AN ENERGY EFFICIENT MEDIUM ACCESS CONTROL PROTOCOL FOR WIRELESS SENSOR NETWORKS

A DISSERTATION

Submitted in partial fulfillment of the
requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
ELECTRONICS AND COMMUNICATION ENGINEERING
(With Specialization in Communication Systems)

By

GAURAV

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2008

CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this dissertation report entitled, **"AN ENERGY EFFICIENT MEDIUM ACCESS CONTROL PROTOCOL FOR WIRELESS SENSOR NETOWRKS"** towards the partial fulfillment of the requirements for the award of the degree of **Master of Technology** with specialization in **Communication Systems**, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee (India) is an authentic record of my own work carried out during the period from July 2007 to June 2008, under the guidance of **Mr. S. CHAKRAVORTY, Assistant Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee** and **Mr. D.A. ROY, Scientific Officer-G, RCnD, BARC.**

I have not submitted the matter embodied in this dissertation for the award of any other Degree or Diploma.

Date: 30-06-2008

Place: Roorkee

Gaurav
GAURAV

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 30/6/08

Place: Roorkee

S. Chakravorty

Mr. S. CHAKRAVORTY
Assistant Professor, E&C Department,
IIT Roorkee,
Roorkee – 247 667 (India).

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my advisor, **Mr. S. Chakravorty**, for his realistic, encouraging, critical and constructive approach throughout my masters study and his efforts during supervision of my dissertation. I would like to thank **Dr. D. K. Mehra** (Head of Department) and all faculty members for their encouragement and constant motivation throughout the course.

I feel fortunate to work with my colleagues. I would like to express my appreciation to Khandu and Jitu for their understanding and support during my academic studies I would again like to thank Khandu and Naveen for their night outs for my dissertation write-up. I would like to thank Arun and Sumit sir for their valuable tea-time advice.

Finally, I would like to express my thanks to my parents, Rajendra Prasad and Sunita Rani, my sisters, Megha and Himani, for their love, trust, understanding and every kind of support not only throughout my dissertation but also throughout my life.

GAURAV

ABSTRACT

Energy is the most critical resource in the life of a wireless sensor node. Therefore, its usage must be optimized to maximize the network life. To increase the life time, sensors should stay in energy saving sleep mode as long as possible, because in sleep mode the radio is either shut down or working with less energy. "Sampling" based schemes improve energy efficiency significantly for low traffic. The performance of sampling based protocols deteriorates at medium and high traffic due to presence of hidden terminals. Elimination of the hidden terminal problem, therefore, would result in improved energy efficiency for medium and high traffic.

In this dissertation work, a new sampling based MAC protocol, Sampling and Scheduling Based MAC (SSMAC), is proposed. In SSMAC, time is divided into time slots during data transmissions. SSMAC minimizes hidden terminals by allowing nodes to transmit in round-robin fashion. The transmitting node initially sends a Wake-Up Frame (WUF) to the destination node to wake it up. The subsequent communication follows CTS/DATA/ACK procedure of 802.11.

SSMAC is simulated in OMNeT++ simulation environment and is compared with WiseMAC, Preamble sampling and SMAC schemes. Simulation results show that SSMAC is most energy efficient in comparison to other schemes. This improvement comes at the cost of latency due to setup time for data transmission.

TABLE OF CONTENTS

List of Figures

Chapter 1

INTRODUCTION

Wireless sensor networks (WSN) are an emerging paradigm posing new challenges for researchers in wireless communication [1]. This new class of networks is very close in behavior to wireless ad-hoc networks. Nevertheless, they have a few unique differences; the principal one is the small size of nodes constituting a wireless sensor network. Although smaller nodes make wireless sensor networks suitable for several existing and emerging applications related to information sensing, this also implies that nodes have limited resources, i.e., CPU speed, memory, battery, and radio interface. Because the nodes are resource constrained, they require network designs that can be customized for different types of application environments, thus placing significant demands on algorithm design, protocol specification, and technologies. Life time of a sensor networks depends on the energy consumed by sensor nodes. Energy consumption can be controlled by proper designing of the Medium access control (MAC). As radio is the main consumer of energy [2], if we control the radio properly so that there is no collisions and overhearing, and overheads are kept low then WSN can be made more energy efficient. So, medium access control is one of the important aspects of WSN which can improve its life time.

1.1. Wireless Sensor Networks

Recent advances in wireless communications and electronics have enabled the development of low cost, low-power, multifunctional sensor nodes that are small in size and communicate in short distances These tiny sensor nodes, which consist of sensing, data processing, and communicating components, leverage the idea of sensor networks. Sensor networks represent a significant improvement over traditional sensors. [3]

A sensor network is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it. The position of sensor nodes need not be engineered or predetermined. This allows random deployment in inaccessible terrains or disaster relief operations. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities. Another unique feature of sensor networks is the cooperative effort of sensor nodes. Sensor nodes are fitted with an onboard processor. Instead of sending the raw data to the other nodes in a route, they use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data. Fig. 1.1 shows the typical structure of WSN. After sensing and translating the aimed phenomenon into a kind of data format from the sensor field, these sensor nodes send their data toward the sink node using the wireless communication system. Sensor nodes have a short transmission range due to their limited radio capabilities. Therefore, the data must be relayed using intermediate nodes towards the sink. In addition, it may be more advantageous to use a multi-hop path to the sink node consisting of shorter links rather than using a single long connection. The sink node forwards the gathered data to the user's computer by the traditional network systems, such as the Ethernet, which provides the specific desired information to the user.

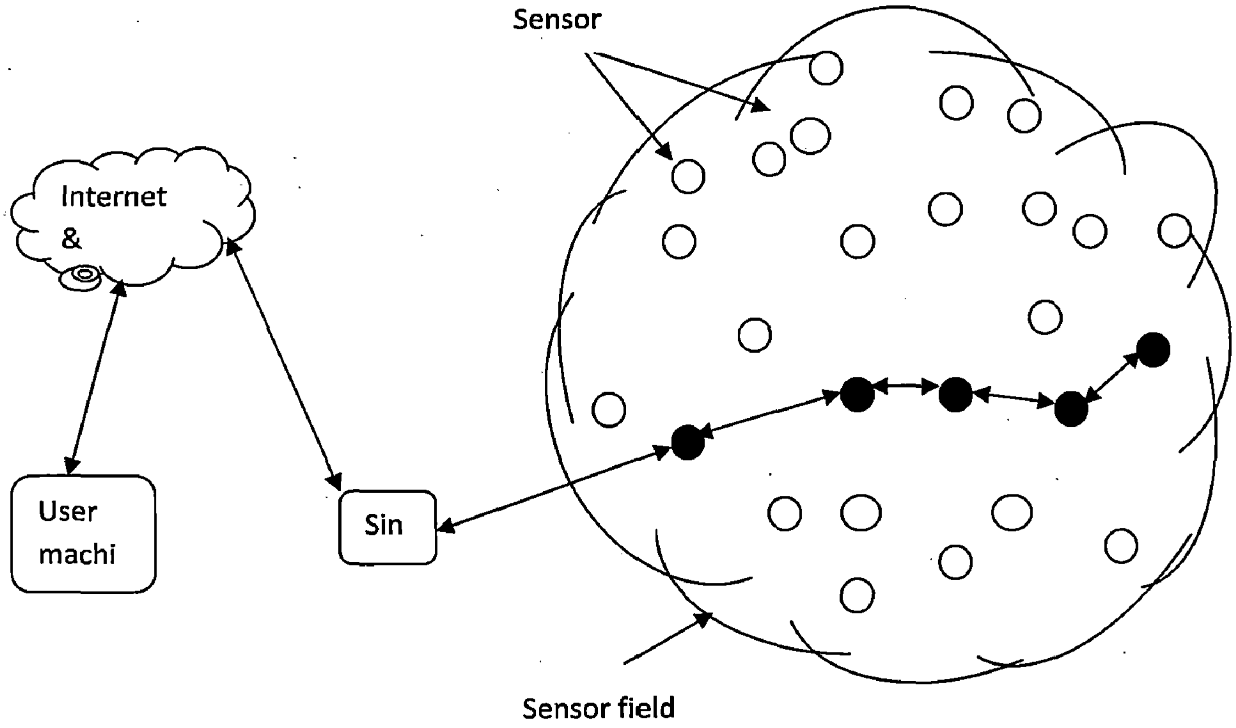

Fig. 1.1: Typical structure of WSN [1]

Wireless sensor network is an application specific and data centric system. The large number of combinations of sensing, processing, and communication require many different and possible application scenarios. And most importantly, the low cost and low energy supply will require, in many application scenarios, redundant deployment of wireless sensor nodes. As a consequence, the most significant concern of any one particular node is considerably reduced as compared to traditional networks. A more important factor is the data that these nodes can observe.

Since these networks have to interact with the environment, their traffic characteristics can be expected to be very different from other, human-driven forms of networks. A typical consequence is that wireless sensor networks are likely to exhibit very low data rates over a large time scale, but can have very heavy traffic when something happens unexpectedly.

1.2. Main Feature of WSN

In ad hoc networks, wireless nodes self-organize into an infrastructureless network with a dynamic topology. Sensor networks share these traits, but also have several distinguishing features. The number of nodes in a typical sensor network is much higher than in a typical ad hoc network, and dense deployments are often desired to ensure coverage and connectivity. For this reason, sensor nodes must be cheap and have stringent energy limitations, which make them more failure-prone. Sensor nodes are generally assumed to be stationary, but their relatively frequent breakdowns and the volatile nature of the wireless channel nonetheless result into a variable network topology. Ideally, sensor network hardware should be power efficient, small, inexpensive, and reliable in order to maximize network lifetime, add flexibility, facilitate data collection and minimize the need for maintenance.

Lifetime (Energy consumption)

Lifetime is extremely critical for most applications, and its primary limiting factor is the energy consumption of the nodes, which need to be self-powering. In sensor networks, energy is consumed mainly for three purposes: data transmission, signal processing, and hardware operation [4]. For most of the applications, data transmission and reception are main source of energy consumption. As highlighted in Pottie and Kaiser [2] that transmission and reception have a big share of energy consumption compared to consumption in processing and hardware operation for a typical WSN platform as MICA.

Medium Access Control (MAC) solutions have a direct impact on energy consumption, as it controls radio transmission and reception. Some of the primary causes of energy waste are found at the MAC layer: collisions, control packet overhead and idle listening. Power-saving forward error control techniques are not easy to implement due to the high amount of computing power that they require.

Responsiveness

A solution to extend network lifetime is to operate the nodes in duty-cycle manner in which nodes periodically switch between sleep and wake-up modes. Although synchronization is the big challenge to adopt this technique but the larger concern is that arbitrary long sleep periods can reduce the responsiveness and effectiveness of sensors. In applications where it is critical that certain events in the environment be detected and reported rapidly, the latency induced by sleep schedules must be kept within strict bounds, even in the presence of network congestion.

Robustness

The large-scale coverage requires the use of large numbers of inexpensive devices. However, inexpensive devices can often be unreliable and prone to failures. Rates of device failure will also be high whenever the sensor devices are deployed in harsh or hostile environments. Protocol designs must therefore have built-in mechanisms to provide robustness. It is important to ensure that the global performance of the system is not sensitive to individual device failures. Further, it is often desirable that the performance of the system degrade as gracefully as possible with respect to component failures.

Scalability

Because a sensor network may contain thousands of nodes, scalability is a critical factor that guarantees that the network performance does not significantly degrade as the network size (or node density) increases.

Self-configuration

Because of their scale and the nature of their applications, wireless sensor networks are inherently *unattended* distributed systems. Autonomous operation of the network is therefore a key design challenge. From the very start, nodes in a wireless sensor network have to be able to configure their own network topology; localize, synchronize, and calibrate themselves; coordinate inter-node communication; and determine other important operating parameters.

1.3. Applications

Wireless sensor networks have many application areas mentioned in the literature [1]. One of the applications of WSN is agriculture and environment monitoring. In precision agriculture, crop and livestock management and precise control of fertilizer concentration can be done. In planetary exploration, exploration and surveillance in inhospitable environments such as remote geographic regions or toxic locations can take place. In geophysical monitoring, seismic activity can be detected at a much finer scale using a network of sensors equipped with accelerometers. The field of hydrochemistry has a compelling need for sensor networks because of the complex spatiotemporal variability in hydrologic, chemical, and ecological parameters and the difficulty of labor-intensive sampling, particularly in remote locations or under adverse conditions. In addition, buoys along the coast could alert surfers, swimmers, and fishermen to dangerous levels of bacteria. Forest fire and floods can be detected early and causes can be localized precisely by densely deployed sensor networks.

In civil engineering, WSN has good applications. WSN can be used for monitoring of structures. Sensors can be placed in bridges to detect and warn of structural weakness and in water reservoirs to spot hazardous materials. The reaction of tall buildings to winds and earthquakes can be studied and material fatigue can be monitored closely. Buildings razed by an earthquake may be infiltrated with sensor robots to locate signs of life.

WSN has many military applications. WSN can be used for asset monitoring and management. Commanders can monitor the status and locations of troops, weapons, and supplies to improve military command, control, communication, and computing. They can be rapidly deployed for surveillance and used to provide battlefield intelligence regarding the location, numbers, movement, and identity of troops and vehicles, and for detection of chemical, biological, and nuclear weapons.

Medical research and healthcare can greatly benefit from sensor networks. Physiological data such as body temperature, blood pressure, and pulse are sensed and automatically transmitted to a computer or physician, where it can be used for health status monitoring and medical exploration. Wireless sensing bandages may warn of infection. Tiny sensors in the blood stream, possibly powered by a weak external electromagnetic field, can continuously analyze the blood and prevent coagulation and thrombosis.

Other general engineering applications are automotive telematics, fingertip accelerometer virtual keyboards, sensing and maintenance in industrial plants, aircraft drag reduction, smart office spaces etc.

A typical example of WSN is designing of WSN for habitat monitoring, researchers at UC Berkeley and the College of the Atlantic in Bar Harbor deployed sensors on Great Duck Island in Maine to measure humidity, pressure, temperature, infrared radiation, total solar radiation, and photosynthetically active radiation. This network has tiered architecture. The lowest level of this architecture consists of sensor nodes that perform general purpose computing and networking in addition to application-specific sensing. The sensor nodes may be deployed in dense patches that are widely separated. The sensor nodes transmit their data through the sensor network to the sensor network gateway. The gateway transmits this data to base station through local transmit network. To provide relevant measurement a sensor board called Mica Weather Board is designed that includes UC Berkeley motes (Mica) as the sensor nodes, temperature, photo resistor, barometric pressure, humidity, and passive infrared (thermopile) sensors. Mica sensor node consists of a single channel at 916MHz with data rate of 40 kbps, an Atmel Atmega 103 microcontroller and considerable amount of non volatile memory.

1.4. Statement of the problem

As discussed earlier, for a WSN, operating at extremely low power is a critical design constraint and MAC layer has a significant impact on energy efficiency as it directly controls radio activities. Although many solutions have been proposed for conventional wireless networks- cellular networks, mobile ad hoc network (MANET) - and short-range wireless local area networks, they are not quite suitable for the unique characteristics of a large scale WSN. So a careful design of the MAC scheme is necessary for the optimal performance and extended lifetime of the network.

Several solutions to MAC problem for WSN have been proposed. Sensor-MAC (SMAC) [5], a hybrid of contention and reservation based schemes, aimed at reducing the energy wastes caused by collisions, control packet overheads, and idle listening but there is still energy consumption due to idle listening for a low data rate WSN. CSMA with preamble sampling [6] and WiseMAC [7] schemes improves energy efficiency by using "Sampling" technique. Due to the presence of hidden terminal problem, these schemes are not energy efficient at medium and high data rate so the performance of these schemes can be improved.

In view of the above, the following tasks have been undertaken in this dissertation:

- Literature survey of existing MAC protocols for WSN
- Detailed study and performance evaluation of SMAC, CSMA with preamble sampling and WiseMAC
- Development of a new energy efficient scheme, its performance evaluation through simulation and comparison with SMAC, preamble sampling and WiseMAC protocols.

1.5. Organization

Including this introductory chapter, the thesis is organized in 6 chapters.

