international support team, who helps me get familiar with Japanese environment both in life and research.

Moreover, I would like to thank all my amazing lab-mates and colleagues at the Graduate University for Advanced Studied and the National Institute of Informatics. They have been given me so much inspiration and collaboration in many works. Among them, I would like to deeply appreciate all members in Professor Yusheng Ji's laboratory and the group of Vietnamese at NII. I am also deeply grateful to Professor Duy-Dinh Le and Doctor

Jumpot Phuritatkul for their supports, guidances, and discussions since my early stage at the university. The memory with all friends during the graduate university period is going to last forever in my mind.

Last but not least, I would like to thank my family member from the bottom of my heart my farther Nguyen Khac Chung, my mother Tran Thi Lan, my wife Phan Nguyen Ngoc Nga, my sister Nguyen Thi Huong Giang, and my son Nguyen Gia. They have been not only bringing happiness to my life but also enduring this long process with me, always fully offering support and love. This dissertation is dedicated to you.

Table of Contents

List of Figures

Chapter 1

List of Abbreviations

WSN	Wireless Sensor Network.
MAC	Medium Access Control.
CPU	Central Processing Unit.
RTS	Request To Send.
CTS	Clear To Send.
ACK	Acknowledgment.
PION	Pioneer.
SCH	Scheduling.
MEMS	Micro-Electro-Mechanical Systems.
SIFS	Short Inter-Frame Space.
DIFS	Distributed Inter-Frame Space.
CCA	Clear Channel Assessment.
LPL	Low Power Listening.
TDMA	Time Division Multiple Access.
CSMA	Carrier Sense Multiple Access.
WLAN	Wireless Local Area Network.
RAM	Random Access Memory.
QoS	Quality of Service.
Tx_Range	Transmission Range.
CS_Range	Carrier Sensing Range.
RCE	Random Correlated Event.
IntServ	Integrated Services.
DiffServ	Differentiated Services.

Chapter 2

Introduction

The major problem prevents a pervasive deployment of wireless sensor networks (WSNs) is that a sensor node generally has a limitation on the battery capacity. Therefore, it is necessary to achieve energy efficient operations of WSNs, as well as to meet the applications' quality of service (QoS). The necessity motivates the development of efficient communication protocols in WSNs. This dissertation focuses on designing efficient medium access control (MAC) protocols for WSNs.

2.1 Wireless Sensor Networks

The ongoing development of the sensing technology and wireless communication yields wireless sensor networks, which have many applications covering all aspects of human life. A typical WSN usually contains a huge number of sensor nodes that are capable of sensing or measuring miscellaneous parameters of their surrounding environments. The sensor nodes are wirelessly connected in order to cooperatively convey the sensing data to one or a limited number of destinations, i.e. sinks, for further processing. Besides that, the sensor nodes are generally expected to be inexpensive and small size, hence the resources on a sensor node are limited.

In this section, we first give discussions of the resource limitation issue. Secondly, we present a short survey of the applications of WSNs.

(a) Micaz/Mica2 (b) TelosB (c) Iris

Figure 2.1: MEMSIC sensor motes

2.1.1 Resource limitations

In order to observe the resources on sensor nodes, we have taken the same approach introduced in [1]. In the first step, we have chosen the sensor networking products, which are popularly used in academia. In the following step, the performance parameters have been either collected from the products' datasheets or calculated by using the metrics calculation method in [2]. As a result, the products of MEMSIC company (former company named Crossbow) [3] are selected. They are Mica2/Micaz [4], TelosB [5], and IRIS motes as shown in Fig. 1.1 (*Note that: in WSNs, a node is similar to a mote*). We then give the below discussions that are oriented by designing MAC protocols. The discussed performance parameters includes Central Processing Unit (CPU), memory, radio, and the power consumption.

Central Processing Unit: Table 1.1 shows several types of processors including 8-bit and 16-bit ones with the clock rate around 8 Mhz. In fact, there are also existing more powerful processors that can support not only a wide range of clock rate but also high or extremely high frequency. For example, the 32-bit Intel PXZ271 on Imote2 mote supports the range

Table 2.1: CPU Specifications

	Mica2	Micaz	TelosB	IRIS
Chip	ATmega128L 8-bit, 8Mhz	MPR2400 8-bit, 8 Mhz	TI MSP430 16-bit, 8 Mhz	XM2100CB 8-bit, 8-Mhz
Power consumption	36 μW sleep 60 mW active	36 μW sleep 60 mW active	15 μW sleep 5.4 mW active	36 μW sleep 31 mW active

Table 2.2: Memory Specifications

	Mica2	Micaz	TelosB	IRIS
RAM	4K bytes	4K bytes	10K bytes	8K bytes
Program Flash	128KB	128 KB	48K bytes	128K bytes

of 13-416 Mhz , moreover the ability of processor maybe gain much more following Moore's law. Therefore, the demands of potential applications are generally satisfied interm of the CPU.

Memory: The memory-related parameters are shown in Table 1.2. We observe that the size of Program Flash memory is generally enough for storing the program code since the size of compiled code mainly depends on the application that normally requires several simple tasks. Moreover, in case of using programing platform and programing language such as TinyOS [6] and nesC [7] the size of compiled code can be further reduced. The remained problem is the amount of RAM available for data processing (e.g., the 4KB on Micaz or 10KB on TelosB). Hence, it is necessary that developers need to put more efforts in oder to minimize the memory footprint of their softwares.

Radio: Radio or wireless module is no doubt a key factor in the formation of WSNs. The specifications of the radios (CC1100 on Mica2, CC2420 on Micaz and TelosB, and RF320 on IRIS) are listed in the Table 1.3. The bandwidth provided by the radios are either 28.4 Kbps or 250 Kbps. Those values are indeed very low if comparing with other wireless technologies such as Wifi or Wimax. However, the bandwidth in most WSN applications is required only a fraction of that, bandwidth is hence not an issue. The big problem here is the poor performance in terms of transmission range (indoor and outdoor) and wireless link quality.

Table 2.3: Radio Specifications

	Mica2	Micaz	TelosB	IRIS
Radio chip	CC1100	CC2420	CC2420	RF230
Indoor range	20 m to 30 m	20 m to 30 m	20 m to 30 m	50 m
Outdoor range	75 m to 100 m	75 m to 100	75 m to 100 m	300 m
Bandwidth	38.4 Kbps	250 Kbps	250 Kbps	250 Kbps
Power consumption	100 μW sleep 36 mW receive 75 mW transmit 2 ms setup	60 μW sleep 63 mW receive 57 mW transmit 1 ms setup	60 μW sleep 63 mW receive 57 mW transmit 1 ms setup	120 sleep 63 mW receive 75 mW transmit 2 ms setup

Therefore, designing MAC protocols that efficiently solve the problem at a certain level is major concern.

Power consumption: Table 1.1 and Table 1.3 show the power consumption parameters of CPU and radio, respectively. A sensor node totally consumes around 100 mW whenever it is active. If the node is equipped by a pair of AA batteries (3000 mAh), it will run out of energy in about 100 hours or five days. Moreover, it is obvious that the lifetime of sensor node depending on how long both the CPU and radio are turned off (i.e., the power consumption in active state is much bigger than the one in sleep state. If comparing the parameters of energy consumption, we can recognize that the radio main module consumed energy in sensor nodes is radio. Hence at the MAC layer that control the operation of radio, power efficiency is the most essential issue.

All the above discussions drive to a conclusion keeping efficient operations of the radio plays the most important role in wireless sensor networks. Specifically, the energy saving MAC protocol, which also efficiently share the wireless channel, is a must.

2.1.2 Applications

Nowadays wireless sensor networks are going into real deployments covering various areas, e.g., military, industry, environment monitoring, and health care for human/structures. Moreover, sensor networking communication plays an important role in the development of Internet of things (IoT) [8, 9]. As mentioned earlier, the common characteristic of WSNs is wireless sensor nodes are deployed to monitor the environment phenomena and relay sensing data to a sink node. Therefore, there exists a classification of WSNs are based on the methods of monitoring applications: periodic monitoring of key parameters and event-based reporting of outliers [1]. However, instead of following that classification, we give a brief survey of the applications in specific areas, which potentially include large number of deployments of WSNs in the future.

Environment monitoring: Environment or habitat monitoring is one of the pioneer applications of sensor networks. By distributing numerous number of sensors over monitoring area, many aspects of environment are monitored in a reliable manner. One of the most typical examples is the ZebraNet project of Princeton University [10]. WSN in the project includes wireless sensor nodes equipped on the necks of wild zebras. The Zebranet, therefore, has capability of monitoring the migration of zebras, inter-species interactions, and even nocturnal behaviors. Another typical example is a bird observation project on Great Duck island [11] leading by University of California at Berkeley. Wireless sensor nodes are deployed in the island in order to measure environment parameters such as the humidity, pressure, temperature, etc., in the birds burrows. The monitoring data is sent to the base station which are connected to Internet. Therefore, researchers can see and understand birds behaviors real-time from remote sites.

Military: Wireless sensor networks can be used in the military for a number of purposes such as monitoring, tracking enemies, building an intrusion detection system [12], etc. The most mentioned application is achieving effective situational awareness in a battlefield [13]. In the battlefield area, a large number of wireless sensor nodes is scattered by soldiers or from an airplane. The wireless sensors then automatically connect and form a network. Hence, the battlefield information are collected and relayed to control centers via the network. WSNs also can help detect snipers based on analyzing the sound of a gun shot [14] by using acoustic sensing and wireless communication. Moreover, underwater WSNs [15–17] can assist naval forces by improving the accuracy of object detection under the oceans.

Structure health monitoring (SHM): SHM becomes another important application of WSNs [18, 19] in order so reduce the huge amount of money, which is spent for maintaining structures e.g., bridges, towers, etc. By using the advanced WSNs, the structures such as heritage building [20], railway bridge [21] are reliably and real time monitored. Precursors of the damages are easily discovered, localized, and efficiently repaired. A typical example is a bridge monitoring project by Hong Kong Polytechnic University [22]. In the projects, new SHM motes are designed and deployed along a bridge, the sensor network guarantees 100% data delivery to a storage center. This lets the maintenance cost is significantly decreased.

Smart energy management: Energy saving has been an emerging global issue. In order to increase the efficient usage of electricity, applying WSNs shows a great potential [23–25]. By using sensor networks in homes and offices, the automation of electrical devices is enabled, the unnecessary operations decrease hence saving energy [26]. Moreover, the capability of wireless communication lets different devices in same building or even in different buildings can communicate forming smart electricity grid. As a result, the information of energy consumption is relayed and remote management operations can be implemented. One of the leader in this WSN direction is also University of Berkeley at California [27].

Others: There are also many areas where sensor networks can help to save deployment, maintenance cost, or reduce carbon footprint [28]. For example, in many industrial food manufactures [29] or agriculture farms [30, 31], using sensor networks achieves automatic monitoring and reporting, the accuracy, safety, and quality of the products are controllable. Besides that, WSNs also show the social impacts in various forms of application. A typical one is disaster recovery when volcano earthquake, tsunami, or fire occur [32–36]. In those disaster scenarios, the essential tasks such as locating survivor, emergency search and rescue can be effectively achieved by pre or post disaster deployments of sensor networks. Moreover, a branch of wireless sensor networks named body sensor networks have been widely used in order to improve the community health and assisting elder people [37–42].

2.2 Challenges in Designing MAC Protocols

In many WSNs, sensor nodes are usually battery-powered and it is often neither feasible nor practical to change or recharge batteries. Therefore, the most essential task is to make the sensor nodes save as much energy as possible. Given that the radio unit is the most power consumer within the sensor node, then a significant amount of energy is expectedly saved through controlling the radio operation. The MAC protocol that directly controls the radio unit necessarily achieves energy efficiency. Besides that, the MAC protocol should consider a set of performance attributes, and make trade-offs between them when necessary. In this section we first discuss requirements of the attribute set. Then we focus on the requirements related to energy efficiency.

2.2.1 Requirements

The requirements in designing MAC protocols for WSNs are thoroughly surveyed in previous works [43–45]. In this section, we readdress and discuss the most important requirements below.

Collision avoidance is the major task and the inherent attribute of all MAC protocols. The MAC protocols need to determine when and how nodes can access the sharing medium and achieve sending/receiving data. However, the collisions in wireless communication are not completely avoided in regular operations. Therefore, the MAC protocols should be designed to avoid frequent collisions, as well as guarantee acceptance levels of collisions.

Energy efficiency is one of the principal task of communication protocols in WSN since energy is a scarce resource. The radio consumes the biggest amount of a node's batteries when it either takes part in long range transmissions or is turned on all the time. Therefore, energy aware MAC protocols are expected to eliminate such energy wasting conditions as much as possible. Specifically, the protocols should save the transmission and reception energy by limiting negative effects such as collisions or transmission of unnecessary messages. Moreover, the protocols should let the radio be in low power or sleep state whenever it is idle. Finally, the protocols need to avoid the excessive transitions among active and sleep states.

Delivery latency is the portion of time required to relay a packet since it is generated by a sender until it is successfully received by a receiver. In sensor networks, the importance of delivery latency depends on applications or traffic patterns. Moreover, the relaying process is always implemented via multiple hops. Therefore, the MAC protocols should be carefully designed to match the latency requirement in most cases trade-offs with other requirement need to be evaluated.

Reliability (or reliable delivery of data) is a classical design goal for all kind of networks. The requirement of reliability in WSNs is unique and more difficult than other wireless network since it should be ensured in case of wireless environments and limited capacity devices. In general wireless networks, the causes of packet drop are mainly buffer overflow and signal interference. The former cause could be avoided through employing a buffer management strategy at MAC protocol. The strategy makes decision on stopping the number of backlogged packets when they exceeds the maximum value of buffer size [46]. The later cause can be minimized through the use of sufficiently high transmission power and the prevention of contention for medium access among nodes. The expected MAC protocol should adopt the mentioned strategies to achieve the reliability.

Scalability and adaptability are the requirements related to changes in network size, node density, and topology. In WSNs, the changes frequently happen since the nodes might die because of running out battery, the network adds new nodes, and the wireless connectivity is varied and interference. A good MAC protocol should efficiently control the radios dealing with such network changes.

Mobility poses a challenge to the MAC protocol design not only in WSNs but also in general wireless networks. The MAC protocol should adapt itself to changes in mobility patterns of sensor nodes or sensing events.

Throughput refers to the amount of data successfully transferred from a sender to a receiver in a given time. Many factors affect the throughput in sensor network including efficiency of collision avoidance, latency, channel utilization, and control overhead. Similar to latency, the importance of throughput depends on application, hence the MAC protocol should consider the trade-offs between the designing requirements.

Fairness reflects the ability of different nodes or applications to share the networking resources equally. In sensor networks, several nodes cooperate for a single common task, one node may have more data to send than other nodes at a particular time. Thus, rather than treating each node fairly, guaranteeing the good performance of the application is more important, hence the fairness is not a dominant issue in designing the MAC protocol.

In short, we discuss the above requirements reflected the characteristics of a MAC protocol in WSNs. The most important factors are effective collision avoidance and energy efficiency.

Others are normally of secondary importance or application-dependent.

2.2.2 Energy Efficiency

Prolonging the sensor node lifetime and keeping network operation viable as long as possible are the important issues in sensor networks. As the previous discussion, energy efficiency is the first design goal of MAC protocols in WSNs. In order to save energy at MAC layer, we should consider the sources that cause energy wastage and make sensors' batteries drain quickly [47–50]. A list of energy wastage sources, which should be tackled in designing MAC protocols, is mentioned below.

Packet collision is the most dominant source of energy wastage, especially in wireless communication. The collision occurs when two or more packets are transmitted at the same time, then become corrupted, and must be discarded. As a result, the retransmissions of the colliding packets are required, the energy consumption is hence significantly increased.

Packet overhearing means that a node receives packets that are not destined to it. The overhearing can be another dominant source of energy wastage under heavy load environments or in dense networks. In WSNs, the dense networks are common since the sensing ranges of many physical sensors are much smaller than the communication range.

Idle listening refers to the wastage caused by keeping a node's radio on without doing anything. In WSN, a radio unit has four distinct modes of operation; idle, receive, transmit, and sleep. The measurements show that the amount of energy consumption is almost similar in the cases of transmit, receive and idle modes. It is thus desirable to completely shut down the radio rather than switching into the idle mode. However, frequent switching between modes, especially switching from a sleep mode to an active mode, leads to more energy consumption than leaving the radio transceiver unit in idle mode because of the start-up power [51]. The MAC protocols should avoid the frequent switching.

Control overhead is the energy consumed by exchanging control packets. The control packets are necessary to avoid collision and share medium in wireless networks. However, the control overhead becomes a major source of energy wastage since the size of control packet is comparable with the one of data packet in WSNs. Moreover, the control packets do not directly convey useful information related to application data, hence they also reduce the effective throughput.

Dynamic traffic reflects the variation and fluctuation of the traffic in place and time. The dynamic traffic is very popular in WSNs since the sensing events are randomly happened and may cause unpredictable peak loads. That may drive the sensor networks into congestion, which consequently raises the collisions probability. Therefore, a huge amount of time and energy are wasted on the back-off procedure.

2.3 Contributions

This dissertation aims to design new efficient MAC protocols tailored for specific applications of sensor networks. By using the term *efficient MAC protocols* we mean that the protocols both achieve energy efficiency, as well as meet the requirements of applications. The contributions of the research include three new proposed MAC protocols and a set of inheritable techniques for designing MAC protocol in WSNs.

In the beginning work, we propose the low control overhead MAC protocol (LO-MAC), which is a low latency, energy efficient, multi-hop MAC protocol. The targeted application of LO-MAC is low data rate wireless sensor networks. Since LO-MAC is a multi-hop MAC (i.e., combining duty cycling and multi-hop forwarding), the protocol reduces not only idle listening but also end-to-end latency of relaying data via multiple hops. Furthermore, LO-MAC leverages common characteristics of wireless communication to save energy consumption in lightweight manners. First, LO-MAC introduces a traffic-adaptive mechanism, which is based on the fact that a node can sense a busy channel when a packet transmission is within the node's carrier sensing range. Hence, the mechanism uses carrier sensing as a binary signal of the data existence; the existence data is effectively notified to other neighbouring nodes. The nodes then either keep their radios on to take part in multi-hop data forwarding or turn them off to save energy. Second, LO-MAC takes full advantage of the broadcast nature of wireless channel and lets a packet convey several meanings when it is in broadcast region of other nodes, e.g., upstream and downstream nodes. Therefore, control overhead and overhearing energy are significantly reduced since the number of transmitting packets is reduced.

In the following work, we introduce a new MAC protocol called MAC^2 (Multi-hop Adaptive with packet Concatenation MAC) for dynamic load environments. MAC^2 achieves energy efficiency, low latency, high throughput, and high delivery ratio by combining three promising techniques into one protocol. First, the idea of forwarding packets over multiple hops within one operational cycle as similar as in LO-MAC. Secondly, an adaptive method that adjusts the listening period according to traffic load minimizing idle listening. Thirdly, a packet concatenation scheme that not only increases throughput but also reduces power consumption that would otherwise be incurred by additional control packets. Furthermore, MAC^2 incorporates the idea of scheduling data transmissions with minimum latency, thereby performing packet concatenation together with the multi-hop transmission mechanism in a most efficient way.

In the final work, we develop the AQ-MAC (Asynchronous QoS-aware MAC) protocol, which is an asynchronous MAC for the dynamic load environments. The environments contains different type of traffic, i.e., high and low priority; each of them has a different QoS requirement. Similar to other MAC, AQ-MAC also achieves duty cycling the usage of radio to avoid idle listening. Besides that AQ-MAC adopts the receiver-initiated manner, which effectively avoids contention in WSNs. By letting receiver to initiate the rendezvous for transmissions between senders and receivers, the protocol effectively avoids collision and shortens end-to-end latency. Moreover, AQ-MAC provides QoS service per packet following the priority imposed in the packet. When a node has an incoming packet with high priority, it immediately turns on the radio in order to wait for a transmission initiated by a receiver. Otherwise, the node keeps the low priority packets in a queue and sends out in a burst until a high priority packet comes or after a timeout value. In addition, AQ-MAC inherits the packet concatenation scheme to improve the energy efficiency by reducing control overhead.

2.4 Dissertation Organization

The remainder of dissertation is organized as follows. Chapter 2 elaborates on the related work in the area of designing efficient MAC protocols for sensor networks. In chapter 2, we discuss the state-of-the-art MAC protocols, their limitations, and well-known techniques widely adopted in the field. Chapter 3 presents our first approach to design a new energy efficient, low latency MAC protocol in low data rate environments. In Chapter 3, we propose LO-MAC, which successfully achieves those design goals. In chapter 4, we introduce MAC^2, an energy efficient, low latency, high throughput MAC protocol for dynamic load wireless sensor networks. Chapter 5 gives the details of AQ-MAC protocol, which is an energy efficient, low latency, and QoS-aware MAC for dynamic load environments. Finally, in chapter 6, we summarize the contributions of the dissertation, and discuss about future work.

Chapter 3

Related Work

3.1 MAC Protocols in WSNs

3.1.1 Duty Cycling

Given that idle listening, i.e., the state in which a node merely awaits incoming packets, is the most significant source of energy wastage [44, 52]. Therefore, it is necessary to find an effective mechanism to minimize this burden. The best and widely adopted solution is a so-called duty cycling mechanism, in which the radio module is frequently turned on and off following the operational cycles. The mechanism is broadly used not only in the MAC layer but also in the other layers of communication stack in WSNs. The nodes can implement packet transmission when their radios are on, otherwise, they put the radios in sleep state, i.e., power saving state. Figure 2.1 shows the original concept of duty cycling; W, S is denoted for the duration of wakeup and sleep period, respectively. The duration values are static regardless of traffic and pre-installed on the nodes, hence the mechanism allows the node be awake for at most W in an operational cycle. In multi-hop communication, the setting may cause large latency, that is the main disadvantage of duty cycling. However, in

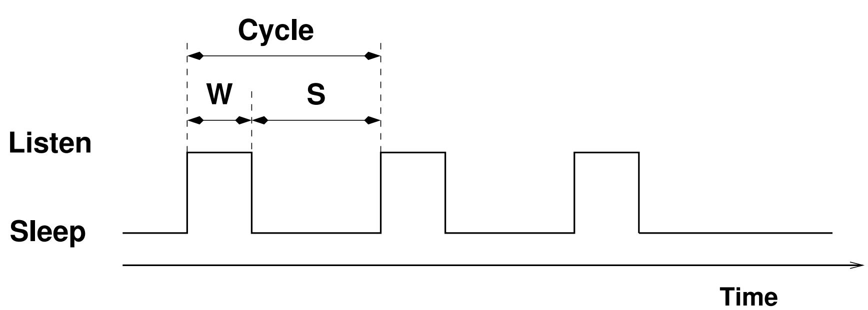

Figure 3.1: Duty Cycling concept

recent proposed MAC protocol the original duty cycling is modified, the values of W, S, or even the length of a cycle can be varied.

The formula $W/(W + S)$ generally defines a parameter named duty cycle. For example, when a duty cycle is at 10%, a node with that value of duty cycle turns its radio on only 10% of running time, resulting substantial energy saving. Therefore, the concept of duty cycle can be interchangeably used in the purpose of reflecting energy consumption.

3.1.2 Taxonomy

Designing MAC protocol is a very active research area in recent years, and it attracts the interests of many researchers. Consequently, a huge number of MAC protocols has been proposed, and there are many ways to classify them. There exist many detail surveys of MAC protocol in WSN, which are literately published every year [53–55]. The most popular method of classification is based on the channel access methods. The MAC protocols are generally divided in to two categories: scheduled access and random access [56]. In the former category, scheduled access protocols organize communications through a Time Division Multiple Access (TDMA) approach. These protocols aim to schedule communications in a manner which prevents collisions, overhearing and idle listening. They can be either distributed, or centralised. Examples for this category are LMAC [57], AI-LMAC [58], TRAMA [59], Crankshaft [60], IEEE 802.15.4 [61], TreeMAC [62] etc. A major drawback of TDMA-based MAC is its low channel utilization when only few nodes have data to send because a node can transmit only in its assigned time slots. In order to improve the channel utilization under low contention, TDMA-based MAC should combine with Carrier Sense Multiple Access (CSMA).

We follow the random access (or CSMA-based) approach in which the nodes access wireless channel through contention resolution mechanisms. *Carrier Sense* describes the fact that a sender node uses feedback from a receiver that detects a carrier wave before trying to send. This approach is recognized to be simple, practical, and suitable in WSNs. Roughly, the random access MACs can be further divided into two subcategories: asynchronous and

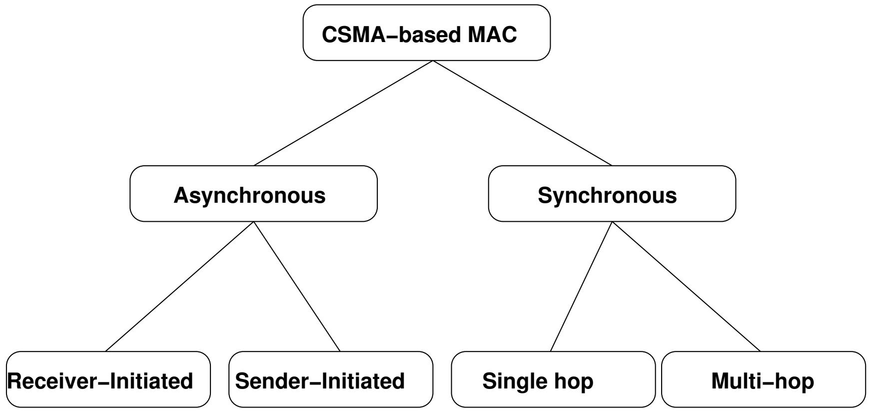

Figure 3.2: Classification of CSMA-based MAC protocols

synchronous protocols. Each subcategory includes smaller branches as shown in Fig. 2.2, i.e., receiver-initiated and sender-initiated in the asynchronous subcategory, and single hop and multi-hop in the synchronous one.

In the remainder of this section, we present a brief survey about several random access MAC protocols, which are closely related to our research.

Asynchronous Sender-Initiated MAC

The first proposed asynchronous MAC is Berkeley MAC (B-MAC) protocol [63], which contains a small core of media access functionality. B-MAC uses clear channel assessment (CCA) and packet backoffs for channel arbitration, link layer acknowledgments for reliability and low power listening (LPL) [64] for low power communication. B-MAC is designed as a link layer protocol, other networking services like organization, synchronization, and routing can be attached above the protocol's core. Moreover, a B-MAC's sender utilizes preamble signaling to initiate its potential receiver. All B-MAC's nodes are pre-installed duty cycling setting, then they periodically wake up at it beginning of their and start checking preamble signals. They keep their radios on when a preamble is sensed. Otherwise, they turn off the radios after a data packet arrives or after a time-out value. A cooperation between a sender and a receiver using B-MAC is shown in Fig. 2.3. We can observe that the sender needs to send a long preamble in order to notify the receiver about the next transmission of data

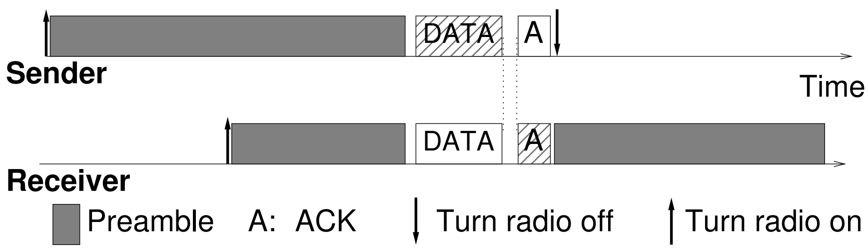

Figure 3.3: Operation of B-MAC with long preamble

packet.The preamble length should be at most about the length of two operational cycle. Consequently, both the sender and the receiver waste much energy during the transmission, and the transmission delay is long in multi-hop communication.

WiseMAC [65] is designed with a similar in B-MAC. The advanced feature in WiseMAC is the sender efficiently reduces the length of the wakeup preamble by knowing the sampling schedule of its neighboring nodes. Initially, the WiseMAC nodes' clock are not synchronized, however a node then synchronize with its neighbours that are potential receiver in order to learn their wakeup schedules. To efficiently enable this learning process, a receiver that successfully receives a data frame includes the remaining time until its next sampling time in the following acknowledgement frame. The sender uses this information, also takes possible clock drifts into account, it can estimate the receiver will wake up next. The process of preamble sampling, and data transiting will be implemented in a more efficient way. However, WiseMAC, as with B-MAC, still suffers from the possibility of simultaneous transmissions from hidden nodes, due to the similar preamble sampling techniques they use. In addition, WiseMAC senders must maintain the same regular wakeup schedule over time, it may cause repeated collisions between competing nodes that wake up at the same time.

X-MAC [66] solves the overhearing problem from hidden nodes in B-MAC and WiseMAC by using strobed preambles. In stead of sending a long preamble, an X-MAC sender transmits sequence of short preambles prior to data transmission, as illustrated in Fig. 2.4. The short preamble contains the target address, which not only helps irrelevant nodes to go to sleep but also allows the intended receiver immediately involve in the data transmission. The receive sends an early ACK frame to the sender, so that the sender stops preamble transmission

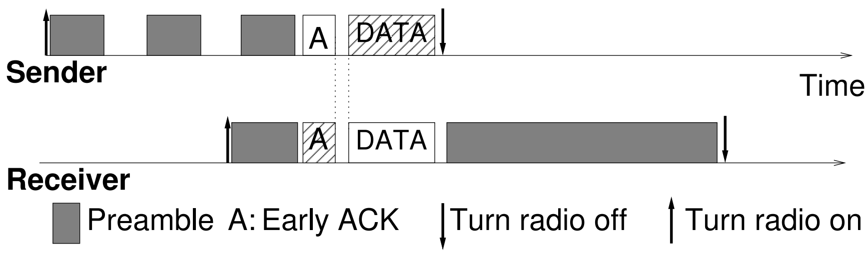

Figure 3.4: Operation of X-MAC, including strobed preamble and early acknowledgement

and start to transmit its data packet. After receiving a data packet, a receiver in X-MAC stays awake for duration equal to the maximum backoff window size to allow queued packets to be continuously transmitted. By doing so, X-MAC saves a huge amount energy by avoiding overhearing while reducing latency almost by half on average. Several protocols are developed based on X-MAC for examples X-MAC with Unified Power Management Architecture (UPMA) [67], Convergent MAC [68], Model-driven Concurrent MAC [69], etc. All of them shows a better performance than X-MAC, however the performance is still not as good as receiver-initiated MAC discussed in the next section.

Receiver-Initiated MAC

Receiver-Initiated MAC (RI-MAC) [70] is a canonical asynchronous MAC protocol, which setups transmission depending on receivers. The receiver-initiated idea is not new in the networking field, however RI-MAC shows its efficiency in WSNs by extensive evaluations including on real sensor motes. Opposite with previously mentioned protocols, a RI-MAC's sender holds a packet and idly listen to the channel. A receiver normally sends a beacon embedded its address, i.e., a ready to receive packet after a sleeping period. If the sender successfully receives the beacon, it will check the embedded address. If the intended receiver is recognized, the DATA/ACK (acknowledgement) exchange will be implemented. RI-MAC achieves a good latency performance since it helps to minimize the rendezvous time occupied the wireless channel between the sender and receiver. Moreover, RI-MAC effectively resolves the contention at receivers, hence RI-MAC can effectively reduce that overhead. Besides that, RI-MAC can use beacon with two roles ACK and ready to receive as shown in Fig.

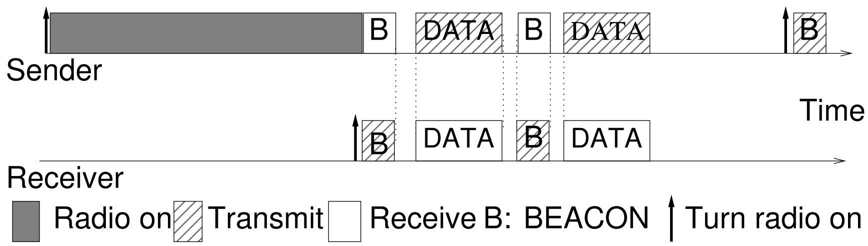

Figure 3.5: Operation of RI-MAC with dual roles beacon

2.5. Therefore, RI-MAC outperforms the previous mentioned protocols in all aspects.

RI-MAC, however, still has a disadvantage at senders. One sender with pending packet needs to keep its radio on until a potential receiver wakes up. There are several protocols address and avoid that problem. In PW-MAC (Predictive-Wakeup MAC) [71] senders predict receivers wakeup time, queue packets and wake up right before the predicted time. CyMAC [72] utilizes a prediction mechanism based on traffic conditions and a relative delay bound. The mechanism let nodes that involve in a communication find each other at the minimum rendezvous time.

All the asynchronous protocols share the same several characteristics. They are simple and it is easy to install them in current real sensor node. The main shortcoming of them is low latency because of they can not support multi-hop transmission in a cycle.The only method bypasses that problem is shortening the length of the cycle at the cost of increasing energy consumption.

Single hop Synchronous MAC

Initially, many proposed protocols focused on the energy efficiency at the expense of other parameters (e.g., latency, fairness). S-MAC [44] is based on the original duty cycle concept to reduce the overhead: collision, overhearing, control packet overhead, and idle listening. It divides time axis into fixed length cycle period, which is further divided into Sync, Data and Sleep periods (as shown in Fig. 2.6). In the Sync period, a node in S-MAC synchronizes its clock with its neighbours, the synchronization guarantees all nodes will wake up at the same time in the following cycle. S-MAC's data transmission is achieved in the Data

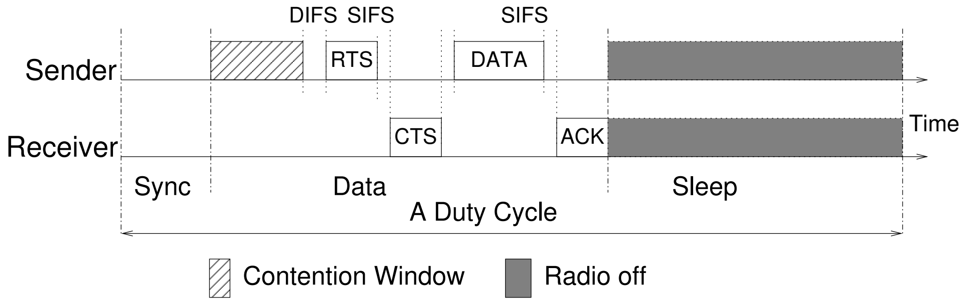

Figure 3.6: Operation of S-MAC: a duty cycle includes Sync, Data and Sleep periods

period, and it fully inherits from IEEE 802.11, i.e., Request-To-Send (RTS)/Clear-To-Send (CTS)/DATA/ACK handshakes. The nodes turn off their radio on the Sleep period to save energy. S-MAC with a periodic sleep/listen mechanism is more energy efficient than the full awake IEEE 802.11 MAC protocol, but it introduced a very large end-to-end latency since it supports only the single hop transmission.

There are also many works try to reduce the latency problem in S-MAC. Sensor MAC with adaptive listening [47] and T-MAC [73] are the typical examples, which are actually the enhanced versions of S-MAC. In S-MAC with adaptive listening, a node that overhears the control packet (e.g., RTS or CTS) of another node's transmission during the Data period will keep awake for a short period after the transmission completes. If this node is the next-hop along a multi-hop path, its neighbour can immediately forward the data packet to this node rather than waiting until the upcoming Data period. On the other hand, T-MAC reduces the latency by adaptively changing the ending time of a Data period when there is no traffic transmission near the node by using a timeout. The Data period is ended whenever both the physical and virtual carrier sensing finds the channel idle for the given duration of the timeout. Both T-MAC and S-MAC with adaptive listening can generally deliver a packet with at most two hops within an operational cycle. Another approach is D-MAC [74] which reduces latency only for data gathering in which multiple nodes try to send data to a sink node through a unidirectional tree of paths. However, D-MAC has the disadvantage that it makes specific assumptions on the communication pattern among nodes (tree-based). Those protocols require an effective synchronization, fortunately, recent

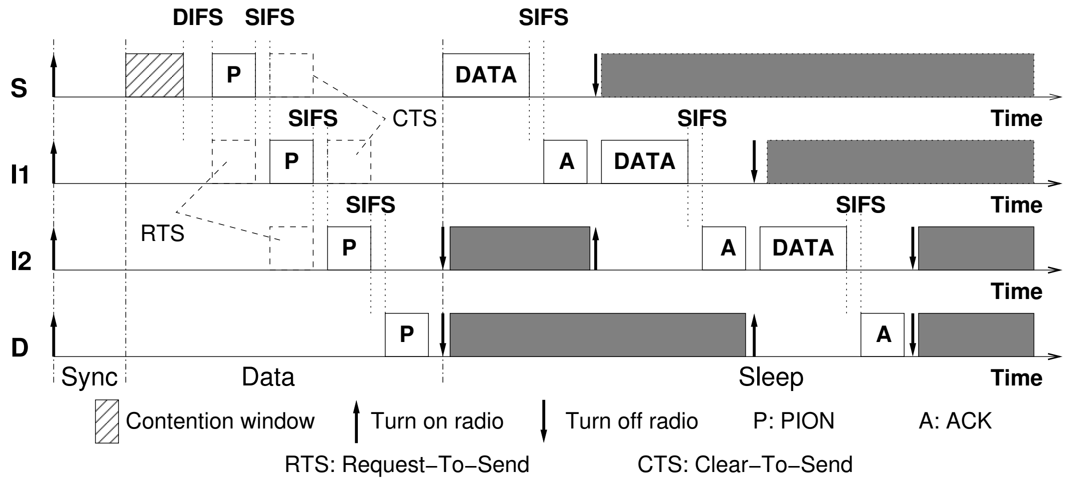

Figure 3.7: Operation of R-MAC

developments on synchronization protocols such as [75] provides promising practical solution for the synchronous MAC protocols.

Multi-hop MAC

Multi-hop MAC protocols are also motivated by the task of shortening the delivery latency by duty cycling. In contrast to each of the aforementioned protocols, multi-hop MAC protocols [76–78] support multi-hop transmission (more than two hops) in a single operational cycle. Like single-hop protocols they usually divide the cycle into the Sync, Data and Sleep periods as introduced by Sensor MAC. A synchronization protocol is employed in the sync period to align the start of the cycle among all nodes in the network. In the subsequent Data period multi-hop MAC protocols let nodes with pending packets compete for corresponding time slots within the subsequent Sleep period. Hence, the actual data is transmitted in the Sleep period.

In order to support multi-hop transmission, the multi-hop protocols exploit cross-layer information from routing layer. The information is then encapsulated in some additional fields in MAC layer's packets. Therefore, the nodes can recognize the ongoing transmissions and schedule their radio to take part in the transmission. Fig. 2.7 shows the operation of R-MAC [77], by embedding cross-layer information, a control packet, i.e., pioneer (PION) can be received as RTS, CTS at an upstream node, and a downstream node respectively. The protocol not only significantly reduces end-to-end delay in multiple hop relaying but also furthermore save energy wastage by control overhead.

3.2 Summary

In this chapter, we give a brief survey of MAC protocols for sensor networks. First, we introduce the duty cycling mechanism, which is widely adopt by almost all MAC protocols. Secondly, we introduce a classification of MAC protocols in wireless sensor networks with a detail discussion on the contention-based MAC protocols. Those contention based protocols are divided into two categories of asynchronous and synchronous. In each category, we have investigated the subcategories and the protocols that belong to each subcategory. We also mention the advantages and disadvantages of those state-of-the-art protocols.

Chapter 4

Minimizing Overhead in Multi-hop MAC Protocols

This chapter introduces an approach in designing MAC protocol for low data rate sensor networks, in which throughput is not a major requirement. The challenges or designing goals are achieving good performances on both energy efficiency and delivery latency. Our approach is based on the state-of-the-art one that combines duty cycling and multi-hop forwarding. However, the unique of our approach is exploiting natural characteristics of wireless channels to achieve those design goals.

4.1 Introduction

4.1.1 Wireless Channel

A channel model for wireless sensor node can be described in the Fig. 3.1. Tx_Range, CS_Range is denoted for Transmission Range and Carrier Sensing Range, respectively. When a wireless sensor S is transmitting, its radio wave covers a circle with radius radiated area, and the wave will affect surrounding wireless sensors. The effect depends on the distance from other sensors to S, denoted as d. Specifically, wireless sensor nodes around S can be partitioned into three classes:

- The nodes at a distance $d \leq Tx\_Range$ are able to correctly decode the signal and receive packet.
- The nodes at a distance d, where $Tx\_Range < d \leq CS\_Range$, are not able to correctly receive data from S. However, when S is transmitting they observer the channel busy, and often defer their transmission.

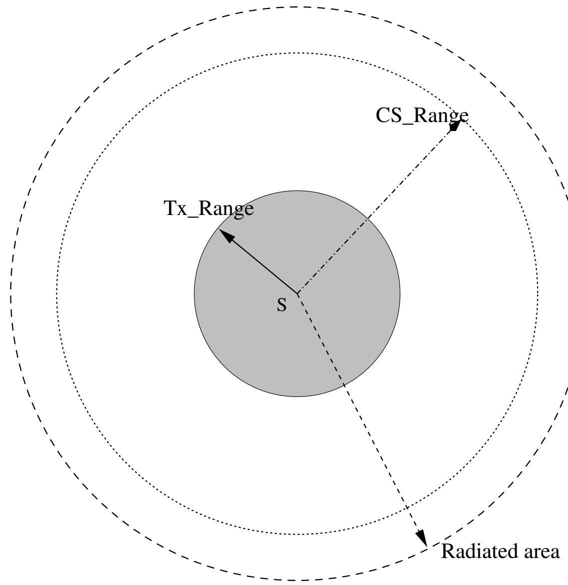

Figure 4.1: Wireless channel model

- The nodes at a distance d, $d > CS\_Range$, can not see any significant change of energy when S is transmitting. Therefore, they can start their transmission, but the quality of transmission may be affected by the energy radiated by S.

This type of channel model is first introduced in IEEE 802.11 [79], it then is adopted by many commercial wireless sensor products and IEEE 802.15.4 [61]. The channel model has strong effects on MAC protocols' performances. It is therefore necessary to understand the advantages and disadvantages of the model when designing MAC protocols.

4.1.2 Multi-hop MAC protocol

Energy efficiency is a requirement when designing a MAC protocol for wireless sensor networks. To reduce needless energy use, most MAC protocols exploit the duty cycling technique, in which the radio frequently turns on and off in each operational cycle. Among those protocols, the multi-hop MAC protocol, e.g., routing enhanced MAC (RMAC), enables multi-hop transmission to minimize the latency burden, which is the main disadvantage of duty cycling. In each cycle, when nodes are awake, the multi-hop protocol can exploit the cross-layer information to initialize the multi-hop flow and the data packet transmission is scheduled in the subsequent sleep period. This technique, however, introduces the long listening period problem in which nodes have to keep the radio on even when no flow data is scheduled. In addition, the protocol still incurs a large control packet overhead. In this chapter, we present LO-MAC (Low Overhead MAC) protocol, which exploits characteristics of the channel model and bypass those problems. LO-MAC solves the long listening problem by adding a short period after the synchronized process. In this period, we use carrier sensing as a binary signal, which lets the nodes know the traffic status of the network. After the period, nodes go to the sleep state when no data exists in the network; otherwise nodes involves in multi-hop data transmission. To minimize the control overhead, we force one packet to play more than one role not only in the initial transmission period but also in the data transmission period.

4.2 Protocol Description

4.2.1 Overview

LO-MAC is a duty cycling contention-based MAC protocol, which employs Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) for accessing channel task. In LO-MAC, the packet structure, Short Inter-Frame Space (SIFS), and Distributed Inter-Frame Space (DIFS) are inherited from IEEE 802.11 [79]. The protocol supports multi-hop transmissions in an operational cycle, which is divided into four periods: Sync, Carrier Sensing, Data, and Sleep. LO-MAC's nodes wake up together at the beginning of the Sync period, during which they exchange SYNC packets to synchronize their local clocks. In the Carrier Sensing period, an adaptive mechanism, which exploits the carrier sensing technique, is introduced. The mechanism notifies the nodes of the existence of traffic in the network, and lets them follow either a busy or idle cycle. In the idle cycle the nodes turn off radios to save energy after the Carrier Sensing period. Otherwise, in the Data period the nodes exchange cross-layer control packets to schedule multi-hop data transmissions in the subsequent Sleep period.

4.2.2 Lightweight Traffic Adaptive Mechanism

The Data period is a key parameter of multi-hop MAC since it is necessary to initialize the multi-hop transmission in oder to achieve a balance between energy efficiency and low delivery latency. However in low data rate environments, it is possible that most cycles the nodes have no packet to be transmitted. In that case, nodes needlessly waste energy during the Data period. The multi-hop MAC is more efficient if it can let the nodes go to sleep when there is no data in the network, and otherwise keep the implementation of multi-hop flow transmission. To accomplish that issue, we add a short period named Carrier Sensing (length T_{cs}) right before the Data period. During the Carrier Sensing period, the nodes without pending packets utilize clear channel assessment (CCA) to determine whether the channel is busy or idle. On the other hand, the node with a pending data and the node sensing channel busy immediately broadcast busy packets. Thus, all nodes that carrier sensed the channel as busy or transmitted a busy packet will recognize the existence of data packet and remain on during the Data period. Rather than every node waking up for the whole Data period, a node only wake up for T_{cs} when there are no packets to be transmitted from its neighborhood nodes.

The benefit of the Carrier Sensing period is that the total length of the carrier sensing process, even in multiple hops, is negligible if compared with the length of the Data period. This is possible when the CCA time for compliant radio is reported less than 15 μs in a typical sensor mote [80] as well as in the specification of IEEE 802.11 [79]. In our evaluation, we used a much larger T_{cs} which is designed to support a very large number of hops in the multi-hop flow. Additionally, busy tones do not contain any information that needs to be decoded. The only function of busy tones is to enable other nodes to detect the channel as busy. The advantage of not having information in a busy tone is that multiple nodes can transmit simultaneously, causing collisions at the receivers without hindering the protocol. If a collision occurs at a receiving node, the node can still detect the channel as busy and remain on for the Data period. We set the maximum number of busy tones a node can send

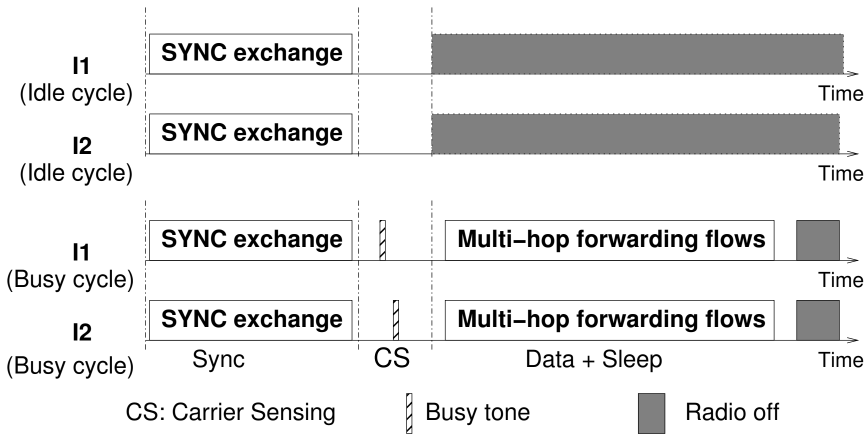

Figure 4.2: Adaptive Scheme in Carrier Sensing Period

in the Carrier Sensing period to one. The nodes those are sending the busy packets cannot sense the channel, but this case still works in LO-MAC because the nodes already know the channel as busy; therefore, they remain awake in the subsequent Data period.