Chapter 2 introduces significance of Medium Access Control (MAC) in wireless networks and relates energy efficiency to MAC. Major causes of energy waste are described next, and power saving schemes employed in other wireless networks are examined for their suitability to WSN.

Chapter 3 presents an overview of energy efficient MAC protocols for WSN. Scheduled and unscheduled MAC are compared and three significant schemes, viz., Sensor MAC (S-MAC), Preamble Sampling and WiseMAC, are discussed in greater detail.

A new energy efficient MAC scheme, Sampling and Scheduling based MAC Protocol (SSMAC), is proposed in chapter 4. The scheme is 'sampling' and 'scheduling' based and avoids the hidden terminal problem.

In Chapter 5 SMAC, WiseMAC, preamble sampling and SSMAC schemes are simulated in OMNet++ simulation environment. Performance of SSMAC is evaluated with respect to Preamble sampling, WiseMAC and SMAC.

Finally, the paper concludes with Chapter 6, which also discusses future work.

Chapter 2

Energy Efficiency and Medium Access Control

In all wireless networks, nodes must share a single medium for communication. Network performance largely depends upon how efficiently and fairly the nodes can share this common medium. Compared to a wired medium, a significant portion of the node's energy is spent on radio transmissions and on listening to the medium for anticipated packet reception. On the other hand, a wireless network always has restricted power sources; thus careful design of the MAC scheme is necessary for the optimal performance and extended lifetime of the network.

2.1 Energy efficiency and Medium Access Control (MAC) in Wireless Networks

Although wireless networks have existed for many years already, explicit concern about their energy efficient operation has emerged only recently. It is quite evident that when the power source is either costly or in short supply, energy efficiency is of paramount importance. In some wireless network applications, energy is actually entirely nonrenewable and is thus an overriding constraint for the design and operation of the network.

In Cellular networks, mobile users are concerned about the longevity of the batteries in their handheld devices, even though these can be replaced or recharged. Furthermore, even at the base station there is a desire for low energy consumption, because, despite the availability of energy supply, there are serious concerns about excessive heat generation. [8, 9]

In wireless LANs, laptops with Bluetooth or 802.11 cards in them are used. As these devices are in close proximity to power supplies. For them, energy efficiency is not a major issue, although there can be WLANs that stand alone outside buildings and for which, therefore, long battery life is also important. [8, 9]

In satellite networks, energy consumption is a serious concern despite the possibility of recharging the solar cells onboard the satellite. And, of course, at the earth stations there are concerns about heat generation (if they are fixed) or battery life (if they are mobile).

In Wireless Sensor Networks [1], it is very difficult to change or recharge batteries for sensor nodes. In fact, some design goals of sensor networks are to build nodes that are cheap enough to be discarded rather than recharged, or that are efficient enough to operate only on ambient power sources. In most of the applications, prolonging the lifetime of each node is a critical issue.

In most of the applications, radio is the main energy consumer. The MAC layer directly controls radio activities, and its energy savings significantly affect the overall node lifetime. So careful design of MAC layer can optimize energy consumption and prolong life time of network.

2.2 Sources of Energy Waste at MAC Layer

Energy efficiency [2, 3, and 4] is one of the most important issues in wireless sensor networks. To design an energy-efficient MAC protocol, we must consider the causes of energy waste from the MAC .The following sources are major causes of energy waste.

Collision

Usually the data gathered by a node are exchanged with others using the radio. Two nodes may transfer data to each other at the same time or several nodes transfer data to the same node at the same time. When a transmitted packet is corrupted in channel, it must be discarded and, thus the follow-on retransmissions increase energy consumption. Collision increases latency as well due to retransmissions.

Overhearing

When a node picks up packets destined for other nodes, overhearing occurs. In an ad-hoc fashion, a transmission from one node to another is potentially overheard by all the neighbors of the transmitting node; thus, all of these nodes consume power even though the packet transmission was not directed to them.

Control packet overhead

Sending and receiving control packets such as routing updates consumes energy and effectively reduces the network bandwidth for data packets. So control packet overhead must be as small as possible.

Idle listening

Nodes must listen to the channel often in order to receive possible traffic that is not sent. This is especially true in many sensor network applications because, if nothing is sensed, nodes are in idle mode most of the time. Actual measurements have shown that idle listening consumes 50 to 100% of the energy required for receiving in such networks

2.3 Power Saving Schemes at MAC

The closest types of networks rendering a similar behavior to WSNs are WAHNs [4], although they have marked differences. Properly standardized MAC protocols designed to cater to the ad hoc and distributed nature of WAHNs have been developed and are in commercial use. Also, some of them focus on energy savings, mainly for mobile applications. These features are highly sought after in WSNs as well. In this section we will discuss power saving schemes of AD-Hoc networks and there merits, demerits with respect to requirement at wireless sensor networks.

2.3.1 WLAN (IEEE 802.11)

IEEE 802.11 [10] is a standard developed for wireless LAN (WLAN) applications intended to replace conventional wired LANs so that the same applications can run seamlessly with media in 802.3 and 802.5 standards. The distributed coordination function (DCF) in IEEE 802.11 is the access method used to support asynchronous data transfer on a best-effort basis when the network functions in an ad hoc mode. It uses carrier sense multiple access with collision avoidance (CSMA/CA).

Each node maintains a backoff counter controlling the channel access. Before a node starts data transmission, it senses the wireless medium. If the medium appears to be idle for a specified period of time (distributed interframe space — DIFS), it starts decrementing the backoff counter. If the carrier is detected during this time, the backoff counter is frozen; otherwise, it starts transmission once the backoff counter reaches zero. The sender and receiver exchange short request-to-send (RTS) and clear-to-send (CTS) control frames to establish a session. Data transmission is followed by an acknowledgment (ACK) frame to confirm successful reception.

IEEE 802.11 standard also defines a power saving (PS) mode in which certain nodes can go to sleep. Under DCF operation, PS nodes "wake up" periodically for a short interval called the Ad Hoc Traffic Indication Map (ATIM) window. It is assumed that hosts are fully connected and synchronized so that the ATIM windows of all PS hosts will start at about the same time. During this window, each node will contend to send a beacon frame. Any successful beacon serves the purpose of synchronizing node clocks and also inhibits other hosts from sending their beacons. After receiving the beacon, an active node can send a direct ATIM frame to a node in PS mode. These transmissions are also contention based and use the same DCF access procedure described earlier. On reception of the ATIM frame, the PS node will reply with an ACK and remain active for the rest of the period. Data transfer will take place after that. The process of waking up a power saving node is shown in fig.2.1.

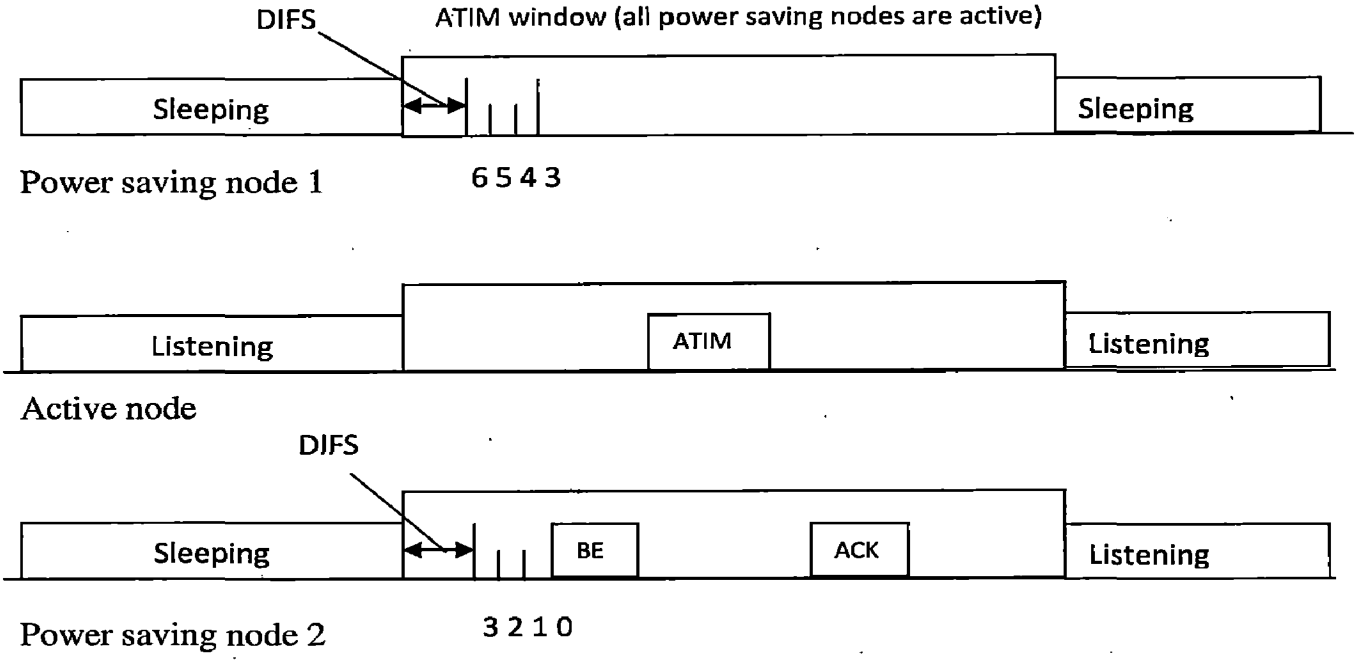

Fig.2.1: Contention between nodes N1 and N2 in IEEE 802.11 DCF.

In fig. 2.1, initially power saving node 1 and 2 are in sleeping mode. When both of these nodes wake up in ATIM window to transmit their beacon, they sense the channel for DIFS time then set their counter. Node 2's counter fires before node 1. Active node after receiving the beacon replies with a ATIM frame and receives acknowledgement from node2. After the end of this ATIM window node 2 switched to listening mode and node 1 goes to sleep mode.

Recent work has shown that the energy consumption using IEEE 802.11 MAC protocol is significantly high because the nodes are listening to the channel most of the time. Although the 802.11 standard defines the PS mode, it provides very limited policy about when nodes should go to sleep. PS mode is designed for single-hop networks in which all nodes can hear each other. When used in multihop networks, IEEE 802.11 may have problems in clock synchronization, neighbor discovery, and network partitioning, thereby degrading the performance.

2.3.2 WPAN (Bluetooth)

Bluetooth [11, 12] is a short-range wireless networking for electronic consumer devices (mobile phones, pagers, PDAs, etc.). It uses a TDMA and CDMA hybrid scheduling-based MAC scheme. The topology is a star network in which several slave nodes are attached to and synchronized with a master node to form a piconet. The number of nodes in a piconet is limited to eight in order to keep a high-capacity link among all the units and to limit the overhead required for addressing. Basic piconet configurations are shown in Fig. 2.2(a) and 2.2(b). Along with the basic TDMA scheme, Bluetooth uses frequency hopping code division multiple access (FH-CDMA), which uses a large number of pseudorandom hopping sequences. Interpiconet communication is achieved by forming ScatterNets as shown in Fig. 2.2(c) and 2.2(d). A single node can be a master in one piconet while it is a slave in another. Also, a node can be a slave in two piconets.

The master node determines hopping sequence, provides clock synchronization information for each slave node, and also controls the traffic in the piconet. The master/slave role is only attributed to a unit for the duration of the piconet. When a piconet is cancelled after a certain period of time, the master and slave roles are also cancelled and new piconets will be formed. Any node can become a master or slave.

(a) Point to point **(b) Point to multipoint**

(a) Scatter Net typel **(d) Scatter Net type2**

Fig.2.2: Piconet configurations in Bluetooth

The time slots are alternately used for master and slave transmissions. The master transmission includes slave address of the unit for which the information is intended. In order to prevent collisions on the channel due to multiple slave transmissions, a polling technique is used: for each slave-to-master slot, the master decides which slave is allowed to transmit.

TDMA schemes have a natural advantage of energy conservation because the duty cycle of the radio is reduced and there are no contention-introduced overheads or collisions. Nodes can be put to sleep to save energy during the off intervals of the duty cycle, thereby making this an obvious candidate for WSNs.

Use of a TDMA protocol usually requires the nodes to form real communication clusters such as the piconets described here. Nodes in such clusters are restricted to communicate within the cluster, except for the master node and possible gateway nodes. In Bluetooth, nodes within a piconet must be synchronized to use the TDMA scheme. Achieving local synchronization within the cluster is not a difficult task. However, network-wide synchronization will be almost impractical, especially in WSNs. So this protocol can not be used for wireless sensor networks.

Nodes in a WSN possess unique characteristics, especially the energy constraints, compact hardware, low transmission ranges, event- or task-based network behavior, and high redundancy. For a WSN, the extension of its lifetime is the most important issue. Therefore, power awareness is prominent in almost every aspect of the operation of WSNs. it is extremely critical that the medium access control scheme be power optimal.

Energy consumption of a WSN occurs in three domains: sensing; data processing; and communications; among these, radio communication is the major consumer of energy. As highlighted in Pottie and Kaiser [2], energy for transmitting 1 kb over a distance of 100 m is estimated as 3 J. With the same amount of energy, a general purpose processor with 100 MIPS/W power could execute 3 million instructions. The sensing circuitry consumes less power than the processor board in a typical WSN platform such as MICA [13, 14]. However, the radio consumes two to three times the power of the processor during packet transmission. Power consumption of the radio during listening to the channel for reception is also higher than the processor at full operation, but relatively lower than the transmitting power.

Thus, it is clear that the research focus should be on optimizing the medium access method in order to extend the lifetime of the network Above MAC schemes can not be used for wireless sensor networks as they are not energy efficient for wireless sensor networks. So development of new energy efficient MAC protocols for WSN is required

Chapter 3

Energy Efficient MAC Protocols for Wireless Sensor Networks

A number of MAC protocols and scheduling algorithms have been proposed in the last few years for different sensor network application. This chapter provides a survey of the state of the art of MAC protocols with an emphasis on energy efficiency. The protocols that are closely related to SSMAC in this dissertation will be described.

The problem of idle listening and collision, a major source of energy waste is addressed by many protocols [2, 3, 4]. In general, some kind of duty cycle is involved, which lets each node sleep periodically. For example, TDMA-based protocols are naturally energy preserving, because they have a duty cycle built-in, and do not suffer from collisions.

Node Activation Multiple Access (NAMA) [15] uses a distributed election algorithm to achieve collision-free transmissions. For each time slot, NAMA selects only one transmitter per two-hop neighborhood and hence all nodes in the one-hop neighborhood of the transmitter are able to receive data collision-free. However, NAMA does not address energy conservation. Traffic Adaptive Medium Access (TRAMA) [16] protocol provides energy efficient and collision free channel access in wireless network. Energy efficiency is achieved by adopting a transmission schedule which will provide collision free transmission, and by keeping nodes in low energy mode if nodes do not have data to transmit. TRAMA is similar to NAMA protocol, in that the identifiers of the nodes within a two-hop neighborhood are used to give conflict free access to the channel to a given node during a particular time slot. However, NAMA does not describe energy efficiency, and nodes that are not transmitting switch to receiver mode. TRAMA ignores synchronization inaccuracy and is energy inefficient for low traffic due to large overhead compared to data. Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks (PEDAMACS) [17] combines the characteristics of the cellular networks with those of sensor networks for continuous data gathering applications. The protocol assumes that a single access point (AP) exists in the network and all nodes communicate with this AP. AP learns whole network topology and traffic at nodes. Then it schedules the nodes to propagate data to AP collision free. However, TDMA has some disadvantages that limit its use in wireless sensor networks. TDMA normally requires nodes to form clusters, analogous to the cells in the cellular communication systems. For large networks, synchronization among clusters is a big problem.

CDMA Sensor MAC (CSMAC) [18] is a combination of DS-CDMA and frequency division in channel allocation. It reduces channel interference and consequently the message latency in the network. It exploits location awareness of sensor nodes to enable efficient network formation for collaborative sensing applications. Although CSMAC is energy efficient but it is quite complex and costly to implement due to small size and large number of sensor nodes in the network.