Figure 3.2 illustrates two types of operational cycles in LO-MAC. Two nodes I1 and I2 exchange SYNC packets during the Sync period. If these two nodes sense the channel as idle during the Carrier Sensing period, they follow an idle cycle. Otherwise, they follow a busy cycle and take part in a multi-hop forwarding flow in the upcoming Sleep and Data periods.

The adaptive mechanism was first introduce in our previous work [78]. We applied to RMAC and introduced a new protocol RMAC-CS (RMAC with Carrier Sending method). The results show that RMAC-CS outperforms RMAC in term of energy efficiency but slightly achieve a longer latency.

4.2.3 Multi-hop Data Transmission in a Busy Cycle

In wireless networks, an inherent characteristic is the broadcast nature, meaning that an active node usually "hears" a packet when it is within transmission range of nodes participating in a transmission. Depending on the content of the packet and the process of handling the received packet at the node, the broadcast nature may be advantageous or become a source of overhearing overhead (i.e., the node has to receive useless packets). We use this

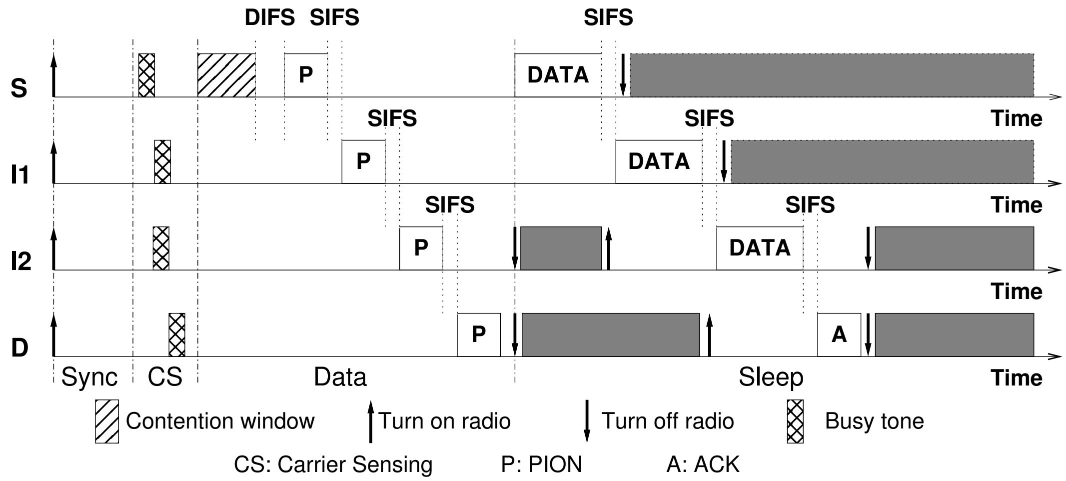

Figure 4.3: LO-MAC in a busy cycle

characteristic to enable multi-hop transmission and conserve energy in designing LO-MAC.

Figure 3.3 gives an overview of multi-hop transmission in a LO-MAC's busy cycle. The scenario includes four nodes: source S, intermediate nodes (I1 and I2), and destination D. All nodes wake up together at the beginning of the Sync period. They keep their radios on at least until the beginning of the Data period since the traffic-adaptive mechanism allows them to detect the existence of pending data at S. During the Data period, a multi-hop transmission is initiated as follows. S starts to transmit the first cross-layer control packet to I1, and I1 stores the cross-layer information then modifies and relays the control packet to I2. The process is repeated at I2, and the control packet reaches D. The cross-layer information is used to schedule the wake up time in the subsequent Sleep period. The nodes are woken up at the scheduled time and implement the multi-hop data packet transmission similar to relaying control packet.

Initializing multi-hop transmission

The initialization of multi-hop transmission is implemented in the Data period by exchanging the PION packets. In LO-MAC, the construction of a PION packet and the exchange process are inherited from RMAC [77]. The PION packet is constructed by adding cross-layer fields to the original IEEE 802.11 RTS packet. The additional fields, which comes

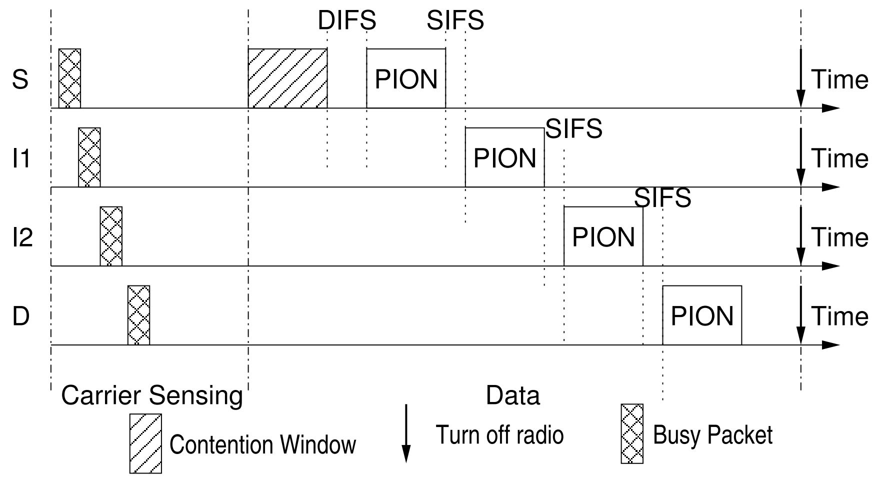

Figure 4.4: PION exchange in LO-MAC

from routing layer, are hop count (the number of hops the PION has traveled) and the final destination. Hence, the new packet PION can achieve both a RTS/CTS role as well as provide a scheduling function.

PION is initiated by the source node and relayed by the intermediate node during the Data period. During its entire life cycle, a PION packet keeps an RTS nature regarding a downstream node and a CTS nature regarding an upstream node. We describe a PION exchange progress for a simple 4-node scenario in Fig. 3.4. The source node S has a data packet, and starts its PION after a contention window (CW). Intermediate node I1 receives and relays the PION to its downstream node I2. The PION from I1 serves as both a CTS to S and an RTS to I2. However, in contrast to the traditional RTS/CTS, after sending a PION packet, the node has to wait at least until the subsequent Sleep period to implement the actual data transmission. Upon receiving I1's PION, I2 performs the same steps as I1. This process of receiving a PION and immediately transmitting another PION continues until either the final destination has received the PION or the end of the current Data period is reached.

As mentioned, the PION packet also provides scheduling information for all nodes in its relay path. The hop count field of a PION packet is used to schedule the wake-up time of nodes in the Sleep period. Unlike RMAC, the LO-MAC scheduling function works as follows:

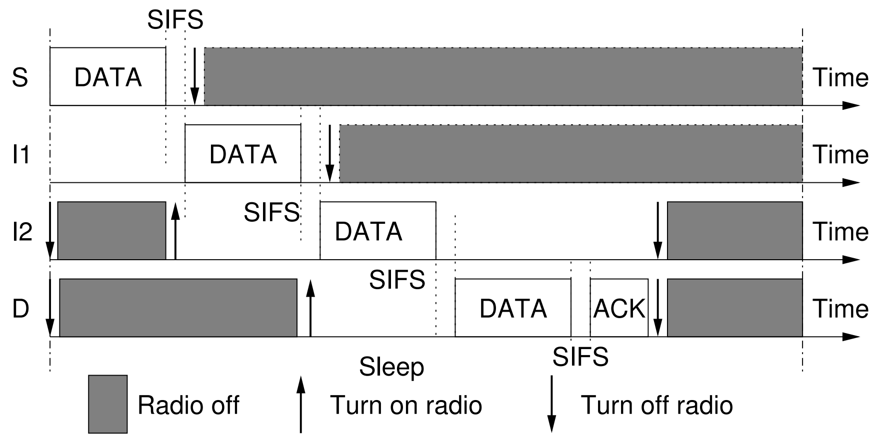

Figure 4.5: Data transmission in LO-MAC

Suppose a node is the i^{th} hop during the PION transmission. We denote its wake-up time in the Sleep period as $T_{wakeup}(i)$. $T_{wakeup}(i)$ is the subsequent time difference from the start of the Sleep period, and is calculated as

$$T_{wakeup}(i) = (i-1)(l_{DATA} + l_{SIFS}) \tag{4.1}$$

Here, l_{DATA} is the time to send a single data packet and l_{SIFS} is the length of SIFS period, respectively. To simplify, LO-MAC assumes all data packets in the sensor network are the same size, so l_{DATA} could be a preset value. Otherwise, the l_{DATA} information can be included in the PION packet, so every node can calculate the correct wake-up time.

In a single cycle, the maximum hop count value also indicates the number of hops over which PION was relayed. This value depends on the length of the Data period, which is primarily determined by the duty cycle if the length of T_{cycle} is fixed. To gain the full benefit of LO-MAC, we need to find the best duty cycle length for the network.

Multi-hop Data Transmission

Similar to RMAC and other multi-hop MAC protocols, LO-MAC's multi-hop data transmission is also implemented in the Sleep period. A source node immediately generates a data packet for a downstream node at the beginning of the Sleep period, and it stays awake at least the SIFS plus a small period to receive an acknowledgment signal from its neighbor. The data packet needs the SIFS for packet processing in each node, and after that it is relayed to a downstream node in the same way as for the PION packet. When the data packet is relayed by each intermediate node, it also plays an ACK function for the upstream node.

To fulfill the ACK function, after transmitting a data packet, the node stays in the awake state for a short period to listen to the channel and verify the next hop transmission when the channel is sensed busy. If the channel is idle that mean the node receives no ACK signal, retransmission of the data packet is requested. When the data packet reaches the destination node, after SIFS the destination node sends an ACK to complete the transmission. Nodes go to sleep when they snoop the ACK signal or receive ACK packets. In the mentioned scenario, since all nodes know their T_{wakeup} after the Data period, as soon as the Sleep period starts nodes S and I1 immediately start their data packet sending/receiving. Other nodes in a multi-hop path that has successful PION transmission in the Data period go to sleep to save energy. Each node later wakes up at the ordered time to receive the data packet from the upstream node and relays the packet to the downstream node. For example, node I2 can go to sleep when the Sleep period begins, but it wakes up at the scheduled time when I1 is ready to forward the data packet to I2. This process is described in Fig. 3.5.

In the LO-MAC protocol, we also keep the Network Allocation Vector and Frame Loss Handling as in RMAC.

4.3 Performance Evaluation

We used the network simulator ns-2 [81] to evaluate LO-MAC 's performance. Each sensor node had a single omni-directional antenna through which the combined free space and two-ray ground reflection radio propagation models were employed. Key networking parameters are shown in Table 3.1, where power consumption parameters were set to typical values for a Mica2 radio (CC1000) [82]. The 250m transmission range and the 550m carrier sensing

Table 4.1: Networking parameters

Bandwidth	20 Kbps	TxRange	250 m
Rx Power	0.5 W	Carrier Sensing Range	550 m
Sleep Power	0.05 W	Contention Window (CW)	64 ms
TxPower	0.5 W	DIFS	10ms
Idle Power	0.45 W	SIFS	5 ms

Table 4.2: Packet parameters

	Packet Size(bytes)	Transmission time(ms)
SYNC/RTS/ACK	10	11
DATA	50	43.0
PION	14	14.2

Table 4.3: Time duration parameters

	T_{cycle}	T_{sync}	T_{data}	T_{sleep}
dc=5%	4465 ms	55.2 ms	168 ms	4241.8 ms
dc=10%	4465 ms	55.2 ms	391.3 ms	4018.5 ms

range were modeled after the 914-MHz Lucent WaveLAN DSSS radio interface; although not typical for a sensor node, we used these parameters to make our results comparable to RMAC. In our evaluation of power efficiency, we focused on the energy consumed by radios, but ignored the energy consumed by other components such as CPU and memory [83]. The transmission time and size of packets are listed in Table 3.2. All the duration parameters of an operational cycle of RMAC are presented in Table 3.3. We denoted the length of a cycle and the durations for Sync, Data, Sleep periods as T_{cycle}, and T_{sync}, T_{data}, T_{sleep} respectively. These durations were calculated with two different duty cycles at 5 and 10%. Note that the duty cycle parameter dc is calculated as follows: $dc = (T_{sync} + T_{data})/T_{cycle}$, whereas T_{cycle} is fixed at 4465 ms. In the evaluation of LO-MAC, we used the same dc-related and duration-related parameters as in the RMAC's, except the T_{sleep}. Since we added a 5-ms Carrier Sensing period in LO-MAC, the length of Sleep period is shortened an amount of T_{cs}.

To simplify our evaluations, we ensured that networks we used were connected networks. In addition, we did not include routing traffic in the simulations. We also assumed that there is a routing protocol deployed to provide the shortest path between any two nodes. We simulated two scenarios: a multi-hop chain and a network scenario. In the chain scenario,

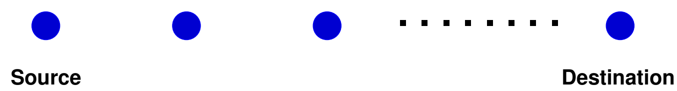

Figure 4.6: Chain scenario

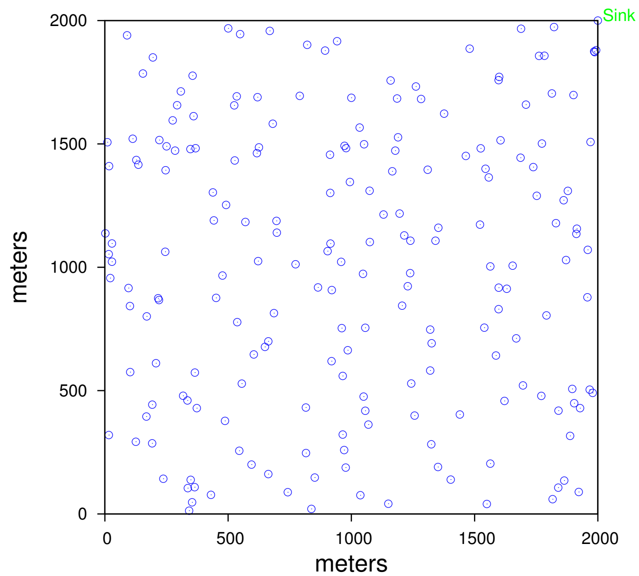

Figure 4.7: Network scenario contains 200 nodes in 2000 × 2000 m area

the nodes were arranged in a straight line and their neighbors were placed 200 meters apart, as shown in Fig. 3.6. In the network scenario, 200 sensor nodes were uniformly in a random pattern within a 2000 meter square area. The sink node was located in the upper right corner of the area, as shown in Fig. 3.7.

4.3.1 Multi-hop chain scenario

We first evaluated LO-MAC in an 11-node chain scenario. In this evaluation, we used a single flow to send packets at a constant rate from node 0 at the beginning of the chain to the destination node 10, which is farthest node from the source. We varied the interval between two consecutive packets from 10 to 60 seconds. Each simulation lasted for a total of 3600 seconds, and the duty cycle was kept to 5% in all nodes. We compared the performance between LO-MAC and RMAC.

Figure 3.8 shows the average energy consumption over all the sensors in the chain. The

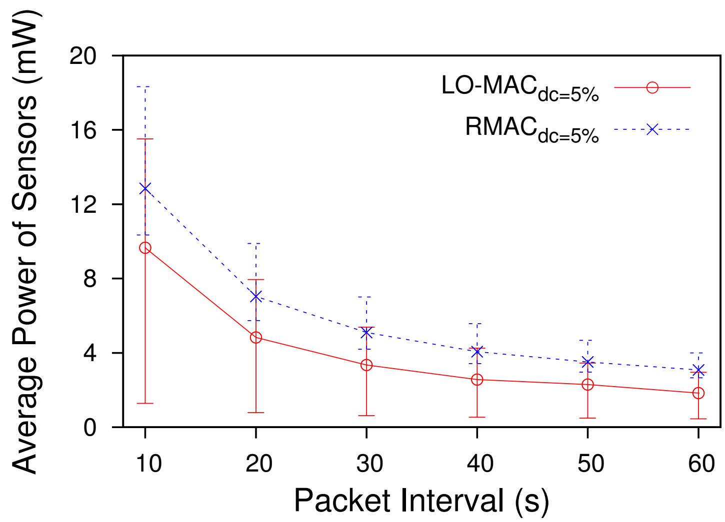

Figure 4.8: Average energy consumption in chain scenario

average energy consumption was calculated by dividing the total energy consumed in the simulation by the total number of sensors. The error bars show the minimum and maximum values for a single sensor's energy consumption. When the traffic load increased, i.e., the packet interval decreased, the nodes in RMAC and LO-MAC increased their energy consumption, but the LO-MAC's nodes consumed less energy than RMAC's. Specifically, when the packet interval was 60 seconds, the average energy consumption in LO-MAC was approximately 75.1% of that of in RMAC, and this value approximated to 59.4% when the packet interval was 10 seconds. We can conclude that LO-MAC outperformed RMAC in terms of energy efficiency. There were two reasons for this. The first one was LO-MAC had the adaptive mechanism, the nodes save power by turning the radio off when no data packet existed in the idle cycles. The second was during the busy cycles in LO-MAC, the nodes transmitted fewer packets than in RMAC. For example, for an N-hop transmission in a cycle, the RMAC's nodes transmitted $(N - 1)$ more ACK packets than the LO-MAC nodes. Moreover, the same amount of energy was consumed for receiving the ACK packets.

Figure 3.9 shows the average delivery latency in the multi-hop chain scenario. The error bar shows the minimum and maximum values of delivery latency. Using an additional period (T_{cs}), the starting point of data transmission (i.e., the beginning of Sleep period) in LO-MAC

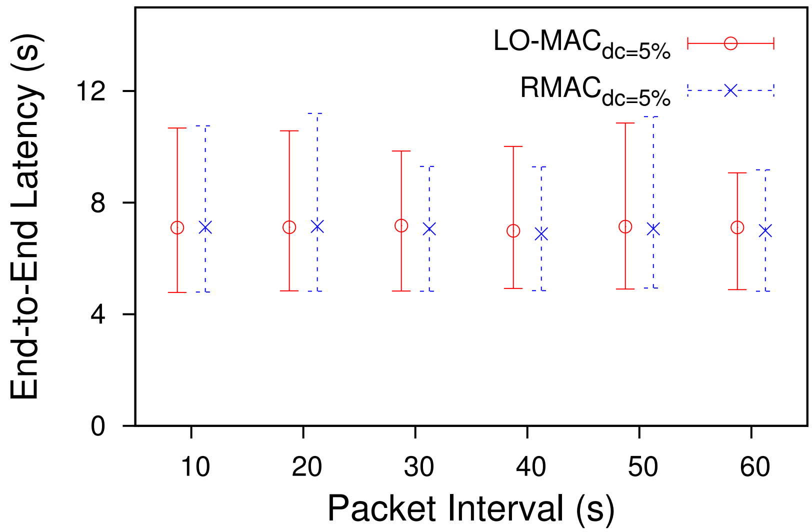

Figure 4.9: Average latency in chain scenario

was later than in RMAC. However in LO-MAC a node finished the data transmission sooner than in RMAC, since LO-MAC's node did not send the ACK packets as mentioned above, that means the node just needed a SIFS to start its data relaying. Moreover, the transmission time of an ACK was even longer than T_{cs}, so the delivery latency in LO-MAC was shorter than in RMAC. However the difference between those two values was negligible if comparing with T_{sleep}. As shown in Fig. 3.9 in this scenario, almost he packets needed more than one cycle to reach the destination, hence they were incurred the sleep latency. In addition, the random process of selecting time slot in the contention window also affected the latency. The sooner the time slot was selected, the better value of latency was achieved as proven in our previous work [84]. Then we conclude LO-MAC and RMAC achieve comparable delivery latency in the multi-hop chain scenario.

4.3.2 Random network scenario

In the evaluation of the network scenario, we adopted the same traffic generation method that was introduced in the original RMAC's paper. The traffic load was generated as follows. At a periodic interval of 50 seconds, a sensor node was randomly selected to send one data

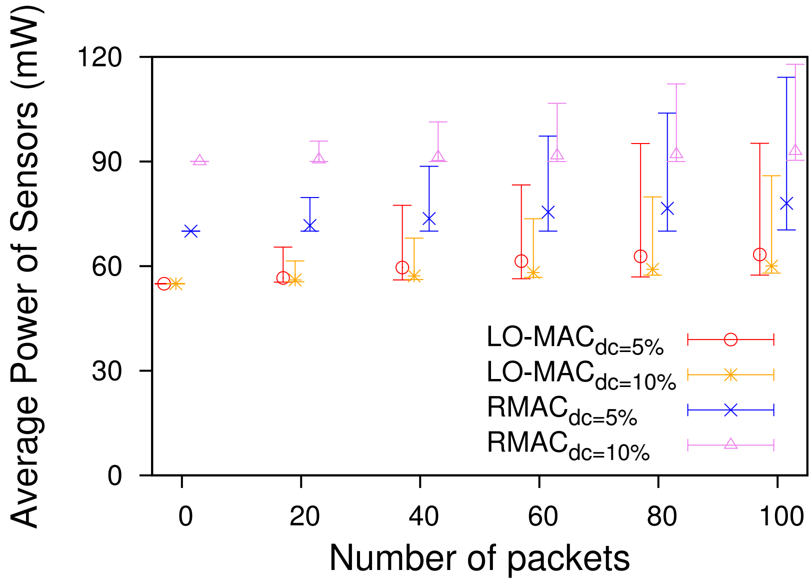

Figure 4.10: Average energy consumption in network scenario

packet to the sink at the top-right corner. If a node was selected to send a packet, it was taken out of the selection pool. The selecting order was similar in both LO-MAC and RMAC's evaluations. The number of packets was varied from 0 to 100; and the total time of each simulation was 5300 seconds. In this evaluation, we used two values of the duty cycle 5% and 10% to investigate the effect of duty cycle to the performance. We denoted LO-MAC$_{dc=5\%}$, RMAC$_{dc=5\%}$ and LO-MAC$_{dc=10\%}$, RMAC$_{dc=10\%}$ as LO-MAC and RMAC in the cases of 5% and 10% duty cycle respectively. The evaluation results for the random network scenario are shown in Fig. 3.10 and Fig. 3.11.

In Fig. 3.10, the middle point is the average value of energy consumption. The average energy consumption value was calculated by dividing the total energy consumption by the total number of nodes. The error bars express the maximum and minimum values of a node's energy consumption during the simulation time. When there was no traffic in the network, the nodes in LO-MAC$_{dc=5\%}$ and LO-MAC$_{dc=10\%}$ consumed the same amount of energy, but less than those in RMAC. That shows the maximum effect of the adaptive method in terms of energy saving. In this case, RMAC$_{dc=10\%}$'s nodes consumed more energy than RMAC$_{dc=5\%}$'s, since T_{data} in RMAC$_{dc=10\%}$ is larger than the one in RMAC$_{dc=5\%}$. When

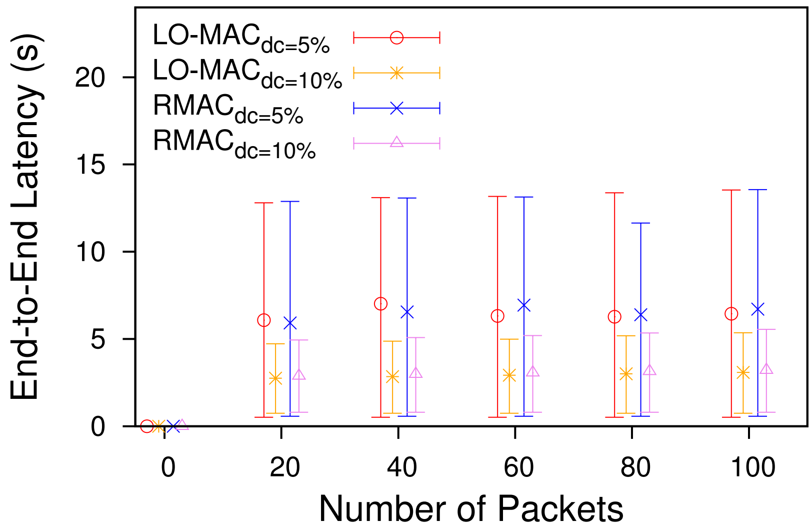

Figure 4.11: Average latency in network scenario

the traffic load increased, the energy consumption in all four scenarios increased. Among them, LO-MAC$_{dc=10\%}$ achieved the best performance in energy saving. Moreover the LO-MAC's protocols have better performance comparing with those of RMAC regardless of duty cycle value. That was because 50 seconds was long enough for a packet to be successfully received at the sink, in most cases only one data flow was transmitted in the network. If there was no packet in the network, sensor nodes still consume energy because they have to exchange synchronized information during the Sync period and "listen" to the channel during the Carrier Sensing period. Another interesting observation from Fig. 3.10 is that with the higher value of duty cycle, the RMAC's nodes consumed more energy but the LO-MAC's nodes did less. That shows the dominant benefit of multi-hop MAC, the longer the Data period is, the more hops a packet can travel in a cycle. Therefore, a packet may need fewer cycles to reach the destination, then the number of idle cycle was increased or the more energy was saved.

Figure 3.11 shows the average value of delivery latency, and the error bar shows the maximum and minimum values. We had the same conclusion as in the chain scenario in the case of 5%-duty cycle, since the difference between the average latency in RMAC$_{dc=5\%}$

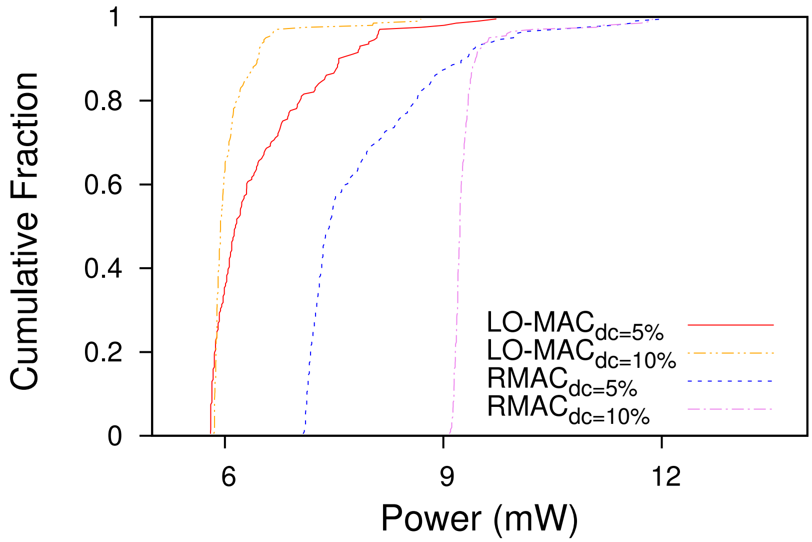

Figure 4.12: CDF of energy consumption in network scenario

and LO-MAC$_{dc=5\%}$ is negligible. However, the latency in LO-MAC$_{dc=10\%}$ is slightly shorter than that in RMAC$_{dc=10\%}$. The reason was that all of the packets were transmitted from the random nodes to the sink in one operational cycle.

To furthermore investigate the performance of the protocols, we simulated in the same network scenario with the same traffic model, but the total number of generated packet was 200. The total simulation time was 10300 seconds. We measured the energy consumption of each node and tracked the delivery latency of all packets. The cumulative distribution function (CDF) of the energy consumption and the delivery latency was shown in Fig. 3.12 and Fig. 3.13 respectively. The results in Fig. 3.12 indicate that even with 5% duty cycle, LO-MAC outperformed the two cases of RMAC, and LO-MAC$_{dc=10\%}$ achieved the best in terms of energy efficiency. Moreover, Fig. 3.13 shows that in LO-MAC$_{dc=10\%}$ the delivery latency was comparable with the one in RMAC$_{dc=10\%}$ but outperformed the others.

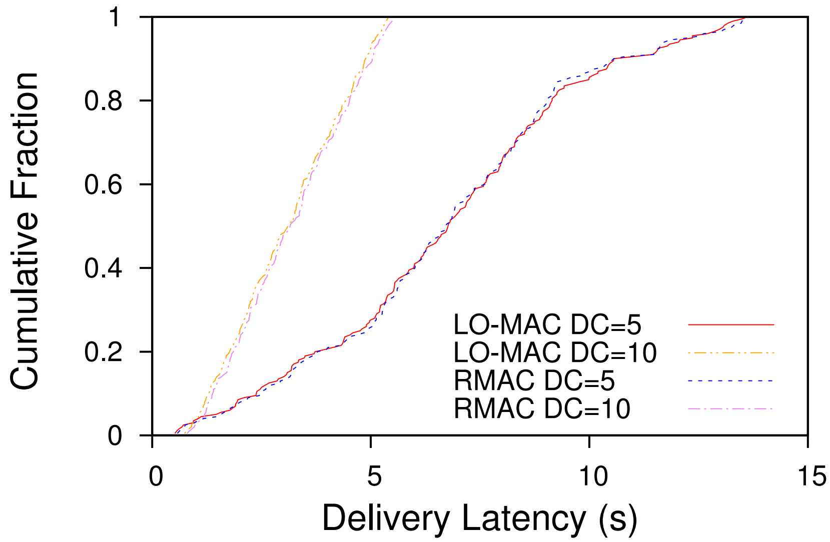

Figure 4.13: CDF of latency in network scenario

4.4 Summary

We proposed LO-MAC, which is an energy efficient, multi-hop MAC protocol for low data rate sensor networks. LO-MAC exploits the characteristics of wireless communication to achieve energy efficiency and low delivery latency. The traffic-adaptive mechanism based on carrier sensing effectively controls the length of the Active period in a cycle; hence, preventing the long Listening period problem. Moreover, LO-MAC relays a packet via multiple hops to reduce end-to-end latency. During the relaying path, a packet from one node often plays two roles to its upstream and downstream neighbors by exploiting the broadcast nature of wireless communication. By doing so, the number of transmissions is significantly reduced; hence, the protocol can effectively prevent overhearing and control overhead. Our simulation results showed that LO-MAC outperformed RMAC in terms of energy efficiency, and achieved comparable delivery latency.

Chapter 5

Improving Efficiency of MAC Protocols using Packet Concatenation

Wireless sensor networks may have dynamic traffic loads since the sensor nodes are distributed and sensing events unpredictably happen. In previous chapter, the multi-hop MAC protocol achieves a good trade-off between energy efficiency and latency in light traffic load environments. However, the original multi-hop MAC degrades its performance under high traffic loads. In this chapter, we present an extension of the multi-hop MAC protocol, which handles well in dynamic load environments.

5.1 Introduction

The duty cycling mechanism achieves energy efficiency at the cost of degrading latency and throughput performance. Meanwhile, an increasing number of prospective applications not only impose requirements on energy efficiency but also on other characteristics such as delay and throughput [85, 86]. Therefore, many improvements to the initial mechanism have been proposed. Among them the idea of forwarding a packet through several hops within one duty cycle, as initially shown in RMAC [77] (Routing enhanced MAC), has proven to be particularly efficient in achieving those goals. The proposed protocols use a single control packet instead of the common RTS/CTS (Request-to-Send/Clear-to-Send) pair to setup a flow across multiple hops. Since next-hop information is required when setting up the flow this is essentially a cross-layer optimization.

The state-of-the art multi-hop protocol named Demand Wakeup MAC (DW-MAC) [76] can support dynamic traffic load environments. It divides the cycle into the sync, data and sleep periods as introduced by Sensor MAC [44]. However, in DW-MAC the sleep period is employed for data transmissions. DW-MAC uses a one-to-one proportional scheduling function to determine the start of data packet transmissions based on the start of the corresponding control packet in the data period. Therefore, nodes in DW-MAC can be involved in multiple traffic flows without causing data-data collisions in the sleep period. However, DW-MAC still incurs idle listening overhead in the data period, allows for data-ack collisions and saturates very fast when the traffic load increases.

In this chapter we propose a new multi-hop MAC protocol, named MAC^2, which joins several techniques to overcome the mentioned disadvantages of DW-MAC. MAC^2's advanced characteristics are:

- MAC^2 utilizes an adaptive scheme that can adjust the length of the listening period in a cycle according to the traffic load.
- MAC^2 inherits the demand wakeup manner from DW-MAC to transmit data, but it optimizes the scheduling function to achieve minimum latency and guarantee collision freeness in the sleep period.
- MAC^2 employs a packet concatenation scheme which combines several packets into a bigger one to reduce control overhead.

5.2 Protocol Description

MAC^2 is a synchronous contention based protocol. The protocol employs Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to access the channel. The packet structure and the concepts of Distributed Inter-Frame Space and Short Inter-Frame Space are taken from IEEE 802.11, but otherwise MAC^2 does not depend on a particular standard. Similar to other duty cycling protocols, the operational cycle of MAC^2 contains three periods: Sync, Data, and Sleep with their lengths denoted as T_{Sync}, T_{Data}, and T_{Sleep}. In the Sync period, MAC^2 adopts an adaptive mechanism which can adapt to the traffic load and lets nodes follow either a busy or an idle cycle. In the idle cycle nodes sleep to save

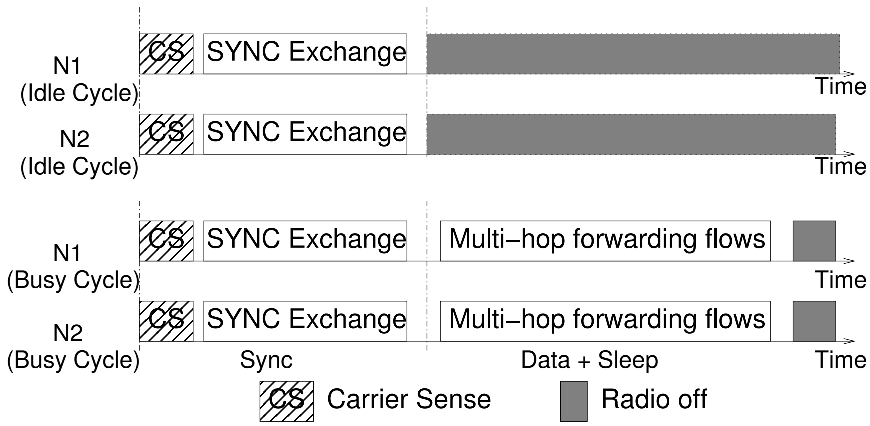

Figure 5.1: SYNC exchange in busy and idle cycle

energy after the sync period. In the busy cycle on the other hand they keep their radio on and exchange cross-layer control packets to reserve time slots for data transmissions in the subsequent Sleep period. MAC^2 inherits Demand Wakeup MAC's (DW-MAC's) [76] on-demand manner in which nodes are woken up during the Sleep period in order to transmit or receive a data packet.

5.2.1 Adaptive Scheme in Sync Period

MAC^2 uses the synchronization protocol proposed in S-MAC just like RMAC and DW-MAC. All nodes use the exchange of SYNC packets to choose and maintain the sleep/awake schedule. In addition, MAC^2 employs an adaptive method in which nodes can adjust their listening periods themselves according to the network traffic. When there is no data in the network, nodes follow an idle cycle with a short listening period (T_{Sync}), otherwise they follow a busy cycle with a long listening period ($T_{Sync} + T_{Data}$). We use the first bit of the SYNC packet to convey this information. When the bit is set, the new SYNC packet, called signaling SYNC, tells all the listeners to extend their listening periods. Thus, if a node has data to transmit, it constructs a signaling SYNC first and broadcasts it to its neighbors. When a node receives a signaling SYNC from one of its neighbors, it sets its clock to the long listening duration. After that, if the remainder of the Sync period is long enough to transmit another SYNC packet, the node will broadcast a new signaling SYNC to its neighbors. We assume that the length of the Sync period is long enough to rebroadcast the SYNC to the entire network, and each node sends only one SYNC packet. We also assume that signaling SYNC packets always win the channel over a normal SYNC (i.e. the first bit is not set). The assumption can be achieved by setting a smaller carrier sense duration for signaling SYNC transmissions. However, if the signaling SYNC does not win the channel, the protocol still works since it repeats the same process in the subsequent cycle.

Figure 4.1 illustrates the operation of MAC^2 in both an idle and a busy cycle. At the beginning of each Sync period, N1 and N2 implement carrier sense. If there is no traffic in the network, N1 and N2 go to sleep at the end of the Sync period; they follow the idle cycle. If node N1 has data to transmit, it sends a signaling SYNC packet to its neighbors including N2. The nodes follow the busy cycle.

This adaptive method can be applied to other synchronous protocols, e.g. RMAC, as we have done in a previous work [87]. The results show that by using an adaptive method RMAC consumes less energy while achieving comparable latency.

5.2.2 Data Forwarding in a Busy Cycle

In this subsection, we introduce the concatenation scheme which concatenates several packets in the queue before scheduling the transmission, and MAC^2's operation in a busy cycle.

5.2.3 Concatenation Scheme

In all duty cycling protocols nodes often have to queue pending packets for an extended amount of time since the sleep period is typically much longer than the active period. This is especially true since although the radio is switched off periodically the sensors are for most sensor network applications kept on at all times. The resulting frequent packet bursts suggest packet concatenation as a means of both improving energy efficiency and latency. Two properties of WSNs make concatenation especially attractive. Firstly, all packets are predominantly addressed to one and the same sink, thereby fulfilling the major requirement of packet concatenation. Secondly, data packets are usually small and therefore the gains

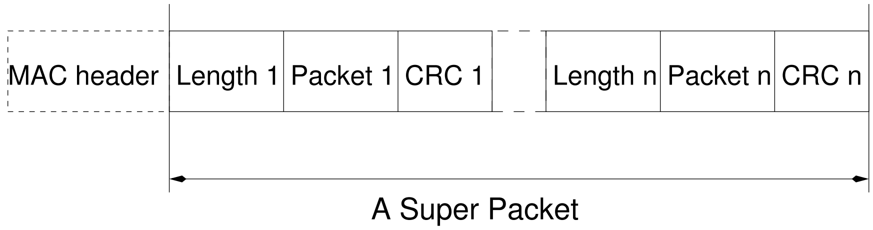

Figure 5.2: Super packet structure

from reducing control packet overhead are comparatively large.

Similar to DW-MAC and S-MAC, the data packet usually contains three parts: payload, length field and CRC field. When the MAC-layer processes the data packet, it attaches the MAC header. As we mentioned before all MAC headers share the same destination address in single sink networks since all data is directed towards the sink. In our scheme, we concatenate several packets into a super packet before adding the MAC-header to it. We denote the length of the resulting super packet as l_{SP}, which is always less than or equal to the concatenation threshold l_{TH}.

Figure 4.2 shows the structure of a super packet. Super packets are constructed from one up to n smaller packets and contain, for each encapsulated packet, its payload plus the length and CRC fields. Although these fields could theoretically serve other purposes as well we only use the length field to reconstruct the original packets from a super packet at the receiver. In case a super packet can not be transmitted because the node does not win the channel, its packets are requeued for later transmission.

Multi-Hop Forwarding in a Busy Cycle

In a busy cycle, nodes keep their radio on after the Sync period. At the start of the Data period nodes with pending data construct a super packet (SP) from one or more packets in their queue and set up the multi-hop flow by sending a control packet. The control packet, named scheduling packet (SCH) as in DW-MAC, is constructed by adding cross-layer fields (next-hop, destination, and number of hops) from the routing layer to the IEEE 802.11 RTS [79] packet. If the multi-hop flow is successfully initialized, node i in the flow calculates

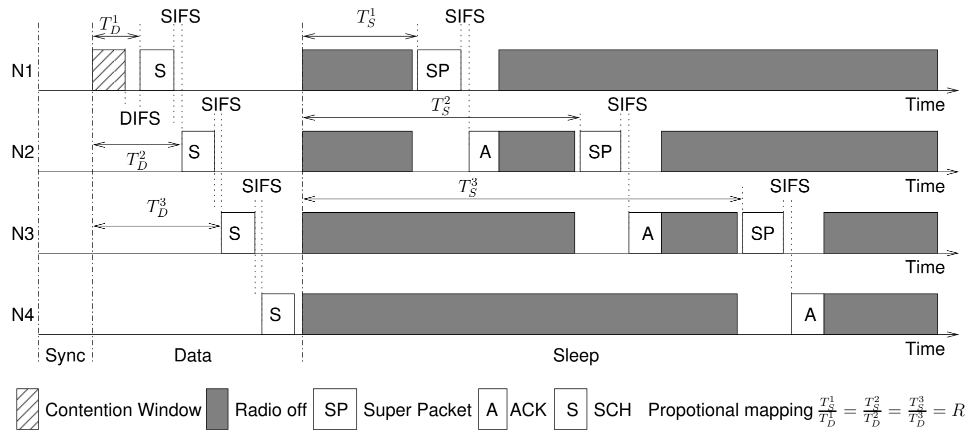

Figure 5.3: MAC^2 in a four-node scenario

its wakeup time in the next Sleep period following the proportional mapping function:

$$\frac{T_S^i}{T_D^i} = R \tag{5.1}$$

where T_D^i is the duration from the beginning of the Data period to the starting moment of the SCH transmission, T_S^i is the duration from the beginning of the Sleep period to the starting moment of the super packet transmission, and R is the mapping function value. In DW-MAC the value of R is $R_{org} = \frac{T_{Sleep}}{T_{Data}}$, but in MAC^2 we use the R_{min} value that we are going to introduce in the next section in order to minimize latency. By using a proportional mapping function, we can schedule more than one SCH exchange in the Data period and therefore more than one data flow.

We illustrate the operation of MAC^2's multi-hop scheduling and multi-hop forwarding in the simple 4-node scenario in Fig. 4.3. Node N1 has data pending in its queue; it uses the concatenation scheme, as explained in the previous section, to construct a super packet. N1 uses the CSMA/CA protocol to contend for the channel. If N1 wins, it will start by sending the 1st SCH after a Distributed Inter-Frame Space (DIFS). The intermediate nodes N2 and N3 relay the SCH packet and at each node, the time taken to process the SCH is one Short Inter-Frame Space (SIFS). During the relay process, the SCH packet serves as an RTS for the next hop and a CTS for the previous hop. In conclusion, the SCH packet from the source node only serves as an RTS and the SCH packet of the destination node only serves as a CTS. Based on R_{min}, nodes will wake up at the correct time to transmit/receive data in the Sleep period.

5.3 Performance Analysis

5.3.1 Achieving Minimum Latency

Using the proportional mapping (4.1) with $R = \frac{T_{Sleep}}{T_{Data}}$, any two data packets (in this case SP packets) never collide at an intended receiver as proven in the original DW-MAC paper [76]. However, there is a possibility of collision between SP and ACK packets. In this subsection we present the conditions of the mapping function to guarantee that there will be no SP-ACK collisions. In the following analysis and the remainder of this paper we assume that the bit error rate is zero for all transmissions.

We investigate a multi-hop relay in a busy cycle. Node i and its neighbor, node $(i+1)$, always satisfy the following equation:

$$R = \frac{T_S^i}{T_D^i} = \frac{T_S^{i+1}}{T_D^{i+1}} \tag{5.2}$$

The intervals between the time transmitting SCH packets and SP packets of two nodes, ΔT_S^i, ΔT_D^i, are defined as follows: $\Delta T_S^i = T_S^{i+1} - T_S^i$ and $\Delta T_D^i = T_D^{i+1} - T_D^i$.We can obtain R by using the property of equal fractions series from (4.2):

$$R = \frac{\Delta T_S^i}{\Delta T_D^i} \tag{5.3}$$

However, when a node receives an SCH packet from its previous hop in the Data period, it needs a duration of $SIFS$ to process and relay the packet to its next hop. Assume l_{SP}, l_{ACK}, l_{SIFS} to be the length of SP, ACK, and SIFS, respectively, hence $\Delta T_D^i = l_{SCH} + l_{SIFS}$. To guarantee no SP-ACK collision in the Sleep period, for example at node $(i+1)$, node

$(i+1)$ should start to transmit its SP packet after node i receives its ACK.

$$\Delta T_S^i \geq l_{ACK} + l_{SP} + l_{SIFS} + \delta \tag{5.4}$$

δ can be considered as the state transition time, and it is negligible compared with other timing parameters. Using (4.3) and (4.4) we can get the condition which guarantees no SP-ACK collision with all possible length of SP in a multi-hop flow:

$$R \geq \frac{l_{ACK} + l_{TH} + l_{SIFS}}{l_{SCH} + l_{SIFS}} \tag{5.5}$$

In the following we will present a theorem and proof for a collision free receiver:

Theorem 1 *There will be no collision at the intended receivers in the Sleep period if R satisfies condition (4.5).*

Proof 1 *If the nodes are the intended receivers in the Sleep period, that means their SCH packets have been successfully exchanged in the previous Data period. We then prove the above theorem by contradiction. Assume that there is a collision at a receiver i. If there is one flow in the network we can easily find that it can only be an SP-ACK collision. And with (4.5) that collision does not occur. If there is a collision between two flows at node i, node i takes part in two SP-ACK transmissions of two neighbors $n1$, $n2$. We denote time to start to transmit these two SP packets (SP1 and SP2) by T_S^{n1} and T_S^{n2}. Since there is a collision at node i, and $\max\{l_{SPj}\} \leq l_{TH}$ $(j = 1, 2)$:*

$$|T_S^{n1} - T_S^{n2}| < l_{TH} + l_{SIFS} + l_{ACK} \tag{5.6}$$

If node i is the intended receiver of two data packets it means that it successfully relayed two SCH packets (SCH1, SCH2) in the previous Data period. Assuming $T1_D^i = T_D^{n1} + l_{SIFS} + l_{SCH}$, and $T2_D^i = T_D^{n2} + l_{SIFS} + l_{SCH}$. Because the node has to wait at least for the *confirmation SCH in a flow before joining another flow, we have:*

$$|T1_D^i - T2_D^i| \geq l_{SIFS} + l_{SCH} \tag{5.7}$$

From (4.6) and (4.7) we have:

$$R = \frac{|T_S^{n1} - T_S^{n2}|}{|T_D^{n1} - T_D^{n2}|} < \frac{l_{ACK} + l_{TH} + l_{SIFS}}{l_{SCH} + l_{SIFS}} \tag{5.8}$$

Hence, we can conclude that the transmissions are collision free at any intended receiver.