Contention protocols have several advantages compared to scheduled protocols. First, because contention protocols allocate resources on-demand, they can scale more easily across changes in node density or traffic load. Second, contention protocols can be more flexible as topologies change. Power Aware Multi-Access Protocol with Signalling for Ad Hoc networks (PAMAS) [19] is one of the earliest contention-based proposals to address power efficiency in channel access. PAMAS saves energy by attempting to avoid over-hearing among neighboring nodes. To achieve this, PAMAS uses out-of channel signaling. Sensor MAC (S-MAC) protocol [5] reduces idle listening by using periodic listen and sleep cycle. It uses RTS/CTS/DATA/ACK mechanism for data transfer and avoids overhearing. To have high energy efficiency duty cycle should be small but it increases latency. Timeout MAC (T-MAC) [20] protocol reduces latency by using variable duty cycle. In this protocol node goes to sleep if no message is received for a fixed period of time. In SMAC and TMAC, if no packet is present for a duty cycle energy is wasted through idle listening.

Scheduling protocols are much energy efficient due to very small duty cycle (1%). STEM [21] proposes using separate radios for data and wake-up in different channels. The data radio is normally in sleep mode and the wake-up radio periodically wakes itself up to listen to the channel. If wake-up radio listen wake-up-signal on wake-up channel, it turns on data radio to receive data. This scheme is inefficient due to more bandwidth requirement. Preamble Sampling [6] is a CSMA-based scheme similar to STEM, but it requires only one RF transceiver for both wake-up and data. The RF transceiver periodically wakes itself up for a very short time to detect whether the channel is busy by a Received Signal Strength Indicator (RSSI). To send a data packet to such a sampling node, the transmitter sends a long wake-up-preamble (WUP) with length of at least one cycle period to wake up the receiver followed by data. The big drawback of preamble sampling as well as of the STEM is that not only the destination node, but also all other neighboring nodes, are woken up, generating many overhearers. WiseMAC [7] is an optimized version of the preamble sampling scheme, in which each node stores each of its neighbors' sampling schedule that is piggybacked on the last received ACK from the neighbor. So, it can use a short WUP to wake up its neighbor according to the stored schedule. However, the length of the preamble is not fixed due to clock drift caused by the quartz instability. This shorter preamble not only drastically decreases the unnecessary waiting time for destination and overhearers, but also increases the channel capacity The Wake-Up-Frame scheme [22] is another preamble sampling-based approach, in which a wake-up-frame (WUF) is used instead of the meaningless WUP to reduce overhearing. The WUF is comprised of multiple short WUFs (SWUFs). Each SWUF includes the destination MAC address and a position field indicating the position of the current SWUF in the whole WUF. When a sampling node is woken up by the WUF and receives one complete SWUF, it can decide whether it is an overhearer. If it is the destination, it can calculate the start time of the data frame according to the position field so that it can go to sleep until the data comes. Although scheduling schemes are energy efficient they do not perform well for high traffic due to presence of hidden terminal problem.

3.1 Scheduled MAC Protocols

The RF transceiver is one of the biggest power consumers in such a small sensor or actuator device. If the RF transceiver stays in receive mode waiting for possible incoming packets, the battery will be exhausted within a few days because the power consumption of idle listening is normally not much smaller than that of transmitting. To this end, it has been estimated that if the duty cycle of the RF transceiver is somehow scheduled down to 1 percent, i.e., the RF transceiver is in receive or transmit mode for only 1 percent of the time, the overall system power consumption can be reduced by a factor of at least 50 [21].

3.1.1 Preamble sampling scheme

In wireless Sensor Networks energy efficiency can be achieved by keeping sensor nodes in sleep state. In this Preamble Sampling scheme [6] each sensor node senses the channel periodically for a very small duration; this process is called as Sampling. After each sampling, node finds the state of the channel by measuring RSSI, if the channel is free then node goes to sleep state again otherwise it continuously listen to the channel to receive data.

When a node receives data packet from upper layer it contend to the channel using some contention based technique. If the channel is free it transmits the long preamble followed by data frame. The length of this preamble should be equal to at least sampling period so that dedicated receiver can sense this preamble and receives the data. After successful completion of data reception receiving node transmits acknowledgement to transmitting node. This technique can be understood by fig. 3.1.

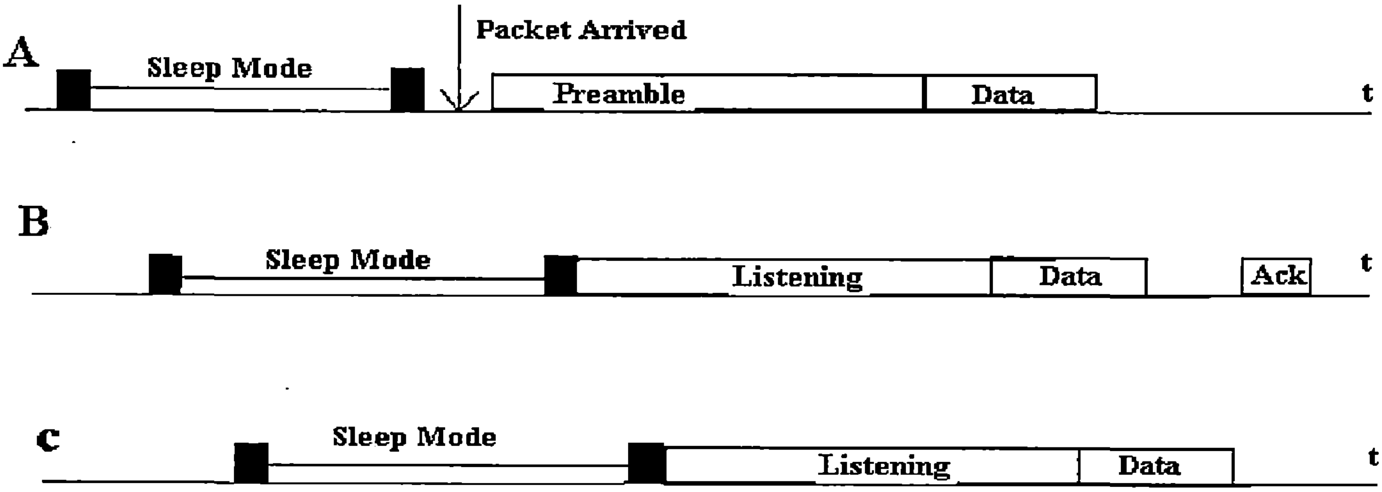

Fig.3.1: Communication process among nodes

When node A receives data packet destined to node B, it transmits preamble followed by data frame. When nodes B and C (neighbor of node A) receive this preamble they remain in listen mode and receives data frame. After receiving data frame node B replied with acknowledgement while node C discards this data frame.

If we combine the property of the preamble sampling with Aloha, the time consumed in idle listening to the channel can be reduced. It is achieved with the cost of increased probability of collisions. Each message is preceded by a preamble. On successful reception of a message, after the time needed to reverse the transceiver from RX to TX, the destination node will send an acknowledgment message back. This is shown in fig. 3.2.

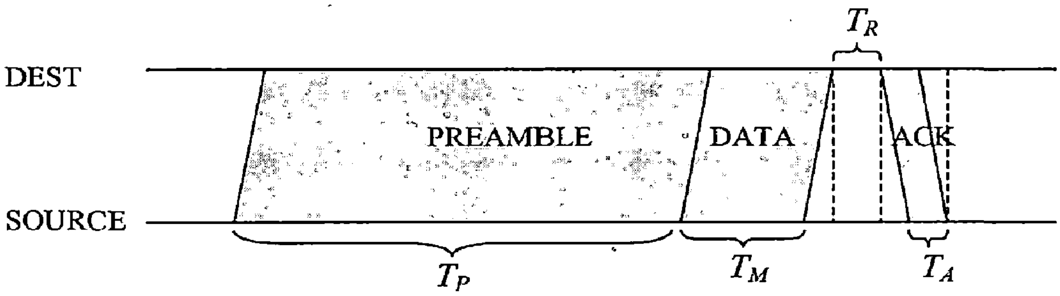

Fig. 3.2: Data – ACK transaction with Preamble [6]

A packet will be destroyed only if another packet is transmitted during data or ack phase. Fig. 3.3 shows the extreme case for collision.

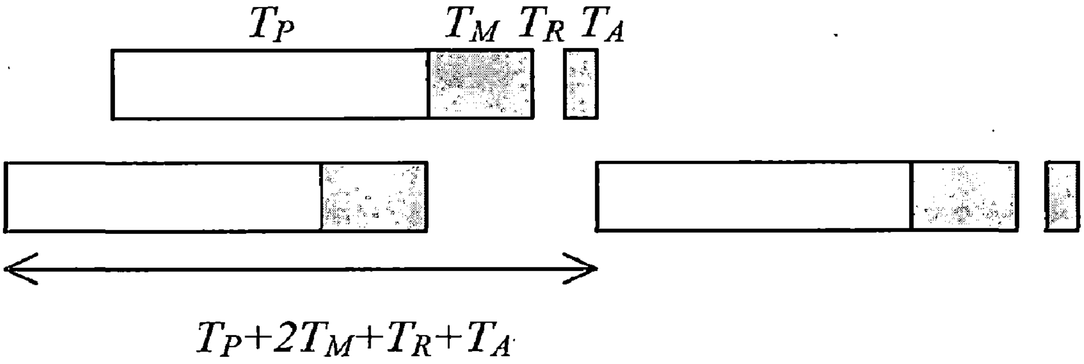

Fig.3.3: Worst case for collision [6]

First packet on the left is not followed by ACK. The reason is that, either the destination of the packet hasn't sent the ACK because it was jammed, or the ACK will not be received because the destination is a hidden node. A packet will hence be transmitted successfully if no other packet has been transmitted in a period of duration $T_P+2T_M+T_R+T_A$.

If N is the no. of neighbors for a node and g is the average packet transmission rate for every node. 1/g is the average inter-arrival time with Poisson distribution. Then the fraction of time when the node is busy is given by

$$b^{PSA} = 1 - e^{-(N+1)g(T_P+T_M+T_R+T_{ACK})} \tag{3.1}$$

The fraction of transmitting time for a node is given by

$$b_1^{PSA} = 1 - e^{-g(T_P+T_M+T_R+T_{ACK})} \tag{3.2}$$

The mean power consumption will be

$$P^{PSA} = b_1^{PSA} P_{TX} + (b^{PSA} - b_1^{PSA}) P_{RX} + \frac{P_{SeRx} T_{SeRx} + P_{RX} T_{Sense}}{T_P} \tag{3.3}$$

In the above equation P_{TX} and P_{RX} are power consumed during transmission and reception respectively. P_{SeRx} is time consumed during receiving set-up and T_{SeRx} is the time consumed in the process. T_{Sense} is the RSSI integration time.

CSMA is another contention based technique. When CSMA technique is used with preamble sampling scheme, it performs better than ALOHA as nodes listen to the channel before transmission.

Fig. 3.4 shows the comparison of CSMA with preamble sampling with respect to ALOHA with preamble sampling, pure ALOHA and genie aided ALOHA in terms of life time of the network. Genie added ALOHA is the advanced version of ALOHA in which node goes to sleep state if the channel id free. Comparison shows that np-CSMA outperforms than other schemes.

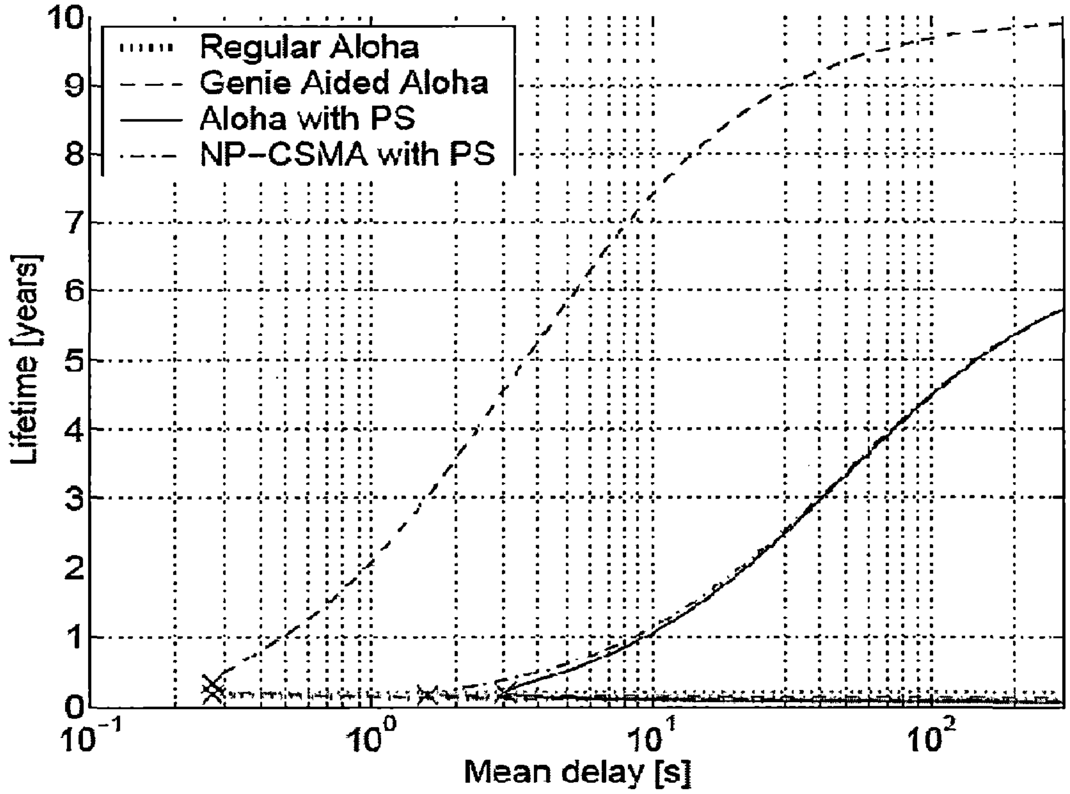

Fig. 3.4: Performance comparison of Preamble sampling with CSMA w.r.t. other schemes [6]

Main disadvantage of preamble sampling scheme is that all the neighbors of sending node receive frame which causes energy wastes in form of overhearing. Channel capacity for this scheme is low due to long length of preamble.

3.1.2 WiseMAC

The Wireless Sensor MAC (WiseMAC) protocol [7] was specially designed for the WiseNET network. The WiseNET project was introduced by the Microelectronics Division to build a power efficient ad hoc network with low power devices that reach a life time for about 5 years. [18]

WiseMAC, which provides energy efficiency, uses preamble sampling techniques and exchange of sampling schedules among the devices. Effective power consumption is gained due to decreasing the overhearing in such networks.

The general technology of WiseMAC is shown in fig.3.5. At regular time intervals, the nodes listen to the data channel for a short time. The time, when they sample the medium is independent but constant for each node. So each node has its own sampling schedule, which it stores but also sends out to its neighbors. Each node, which receives a sampling schedule from a neighbor, updates its own with this information. Therefore, all network nodes know when the other nodes are awake and sample the medium or when they are in sleep mode.

When one node has to transmit data to another node within the network, the source checks its sampling schedule to know when the destination is awake to receive the packet. The sampling schedule and the knowledge about other nodes avoid the long period of idle listening. If the source wants to start transmission, it will listen to the data channel and if it is busy, the node will go back to sleep mode and try transmission at the next sampling time. If the channel is free, the source node will wait until the destination wakes up and then it sends out the data packets with a preamble in front of each packet. The preamble ensures that the destination stays awake and receives the forwarded data. Due to the knowledge of each nodes' schedule, the preamble size can be very small, which reduces energy waste, that appears because of the sending of longer packets.

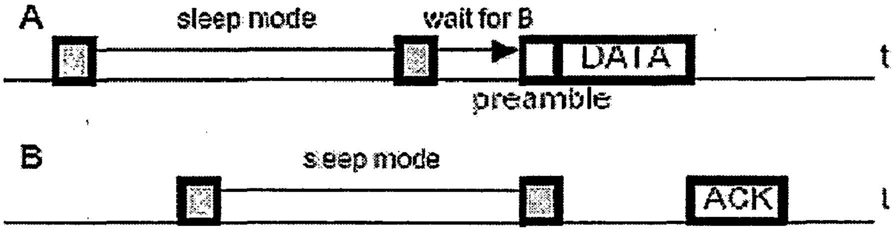

Fig.3.5: Communication process among nodes

WiseMAC does not use synchronization, although the internal clocks in each device drift apart because of the long and numerous sleeping of each node. To keep successful transmissions, the preambles, which are also used as a reservation of the medium, are sent before the data packet and ensure that nodes really wake up. If two nodes wanted to transmit at the same time, collisions would occur and in addition to mitigate those collisions, WiseMAC also uses non-persistent carrier sensing with a backoff time like WLAN. This increases the power consumption of each node due to the waiting in the backoff time until the destination node wakes up.