We set $R_{min} = \frac{l_{ACK}+l_{TH}+l_{SIFS}}{l_{SCH}+l_{SIFS}}$ and show that among all values of R, R_{min} gives the minimum delay. The delivery latency is proportional to the number of hops from the source node. In duty cycling protocols, the data transmission can only be initialized during the active period of the radio. Thus, a newly generated packet is initially delayed by $T_{Cycle}/2$ on average (denoted as T_{Init}), $T_{Cycle} = T_{Sync} + T_{Data} + T_{Sleep}$.

Assuming that the destination is h hops away from the source node, the average delivery latency of a node h hops away from the source node $L(h)$ is:

$$L(h) = T_{Init} + N_C(h, N) \times T_{Cycle} + T_{last}(h, N) \tag{5.9}$$

Here, N is the number of hops a packet passes in one cycle and $N_C(h, N)$ is the number of cycles a packet passes before the last cycle, and $N_C(h, N)$ is specified as follows:

$$N_C(h, N) = \begin{cases} \lfloor \frac{h}{N} \rfloor & \text{if } h\%N > 0 \\ \lfloor \frac{h}{N} \rfloor - 1 & \text{otherwise} \end{cases} \tag{5.10}$$

where $\lfloor . \rfloor$ is the floor function and % is the modulo operation. $T_{last}(h, N)$ is the latency in the last cycle and can be calculated as:

$$T_{last}(h, N) = T_S^{N_{last}(h,N)} + l_{SP} + l_{SIFS} + l_{ACK} \tag{5.11}$$

where $N_{last}(h)$ is the number of hops that the packet passed in the last cycle.

$$N_{last}(h,N) = \begin{cases} h\%N & \text{if } h\%N > 0 \\ N & \text{otherwise} \end{cases} \tag{5.12}$$

Since SCH relays are similar for all R, from (4.1)(4.9)(4.10)(4.11)(4.12) we obtain that the minimum latency is achieved when R is at a minimum. The same proof is shown in our previous work for DW-MAC [84]. Since we use l_{TH} to express the packet length in the mapping function, the delivery latency of a single packet becomes longer in case $l_{SP} < l_{TH}$. However, this slight inefficiency exists only in low traffic environments. Once the traffic load increases, l_{SP} will often be equal to l_{TH} and the difference in latency will become negligible. In any case, the throughput per cycle is not affected.

5.3.2 Upper and Lower Bound of Delivery Latency

In this section we give bounds of delay under low traffic conditions. In MAC^2 and other multi-hop protocols, the number of hops (N) that a data packet can reach in a duty cycle mainly depends on T_{Data}. The equation below expresses the relationship between N and T_{Data}, which can be considered as the most important factor in the protocol.

$$T_{Data} = CW + l_{DIFS} + N \times (l_{SIFS} + l_{SCH}) \tag{5.13}$$

Note that because of the small SCH packet, the duration of the contention window (CW) could be used to send the SCH. In this case, the maximum and minimum values of N can be determined as: $N_{max} = \lfloor \frac{T_{Data}}{l_{SIFS}+l_{SCH}} \rfloor$ and $N_{min} = \lfloor \frac{T_{Data}-CW}{l_{SIFS}+l_{SCH}} \rfloor$.

First, we investigate the single packet scenario in which the destination is h hops away from the source node. From (4.9) the minimum delivery latency of a node h hops from the source node $L(h)$ can be calculated as follows:

$$L(h) = T_{Init} + N_C(h, N_{max}) \times T_{Cycle} + T_{last}(h, N_{max}) \tag{5.14}$$

In addition the $T_S^{N_{last}(h)}$ can be calculated from (1): $T_S^{N_{last}(h)} = T_D^{N_{last}(h)} \times R_{min}$, and $T_D^{N_{last}(h)}$ can be calculated from T_D^1, which is the time of transmission of the first SCH in the last cycle.

$$T_D^{N_{last}(h)} = l_{DIFS} + (N_{last}(h) - 1) \times (l_{SIFS} + l_{SCH}) \tag{5.15}$$

On the other hand, the minimum value of T_{Init} is zero, so from (4.14)(4.15), we have the lower bound of the latency to be:

$$\begin{aligned} L_{lower}(h) &= N_C(h, N_{max}) \times T_{Cycle} + ((N_{last}(h, N_{max}) \\ &\quad -1) \times (l_{SIFS} + l_{SCH})) \times R_{min} + l_{DIFS} + \\ &\quad l_{SP} + l_{SIFS} + l_{ACK} \end{aligned} \tag{5.16}$$

Now we find the value of the upper bound. A packet incurs the maximum latency when it can traverse only N_{min} hops in a cycle. Moreover, T_{Init} will be at its maximum when the packet is generated at the beginning of the Data period in a idle cycle. Thus, the upper bound of latency is:

$$\begin{aligned} L_{upper}(h) &= T_{Sleep} + T_{Data} + N_C(h, N_{min}) \times T_{Cycle} + \\ &\quad (l_{DIFS} + (N_{last}(h, N_{min}) - 1) \times (l_{SIFS} + \\ &\quad l_{SCH})) \times R_{min}) + l_{SP} + l_{SIFS} + l_{ACK} \end{aligned} \tag{5.17}$$

Secondly, we investigate a scenario in which M transmitters try to send a packet to one sink, and the number of retransmissions allowed is $Nret$. We assume no packet is generated while the previously generated packets are being transmitted. When a node fails in the channel contention process, it waits until the next cycle to contend again. Hence, the maximum latency will be at the node which wins the channel last. After $min\{Nret, M\}$ cycles, it will start to setup the data flow:

$$L_{upper}^{Nret,M}(h_{max}) = \min\{Nret, M\} \times T_{Cycle} + L_{upper}(h_{max}) \tag{5.18}$$

Here, h_{max} is the maximum hop-count to the destination among the M transmitters. If M is greater than $Nret$, $(M - Nret)$ nodes will fail to transmit data.

Now let us investigate a scenario in which there are some SCH packets that failed to be transmitted to next hops. If a node receives a SCH packet but cannot relay it to its neighbor, it receives the corresponding data packet and keeps it in its queue. The reason is that there is no collision during this Sleep period. The node can be considered as the transmitters in the upcoming cycle. If the scenario contains several transmitters, we can use (4.18) to investigate the upper bound of latency. Since this analysis is applied for a low traffic environment, we can assume there will be few failed SCH packets. The upper bound can be derived from (4.18) by replacing M with M_{max} the maximum number of transmitters or the maximum number of neighboring nodes for the specific topology.

5.4 Performance Evaluation

We use the network simulator ns-2 [81] to evaluate our analysis under a simple scenario and the protocol's performance under various scenarios. Each sensor node had a single omnidirectional antenna through which the combined free space and two-ray ground reflection radio propagation models were employed. Key networking parameters are shown in Table 4.1, power consumption parameters were set to typical values for a Mica2 radio (CC1000). The 250m transmission range and the 550m carrier sensing range were modeled after the 914-MHz Lucent WaveLAN DSSS radio interface; although not typical for a sensor node,

Table 5.1: Networking parameters

Rx Power	22.2 mW	Sleep Power	3 μW
State Transition Power	31.2 mW	Tx Power	31.2 mW
Idle Power	22.2 mW	Tx Range	250 m
Carrier Sensing Range	550 m	SIFS	5 ms
Contention Window (CW)	64 ms	DIFS	10 ms
Retry Limit	5	l_{DATA}	43 ms
l_{ACK}	11 ms	l_{SCH}	14.2 ms
l_{TH}	243 ms	ifq	2500 bytes

we used these parameters to make our results comparable to DW-MAC. In our evaluation of power efficiency, we focused on the energy consumed by radios, but ignored the energy consumed by other components such as CPU and memory [83]. The original size of one data packet is 50 bytes, which leads to a transmission time of $l_{DATA} = 43ms$. The super packet threshold is set to 300 bytes or $l_{TH} = 243ms$. Note that in our simulations the transmission time of a packet is calculated the same way as described in the RMAC and DW-MAC papers as follows:

$$l_{packet} = \frac{packetsize}{EB} + l_{preamble} + l_{processingtime} \tag{5.19}$$

where $EB = 10Kbps$ stands for the effective bandwidth, $l_{preamble} = 2ms$ for the transmission time of a preamble, and $l_{processingtime} = 1ms$ for the processing time. The capacity of the queue between the network and link layers (ifq) is set to 2500 bytes and other values are kept similar to those used in the evaluation of DW-MAC. All time duration parameters are shown in Table 4.2.

To simplify our evaluations, we did not include the routing traffic in the simulations. We also assumed that there was a routing protocol deployed to provide the shortest path between any two nodes and ensured that the networks we used in our simulations are connected networks. As it has been shown in DW-MAC's original paper, DW-MAC outperforms other protocols in a wide range of traffic load conditions. Therefore, we based our protocol on DW-MAC and also used it as a benchmark in our evaluation.

5.4.1 Latency Bounds

For the sake of simplicity, we used a chain scenario with 15 nodes. The distance between two neighboring nodes is 200 meters. We compared the performance of MAC^2 with two values

Table 5.2: Time duration parameters

	T_{Cycle}	T_{Sync}	T_{Data}	T_{Sleep}
DW-MAC	4465 ms	55.2 ms	168 ms	4241.8 ms
MAC^2	4465 ms	55.2 ms	168 ms	4241.8 ms

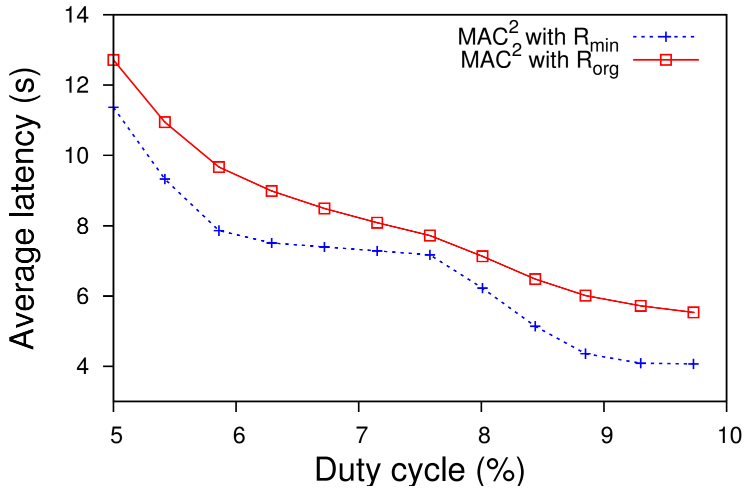

Figure 5.4: Achieving minimum latency

of R: R_{min} and R_{org} (where $R_{org} = T_{Sleep}/T_{Data}$). As we discussed before, the number N in (4.11) is one of the key factors in designing the multi-hop MAC protocol. We kept the value of T_{Cycle} at $4465ms$ and varied T_{Data} in steps of $l_{SIFS} + l_{SCH}$ to allow more hops to be passed in one cycle. Hence, the duty cycle value (DC) also varied. Note that the DC is calculated as follows: $DC = (T_{Sync} + T_{Data})/T_{Cycle}$.

In this evaluation, the source node 0 at the beginning of the chain sends to the destination node 14 which is the farthest node from the source, once every 30 seconds, and the total number of packets sent is one hundred. We measured the average delivery latency and plotted it in Fig. 4.4. The average delivery latency decreased when increasing DC. With a DC smaller than 6% almost all packets needed three cycles to reach the destination. Furthermore, with a DC in the range $7\% - 10\%$ they needed two cycles and one cycle otherwise. However, in all cases, we found that MAC^2 (with R_{min}) has lower latency than MAC^2 with R_{org}.

We evaluated the lower and upper bounds of latency in various scenarios. First, we considered the chain scenario with a single source. The length of the chain varied from 4 to 15 hops, and the source sent a packet every 30 seconds to the destination at the end of the chain. We kept the duty cycle at 5%, the default value in MAC^2. Fig. 4.5 and Fig. 4.6 show

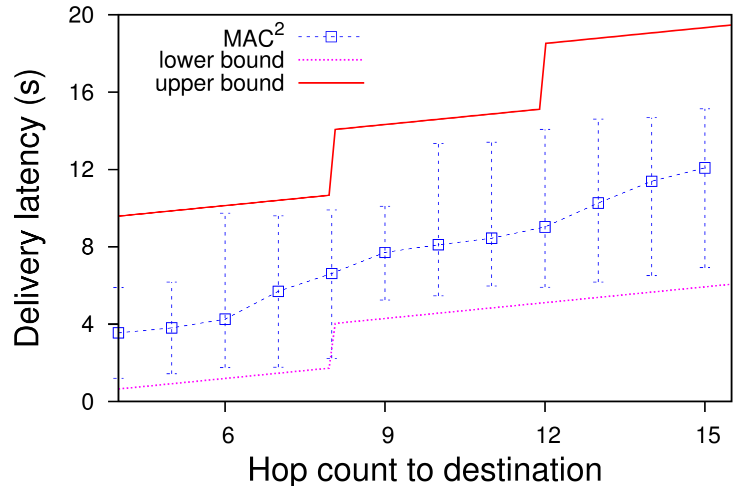

Figure 5.5: Latency bounds in single source scenario

the simulations' results and the numerical bounds' results. The graphs of the lower and upper bounds show that a packet can reach a node 8 hops away in the best case and 5 hops away in the worst case, respectively. We can see that the maximum, minimum, and average values of the delivery latency are in the range of the upper and lower bounds. The difference between the maximum and minimum values depends on T_{Init}. The delay never reached the upper bound because there were no collisions in any of the cycles. This evaluation can be a guide for us to investigate the hop length to the sink after deployment and to choose a suitable duty cycle to satisfy the delay requirements.

After that, we investigated a scenario with two transmitters. We assumed that between node 0 and node 1 there was an event that occurred during each 30-second interval. The sensing range of both nodes was 200 meters. After sensing the event, nodes sent a packet to the sink node, the farthest node in the chain. The chain length varied from 4 to 15 hops. Node 0 and node 1 contended for the channel every 30 seconds. That is, when there was an event, only one node could send a packet; the other one had to wait until the next cycle. Since $Nret = 5$ in this case, we have $min(M, Nret) = 2$. We plotted the upper bound as given in (4.18). The results shown in Fig. 4.6 lead us to the same conclusion as in the previous experiment.

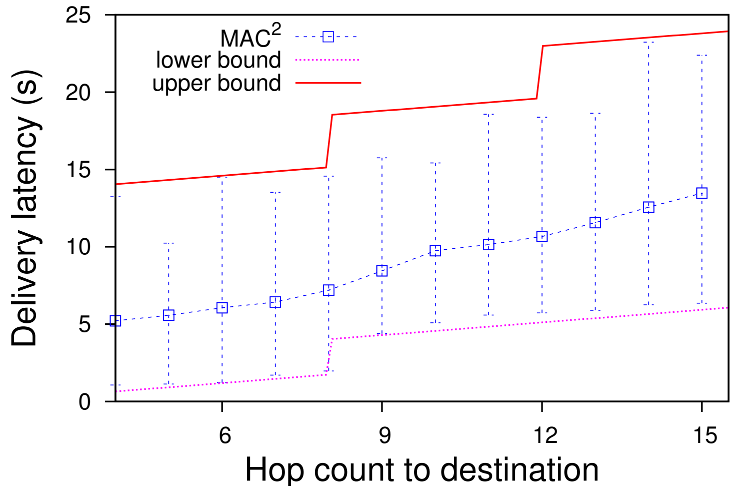

Figure 5.6: Latency bounds in two sources scenario

5.4.2 Results in Grid Scenarios

In this scenario we use a grid of 7×7 cells. Each node is 200 meters apart from its direct neighbors and the sink node is at the center of the grid. We use the Random Correlated-Event (RCE) traffic model introduced in [76], which creates events at random locations. All nodes that can sense a given event, i.e. nodes for which the event lies within the sensing range, generate one data packet in response. We investigated scenarios with all nodes having the same sensing range of 100, 150, 200, 250, 300, 350, 400, 450 and 500 meters. For each of the sensing range's value we ran a simulation with 200 events. With the specific number of events we set the interval between two events (IE) to 25, 50, and 100 seconds and compared the performance between MAC^2 and DW-MAC.

Figure 4.7 shows the average energy consumption in the evaluation. In DW-MAC, the average energy consumption increases slightly when the sensing range increases as well as the value of IE. In MAC^2, when the sensing range increases the average energy consumption increases but the value is lower than in DW-MAC because more data packets are generated and the nodes have to keep their radio on for more cycles. We come to the same conclusion when IE increases. If IE is large enough (IE = 50, 100 seconds), most packets generated by a previous event reach the sink before the next event occurs and the adaptive mechanism

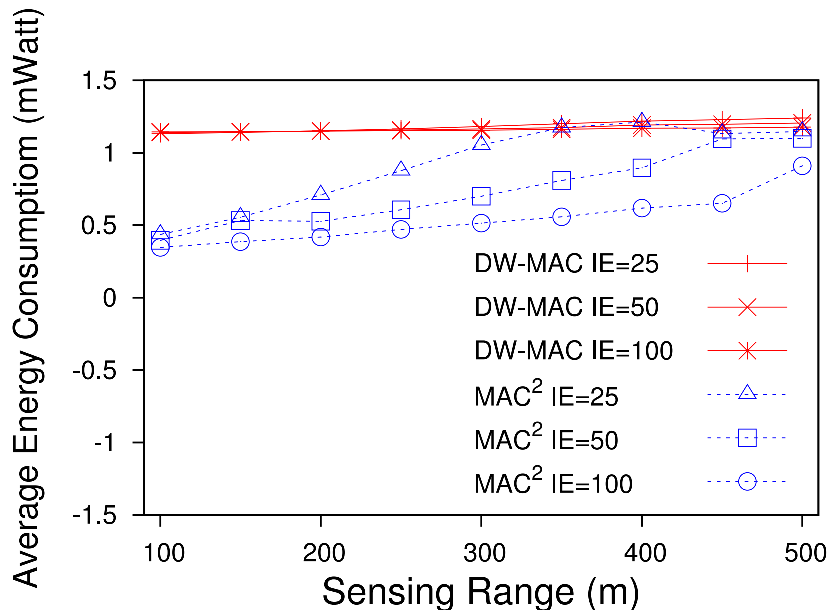

Figure 5.7: Average energy consumption in grid scenario

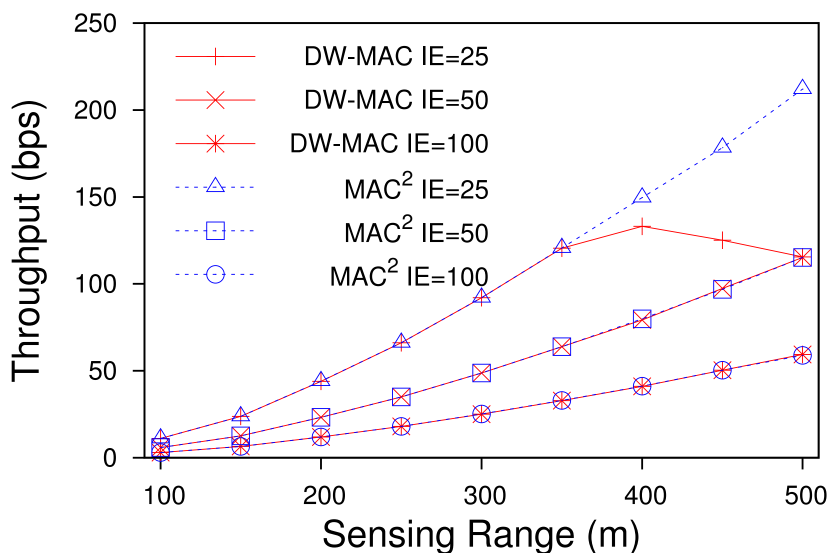

Figure 5.8: Throughput at the sink in grid scenario

can save power in some idle cycles. Therefore, we can conclude that MAC^2 achieves better energy efficiency than DW-MAC.

When the IE value is at 50 or 100 seconds, the throughput at the sink keeps increasing along with the sensing range as shown in Fig. 4.8 and Fig. 4.9. But when IE is smaller, there are still some packets that can not go to the sink when new events occur, DW-MAC starts to

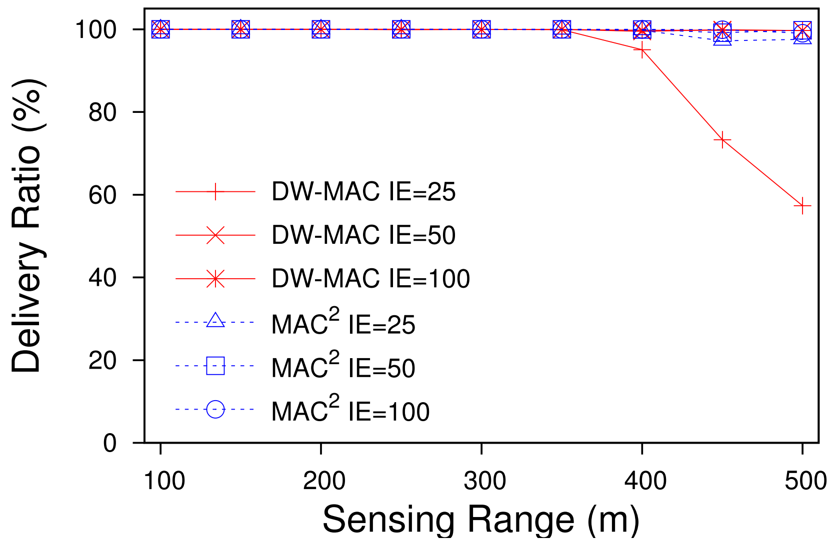

Figure 5.9: Delivery ratio in grid scenario

drop packets when the sensing range is at 350 meters, and the throughput slightly decreases due to packet collision. But in the case of MAC^2, the concatenation scheme is applied to handle this situation, the throughput continuously increases and the delivery ratio is kept at almost at 100%.

5.4.3 Results in Network Scenarios

We also evaluated MAC^2's performance under a random network scenario. A network contains 100 nodes which are randomly deployed in the square area 1500×1500 meter. The sink node is located at the top right corner of the area. We generated 10 different scenarios with 200 events each inside the area. We slightly modified the RCE traffic model to make it more realistic. Each event is randomly generated and the interval between two subsequent events was set to a random interval between 0 and 50 seconds. After the sink received the final packet, we kept the network running for another 100 seconds and compared the performance of MAC^2 against DW-MAC. The evaluation results are shown in Fig. 4.10, Fig. 4.11, Fig. 4.12, and Fig. 4.13; each average value is calculated from the results of 10 random runs and the error bars show the 95 % confidence interval.

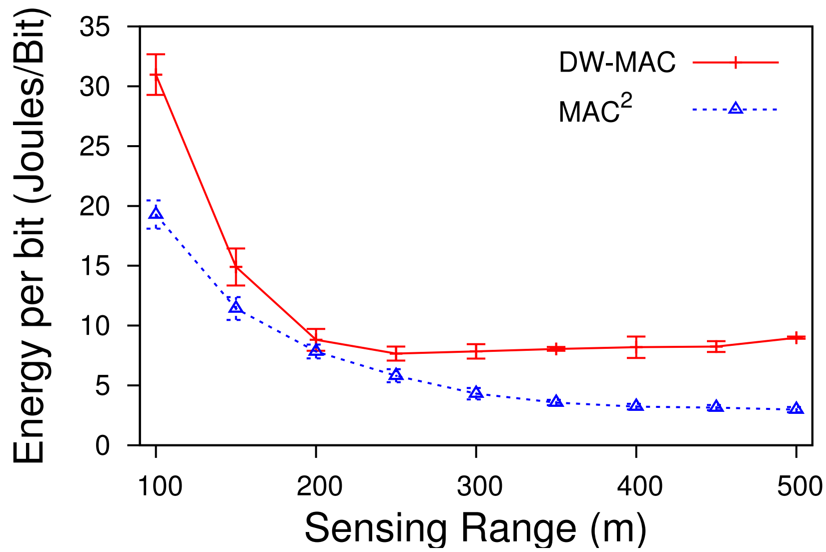

Figure 5.10: Energy consumption per bit in network scenario

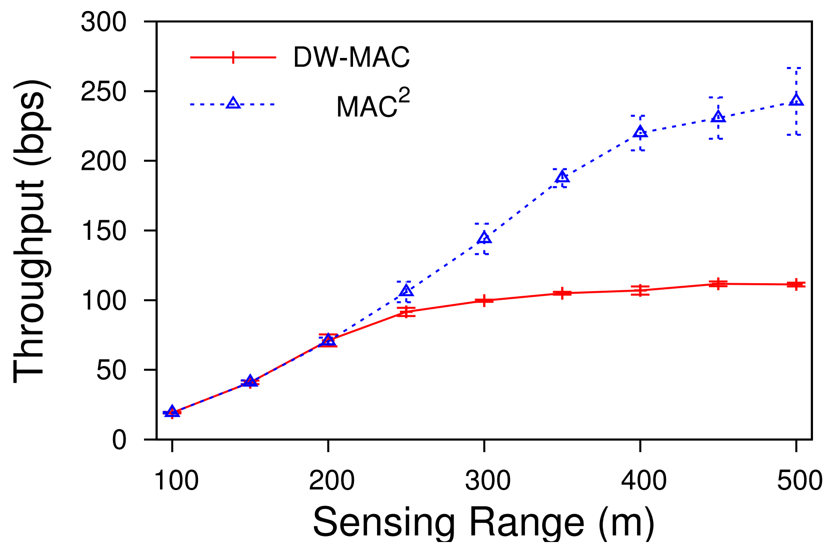

Figure 5.11: Throughput at the sink in network scenario

Figure 4.10 shows the energy consumption per bit which is calculated by the ratio of total energy consumption to the throughput. We use this value to reflect the energy efficiency of the network. When the sensing range is smaller than 200 meters, traffic load is low and MAC^2 is more energy efficient than DW-MAC since MAC^2 employs an adaptive mechanism, the nodes keep their radio off when there is no data and therefore save energy. At this level

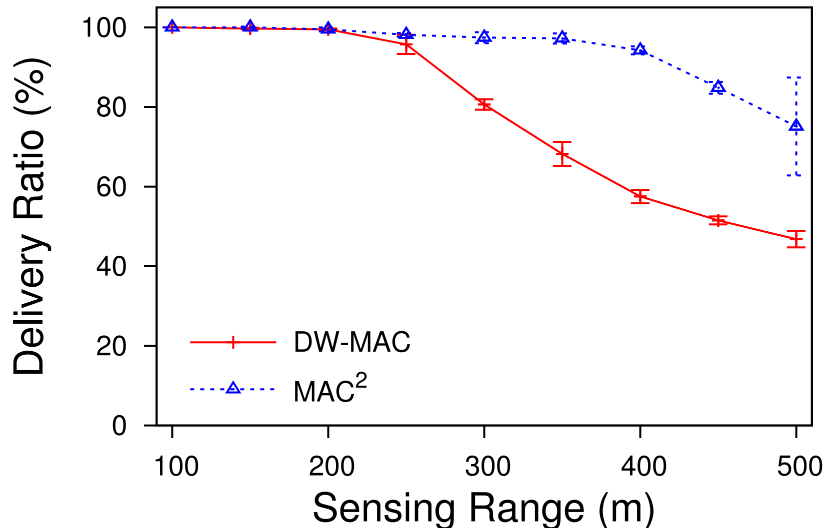

Figure 5.12: Delivery ratio in network scenario

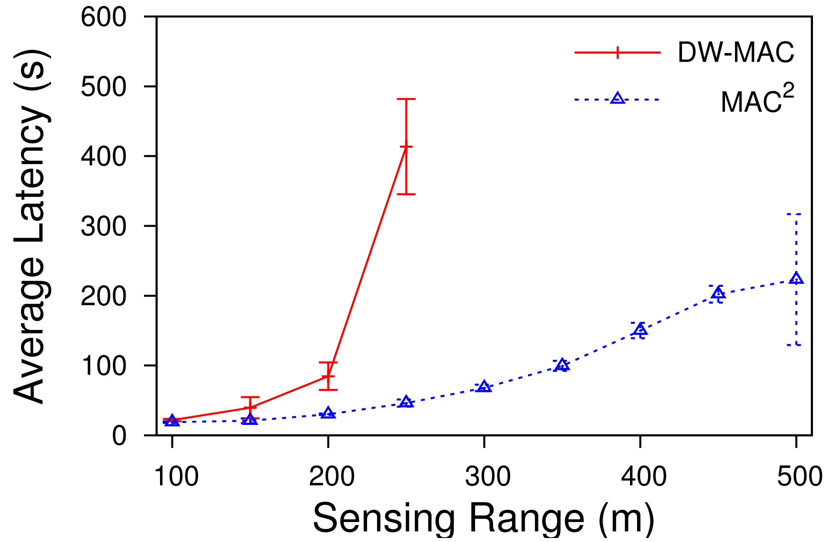

Figure 5.13: Average latency in network scenario

of traffic load, the two protocols share the same throughput performance and handle 100% delivery ratio. When the sensing range increases, the traffic load becomes heavier and MAC^2 still consumes less power than DW-MAC. The reason is the concatenation scheme is applied and the control overhead is largely reduced. Moreover, since there are fewer packets in MAC^2's network, the collision probability is smaller.

The throughput performance of the two protocols is shown in Fig. 4.11. When the sensing range grows larger than 250 meters, in MAC^2 the throughput keeps increasing. In the case where the sensing range is 500 meters, MAC^2's throughput is even 2.5 times higher than DW-MAC's. That is because of the effectiveness of the concatenation scheme. In DW-MAC, although more packets are generated, the amount of data received at the sink stays almost the same. That means more packets are dropped. The same conclusion can be drawn from Fig. 4.12, the delivery ratio sharply decreases in DW-MAC's case when the traffic load gets higher. That means DW-MAC's network reaches the saturated state faster then MAC^2's network does. Figure 4.13 shows the end-to-end latency of the two protocols where MAC^2 also outperforms DW-MAC. When the sensing range is larger than 250 meters DW-MAC's latency increases excessively due to the saturation of nodes' buffers. Therefore, those values are not shown in the figure. Note that, in MAC^2 we use the earliest generated packet in the super packet as the start point of super packet transmission. We can conclude that MAC^2 is more energy, latency and throughput efficient than DW-MAC for all investigated traffic loads.

5.5 Summary

In this chapter we present MAC^2, an energy efficient, high throughput multi-hop MAC protocol. The protocol employs multi-hop forwarding to overcome the latency burden of duty cycling. Besides that, it is optimized to perform well in a wide range of traffic load conditions. MAC^2 derives its wake up on demand manner from DW-MAC to support multiple traffic flows in a single cycle. Furthermore, MAC^2 adopts an adaptive mechanism that adjusts the listening period according to traffic load, minimizing idle listening. Finally, MAC^2 utilizes a concatenation scheme that can combine several queued packets for the same destination and send them as one super packet. Throughput is significantly increased and control overhead is reduced and evaluations of MAC^2 show that it outperforms other state-of-the-art multi-hop protocols.

Chapter 6

Providing QoS Awareness in Receiver-Initiated MAC Protocol

In this chapter, we present background knowledge of Quality of Service (QoS) concept in communication networks. We then describe the QoS challenges, which are especially addressed for MAC protocols in sensor networks. Finally, we present our approach in designing a QoS Aware MAC protocol for WSNs.

6.1 Background

The term QoS is broadly used in the area of wired and wireless networks but there is a lack of an exact definition covering all QoS meanings. International Telecommunication Union (ITU) Recommendation E.800 (09/08) has defined QoS as: "Totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service" [88]. The original definition refers QoS comprising requirements on all the aspects of a connection, such as service response time, loss, signal-to-noise ratio, cross-talk, echo, interrupts, frequency response, loudness levels etc. on telephone networks. However, the definition is recently retrieved and updated; more QoS-related terms are added, hence it can be adopted as a reference. QoS previously refers to an orchestration of resource reservation rather than the provided service quality itself. Recently, QoS is widely accepted as the ability of giving different priorities to various different patterns in the networks such as users, applications, data flows, frames or packets. Hence the network users can get higher level of performance over others through a set of measurable service parameters such as delay, effective bandwidth, and packet loss. In traditional communication networks, the traffic is transfered under only best-effort manner, which has been proven to have many disadvantages. Moreover, the development of communication medium and the diversity of users' demands yield various types of traffic in the networks. Therefore, it is necessary to have QoS-guaranteed mechanism with different types of data in order to increase the network performance and network benefit.

6.1.1 QoS Models in Communication Networks

QoS expresses the ability of a network to satisfy the requirements of the user or application at certain levels. In a different speaking, QoS describes the level to which particular network traffic is prioritized over other types of traffic. In traditional wire and wireless networks, there are two main types: hard QoS and soft QoS. The applications that require hard QoS should be provided deterministic QoS guarantees, such as strict bounds on delays, effective bandwidth or packet losses etc. An example would be Asynchronous Transfer Mode (ATM). The QoS service level and bandwidth delivered by an ATM network is agreed to and guaranteed by contract. On the other hand, the applications also has tight QoS requirements in soft QoS approach. However, the temporal violations on QoS provisioning can be tolerated to a certain extent [89]. There are various applications related to soft QoS in the current Internet such as VoIP, real-time network application. In order to provide hard or soft QoS guarantees, service differentiation is the widely adopted. From theoretical domain, there are two service differentiation models proposed for conventional computer networks: Integrated services (IntServ) [90] and differentiated services (DiffServ) [91]. The differentiation models are used to prioritize traffic patterns (i.e., flows or packets), map the priorities into different service qualities. Moreover, they provide required service quality by sharing limited resources among the traffic patterns. Fig. 5.1 presents the concepts of both IntServ and DiffServ.

IntServ model specifies a fine-grained QoS system and follows the hard QoS approach [92]. The model maintains service on a per-flow basis; the flows can be either data-centric or host-centric. In the context of sensor networks, the data-centric flow can be information generated by motion sensors reacting to a common sensing event; the host-centric flow can be the sequence of packets between a particular source and destination. However, IntServ model has several disadvantages, which make it inappropriate for WSNs. First, it is hard to provide guaranteed service quality due to time varying channel capacity on the wireless medium. Second, maintenance of the per-flow states of the sensor nodes and scalability for dense networks are challenges. Third, IntServ model requires a reliable in-band or out-of-band QoS signaling within the sensor network for resource reservation which is very hard or expensive to assure in WSNs.

On the other hand, DiffServ model maintains service on a per-packet basis. The model has a major drawback of costly memory requirement since the nodes taken part in the model's communication will behave as a source and an intermediate hop. However, DiffServe model is lightweight and easy-to-implement; moreover the model operates in a multi-hop manner. Therefore, the model is suitable and can be easily adapted to WSNs. Each type of packet will be assigned a level of importance; and the information will be apparent for every nodes in the network. Following this approach, not only MAC protocols but also different layers protocols can treat the packet relying on its priority imposes. In our work, we also consider the DiffServ model at MAC layer.

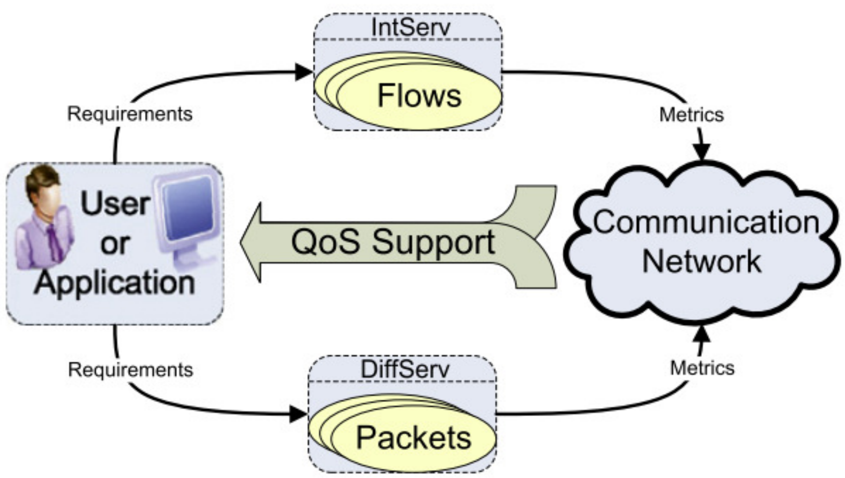

Figure 6.1: Concepts of IntServ and DiffServ models

6.1.2 QoS Challenges in WSNs

QoS perspectives

QoS perspective refers to the aspects of QoS, which users or system designers are interested. In WSNs, the QoS perspectives can be divided into two categories as application-specific and network-specific [93]. These two perspectives represent the two different approaches already followed in the literature. The application-specific perspective focuses on the quality of the application itself. QoS is again assured by fulfilling the requirements imposed by the application such as lifetime [94], network coverage, efficient deployment, quality of the sensing, etc. The network-specific perspective provides service quality during delivery of the data. From this perspective, network resources are utilized efficiently in each layer of the communication protocol stack to fulfil the requirements imposed by the carried data, such as latency, packet loss, reliability. Since our goal is on designing QoS-aware MAC protocols, we have been approaching from the network-specific perspective to QoS provisioning. The application specific perspective is therefore out of our scope in this work.

QoS Challenges

In sensor networks, achieving QoS provisioning inherit most of the well-known QoS challenges from traditional wireless networks [95]. Moreover, the achievements should takes into account typical characteristics of sensor networks, such as severe resource constraints and harsh environmental conditions that pose additional unique challenges for QoS-support. Specifically, a QoS-aware MAC protocol for WSNs should be an energy efficient MAC with QoS awareness. Therefore, all requirements mentioned in Chapter 1 should be carefully considered along with QoS challenges.

6.1.3 QoS-Aware MAC protocols

Again, the MAC protocols generally employ a duty cycling mechanism according to which the radio module of the sensors is turned on and off frequently in order to enhance energy efficiency. On the other hand, a sensor mote always include a communication part and a sensing part. The sensing part contains several sensors, e.g, the sensorboard MTS4200 [96] has humidity, temperature, light, pressure sensors etc. The sensors may generate different types of traffic that have different requirements. Moreover the traffic can be predictable or unpredictable in WSNs. Hence the MAC protocols necessarily handle well with the variation of traffic as well as meet the requirement of energy efficiency. There are several duty cycling MAC protocols that are proposed with QoS handling mechanisms. However most of them are based on the synchronous duty cycling or/and TDMA (Time Division Multiple Access) mechanisms [97–99]. Those protocols are always based on an assumption in which all nodes are perfectly synchronized.

6.2 Protocol Description

6.2.1 Overview

In this section we propose a new asynchronous receiver-initiated QoS-aware MAC protocol named AQ-MAC. AQ-MAC simply utilizes QoS with a single queue architecture. The protocol determines the transmission strategy depending on the priority of packets. If an incoming packet has high priority, AQ-MAC decides to send the packet immediately. Then the sender turns its radio on and wait for a potential receiver. Otherwise the node keeps the packets which have low priority in its queue in order to avoid the problem of energy wastage at sender. The queued packets can be sent out in a burst when the queuing time reaches a timeout value or a high priority packet arrives. Another advantage feature of AQ-MAC is the protocol adopts the concatenation scheme from our previous work [100]. The scheme concatenates several small packets that are destined to the same sink into a bigger one before sending out. Therefore the control overhead is significantly reduced.

6.2.2 QoS Provisioning

AQ-MAC protocol follows the differentiated service (DiffServ) mechanism [91] to achieve QoS. The mechanism is revealed to be most suitable for WSNs [101] since it is lightweight, easy-to-implement and can operate in a multi-hop manner. DiffServ usually consists of two phases: priority assignment and differentiation between priority levels. In AQ-MAC, a static priority assignment method classifies the traffic into two classes based on its degree of importance. Accordingly, a packet is either low priority or high priority. The similar traffic classes can be found in many WSN's applications in which a packet is either a time critical or a normal one. If the super packet which is described in detail in the previous chapter consists a part of a high priority packet, it is assigned high priority. To provide different QoS, the differentiation method of AQ-MAC works as follows: when a high priority packet arrives at a node, it is going to be sent out as soon as possible. If the node is in sleeping mode, it immediately wakes up and becomes a sender. On the other hand, a low priority packet is often stored in a node's queue for a period of time. During that period, the node holding the low priority packet works as a normal non-sender. The length of the period is often determined by a predefined timeout value unless the node is triggered by an incoming high priority packet. Because of the resource limitations of sensor nodes, a WSN's QoS model should be lightweight. For that reason, AQ-MAC adopts the single queue architecture, hence there is no need to use a packet scheduler.

Receiver-Initiated Transmission

AQ-MAC is an asynchronous duty cycling protocol. In a normal operational cycle, a node alternately turns on/off its radio to reduce idle listening. The protocol inherits the receiver-initiated transmission from RI-MAC. This type of transmission is proven to outperform the traditional sender-initiated one [70]. Moreover it becomes popular in designing MAC protocols on real sensor motes [102].

In a receiver-initiated protocol, after waking up each non-sender node immediately broad-

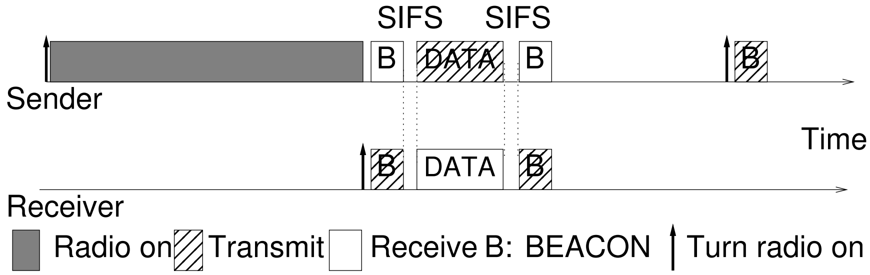

Figure 6.2: Receiver-Initiated Transmission

casts a beacon packet which contains its address. By doing so, it announces to its neighbors that it is active and ready for an incoming data packet. The node then samples the channel for a short period (called dwell time) to determine whether there is incoming packet or not. On the other hand, a sender that is holding a data packet keeps in the listening mode and waits for an intended receiver to wake up. Upon receiving the beacon from the receiver, the sender immediately transmits the pending data. A successful transmission is completed when a beacon with ACK function arrives at the sender. This beacon however can serve not only as an ACK packet but also as a new receiver-initiated beacon. When the sender has no queued packet, it becomes a non-sender. The node broadcasts a beacon right after its next wakeup time. Fig. 5.2 shows a typical receiver-initiated transmission. In the figure, SIFS is abbreviated for short inter-frame space, the duration needed to process a packet and switching radio mode.

By keeping the sender active and letting the receiver to schedule the transmission, the receiver-initiated protocol provides a nearly minimum value of the delivery latency. This benefit is suitable for time critical traffic, i.e., the high priority traffic in AQ-MAC, but not for the low priority one since a huge amount of energy consumed by idle listening at the sender. Therefore it is necessary to reduce that energy consumption, especially in the environment with the low priority traffic. AQ-MAC is designed to handle with different types of traffic and solves that problem by enabling QoS per packet. In AQ-MAC, a node queuing a low priority packet works as a non-sender node as mentioned in previous section. That means it doesn't waste energy to idly listen the channel whenever a low priority packet arrives.

Another benefit of the receiver-initiated transmission is a possibility for a beacon from a receiver can be recognized both as an ACK and a ready-to-receive packet. The benefit comes along with the improvement in both of energy efficiency and latency performance. However it occurs only in the case the sender that receives an ACK beacon has an extra queued packet. Since the low priority packets are often queued, AQ-MAC 's nodes frequently enjoy the benefit. In addition, AQ-MAC has a concatenation scheme which concatenated several packets into a bigger one before sending out. The energy efficiency is furthermore improved.

AQ-MAC and RI-MAC also share the same collision detection and retransmission schemes. When a collision occurs at the receiver, it retransmits a new beacon which includes a value of backoff window. The contending senders utilizes a random backoff period before retransmission to avoid collisions.

Concatenation scheme

This concatenation scheme is derived from our previous work [100] in which we apply the scheme for a synchronous duty cycling MAC protocol. However, we found that the scheme can be adopted by other asynchronous duty cycling MAC protocols whenever they have to handle with queuing packets. The improvement of performance is expected because of three reasons, which is also mentioned in the previous chapter. First, queuing packets are popular since the sensing part of a sensor node kept on at all the time even though the communication part is switch off periodically. Secondly, all packets are predominantly addressed to one and the same sink, this characteristic is used as the major requirement of packet concatenation. Thirdly, the sizes data packets are usually small and therefore the gains from reducing control packet overhead are comparatively large.

Timeout Value

To trigger the transmission of low priority traffic in AQ-MAC, a node maintains a timeout timer. The timeout timer of a node restarts when the node successfully implements a

Table 6.1: Networking parameters

Bandwidth	250 Kbps	Slot time	320 μs
CCA Check Delay	128 μs	Tx Range	250 m
Carrier Sensing Range	550 m	SIFS	191 μs
Backoff Window	0–255	Beacon size	6–9 Bytes
Retry Limit	5	Dwell Time	10 ms
T	10s	L_{TH}	112 bytes

transmission, i.e, the node receives an ACK beacon. The timer fires after a pre-determined timeout value or in the case of an high priority packet arrives. For example, the sink node's neighbors checks their queue for a pending packet after a timeout value $T_1 = T$ which is supposed to be set by an administrator. Other node updates the timeout value based on the distance to the sink and the value T following:

$$T_h = \frac{T}{h}$$

where h is the hop-count from the node to the sink. After the timeout timer is fired, the node turns its radio module on and transmits a pending packet following the receiver-initiated manner.

6.3 Performance Evaluation

We use the network simulator ns-2 [81] to evaluate our protocol under various scenarios. The networking parameters are listed in Table 5.1, where most of the parameters are based on the value of Micaz mote or CC2420 radio [103]. The original size of one data packet is 28 bytes, since the maximum size CC2420 supports is 128 bytes packet, we set the threshold for super packet with $n = 4$, that leads to L_{TH} equals 112 bytes. The value of T is 10 seconds. Due to a sensor consumes a similar power amount regardless whether it is transmitting, receiving or idle listening [44], we use the duty cycle value to indicate the energy consumption. This value is the total length of wakeup period divided by the total simulation time. We evaluate the protocol with different types of traffic (low priority, high priority and mixes of them) in

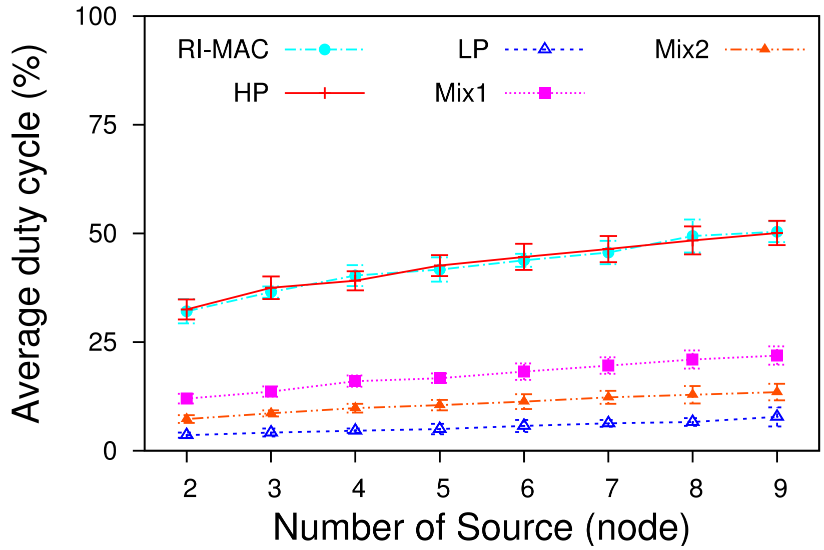

Figure 6.3: Average duty cycle in one-hop scenario

multiple types of networks. Even though RI-MAC does not support different types of traffic, we still show its performance as a baseline. To account clock drift we let a node choose a random value between 0.5 and 1.5 seconds for its next wakeup time.