The general technology of WiseMAC is shown in fig.3.5. At regular time intervals, the nodes listen to the data channel for a short time. The time, when they sample the medium is independent but constant for each node. So each node has its own sampling schedule, which it stores but also sends out to its neighbors. Each node, which receives a sampling schedule from a neighbor, updates its own with this information. Therefore, all network nodes know when the other nodes are awake and sample the medium or when they are in sleep mode.

When one node has to transmit data to another node within the network, the source checks its sampling schedule to know when the destination is awake to receive the packet. The sampling schedule and the knowledge about other nodes avoid the long period of idle listening. If the source wants to start transmission, it will listen to the data channel and if it is busy, the node will go back to sleep mode and try transmission at the next sampling time. If the channel is free, the source node will wait until the destination wakes up and then it sends out the data packets with a preamble in front of each packet. The preamble ensures that the destination stays awake and receives the forwarded data. Due to the knowledge of each nodes' schedule, the preamble size can be very small, which reduces energy waste, that appears because of the sending of longer packets.

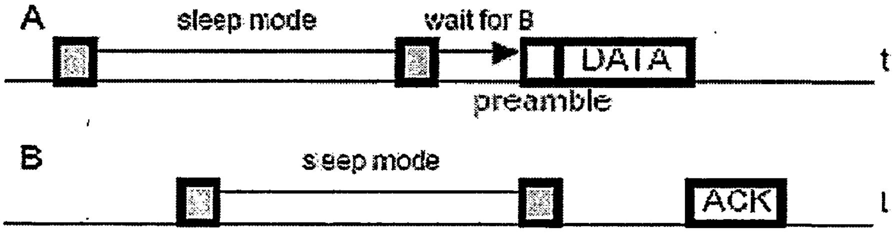

Fig.3.5: Communication process among nodes

WiseMAC does not use synchronization, although the internal clocks in each device drift apart because of the long and numerous sleeping of each node. To keep successful transmissions, the preambles, which are also used as a reservation of the medium, are sent before the data packet and ensure that nodes really wake up. If two nodes wanted to transmit at the same time, collisions would occur and in addition to mitigate those collisions, WiseMAC also uses non-persistent carrier sensing with a backoff time like WLAN. This increases the power consumption of each node due to the waiting in the backoff time until the destination node wakes up.

The WiseMAC protocol can also be used for infrastructure networks. The access point sends out the preambles and data packets.

This protocol was simulated on the GloMoSim platform with radio layer modeled on WiseNET transceiver. Simulation was done for a network of 81 nodes in a 9 by 9 grid. Traffic is generated following a Poisson process by the 9 nodes on the left of the lattice, and carried in a multi-hop fashion towards the right. Data frames have length of 64 bytes. The channel data rate is 25 kbps. Wake-up period was chosen to be 200 milliseconds. The average power consumed by central forwarding node is 25 µW when message is forwarded every 100 seconds. With a single alkaline battery of capacity C=2.6 Ah and a constant power leakage of 27 µW, lifetime over 5 years can be achieved. For a high traffic (inter-arrival between 100 seconds and 5 seconds), the average power consumption of WiseMAC is 200 µW. Energy efficiency for high load reaches above 75% where energy efficiency is defined as the ration between the energy consumed by an ideal protocol to forward a packet (without any wake-up, idle listening, overhearing or collision overhead) divided by the energy consumed by WiseMAC to forward the same payload. The average hop latency was measured between 140 and 240 ms, depending on the traffic conditions.

3.2 Unscheduled MAC Protocols

In these protocols a common channel is shared by all nodes and it is allocated on-demand. A contention mechanism is employed to decide which node has the right to access the channel at any moment.

These protocols have several advantages compared to scheduled protocols. First, because unscheduled protocols allocate resources on-demand, they can scale more easily across changes in node density or traffic load. Second, unscheduled protocols can be more flexible as topologies change. There is no requirement to form communication clusters, and peer-to peer communication is directly supported. Finally, unscheduled protocols do not require fine-grained time synchronizations as in TDMA protocols.

The major disadvantage of unscheduled protocol is its inefficient usage of energy. It normally has all the sources of energy waste we discussed in chapter 2: nodes listen at all times and collisions and contention for the media can waste energy. Overcoming this disadvantage is required if contention-based protocols are to be applied to long-lived sensor networks.

3.2.1 Sensor MAC (SMAC)

The main objective of the S-Mac [5] is to conserve energy in sensor networks; it takes in to consideration that fairness and latency are less critical issues compared to energy savings. Thus this scheme compromises fairness and latency to a certain degree. In order to save energy, S-MAC establishes a low duty cycle operation in nodes. It reduces idle listening by periodically putting nodes into sleep in which the radio transceiver is completely turned off.

During the design of the S-MAC, the following assumptions have been considered:

- Short range multihop communications will take place among a large number of nodes.
- In-network data processing is used to reduce traffic.
- Applications will have long idle periods and can tolerate some latency.
- Network lifetime is critical for the application.

All nodes in the network will be following a sleep and listen cycle called a frame, as shown in fig.3.6. The duration of the listen period is normally fixed and the sleep interval may be changed according to application requirements, changing the duty cycle.

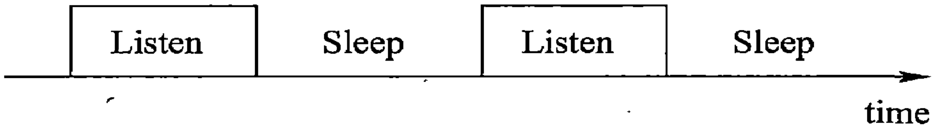

Fig.3.6: Periodic listen and sleep

The same RTS/CTS/DATA/ACK procedure as that in IEEE 802.1 is adopted here for unicast packets in order to ensure collision avoidance and to avoid hidden terminal problem. Broadcast packets are sent without using RTS/CTS; NAV timer update information is included in all four types of packets. Thus this scheme uses virtual and physical carrier sensing. After a successful exchange of RTS and CTS, the sender will start transmission and will extend it into the sleeping duration as well, if required. The nodes do not follow their sleep schedules until they finish the transmissions, thus increasing the performance.

Coordinated sleeping

Although a node can freely choose its own active/sleep schedules in S-MAC, it attempts to reduce the overhead by synchronizing schedules of neighboring nodes together. Nodes exchange their schedules by periodically broadcasting a SYNC packet to their immediate neighbors at the beginning of each listen interval. A set of nodes synchronized together will form a virtual cluster. Because the whole network can not be synchronized together, neighboring nodes are allowed to have different schedules. However, neighboring nodes are free to talk to each other, no matter to which listen schedule they adhere. A considerable portion of the nodes will belong to more than one virtual cluster, enabling intercluster communication. Thus, this scheme is claimed to be adaptive to topology changes.

When the network is first deployed, each node tries to retrieve a sleep schedule from a neighbor first. In the case of failure, it adopts one of its own and also tries to announce it to neighbors by broadcasting a SYNC packet. Broadcasting SYNC packets may also follow the normal carrier sense and random backoff procedure. If a node receives a different schedule after it announces its own schedule, it must adopt one of the following:

- If the node detects no other neighbors, it can discard the current the current schedule and adopt the new.
- If it has one or more neighbors and is already a part of an existing virtual cluster it can adopt both schedules by waking up at the listen intervals of both.

The active interval of a node is divided into three parts for SYNC, RTS and CTS as shown in fig.3.7. If CTS is received, data transmission will be immediately followed. Here node 2 and node 3 are synchronized to the schedule of the node 1 by receiving its SYNC packet, thus falling into the same virtual cluster. When node 2 and node 3 sends their RTS packet, one who initiates first will get CTS packet. When a new node powers on, it listens to the channel in anticipation of a SYNC packet. However, it is possible that a new node fails to discover an existing neighbor because of collisions or delays in sending SYNC packets by neighbor due to busy medium. To prevent a case in which two neighbors cannot find each other when they follow completely different schedules, SMAC protocols employs a simple periodic neighbor discovery procedure by requiring each node to listen periodically to the channel for the whole synchronization period. The frequency can be varied depending on the network conditions etc.

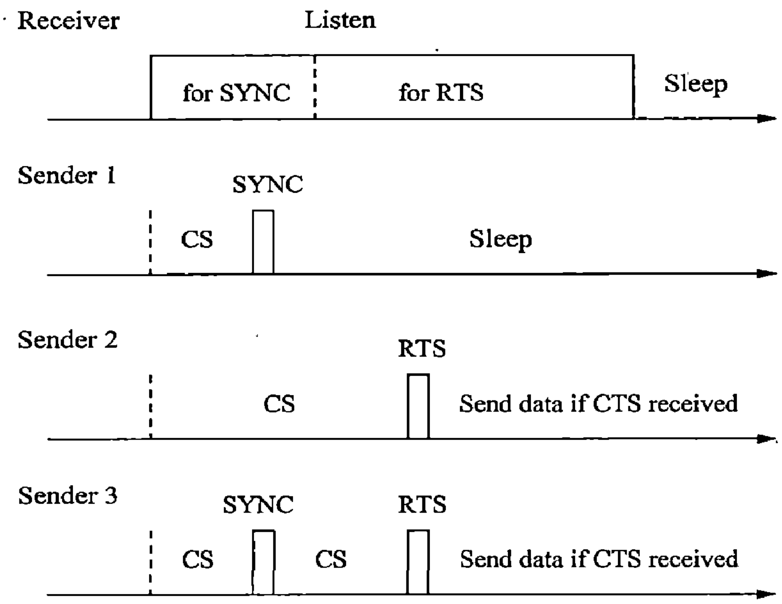

Fig.3.7: Communication between nodes in SMAC [5]

This protocol is experimented with the topology shown in fig. 3.8. This is a two-hop network with two sources and two sinks. Packets from source A reach at sink D through node C while packets from source B reach at sink E through node C.

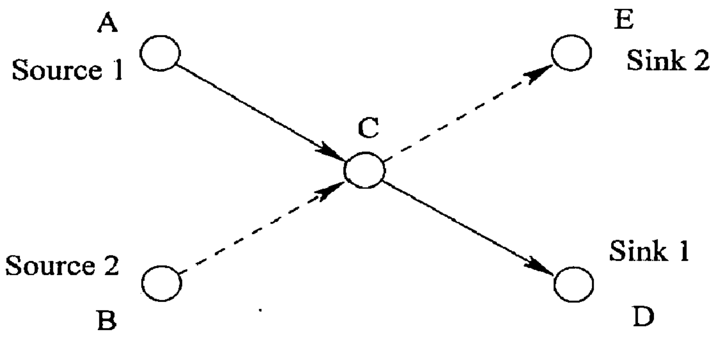

Fig. 3.8: Topology used for SMAC experiment [5]

Each node is based on Rene Motes. This mote consists of an Atmel AT90LS8535 microcontroller, TR1000 radio transceiver. This node operates at a transmission rate of 19.2 Kbps. The experiments are carried out for three schemes: SMAC with 30% duty cycle, IEEE 802.11 and 'Overhearing avoidance'. 'Overhearing avoidance' is the extended version of IEEE 802.11. In this scheme nodes go to sleep it they receive any

RTS of CTS frame while in IEEE 802.11 nodes go for virtual sensing and keep their radio "on".

Fig. 3.9 and 3.10 compare these schemes with respect to energy consumption and percentage sleep time at source nodes.

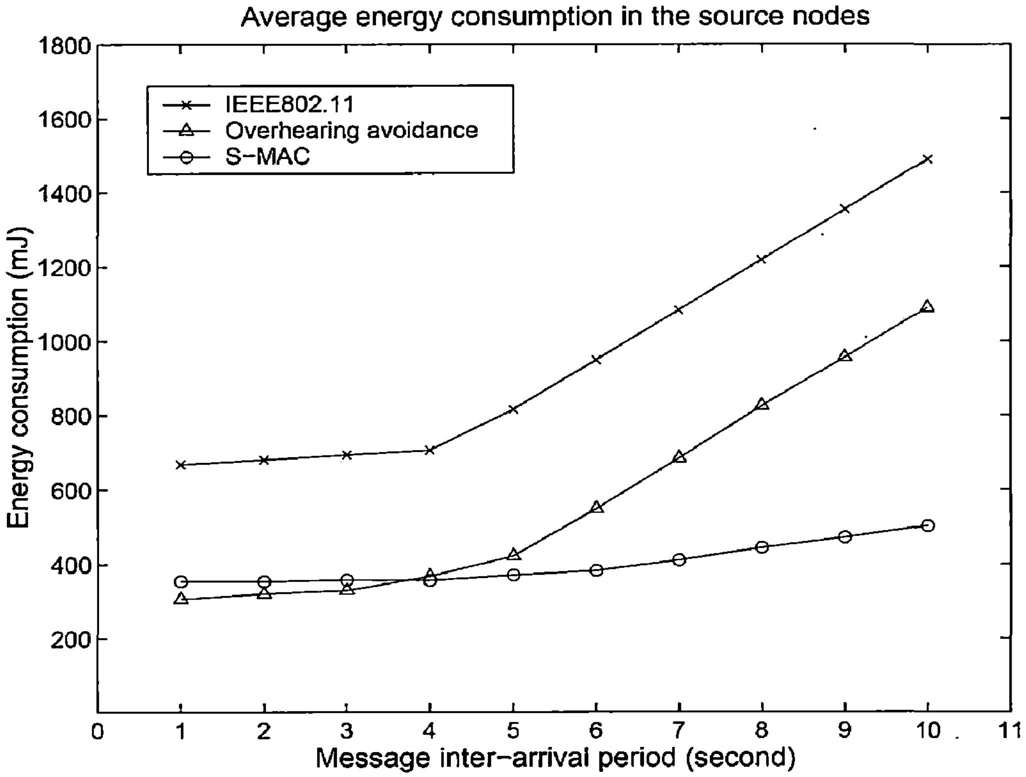

Fig. 3.9: Comparison of energy consumption between different MAC schemes [5]

Fig. 3.5 shows that IEEE 802.11 protocol is less energy efficient with respect to other protocols because radio is always turned on for IEEE 802.11. "Overhearing avoidance" scheme perform better than IEEE 802.11 because each node goes to sleep mode if they receive any RTS or CTS packet. SMAC protocol gives the best results because it avoids both overhearing and idle listening.

Fig.3.6 compares percentage sleep time for "Overhearing avoidance" and SMAC scheme. SMAC protocol exhibits high percentage sleep time due to minimized idle listening by switching nodes to sleep mode.