6.3.1 Results in One-hop Scenario

In this scenario, multiple sources send packets to a destination. All nodes are in the others' communication range. The scenario is similar to the case when the sink's neighbors communicate with it. Each source node generates packets at an interval is randomly chosen between 0.5 and 1,5 seconds. The packet sources stop at 230 seconds, and the total simulation time is 240 seconds to make sure all queued packet are sent out. In our evaluations, the latency of a super packet is the duration between the earliest generated packet in the super packet and its received time. We evaluated AQ-MAC with 4 types of traffic: low priority (LP), high priority (HP), Mix1, Mix2. Mix1, Mix2 mean the traffic is generated one high priority packet over two packets, one high priority over four low priority packets, respectively. We varied the number of sources from two to nine. The results of the average latency, the average of duty cycle, and the throughput are shown in Fig. 5.3, Fig. 5.4, Fig. 5.5. In the figures, each value is averaged over ten runs, the error bars show 95% confidence interval.

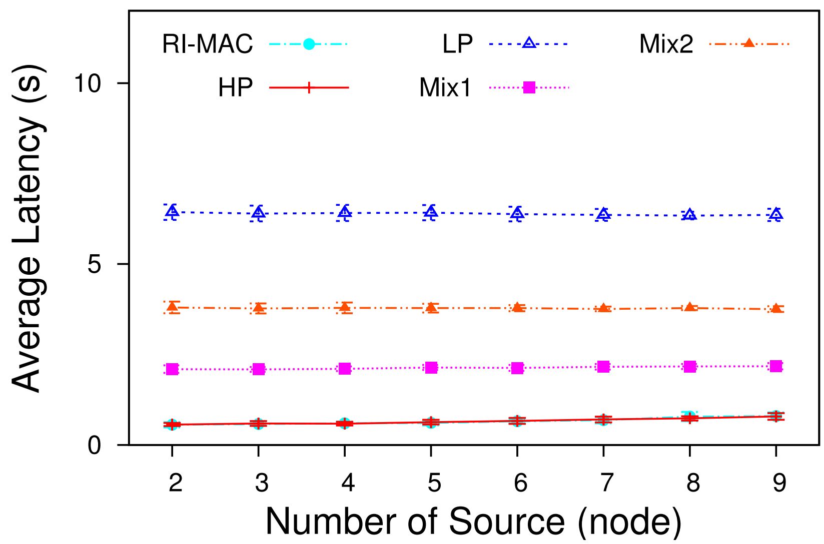

Figure 6.4: Average latency in one-hop scenario

Figure 5.3 shows the average duty cycle of all nodes in the network. We can see that with the high priority packet AQ-MAC works just like RI-MAC, and consume a large amount of energy (more than 30% in 2-source scenario). In this case the packet concatenation has less effect since the nodes transmit the packet very fast, so mostly packets are not queued. On the other hand, in the case of low priority traffic, the packets have to wait until the deadline to be transmitted out. The chance of sending out in a burst plus the concatenation scheme gives a very low value of duty cycle (below 8 % in 9-source scenario), that means the energy consumption is small. The big gap between the values of HP and LP indicates that a huge amount energy is consumed by the idle listing at the senders. When the traffic varies, from the graph of Mix1 and Mix2 we can conclude the larger amount of high priority traffic appears, the more energy consumes in the network. Low priority packets got the penalty at latency as shown in Fig. 5.4. However the value of latency is smaller than the timeout value, that is because only few packets (usually the first packet in a burst) incur the large latency. On the other hand, the latency of high priority packets are very small, nearly minimum.

Since the delivery rate values are always 100 %, we don't show those values. In Fig. 5.5,

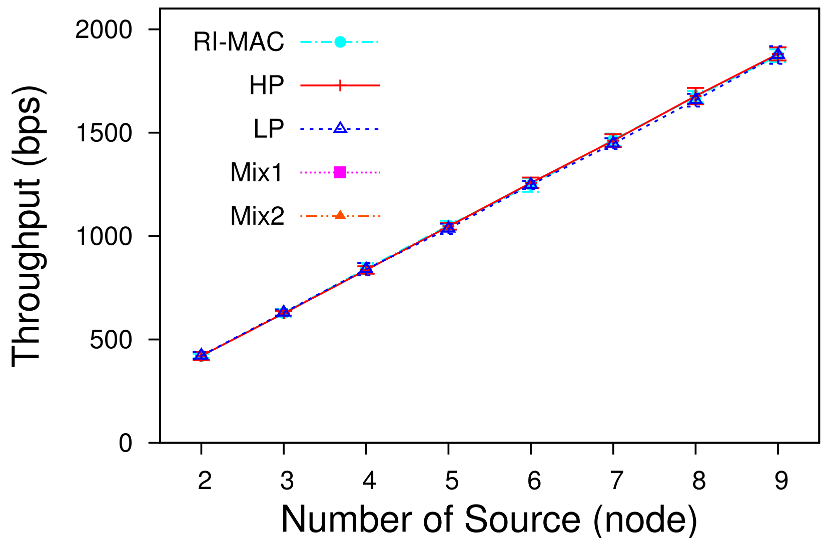

Figure 6.5: Throughput in one-hop scenario

the throughput increases along with the number of source. The values of different types of traffic are similar. That confirms the effect of collision avoidance using the receive-initiated transmission.

6.3.2 Results in Grid Scenario

To evaluate the performance of AQ-MAC under multi-hop fashion, we choose a grid network. The network contains 49 nodes forming a 7×7 grid with the distance between two neighbors is 200 meters Fig. 5.6. The sink node is located at the center of the grid, therefore the length of path from a node to the sink varies from 1 to 6 hops. In this evaluation, we use the random correlated event (RCE) traffic model which is firstly proposed in [76]. In this model an event is occurred at a random location, the nodes which are located in the circle with the radius R (called sensing range) center at the event's coordinate, are going to generate a packet. Each generated packet is toward to the sink. We generated ten different scenarios of 100 random events within the network area. The interval between two events is randomly chosen between 0 to 5 seconds. Each 100-event scenario then is evaluated under three types of traffic LP, HP and Mix. LP, HP stand for low priority, high priority and corresponding

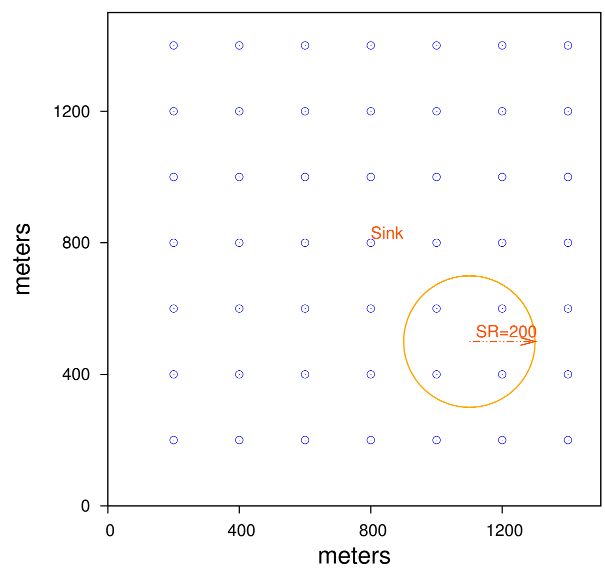

Figure 6.6: 49-node grid scenario with RCE traffic model

to the importance level of packets in the network. The Mix traffic includes 50 events which generate high priority packets, and 50 events which generate low priority packets. We varied the traffic load by varying the sensing range R from 100 to 500 meters. All the results are calculated within the period from the beginning of simulation to the time when the last packet is received. The evaluation results of average duty cycle, average latency and throughput at the sink are shown in Fig. 5.7, Fig. 5.8, Fig. 5.9, respectively. Each value is averaged over ten run, and the error bars show 95% confidence interval.

Note that the duty cycle value is proportional with the amount of energy consumption in the network. When the traffic is low (the sensing range is 100 or 200 meters), AQ-MAC with three types of traffic (LP, HP, Mix) and RI-MAC share similar values of energy consumption as in Fig. 5.7. That is because the number of generated packet is small, the packets can independently reach to the sink. But when the sensing range increases the energy consumption of the nodes increases accordingly. However the network with low priority traffic always achieves the best of energy efficiency since they can save the energy wasted at senders and get the benefit of the concatenation scheme. According to Fig. 5.7 the nodes equipped RI-MAC always consume the largest amount of energy. The AQ-MAC's nodes consume less

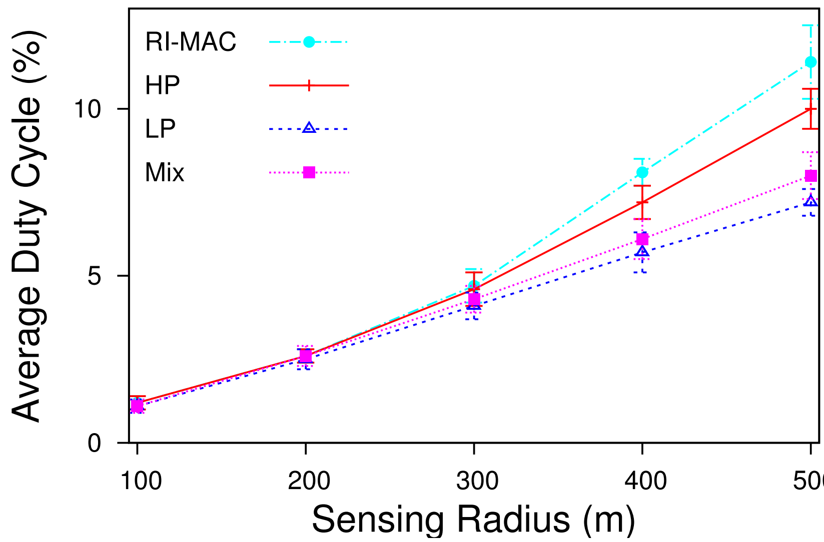

Figure 6.7: Average duty cycle in multi-hop scenario

energy than RI-MAC's, even when they transmit the high priority packets. The reason is in the multi-hop network, a packet normally needs to traverse several hops to reach the sink. Hence it has chance to be concatenated with others during the forwarding path. Moreover in the case of the low priority traffic or the Mix traffic, the amount of energy consumption is smaller since the packets have more chance to be concatenated in super packets, and more chance for the super packets are sent in sequences.

The values of average latency are shown in Fig. 5.8. AQ-MAC with the traffic type HP, Mix and RI-MAC have smaller latency than the protocol with the LP type. In all cases, the HP traffic always has the best latency performance as expected. When the traffic load is light (sensing range is at 100 meters), the values of AQ-MAC with the Mix and LP traffic are similar. But under the high traffic load, the trigger of transmission is more frequent in the network with the traffic type Mix. That is the reason why the Mix's latency is smaller than the LP's. The concatenation scheme also has effect on latency, especially when the sensing range is 400, 500 meters. It can be recognized by comparing AQ-MAC's value under the HP traffic with RI-MAC's. In this case, AQ-MAC has a smaller latency than RI-MAC. When the traffic varies both in types and in load, AQ-MAC shows the good performance. The values

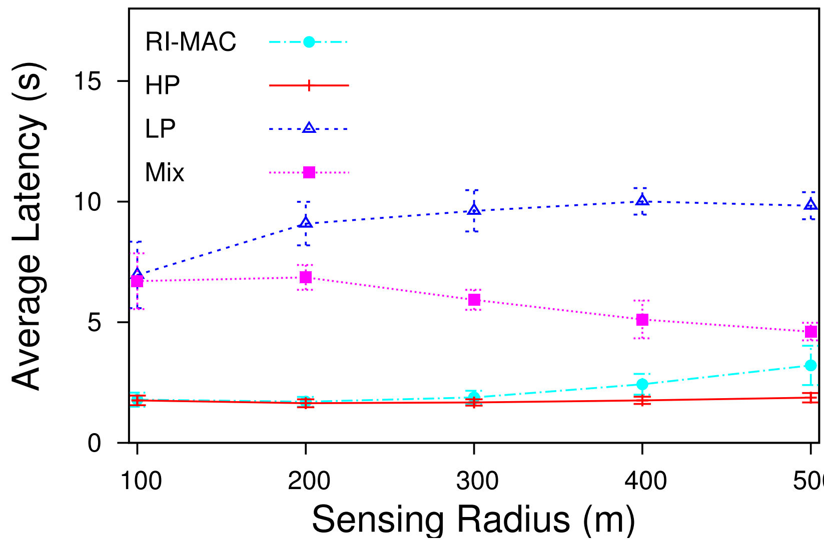

Figure 6.8: Average latency in multi-hop scenario

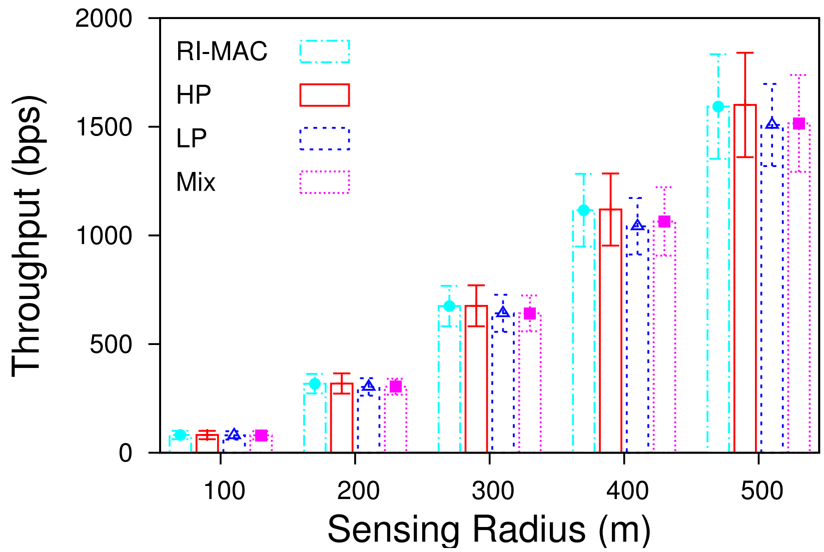

Figure 6.9: Throughput at the sink in multi-hop scenario

of throughput at the sink node are shown in Fig. 5.9. The throughput generally increases when the traffic load increases. But at a same value of the sensing range, there is a slightly difference between the values of LP, HP and Mix traffic. That is because we calculated the throughput at the time when the final packet was successfully received. However the difference is small since only the latency of few packets which are generated by the last event can effect the values of throughput.

6.4 Summary

In this chapter we have proposed AQ-MAC, an energy-efficient asynchronous MAC protocol with QoS awareness for wireless sensor network. AQ-MAC adopts the receiver-initiated transmission and provides QoS service per packet according to the priority imposed. AQ-MAC's data packets are simply classified into low and high priority. The packets which has low priority are kept in a node's queue until a timeout value or a more important packet arrives. The queued packets are often concatenated into a bigger one before sending out in a burst. By queuing the low priority packets, AQ-MAC significantly saves the energy consumption caused by idle listening at senders. Moreover by using the concatenation scheme, AQ-MAC notably improves the latency performance as well as reduces control overhead. The protocol has been evaluated in multiple scenarios under different types of traffic. The results reveal that AQ-MAC adapts well with the variation of traffic, achieves QoS efficiently while keeps good performances of energy efficiency and delivery latency.

Chapter 7

Conclusion and Future Research

7.1 Summary of Contributions

In the dissertation three new efficient MAC protocols have been proposed. The protocols contains several efficient techniques, that can be usable in other MAC protocols. We summarize the techniques below.

Minimizing control overhead for multi-hop MAC protocols: We have minimized control overhead by extensively utilizing the broadcast nature of wireless channel. In the utilization, a single packet can serve as many roles (e.g., a data packet can be used as acknowledge function), then the energy consumed by control packets is reduced.

Two efficient traffic adaptive methods: We have proposed two adaptive methods, whose task is adjusting the length of the nodes' active period in an operational cycle according to the traffic load. The modified sync packet in the first method and the carrier sensing in the second one are used as binary signals, which let the nodes know the traffic status of the networks. The nodes then keep their radio on to receive/transmit data; otherwise, they turn their radio off to save energy.

Concatenation scheme: We have proposed a concatenation scheme, which can concatenates several packets into a super packet before sending out of a node. The motivations of the scheme are: (1) the data packets are always destine to a same sink, (2) the size of a data packet is small. The scheme not only improves the multi-hop MAC's throughput but also significantly reduces control overhead. Moreover, the scheme can be applied in any duty cycling MAC protocol.

Theoretical analysis of the multi-hop protocols' performance: We have theoretically analyzed multi-hop MAC protocols to achieve minimum delay and guarantee collision free in data transmissions. The analysis results also reveal bounds of delivery latency in multi-hop transmissions.

QoS provisioning: We have introduced an approach to support QoS in asynchronous receiver-initiated MAC protocols. By prioritizing packets and providing different transmission strategies, the protocol adapts well with different types of traffic, and achieves energy efficiency, i.e., reducing idling energy at senders under low priority traffic.

7.2 Discussion

The proposed protocols achieve good performances in not only energy saving but also other parameters, e.g., latency, throughput and QoS provision depending on applications. We now discuss the advantages and disadvantages of the techniques as well as the proposed protocols.

LO-MAC in Chapter 3 is most suitable with low data rate applications, in which an sensing event is rarely happened. In the such applications, network lifetime and end-to-end latency but not throughput are the major concerns. In LO-MAC, the concerned parameters are guaranteed by utilizing multi-hop forwarding, duty cycling mechanism, and exploiting carrier sensing, broadcast properties. However, the protocol only supports sending only one data packet at a node in a duty cycle. This is because the scheduling function lets all communication starts at the beginning of Sleep period. Moreover, the wrong detected results of carrier sensing technique may degrade the protocol's performance. Fortunately, there is only possible case, in which the technique reports busy in stead of idle channel. Therefore, it is better if false positive probability of the detection is introduced. Beside that, the process of handling data a packet and recognizing as an acknowledgement can be done in another way: recognizing Start Frame Delimiter of the data packet.

MAC^2 in Chapter 4 can efficiently work in dynamic traffic load applications. By using duty cycling, aggregation scheme and multi-hop forwarding, the protocol aims to satisfy basic requirements of the applications such as throughput, latency, energy efficiency and delivery ratio. The main disadvantage of the protocol is the aggregation scheme may degrade the performance in noisy environment. A lightweight error correction or recovery mechanism is definitely needed, however the size of super packet is still small so the trade off of complexity should be considered. Moreover, MAC^2 and LO-MAC are based on a perfect synchronization process, to make them more realistic clock drifts are necessary considered.

AQ-MAC in Chapter 5 follows the approach of using asynchronous duty cycling, which can avoid the overhead of synchronization. Moreover, the receiver-initiated manner and the concatenation scheme are combined to achieve both energy efficiency and QoS provisioning. The protocol is supposed to work in a dynamic traffic environment, which contains several types of data with different QoS requirements. In the protocol, the process of choosing value of timeout is still heuristic, a possible extension is calculating the value according to some traffic prediction model. However, in realistic applications the traffic is mostly unpredictable, then we observe that value is a configurable parameter. The value can be preset or changed during operation processes by a network administrator.

7.3 Future Work

In this section, we briefly discuss our future research based on the dissertation. We outline here two main directions: implementing the proposed protocols on real motes; and designing new protocols to meet requirements of a realistic application, i.e., disaster management.

7.3.1 Implementing the Proposed Protocols on Real Motes

So far, most of our work has been done on the network simulator ns-2. To qualify and quantity the performance of the proposed protocols, we plan to implement the proposed protocols on hardware using TinyOS [104]. The hardwares, which are selected in this work, are state-of-the-are sensor nodes from the MEMSIC company, including Micaz motes, Iris motes. The basic unit of the protocols are built upon the Unified Power Management Architecture framework [67]. The experimental parameters are going to be changed to meet the conditions of hardware, for example, the length of synchronization period will be longer to guarantee clock drift. However, the algorithms embedded in the protocols are kept.

We also plan to construct a real test bed for multi-hop networks where the proposed protocols are installed on real sensor motes. The protocols is willing to run in a long term to get benchmarks of performances. In addition, we expect to collect all sensing data from the effect of surrounding environment, the data then will be carefully analyzed.

7.3.2 Sensor Network for Disaster Management

Living in Japan during the Great Tohoku Earthquake and Tsunami [105], as well as, experiencing the Fukushima Nuclear Disaster motivate us to bring the advantages of sensor networking to the disaster management applications. The capability of accurate sensing and the wireless communication of sensors show very great potentials. However, in such disaster most of state-of-the-art sensor networks protocols significantly reduce or even lose their performances. The first reason is the communication medium had been changed, i.e., the natural air was changed to the water when the tsunami came. The second one is the surrounding environment of sensor nodes became extremely hostile, i.e., around the Fukushima Daiichi.

The objective of this work is to propose efficient communication protocols for wireless sensor networks under various severe conditions of the disasters. Going into more abstract level, the proposal will address a number of up-to-date technological issues and propose novel protocols to make sensor networks first enable a more convenient way of early waning message delivery. Secondly, the sensor networks keep, store, process the sensing data and provide systems which are able to learn about the phenomena of natural disasters. To achieve these goals, the wireless sensor networks should efficiently fulfill a number of constraints not only in communication but also in data processing and sensing part. While up-to-date technologies have been proven to have capability in physical sensing with high resolution, exact/instant processing huge amount of data, the wireless communication still raises a number of unsolved problems. The proposal faces these problems in a comprehensive way, by addressing the following aspects of communication protocols.

Robustness: The protocols should keep a good performance in any condition of surrounding environments, for example, the environment is changed from natural air to water when a tsunami comes.

Performance improvement: The design goals of the new protocols include not only energy efficiency but also other parameters, e.g., latency, throughput, and mobility.

Adapting to traffic fluctuation: In the disasters, traffic may fluctuate by time and space in a wide range of amplitude. We shall design new protocols, which can adapt well with that specific traffic fluctuation.

Security Issues: The new proposed protocols will consider the security issue to avoid malicious behaviours in the network.

Working well with other technologies: The new protocols are expected to work peacefully with different protocols and even different technologies.

Appendix

List of Publications

Referred Journals

1. **Kien Nguyen**, Ulrich Meis, and Yusheng Ji, *"MAC^2: A Multi-hop Adaptive MAC Protocol with Packet Concatenation for Sensor Networks,"* IEICE Special Section: Architectures, Protocols, and Applications for the Future Internet, Vol.E95-D, No.2, pp. 480-489 (2012).