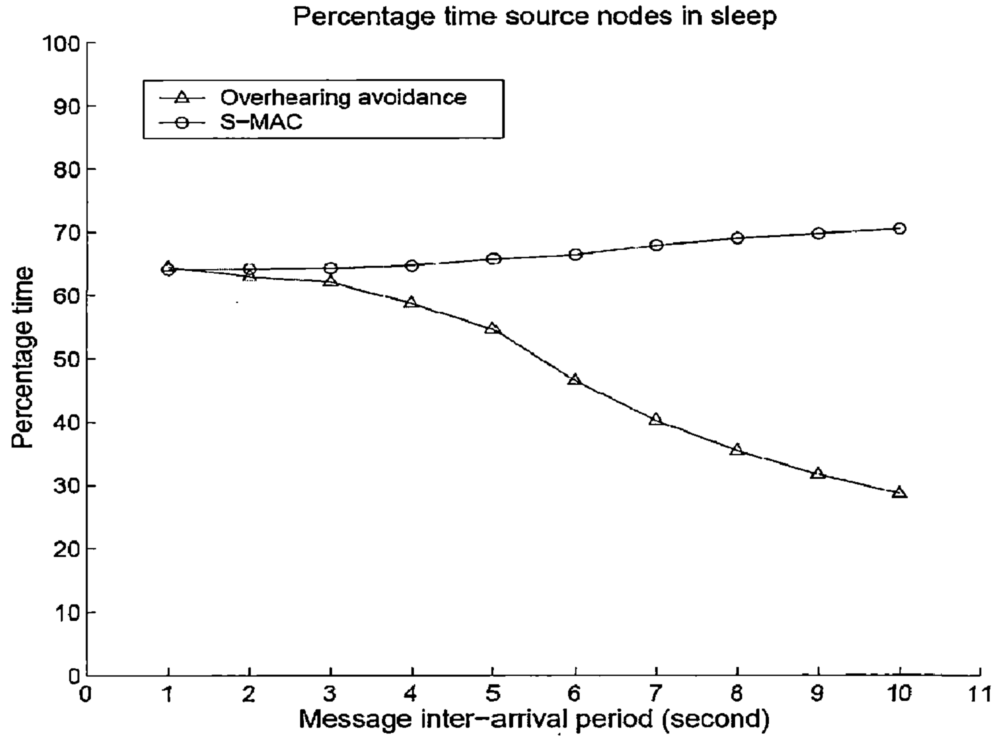

Fig. 3.10: Comparison of percentage sleep time between different MAC schemes [5]

Chapter 4

SAMPLING AND SCHEDULING BASED MAC PROTOCOL

MAC layer design is an important issue that governs the energy efficiency of a WSN. Radio operations, that are the main consumers of energy, are directly controlled by MAC layer. The main sources of energy waste at MAC are collision, idle listening, overhearing and large overheads. An efficient MAC design should, therefore, attempt to minimize the energy waste or stop it altogether.

As already discussed in section 3.1, Sampling based protocols show energy efficient performance for low traffic. Among the Sampling based schemes, WiseMAC performs better as each node has knowledge of the wake-up time of its neighbors and consequently a small preamble length. The performance of this protocol degrades at medium and high rate due to hidden terminal problem. Elimination of the hidden terminal problem, therefore, would result in improved energy efficiency for medium and high traffic.

4.1. Overview of SSMAC

Energy efficiency is a major consideration in the design of MAC layer for Wireless Sensor Networks. The energy efficient protocol proposed in this dissertation uses the concept of "Sampling". In other words, each node in the network samples the channel periodically for a very small time. Each node finds the "free" or "busy" state of the channel by measuring RSSI. If the channel is found free, the node goes back to sleep state; otherwise it remains "On".

In SSMAC, neighbors of a receiving node transmit their packets in round-robin fashion, and so, no collisions occur among these nodes. Each node, as a consequence, receives data only from one of its neighbors during "sampling" period. Collisions, however, may still occur between some selected hidden terminal nodes, as shown in fig. 4.1. Fig. 4.1 shows a network of 12 sensor nodes. Number inside the circle represents the node number. Tx range 1 and Tx range 2 represents the transmission range of node 7 and node 4 respectively. Let node 8 and node 3 have data to transmit for node 7 and node 4 respectively. If transmitting nodes transmit their data at the same time then the collision will occur. This problem may be called the "Selected Hidden Terminal Problem".

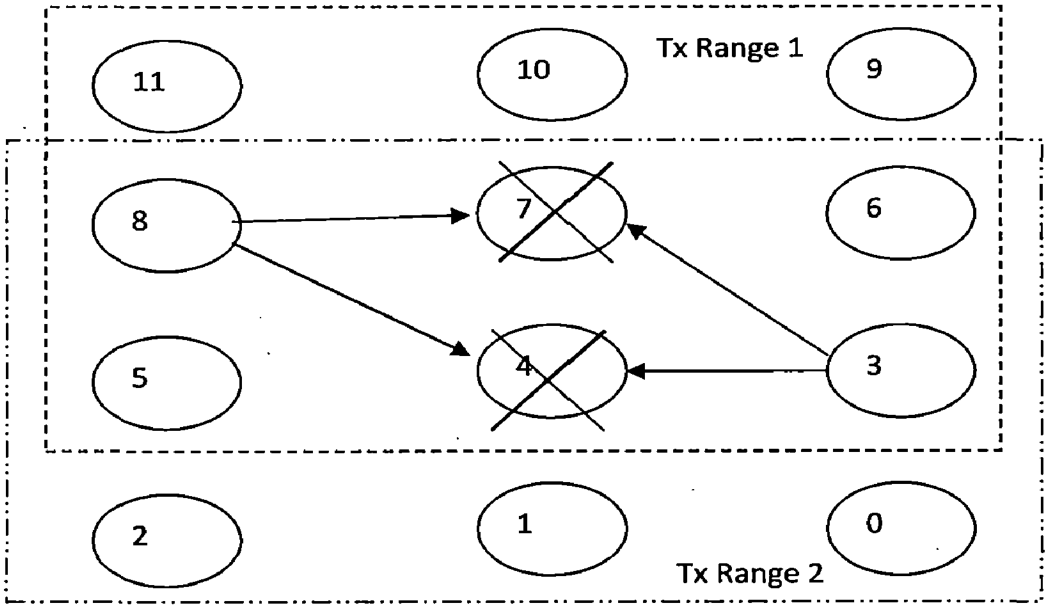

Fig. 4.1: Selected hidden terminal problem

The resulting collisions can be minimized if the nodes are made to wake-up in a particular fashion as explained later.

Each node in the network, that wants to transmit its data packet, wakes up the destination node by transmitting a Wake-up frame (WUF). Let L_{WUF} be the length of the WUF. L_{WUF} is not fixed, but it is prolonged to compensate for the clock drift caused by quartz instability. If the frequency tolerance of the quartz is θ, L_{WUF} should be at least $4\theta T$, where T is the time elapsed after the last received ACK. The value of L_{WUF} accounts for the worst case scenario: If the sender lags by θT and the receiver leads by θT, the WUF has to be sent $2\theta T$ before the saved schedule of the sending node to ensure that it wakes up the receiver; if it is the other way around, the WUF has to last for at least $2\theta T$ more beyond its saved schedule. WUF is broken into a number of smaller intervals called SWUF as shown in fig. 4.2. Each SWUF consists of the mac address of the destination node and number of remaining SWUFs. After sensing a WUF, a destination node keeps its radio 'ON' until it receives a complete SWUF and then goes to sleep mode without receiving rest of the SWUF. The destination node can go to 'Sleep' till the end of WUF as it has information about remaining number of SWUFs. This will reduce unnecessary waiting time.

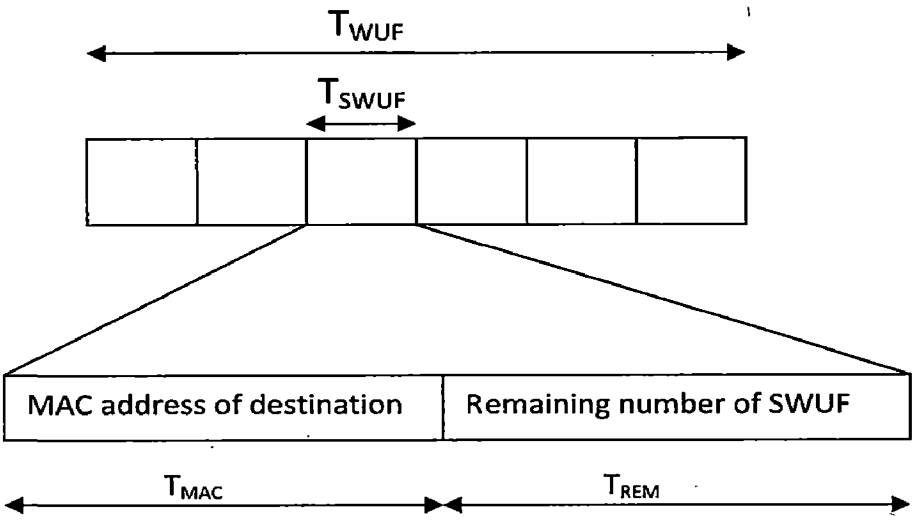

Fig. 4.2: Structure of WUF

SSMAC is organized in two phases: set-up phase and communication phase. Set-up phase is used for transmission of signaling information during network initialization while communication phase is used for data transmission.

In the Communication phase, time is divided into periods which are again divided into three time slots, numbered from 0 to 2 as shown in fig. 4.3. Each time slot consists of three parts: wake-up slot, communication slot, and synchronization slot. Let the time slot number is represented by T_S. Each node samples the channel only during wake-up slot.. In each period every node wakes up only once in one of the slot in predefined way. Data transmission takes place during Communication phase. Each node uses WUF/CTS/DATA/ACK procedure for communication. This procedure resembles the RTS/CTS/DATA/ACK standard procedure where RTS is replaced by WUF. Nodes turn on their radio only for "Sampling" and data transmission.

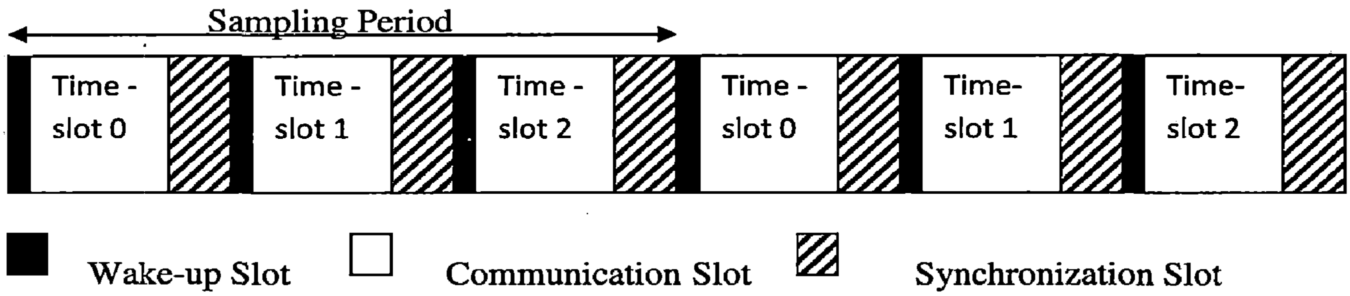

Fig. 4.3: Time slot organization

In the set-up phase, to minimize the "Selected Hidden Terminal Problem" , each node is made to select a time slot for the communication phase in a particular fashion, and to gather information (neighbor's MAC address, time slot number, wake-up time) about their one and two hop neighbors in order to calculate their turn to transmit to their neighbor. For whole set-up phase, each node remains in receiving mode except during transmission. Each node randomly selects its time for transmission. Each node uses np-CSMA technique to transmit its signaling packet. This protocol starts with Set-up phase followed by Communication phase.

4.2. Set-up Phase

In the set-up phase nodes select their time slot, and gathers information about their one hop and two hop neighbors. Each node has a hop counter register that helps in calculating T_S. Initially each node's hop counter value (T_C) is set to 0. Each node randomly select their wake-up time (t_W) for communication phase which is the relative time with respect to starting point of wake-up slot and lies within wake-up slot. The length of wake-up slot is taken equal to the length of one SWUF so that all the two hop neighbors contend for the channel in the same slot. If it is not done then a node may interrupt already existing transmission process.

Fig. 4.4 shows a network of 9 nodes arranged in a grid of 3x3 at the starting of set-up phase. Each node is represented by a circle. The leftmost number in circle represents the mac address of the node, middle number represents T_C and the right most represents T_S. Let node 0 be the sink node.

Fig. 4.4: Network structure at the start of set-up phase

Node 0 starts the set-up phase by broadcasting SYNC frame. This SYNC frame consists of its MAC address, hop counter value (T_{CS}) which is equal to T_C and wake-up time (t_W). Sink node implicitly selects 0th timeslot and sets Tcs to Tc.

All the neighbors of node 0 receive this SYNC frame, and calculate and save the information (time slot number, wake-up time, and MAC address) about the sink node. Receiving nodes which have T_C equal to 0 (not applicable for sink node) or greater than (T_{CS} +1) calculate their T_S values using (4.1) and update T_C value using (4.3). Receiving nods calculate sink node's T_S using (4.2).

$$T_{S\cdot} = (T_{CS} + 1) \bmod 3 \qquad (4.1)$$

$$T_{S\cdot} = T_{CS} \bmod 3 \qquad (4.2)$$

$$T_C = T_{CS} + 1 \qquad (4.3)$$

The modified network after reception of SYNC frames by sink node's neighbors is shown in fig. 4.5.

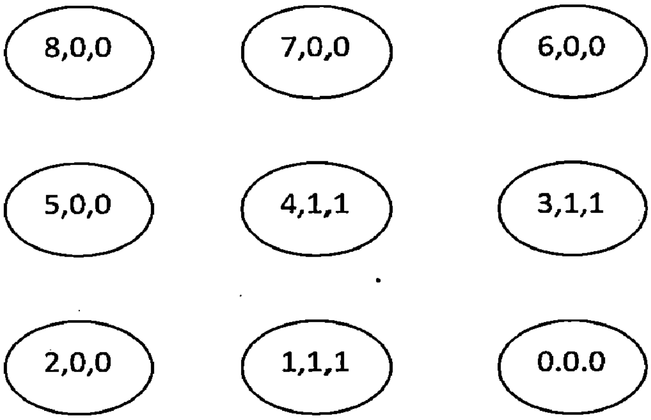

Fig. 4.5: Modified Network Structure - 1

Now neighbors of sink node 0 i.e. nodes 1, 3, and 4 set a timer and contend for the channel using np-CSMA. Let node 1 gets access to the channel and transmits its SYNC packet with Tcs equal to its Tc and already selected wake-up time. This packet is received by the node 0,2,3,4 and 5. These nodes add node 1 in their neighbor list with its MAC address, wake-up time and T_S calculated from (4.2). Now all these receiving nodes compare (Tcs +1) with their T_C values and update their T_C and T_S values as explained before.. Thus nodes 2 and 5 set their T_S and T_C values to 2, and other receiving nodes do not make any changes in T_C and T_S. Now the modified network is shown in fig. 4.6.

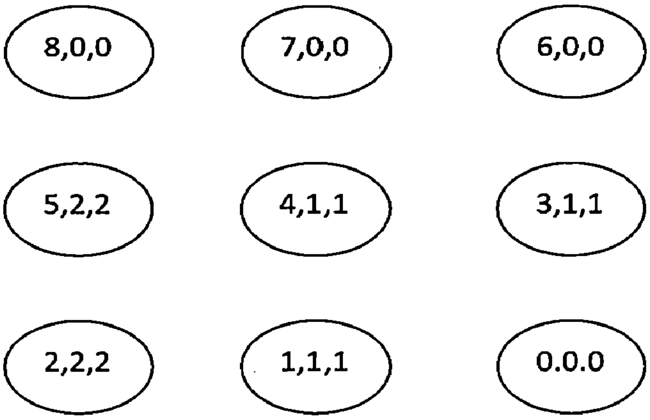

Fig. 4.6: Modified Network Structure - 2

Now nodes 2 and 5 set their timer and contend for the channel to transmit their SYNC frame. Let node 5 gets access to the channel. It transmits SYNC frame with T_{CS} equal to T_C and already selected wake-up time. The receiving nodes 1, 2, 4, 7, and 8 add node 5 in their neighbor list with its information. In the network nodes 7 and 8 have T_C equal to 0 so they set their T_S to 0 using (4.1) and T_C to 3 using (4.3). Other nodes do not make any modification. The modified network is shown in fig. 4.7.

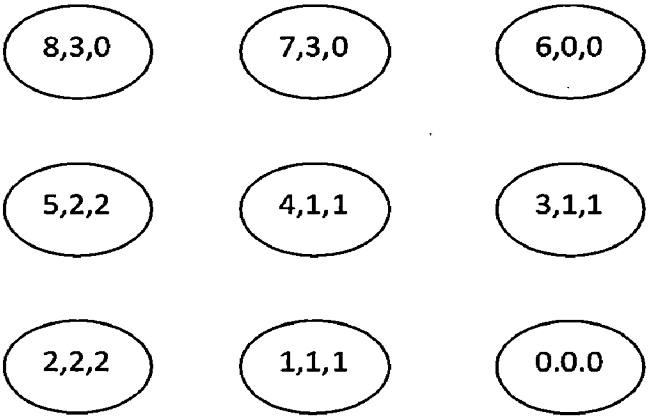

Fig. 4.7: Modified Network Structure - 3

Let node 4 gets access to channel. It transmits its SYNC with Tcs equal to T_C and other fields mentioned already. All receiving nodes add this node with its information in their neighbor list. Nodes 7 and 8 have T_C greater than (Tcs +1) so they update their T_C and T_S

values to 2. Node 6 also updates its T_C and T_S values to 2 as its T_C value is 0. Thus the modified network is shown in fig. 4.8.

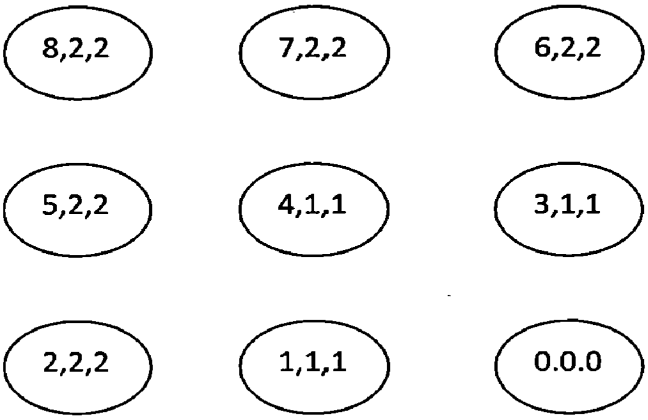

Fig. 4.8: Modified Network Structure - 4

Now all the remaining nodes transmit their SYNC frames that make no changes in T_C and T_S values of receiving nodes because each receiving node will get a SYNC frame with (T_{CS} +1) greater or equal to T_C but the receiving nodes will add SYNC frame' owner as one hop neighbors in their neighbor list.

If the above described procedure is applied to a network of 36 nodes arranged in 6x6 grids with assumption of no collision then the time slots are allocated in the manner shown in fig. 4.9.Number in the circle represents T_S of the corresponding node.

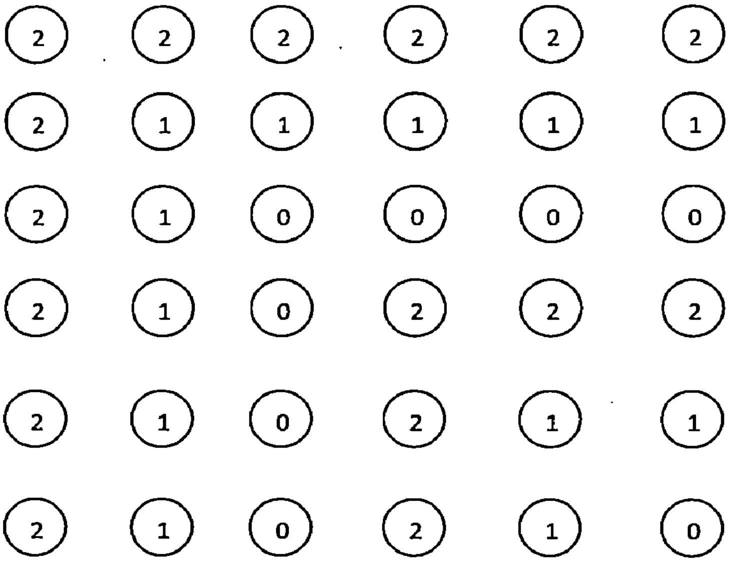

Fig. 4.9: Nodes time slot in lattice structure

In the above, we assumed a collision free transmission. But, in practice, collisions will occur due to hidden terminals. So to guarantee consistent neighborhood information, all the nodes set their timer to transmit again their SYNC packet using np-CSMA. Hop counter value of SYNC frame will be taken from current value of T_C. This SYNC packet updates the nodes parameters as described earlier. This process is repeated k times where k depends on the size of network.

After receiving one hop neighbor's information, each node broadcasts information (mac address, T_S) about all its one hop neighbors using np-CSMA by transmitting a frame called as Information frame. When a node receives an information frame from one of its one-hop neighbors, it makes a neighbor list for its one hop neighbors. To guarantee consistent two-hop neighborhood information all the nods transmits information frame k times as explained before.

After this, each node will have information about its entire one hop and two hop neighbors. Now each node calculates its turn to transmit data packet to its one hop neighbors. This method can be understood using fig. 4.8. Let node 4 be selected as a receiving node. One hop neighbors of the node 4 select their turn according to their time slot number and mac address. Nodes with lower timeslot number will have higher priority to transmit than the nodes with higher timeslot number, while among the nodes with same timeslot number, node having lower mac address will have higher priority. The reason for giving priority with respect to time slot is will be apparent from later discussions. In fig. 4.8, node 0 will have highest priority and it gets first turn to transmit to node 4. Nodes 1 and 3 will have the next higher priority. However, node 1 will have higher priority to transmit due to its smaller mac address. Remaining nodes have lowest priority, these nodes also selects their turn according to their mac address. So nodes 2, 5, 6, 7 and 8 get 4th, 5th, 6th, 7th and last turn respectively. If a receiving node has n neighbors then each one hop neighbor will get a chance to transmit in every nT_P time where T_P is the sampling period.

The choice of the number of time slots i.e. 3, and their allocation and prioritization rules are aimed to minimize "Selected Hidden Terminal Problem". Using allocation of time slots as described above and prioritization will allow maximum number of data transmission from nodes with same time slot. For example in fig. 4.9, at any time period, nodes with time slot 0 will get most of the transmissions from nodes with 0th or 1st or 2nd timeslot. This will almost avoid "Selected Hidden Terminal Problem" among nodes with different time slots. If we use only two time slots then, time slot allocation is shown in fig. 4.10.

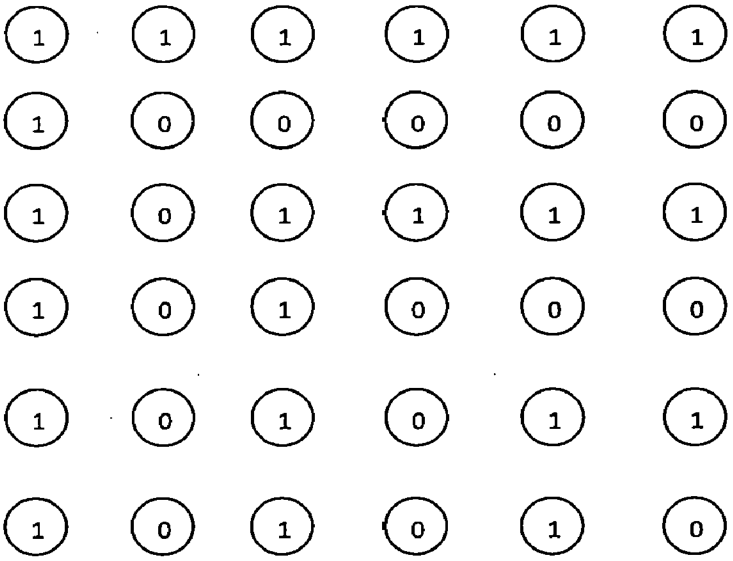

Fig. 4.10: Time slot organization for two timeslot network

Now the probability of having selected hidden terminal problem will be high as nodes with same time slots are surrounded by the nodes of only one type of time slot (different from previous one) and their transmissions may introduce hidden terminals. If we have more than three time slots then there will be no impact on selected terminal problem as nodes with same time slot will be surrounded by the nodes with only two different time slots. With the actions described above, the probability of selected hidden terminal problem will be very low as data generation rate in WSN is very low in comparison to other wireless networks.

4.3. Communication Phase

Actual data transmission between the nodes takes place during communication phase. The data packet to be transmitted from a node can be self generated or received from near- by nodes.

Fig. 4.11 depicts data transmission process between nodes A and B from the point of transmission of WUF.

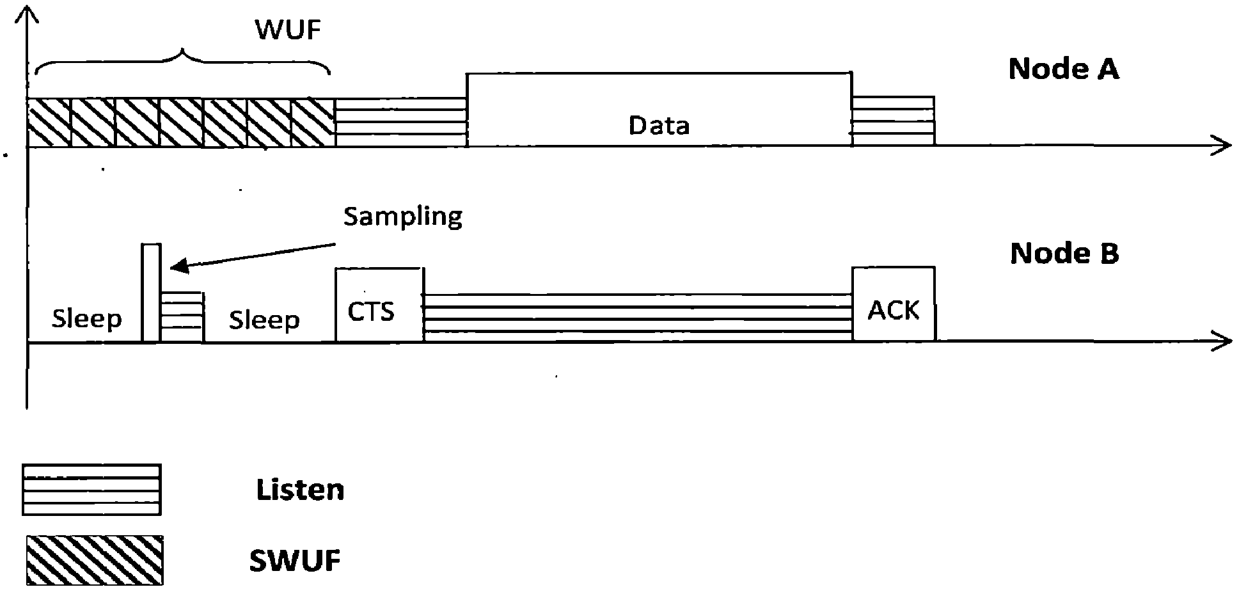

Fig. 4.11: Communication pattern between two nodes

When node A receives a packet from upper layer destined to node B, it calculates remaining time to start transmission process using its already calculated turn and backoff time counter value. When the time instance arrives, node A turns on its radio to start transmission. Initially the node A senses the channel, if the channel is found free, Node A transmits wake-up frame (WUF). When the node B wakes up to sample the channel, it finds the channel in "Busy" state by measuring the RSSI. After finding the channel in busy state, radio remains turns on to receive a SWUF. If the node B receives a SWUF it checks the destination address field of SWUF to find whether the SWUF is destined to it. After receiving a SWUF, node B estimates the remaining time for completion of WUF using remaining of SWUF field and goes to sleep mode for this time period. When node B wakes up, it replies with a Clear To Send (CTS) frame. When node A receives CTS frame it means channel has been occupied for the current slot for data frame transmission. After receiving the CTS frame, node A transmits its data frame. Node B replies with acknowledgement after receiving complete data frame. If node A does not receive CTS, it calculates remaining time to transmit in its next turn and turn off the radio.

If the selected hidden terminal problem occurs then the two simultaneously transmitting nodes will never receive the CTS frame. To solve this problem a WUF transmission counter is provided which is set to a threshold value. When WUF transmission counter crosses the threshold value for a data frame then each node randomly selects a number (as in exponential backoff mechanism) and then these nodes do not transmit for that number of turns.

Synchronization slot is provided to mitigate clock drift problem. When clock drift occurs, the transmission process is shifted to synchronize slot. As there is no communication process of previous slot is existing, collision is avoided. If θ is clock drift per second and Δ is the length of synchronization slot then for worst possible case, collision may start after $(\Delta/(2*\theta))$ seconds. To synchronize nodes again, sink node broadcasts synchronizing packet after every $(\Delta/(2*\theta))$ seconds. Nodes that receive this packet synchronize their clock and broadcast this packet. Thus all nodes again synchronize to operate perfectly.

crosses the threshold value for a data frame then each node randomly selects a number (as in exponential backoff mechanism) and then these nodes do not transmit for that number of turns.

Synchronization slot is provided to mitigate clock drift problem. When clock drift occurs, the transmission process is shifted to synchronize slot. As there is no communication process of previous slot is existing, collision is avoided. If θ is clock drift per second and Δ is the length of synchronization slot then for worst possible case, collision may start after $(\Delta/(2*\theta))$ seconds. To synchronize nodes again, sink node broadcasts synchronizing packet after every $(\Delta/(2*\theta))$ seconds. Nodes that receive this packet synchronize their clock and broadcast this packet. Thus all nodes again synchronize to operate perfectly.

Chapter 5

SIMULATION SET-UP, RESULTS AND DISCUSSION

We evaluate the performance of SMAC with 50% duty cycle, WiseMAC, Preamble Sampling and SSMAC protocols through simulation. All simulations are built on the network simulator OMNeT++ [23]. The schemes are compared in terms of percentage sleep time, percentage transmit time, latency, percentage collisions and power consumption.

5.1. OMNeT++

Objective Modular Network Test-bed in C++ (OMNeT++) is a public-source, component-based, modular simulation framework [23]. It is has been used to simulate communication networks and other distributed systems. The OMNeT++ model is a collection of hierarchically nested modules as shown in Figure 5.1. The top-level module is also called the System Module or Network. This module contains one or more sub-modules each of which could contain other sub-modules. The modules can be nested to any depth and hence it is possible to capture complex system models in OMNeT++.

Modules are distinguished as being either simple or compound. A simple module is associated with a C++ file that supplies the desired behaviors that encapsulate algorithms. Simple modules form the lowest level of the module hierarchy. Users implement simple modules in C++ using the OMNeT++ simulation class library. Compound modules are aggregates of simple modules and are not directly associated with a C++ file that supplies behaviors.

Modules communicate by exchanging messages. Each message may be a complex data structure. Messages may be exchanged directly between simple modules (based on their unique ID) or via a series of gates and connections. Messages represent frames or packets in a computer network. The local simulation time advances when a module receives messages from another module or from itself. Self-messages are used by a module to schedule events at a later time.

The structure and interface of the modules are specified using a network description language. They implement the underlying behaviors of simple modules. Simulation executions are easily configured via initialization files. It tracks the events generated and ensures that messages are delivered to the right modules at the right time.

We have chosen OMNeT++ as the framework for Sensor Network Simulations due to following salient features:

- OMNeT++ allows the design of modular simulation models, which can be combined and reused flexibly.
- It is possible to compose models with any granular hierarchy.
- The object-oriented approach of OMNeT++ allows the flexible extension of the base classes provided in the simulation kernel.
- OMNeT++ offers an extensive simulation library that includes support for input/output, statistics, data collection, graphical presentation of simulation data, random number generators and data structures.
- OMNeT++ simulation kernel uses C++ which makes it possible to be embedded in larger applications.
- OMNeT++ models are built with NED and omnetpp.ini and do not use scripts which makes it easier for various simulations to be configured.