2. Jumpot Phuritatkul, **Kien Nguyen**, Michihiro Koibuchi, Yusheng Ji, Kensuke Fukuda, Shunji Abe, Jun Matsukata, Shigeo Urushidani, and Shigeki Yamada,*"Impact of QoS Operations on a Experimental Testbed Network,"* Simulation Modelling Practice and Theory, Vol. 17, Issue 3, pp. 528-537, Wiley InterScience (2009).

Referred Proceedings

1. **Kien Nguyen** and Yusheng Ji, *"Asynchronous MAC Protocol with QoS Awareness in Wireless Sensor Networks,"* **to be appeared** in Proceedings of IEEE GLOBECOM 2012, (2012).

2. **Kien Nguyen**, Ulrich Meis, and Yusheng Ji, *"An Energy Efficient, High Throughput MAC Protocol Using Packet Aggregation,"* in Proceedings of the second Workshop on Ubiquitous Computing and Networks (UbiCoNet 2011) in conjunction with IEEE GLOBECOM 2011, pp. 1236-1240 (2011).

3. **Kien Nguyen** and Yusheng Ji, *"Impact of contention on performance of flows in multi-hop MAC protocol for sensor networks,"* in Proceedings of the ACM 2nd International Symposium on Information and Communication Technology (SoICT 2011), pp. 42-46 (2011).

4. **Kien Nguyen** and Yusheng Ji, *"Achieving minimum latency in multi-hop MAC protocol for Wireless Sensor Networks,"* in Proceedings of IEEE 73rd Vehicular Technology Conference: VTC Spring 2011, pp. 1-5 (2011).

5. **Kien Nguyen** and Yusheng Ji, *"Using carrier sensing to improve energy efficiency of MAC protocol in sensor networks,"* in Proceedings of the IEEE International Symposium on Communications and Information Technologies (ISCIT10), pp. 70-74(2010).

6. **Kien Nguyen** and Yusheng Ji, *"AM-MAC: An Energy Efficient, Adaptive Multi-hop MAC Protocol for Sensor Networks,"* in Proceedings of the ACM 6th International Wireless Communications and Mobile Computing Conference (IWCMC10), pp. 432-436(2010).

7. **Kien Nguyen** and Yusheng Ji, *"An adaptive energy-efficient multi-hop MAC protocol for sensor networks,"* in Proceedings of APAN Network Research Workshop 2009, pp. 91-95 (2009).

8. **Kien Nguyen** and Yusheng Ji, *"LCO-MAC: A low latency, low control overhead MAC protocol for Wireless Sensor Networks,"* in Proceedings of IEEE 4th International Conference on Mobile Ad-hoc and Sensor Networks (MSN'08) (2008).

9. Jumpot Phuritatkul, **Kien Nguyen**, Michihiro Koibuchi, Yusheng Ji, Kensuke Fukuda, Shunji Abe, Jun Matsukata, Shigeo Urushidani, and Shigeki Yamada, *"Investigating QoS Performance on a Testbed Network,"* in Proceedings of International Workshop on Performance Modeling and Evaluation in Computer and Telec ommunication Network (PMECT07) (2007).

Non-referred Proceedings

1. **Kien Nguyen** and Yusheng Ji, *"Toward QoS Provisioning in Energy Efficient MAC Protocols for Sensor Networks,"* in Proceedings of IEICE Society Conference 2012, (2012).

2. **Kien Nguyen** and Yusheng Ji, *"Multi-hop MAC protocol for Energy Harvesting Sensor Networks,"* National Institute of Informatics Open house 2012, (2012).

3. **Kien Nguyen** and Yusheng Ji, *"Improving Flow Capacity of Multi-Hop MAC Protocol for Sensor Networks,"* in Proceedings of IEICE General Conference 2012, BS-3-37 (2012).

4. **Kien Nguyen**, Ulrich Meis, and Yusheng Ji, *"Improving the performance of duty-cycling MAC protocol using packet aggregation,"*, in Proceedings of IEICE Society Conference 2011, BS-6-6 (2011).

5. **Kien Nguyen** and Yusheng Ji, *"A traffic adaptive method for MAC protocol in Sensor Networks,"* in Proceedings of IEICE Society Conference 2010, BS-7-19 (2010).

6. **Kien Nguyen** and Yusheng Ji, *"A reseach on MAC protocols for WSNs,"* National Insitute of Informatics Open house 2010, (2010).

7. **Kien Nguyen** and Yusheng Ji, *"Improving the latency and energy efficiency in MAC protocol for wireless sensor networks,"* in Proceedings of IEICE Society Conference 2008, BS-12-6 (2008).

8. **Kien Nguyen** and Yusheng Ji, *"LO-MAC: low overhead MAC protocol for WSNs,"* JSPS Summer Program Poster Presentation 2008, (2008)

9. **Kien Nguyen** and Yusheng Ji, *"QoS on experiment testbed on SINET3,"* ICT Asia Seminar 2007, (2007).

References

[1] K. Langendoen, *Medium Access Control in Wireless Sensor Networks.* Nova Science Publishers, Inc., May 2008.

[2] J. Beutel, "Metrics for Sensor Network Platforms," in *Proc. ACM Workshop on Real-World Wireless Sensor Networks 2006*, June 2006.

[3] "The memsic company http://www.memsic.com."

[4] J. L. Hill and D. E. Culler, "Mica: A Wireless Platform for Deeply Embedded Networks," *IEEE Micro*, vol. 22, pp. 12–24, Nov 2002.

[5] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proc. ACM/IEEE 4th Intl. Symposium on Information Processing in Sensor Networks*, pp. 364 – 369, April 2005.

[6] P. Levis, S. Madden, J. Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An operating system for sensor networks," *Ambient Intelligence*, 2004.

[7] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC language: A holistic approach to networked embedded systems," in *Proc. ACM SIGPLAN Programming Language Design and Implementation*, PLDI '03, pp. 1–11, 2003.

[8] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, pp. 2787–2805, Oct 2010.

[9] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's INTRAnet of things to a future INTERnet of things: a wireless and mobility-related view," *IEEE Trans. on Wireless Communications*, vol. 17, pp. 44 –51, December 2010.

[10] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," in *Proc. ACM 10th Intel. Conf. on Architectural Support for Programming Languages and Operating systems (ASPLOS-X 2002)*, pp. 96–107, Oct 2002.

[11] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. 1st ACM International Workshop On Wireless Sensor Networks And Applications*, WSNA '02, pp. 88–97, 2002.

[12] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking," *Elsevier Computer Networks*, vol. 46, pp. 605–634, 2004.

[13] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications: remote large-scale environments," in *Proc. IEEE 28th Conf. on Military communications*, MILCOM'09, pp. 911–917, 2009.

[14] A. Ledeczi, A. Nadas, P. Volgyesi, G. Balogh, B. Kusy, J. Sallai, G. Pap, S. Dora, K. Molnar, M. Maroti, and G. Simon, "Countersniper system for urban warfare," *TOSN*, vol. 1, no. 2, pp. 153–177, 2005.

[15] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[16] D. Pompili, T. Melodia, and I. F. Akyildiz, "A CDMA-based Medium Access Control for UnderWater Acoustic Sensor Networks," *IEEE Trans. on Wireless Communications*, vol. 8, no. 4, pp. 1899–1909, 2009.

[17] K. Romer and F. Mattern, "The design space of wireless sensor networks," *IEEE Trans. on Wireless Communications*, vol. 11, pp. 54–61, Dec 2004.

[18] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proc. 6th Intl. Symposium on Information Processing in Sensor Networks*, IPSN 2007, pp. 254–263, April 2007.

[19] G. Hackmann, F. Sun, N. Castaneda, C. Lu, and S. Dyke, "A holistic approach to decentralized structural damage localization using wireless sensor networks," *Computer Communications*, vol. 36, no. 1, pp. 29 – 41, 2012.

[20] M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon, "Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment," in *Proc. 8th Intl. Conf. on Information Processing in Sensor Networks*, IPSN '09, pp. 277–288, IEEE Computer Society, 2009.

[21] K. Chebrolu, B. Raman, N. Mishra, P. K. Valiveti, and R. Kumar, "Brimon: a sensor network system for railway bridge monitoring," in *Proc. ACM 6th Intl. Conf. on Mobile systems, applications, and services*, MobiSys '08, pp. 2–14, 2008.

[22] M. Wang, J. Cao, M. Liu, B. Chen, Y. Xu, and J. Li, "Design and implementation of distributed algorithms for wsn based structural health monitoring," *Int. J. Sen. Netw.*, vol. 5, pp. 11–21, Feb 2009.

[23] J. Taneja, R. Katz, and D. Culler, "Defining CPS Challenges in a Sustainable Electricity Grid," in *Proc. IEEE/ACM 3rd International Conference on Cyber-Physical Systems*, ICCPS '12, pp. 119–128, 2012.

[24] V. Gungor, B. Lu, and G. Hancke, "Opportunities and Challenges of Wireless Sensor Networks in Smart Grid," *IEEE Trans. on Industrial Electronics*, vol. 57, pp. 3557–3564, Oct. 2010.

[25] Z. Fadlullah, M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward intelligent machine-to-machine communications in smart grid," *IEEE Communications Magazine*, vol. 49, pp. 60 –65, April 2011.

[26] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proc. 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, BuildSys '10, pp. 1–6, 2010.

[27] K. Randy, C. David, S. Seth, A. Sara, C. Yanpei, D.-H. Stephen, D. Prabal, H. Mike, J. Xiaofan, K. Laura, K. Andrew, L. Ken, O. Jorge, M. Prashanth, R. Evan, T. Jay, H. Jeff, and S. Sushant, "An information-centric energy infrastructure: The berkeley view," *Journal of Sustainable Computing: Informatics and Systems*, vol. 1, pp. 7–22, March 2011.

[28] A. Ilic, T. Staake, and E. Fleisch, "Using Sensor Information to Reduce the Carbon Footprint of Perishable Goods," *IEEE Trans. on Pervasive Computing*, vol. 8, pp. 22–29, Jan 2009.

[29] N. Wang, N. Zhang, and M. Wang, "Wireless sensors in agriculture and food industry–Recent development and future perspective," *Computers and Electronics in Agriculture*, vol. 50, pp. 1–14, Jan. 2006.

[30] K. Langendoen, A. Baggio, and O. Visser, "Murphy Loves Potatoes: Experiences from a Pilot Sensor Network Deployment in Precision Agriculture," in *Proc. 14th Int. Workshop on Parallel and Distributed Real-Time Systems (WPDRTS)*, pp. 1–8, apr 2006.

[31] T. Wark, P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley, "Transforming Agriculture through Pervasive Wireless Sensor Networks," *IEEE Trans. on Pervasive Computing*, vol. 6, pp. 50–57, April-June 2007.

[32] J. M. Lees, J. B. Johnson, M. Ruiz, L. Troncoso, and M. Welsh, "Reventador Volcano 2005: Eruptive activity inferred from seismo-acoustic observation," *Journal of Volcanology and Geothermal Research*, vol. 176, pp. 179–190, Sep 2008.

[33] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. LaHusen, "Air-dropped sensor network for real-time high-fidelity volcano monitoring," in *Proc. ACM 7th Intl. Conf. on Mobile systems, applications, and services*, MobiSys '09, pp. 305–318, 2009.

[34] K. Sha, W. Shi, and O. Watkins, "Using Wireless Sensor Networks for Fire Rescue Applications: Requirements and Challenges," in *Proc. IEEE Intl. Conf. on Electro/information Technology 2006*, pp. 239 –244, may 2006.

[35] S. Wen-Zhan, H. Renjie, M. Xu, B. Shirazi, and R. LaHusen, "Design and Deployment of Sensor Network for Real-Time High-Fidelity Volcano Monitoring," *IEEE Trans. on Parallel and Distributed Systems*, vol. 21, pp. 1658 –1674, Nov. 2010.

[36] T. Rui, X. Guoliang, C. Jinzhu, S. Wen-Zhan, and H. Renjie, "Quality-Driven Volcanic Earthquake Detection Using Wireless Sensor Networks," in *Proc. 31st IEEE Conf. Real-Time Systems Symposium*, RTSS 2010, pp. 271–280, Dec 2010.

[37] Y. Hao and R. Foster, "Wireless body sensor networks for health-monitoring applications," *Physiological Measurement*, vol. 29, pp. 27–56, Nov. 2008.

[38] S. Ullah, H. Higgins, B. Braem, B. Latre, C. Blondia, I. Moerman, S. Saleem, Z. Rahman, and K. S. Kwak, "A Comprehensive Survey of Wireless Body Area Networks," *Journal of Medical Systems*, vol. 36, pp. 1065–1094, Jun 2012.

[39] L. Huaming and J. Tan, "Heartbeat-Driven Medium-Access Control for Body Sensor Networks," *IEEE Trans. on Information Technology in Biomedicine*, vol. 14, pp. 44–51, Jan. 2010.

[40] Z. Gang, L. Jian, W. Chieh-Yih, M. Yarvis, and J. Stankovic, "Bodyqos: Adaptive and radio-agnostic qos for body sensor networks," in *Proc. 27th IEEE Conference on Computer Communications*, INFOCOM 2008, pp. 565 –573, April 2008.

[41] M. Hanson, H. Powell, A. Barth, K. Ringgenberg, B. Calhoun, J. Aylor, and J. Lach, "Body area sensor networks: Challenges and opportunities," *IEEE Computer*, vol. 42, pp. 58–65, jan 2009.

[42] H. Alemdar and E. Cem, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.

[43] I. Demirkol, C. Ersoy, and F. Alagoz, "Mac protocols for wireless sensor networks: a survey," *IEEE Trans. on Communications Magazine*, vol. 44, no. 4, pp. 115–121, 2006.

[44] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proc. 21st IEEE INFOCOM*, pp. 1567–1576, 2002.

[45] B. Yahya and J. Ben-Othman, "Towards a classification of energy aware MAC protocols for wireless sensor networks," *Wireless Communications and Mobile Computing*, 2009.

[46] A. Boukerche and S. Samarah, "A Novel Algorithm for Mining Association Rules in Wireless Ad Hoc Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 7, pp. 865–877, 2008.

[47] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 493–506, 2004.

[48] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J.-C. Chen, "A Survey of Energy Efficient Network Protocols for Wireless Networks," *Wireless Networks*, vol. 7, pp. 343–358, 2001.

[49] P. Lettieri and M. B. Srivastava, "Advances in Wireless Terminals," *IEEE Personal Communications*, vol. 6, pp. 6–19, 1999.

[50] P. Havinga and G. Smit, "Energy efficient wireless atm design," 1999.

[51] M. Matthew and V. Nitin, "Minimizing energy consumption in sensor networks using a wakeup radio," in *Proc. IEEE International Conference on Wireless Communications and Networks (WCNC 2004)*, (Sweden), March 2004.

[52] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. 20th IEEE INFOCOM*, pp. 1548–1557, 2001.

[53] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC Essentials for Wireless Sensor Networks," *IEEE Trans. on Communications Surveys Tutorials*, vol. 12, pp. 222–248, quarter 2010.

[54] D. Messaoud, D. Djamel, and B. Nadjib, "Survey on Latency Issues of Asynchronous MAC Protocols in Delay-Sensitive Wireless Sensor Networks," *IEEE Trans. on Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1 –23, 2012.

[55] P. Huang, L. Xiao, S. Soltani, M. Mutka, and N. Xi, "The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey," *IEEE Trans. on Communications Surveys Tutorials*, no. 99, pp. 1–20, 2012.

[56] K. Kredo, II and P. Mohapatra, "Medium access control in wireless sensor networks," *Comput. Netw.*, vol. 51, no. 4, pp. 961–994, 2007.

[57] L. v. Hoesel and P. Havinga, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks," in *Proc. INSS04*, June 2004.

[58] S. Chatterjea, L. van Hoesel, and P. Havinga, "AI-LMAC: an adaptive, information-centric and lightweight MAC protocol for wireless sensor networks," in *Proc. Intelligent Sensors, Sensor Networks and Information Processing Conference*, ISSNIP04, pp. 381 – 388, dec 2004.

[59] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," in *Proc. 1st ACM SenSys*, pp. 181–192, NOV 2003.

[60] G. Halkes and K. Langendoen, "Crankshaft: An Energy-Efficient MAC-Protocol For Dense Wireless Sensor Networks," in *Proc. European Conference on Wireless Sensor Networks 2007 EWSN07*, Jan 2007.

[61] "IEEE Standard 802.15.4 Medium Access Control (MAC) and Physical Layer (PHY) Specifications." IEEE Standard 802.15.4, 2003.

[62] W.-Z. Song, R. Huang, B. Shirazi, and R. LaHusen, "TreeMAC: Localized TDMA MAC protocol for real-time high-data-rate sensor networks," *Pervasive Mob. Comput.*, vol. 5, pp. 750–765, Dec 2009.

[63] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM 2nd SenSys*, pp. 95–107, 2004.

[64] A. El-Hoiydi, "Spatial TDMA and CSMA with Preamble Sampling for Low Power Ad Hoc Wireless Sensor Networks," in *Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)*, (Washington, DC, USA), p. 685, IEEE Computer Society, 2002.

[65] A. El-Hoiydi and J. D. Decotignie, "WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," in *Proc. 9th International Symposium on Computers and Communications (ISCC 2004)*, vol. 1, pp. 244–251, 2004.

[66] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. 4th ACM SenSys*, pp. 307–320, 2006.

[67] K. Klues, G. Hackmann, O. Chipara, and C. Lu, "A Component-Based Architecture for Power-Efficient Media Access Control in Wireless Sensor Networks," in *Proc. 5th ACM Sensys* , 2007.

[68] S. Liu, K.-W. Fan, and P. Sinha, "Cmac: An energy-efficient mac layer protocol using convergent packet forwarding for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 5, pp. 2901–2934, Nov 2009.

[69] S. Mo, X. Guoliang, Z. Gang, L. Shucheng, and W. Xiaorui, "C-mac: Model-driven concurrent medium access control for wireless sensor networks," in *Proc. IEEE INFOCOM 2009*, pp. 1845–1853, 2009.

[70] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proc. 6th ACM SenSys*, pp. 1–14, 2008.

[71] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "Pw-mac: An energy-efficient prediction-wakeup mac protocol for wireless sensor networks," in *Proc. 30th IEEE INFOCOM*, pp. 1305–1313, April 2011.

[72] Y. Peng, Z. Li, D. Qiao, and W. Zhang, "Delay-bounded mac with minimal idle listening for sensor networks," in *Proc. 30th IEEE INFOCOM*, pp. 1314–1322, April 2011.

[73] T. van Dam and K. Langendoen, "An adaptive energy efficient and low-latency mac for data gathering in wireless sensor networks," in *Proc. 1st ACM SenSys*, pp. 171–180, 2003.

[74] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks," in *Proc. 18th IEEE International Parallel and Distributed Processing Symposium*, IPDPS'04, pp. 224–231, 2004.

[75] A. Rowe, V. Gupta, and R. R. Rajkumar, "Low-power clock synchronization using electromagnetic energy radiating from ac power lines," in *Proc. 7th ACM SenSys*, pp. 211–224, 2009.

[76] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proc. 9th ACM MobiHoc*, pp. 53–62, 2008.

[77] S. Du, A. Saha, and D. Johnson, "RMAC: a Routing-Enhanced Duty-Cycle MAC protocol for wireless sensor networks," in *Proc. 26th IEEE INFOCOM*, pp. 1478–1486, 2007.

[78] K. Nguyen and Y. Ji, "Using carrier sensing to improve energy efficiency of mac protocol in sensor networks," in *Proc. 10th IEEE International Symposium on Communications and Information Technologies*, ISCIT '10, pp. 432–436, 2010.

[79] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications." IEEE Standard 802.11, June 1999.

[80] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. 1st ACM SenSys*, pp. 1–13, 2003.

[81] "The network simulator – ns2 http://www.isi.edu/ns/index.html."

[82] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," in *Proc. 4th ACM SenSys*, pp. 321–334, 2006.

[83] V. Shnayder, M. Hempstead, B. rong Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. 2nd ACM SenSys*, pp. 188–200, 2004.

[84] K. Nguyen and Y. Ji, "Achieving minimum latency in multi-hop mac protocol for wireless sensor networks," in *Proc. IEEE 73rd Vehicular Technology Conference*, VTC Spring '11, pp. 1–5, May 2011.

[85] M. Ali, U. Saif, A. Dunkels, T. Voigt, K. Römer, K. Langendoen, J. Polastre, and Z. Uzmi, "Medium Access Control Issues in Sensor Networks," *ACM SIGCOMM Computer Communication Review*, vol. 36, pp. 33–36, April 2006.

[86] B. Yahya and J. Ben-Othman, "Towards a classification of energy aware mac protocols for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 9, pp. 1572–1607, 2009.

[87] K. Nguyen and Y. Ji, "AM-MAC: an energy efficient, adaptive multi-hop mac protocol for sensor networks," in *Proc. 6th ACM International Wireless Communications and Mobile Computing Conference*, IWCMC '10, pp. 432–436, 2010.

[88] "E.800: Terms and definitions related to quality of service and network performance including dependability," August 1994.

[89] O. Tsigkas and F.-N. Pavudou, "Providing QOS support at the distributed wireless MAC layer: a comprehensive study," *IEEE Tran. on Wireless Communications*, vol. 15, pp. 22–31, Feb. 2008.

[90] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," 1994.

[91] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Service," 1998.

[92] W. Pattara-atikom, P. Krishnamurthy, and S. Banerjee, "Distributed mechanisms for Quality of Service in wireless LANs," *IEEE Trans. on Wireless Communications*, vol. 10, pp. 26–34, 2003.

[93] D. Chen and P. K. Varshney, "QoS Support in Wireless Sensor Networks: A Survey," in *International Conference on Wireless Networks'04*, pp. 227–233, 2004.

[94] Y. Xiao, H. Chen, K. Wu, C. Liu, and B. Sun, "Maximizing Network Lifetime under QoS Constraints in Wireless Sensor Networks," in *Proc. GLOBECOM 2006*, 2006.

[95] T. B. Reddy, I. Karthigeyan, B. S. Manoj, and C. S. R. Murthy, "Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions," *Ad Hoc Networks*, vol. 4, no. 1, pp. 83–124, 2006.

[96] "MTS420/400 Environmental SensorBoard."

[97] J. Ben-othman, L. Mokdad, and B. Yahya, "An Energy Efficient Priority-Based QoS MAC Protocol for Wireless Sensor Networks," in *Proc. IEEE ICC 2011*, pp. 1–6, 2011.

[98] Y. Liu, I. Elhanany, and H. Qi, "An Energy-Efficient Qos-Aware Media Access Control Protocol for Wireless Sensor Networks," in *Proc. 2nd IEEE MASS*, pp. 189–191, 2005.

[99] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: a Hybrid MAC for wireless sensor networks," in *Proc. 3rd ACM SenSys*, pp. 90–101, 2005.

[100] K. Nguyen, U. Meis, and Y. Ji, "MAC^2: a Multi-hop Adaptive Mac Protocol with Packet Concatenation," *IEICE Transaction on Infomation and Systems*, vol. E95-D, no. 2, pp. 480–489, 2012.

[101] M. A. Yigitel, Ö. D. Incel, and C. Ersoy, "QoS-aware MAC Protocols for Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 55, no. 8, pp. 1982–2004, 2011.

[102] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," in *Proc. 8th ACM SenSys*, pp. 1–14, 2010.

[103] "CC2420 Datasheet http://www.ti.com."

[104] "Tinyos 's website http://www.tinyos.net/."

[105] "http://en.wikipedia.org/wiki/2011_tohoku_earthquake_and_tsunami."