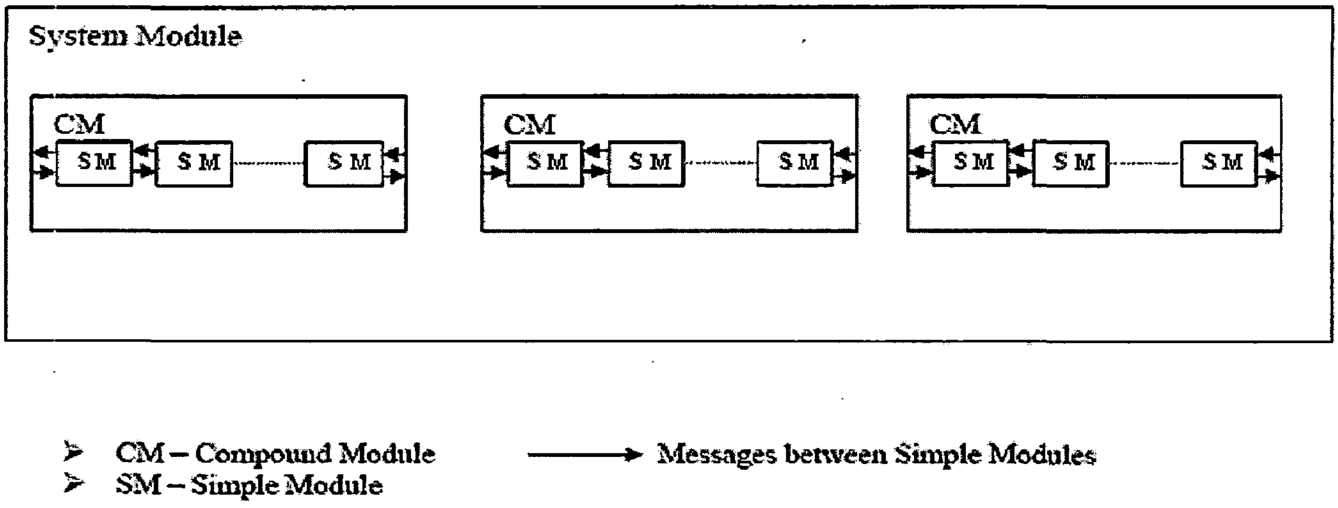

Fig. 5.1: Module hierarchy in OMNeT++

5.2. Simulation model

When a network is simulated in OMNeT++, the network is modeled in modules. In this section, we presented our simulation model and the functioning of this model is presented by message flow diagram.

Fig. 5.2 shows the simulation model for a network of 2 nodes. This model consists of 5 simple modules and 1 compound module.

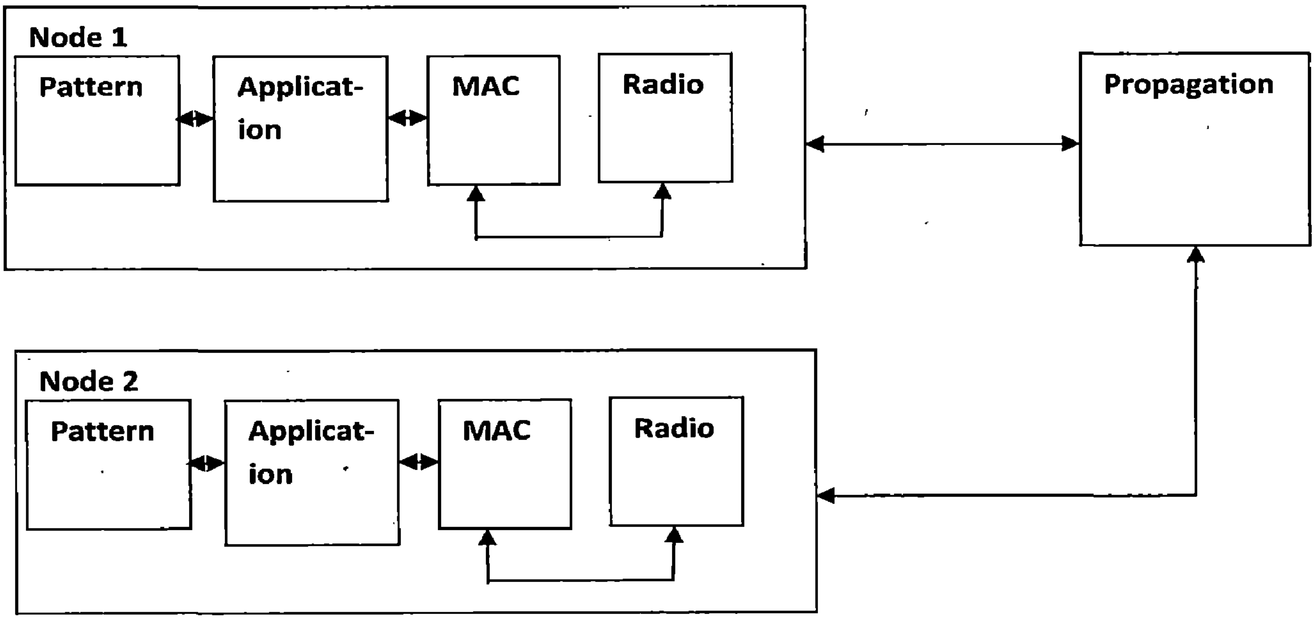

Fig. 5.2: Simulation model

The compound module representing a sensor node consists of four simple modules: pattern, application (APP), MAC and radio. Pattern module generates data for the sensor. APP represents the application layer for the network. This module consists of buffer of length 10. MAC module supervises the MAC functioning and directly controls the radio module which is the model of transceiver TR1001. Compound module of each node is connected to propagation module (PROP) which supervises the channel activities. It routes the data from a transmitting node to those nodes that lie in its radio range. One can simulate the network without the PROP by connecting each node to its neighbors through connectors but that makes the model too complex.

Fig. 5.3 shows the model for message transmissions in the network for the case of 2 nodes. This model has taken basic MAC to describe message flow between modules. In this basic MAC scheme, a node transmits the data packet whenever it has data to transmit with no acknowledgement requirement.

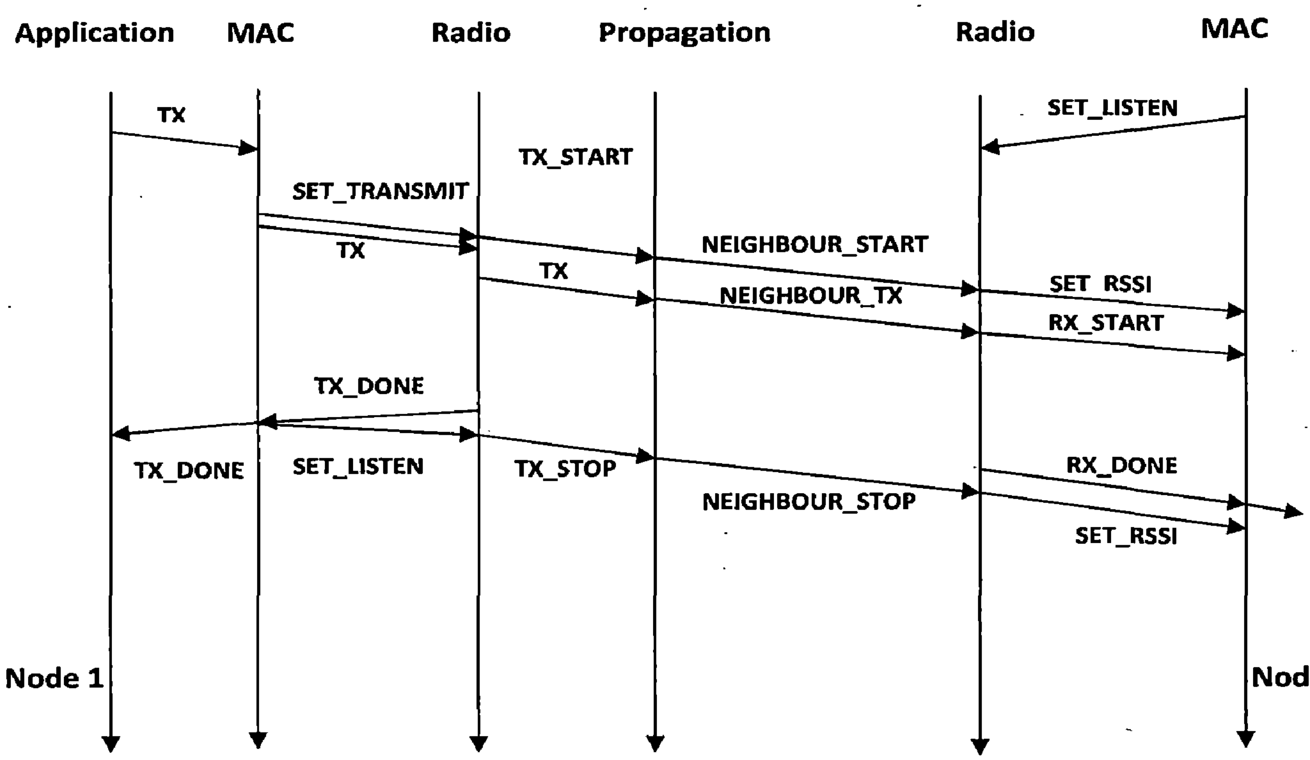

Fig.5.3. Model for message transmission

Initially Node 2's MAC module sends SET_LISTEN message to radio module to set radio in listen mode. Pattern module of Node 1 generates the data packet and transfers it to the application module. If there is no data packet to transmit at MAC module, Application module sends this data packet to MAC module using TX message; otherwise, the data packet is buffered at the Application module. When MAC is ready for this data packet, sends SET_TRANSMIT message to radio to switch radio in transmitting mode. SET_TRANSMIT message is followed by TX message which consists of data packet. Radio transmits TX_START message to PROP followed by TX message. After receiving TX_START message, PROP sends NEIGHBOR_START message to node 2 to inform presence of signal in the channel. Node 2's radio sends RSSI information through SET_RSSI message. This information can be used for channel sensing based MAC protocols. NEIGHBOR_START message is followed by NEIGHBOR_TX message which consists of data packet. When node 2's radio receives NEIGHBOR_TX message, the radio starts receiving data packet and sends RX_START message with data packet to MAC module. When radio receives full packet it sends RX_DONE message to MAC module which, after receiving, forwards data packet to application module. When the node 2's radio completes packet transmission it sends TX_DONE message to MAC module which forwards the message to application module. MAC module, after receiving TX_DONE, sends SET_LISTEN to radio module to switch it to receive mode. Radio sends TX_STOP message to propagation module to convey the end of transmission after receiving SET_LISTEN. When propagation module receives TX_STOP message it sends NEIGHBOR_STOP message to node 2's radio to convey that node 2's transmission is over. Radio modifies RSSI value at MAC by sending new SET_RSSI message. If a radio receives another NEIGHBOR_START message prior to NEIGHBOR_STOP then collision will occur and MAC is informed about collision by sending RX_STOP message. There is time difference between reception of RX_DONE and SET_RSSI due to extra transmit time the radio takes to switch to the sleeping or listening mode.

5.3. Simulation Parameters

Simulation has been carried out for a network of 100 nodes in a 10 by 10 grid. We have chosen a radio range so that all non-edge nodes have 8 neighbors. We have assumed that there is no fading in the channel. This assumption is valid for a sensor network in which nodes are placed very close to their neighbors. The parameters used for the simulation are tabulated in table 5.1.

Parameter name	**Parameter value**
Power consumed during transmission	24.75mW
Power consumed during reception	13.5mW
Power consumed during sleep mode	15μW
Radio data rate	115.2KBps
Data payload size	60 Byte
SWUF length	2 Byte
Header length (CTS,ACK)	6 Byte
Sampling time (WUP, WiseMAC, SSMAC)	1 Second
Time slot length	33.33 millisecond
Synchronization slot length	22 millisecond
SMAC period length	1 sec

Table 5.1: Simulation parameters

The physical layer model used for all the simulations was based on the TR1001, a typical radio used in sensor networks. The TR1001 [24], the radio used by the UC Berkeley Motes [25], are short range, low data-rate radios with built-in support for low-power sleep state. This transceiver uses ASK modulation for signal transmission but it is not significant for this simulation.

5.4. Simulation Results

The network is simulated with different traffic loads. All nodes generate data packets with Poison distribution and transmit these to one of their neighbors. The neighbors are selected with uniform distribution.

Fig.5.4 compares the average percentage sleep time of different MAC schemes. SMAC's sleep time decreases with increasing message inter-arrival time because, in high traffic, nodes go to sleep mode using NAV (Network Allocation Vector) and wake up only when

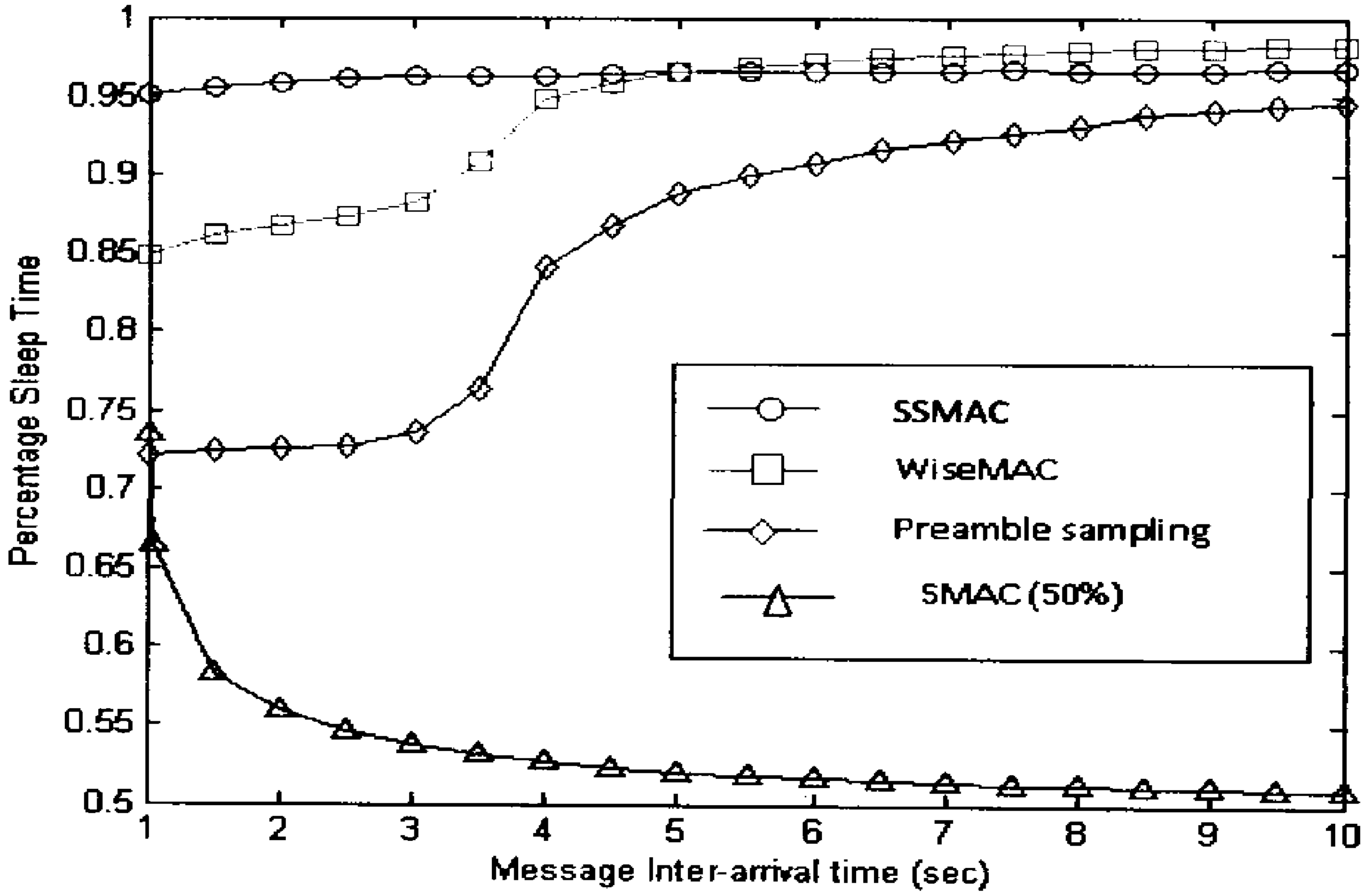

Fig. 5.4: Comparison of percentage sleep time

the transmission is over. Percentage sleep time for WiseMAC and Preamble sampling schemes are higher than that of SMAC due to their very low duty cycle and it increases with increasing inter-arrival time due to their less involvement in transmissions and receptions. WiseMAC has higher sleep time than Preamble sampling scheme because of small length of preamble used which also causes less overhearing. SSMAC has highest percentage sleep time because, due to collision free transmission, no time is wasted in retransmission.

Fig. 5.5 compares power consumed per node in the network. It is clear that the SSMAC protocol outperforms WiseMAC, Preamble sampling and SMAC protocols. SMAC protocol consumes highest amount of energy due to high duty cycle. Although WiseMAC and Preamble sampling schemes have very low duty cycle, they exhibit high power consumption at high and moderate load due to collision from hidden terminals.

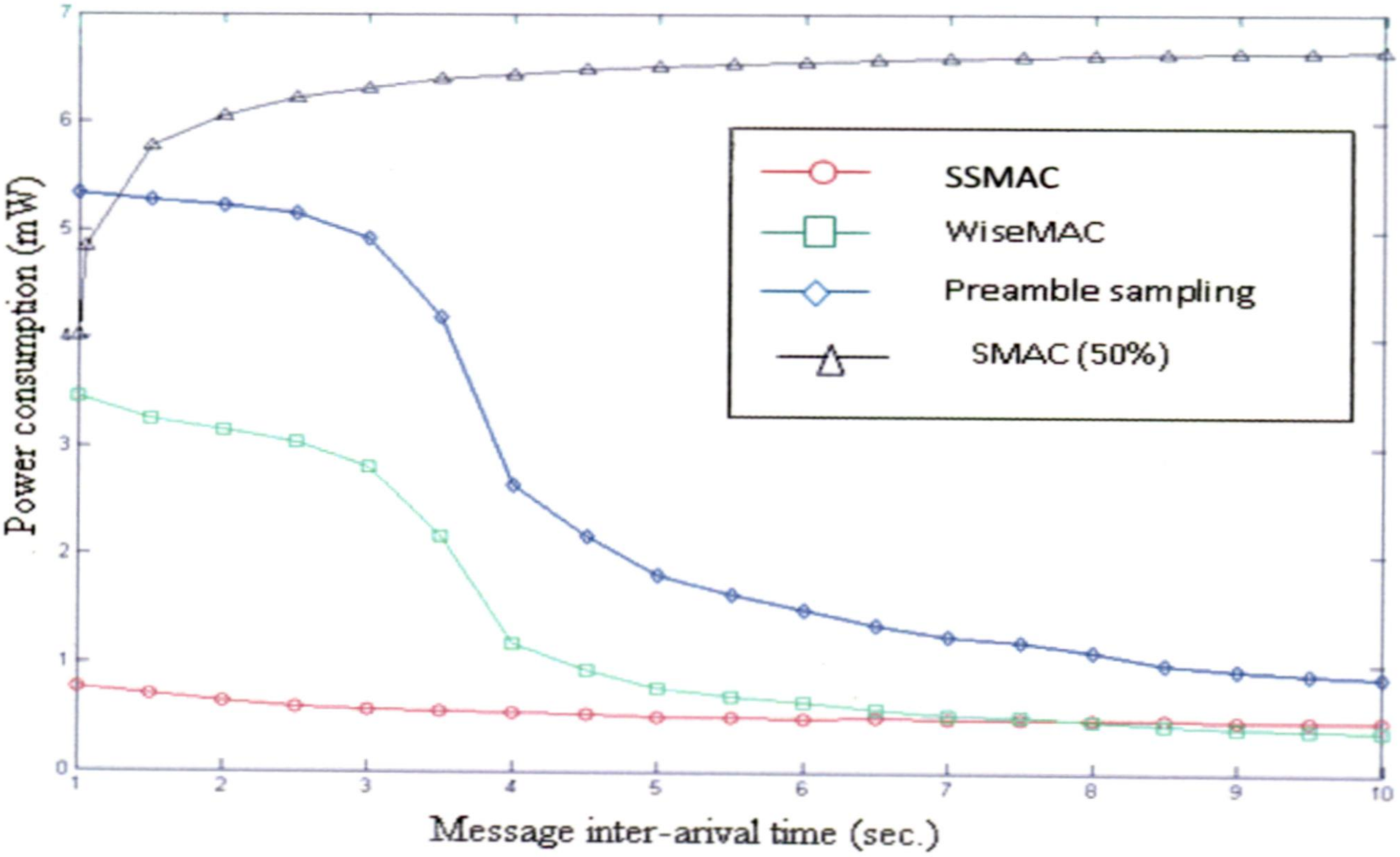

Fig. 5.5: Comparison of power consumption for different MAC schemes

At low data rate, WiseMAC outperforms Preamble sampling scheme because of its small preamble length. SSMAC gives best result as it has vey low duty cycle and provides collision free operation.

Percentage data frame collision with respect to total data frame transmission for different schemes is shown in fig. 5.6. As already described, SMAC and SSMAC protocols have no collisions while collisions occur in WiseMAC and Preamble sampling schemes due to hidden terminals.

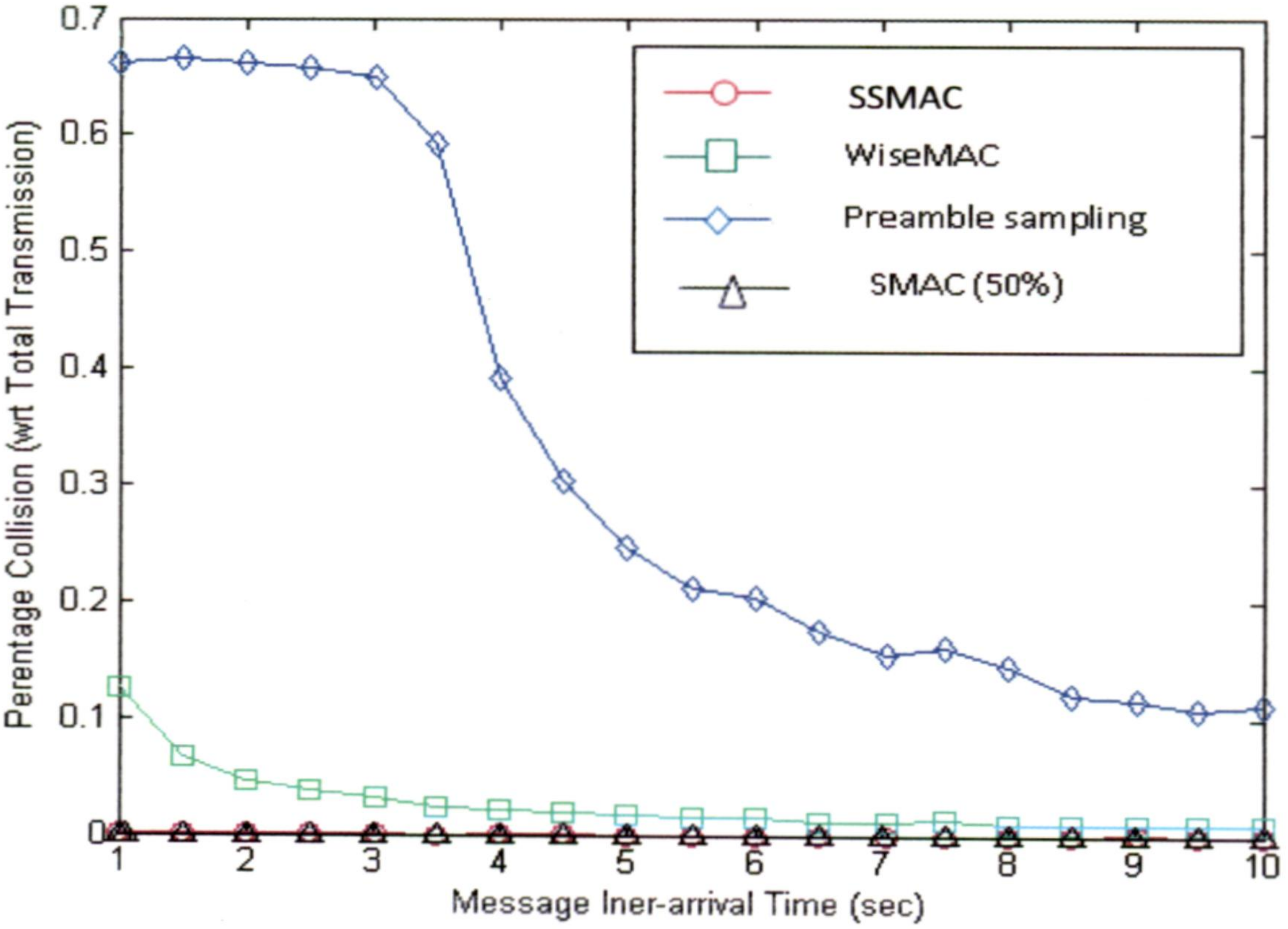

Fig. 5.6: Comparison of percentage collisions for different MAC schemes

The good performance of SSMAC comes at the cost of latency. Fig. 5.7 compares latency for different schemes. Latency in SSMAC is high because each node has to wait for its turn to transmit its packet. If the node is not successful to transmit in the current turn, then it has to wait for the next turn. At high traffic, Preamble sampling scheme exhibits maximum delay because the probability of collision is high due to long preamble.

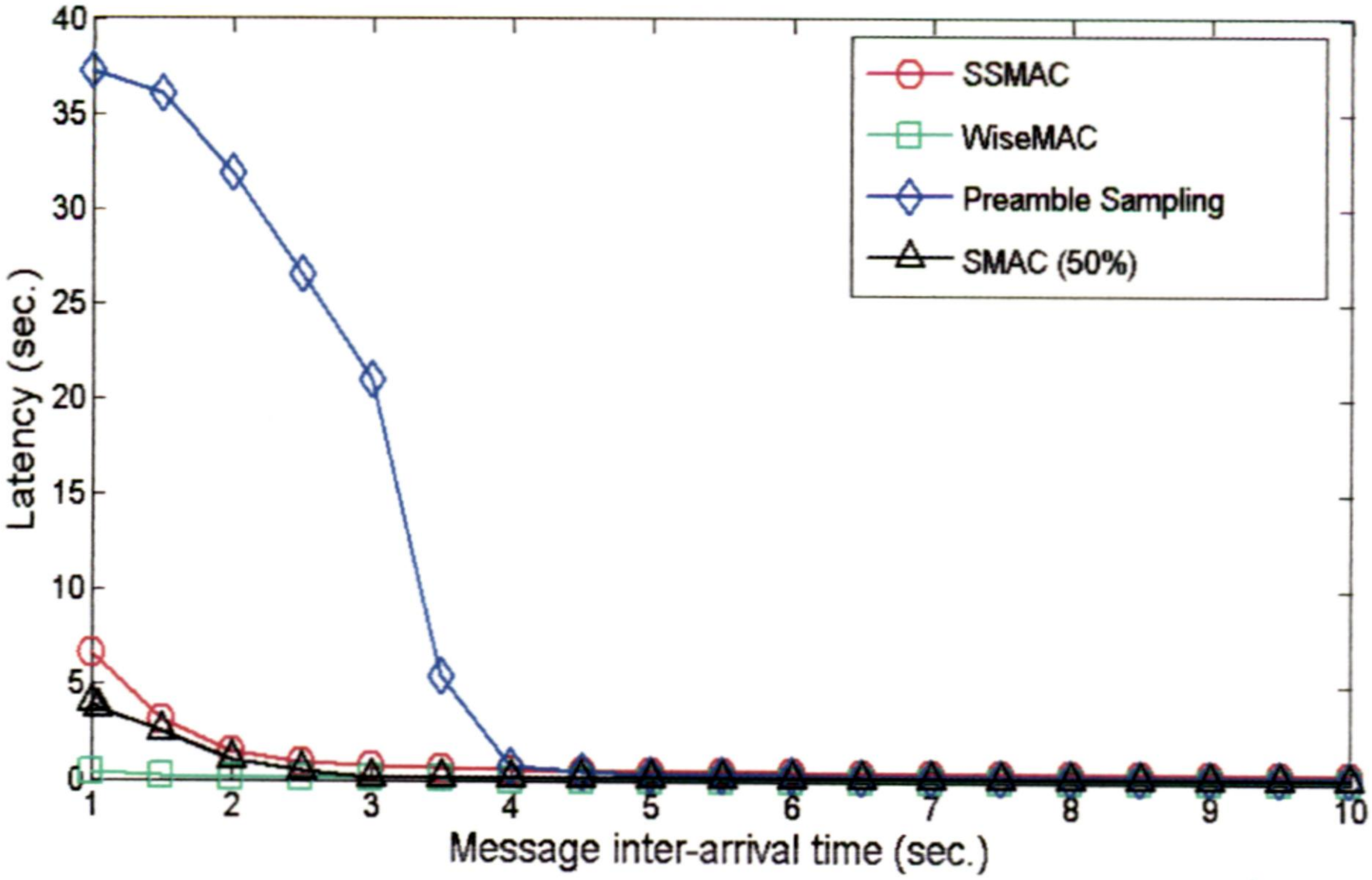

Fig. 5.7: Comparison of latency for different MAC protocols

Chapter 6

CONCLUSION AND SCOPE FOR FUTURE WORK

Energy efficiency is the primary issue in the design of wireless sensor networks. A proper designing of MAC layer is required to improve the network performance as it directly controls the radio which is the main power consumer. SSMAC improves the energy efficiency by avoiding data frame collisions due to hidden terminals. Hidden terminal problem is solved by allowing nodes to transmit their data in round-robin fashion, by allocating time slots in specific fashion, and by using WUF/CTS/DATA/ACK procedure for data transmission. SSMAC outperforms SMAC, WiseMAC and Preamble sampling schemes in terms of energy efficiency. SSMAC improves the energy efficiency by approximately 66 percent with respect to WiseMAC at the message inter-arrival time of 3 second. At the same traffic, percentage sleep time of SSMAC improves by approximately 10 percent compared to that of WiseMAC.

This improved performance is achieved at the cost of latency introduced due to setup process for data transmission. For the above mentioned traffic, the latency is increased by approximately .4 second. For most of the applications such amount of latency does not matter and it is acceptable with improvement in the lifetime of the network.

This dissertation work can be extended for future work in many ways. Performance of SSMAC can be studied for a fading channel between the sensors. SSMAC requires three transmissions to broadcast any data frame. So SSMAC can be improved and studied for broadcasting case. A detailed study can be done for SSMAC with more than 3 time slots. SSMAC can further improve network performance if it is combined with an efficient routing protocol as routing is also a major factor in determining energy efficiency. Although this cross layer designing can improve energy efficiency, it may increase the computational complexity at sensor nodes. So a joint study of SSMAC and efficient routing protocols can be done to improve energy efficiency.

REFERENCES

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine, Volume 40, Issue 8, pp. 102 – 114, August 2002.
2. G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors", Communication. ACM, Volume 43, Issue 5, pp. 51-58, May 2000.
3. K. Römer and F. Mattern, "The Design Space of Wireless Sensor Networks", IEEE Wireless Communications Magazine, Volume 11, Issue 6, pp. 54-61, December 2004.
4. I. Mahgoub and M. Ilyas, "Smart Dust: Sensor network Applications, Architecture, And Design", 1st Edison, CRC press.
5. W. Y. J. Heidemann and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks", IEEE INFOCOM 2002, Joint Conference of the IEEE Computer and Communications Societies. IEEE Proceedings, Volume 3, pp. 1567-1576, CA, USA.
6. A. El-Hoiydi, "Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks", Proc. IEEE Int'l Conf. Comm. (ICC), 2002, Volume 5. pp. 3418 – 3423, Neuchatel, Switzerland
7. A. El-Hoiydi and J.D. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks", Proc. Int'l Symp. Computers and Comm. (ISCC), Volume 1, July 2004, pp. 244-251, Neuchatel, Switzerland.
8. K. Pahlavan and P. Krishnamurthy, "Principles of Wireless Networks: A Unified Approach", 2nd Edison, Pearson Education.
9. T. Rappaport, "Introduction to Wireless Communication Systems", 2nd Edison, Prentice-Hall.
10. The Institute of Electrical and Electronics Engineers, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Standard 802.11, June 1997.
11. J.C. Haartsen, "The Bluetooth Radio System", IEEE Personal Communication Magazine, pp. 28–36, February 2000.
12. Bluetooth SIG Inc., Specification of the Bluetooth system: core, http://www.bluetooth.org, 2001.

13. Crossbow Inc., Expected Battery Life vs. System Current Usage and Duty Cycle, www.xbow.com/Support/Support_pdf_files/PowerManagement.xls, Energy specifications for MICA motes, 2003.
14. Crossbow Inc., Data sheet for MICA2 Wireless Measurement System, 2003.
15. L. Bao and J. J. Garcia-Luna-Aceves, "A New Approach to Channel Access Scheduling for Ad hoc Networks", Conference on Mobile Computing and Networking, pp. 210-221, November 2001, Rome, Italy.
16. V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, "Energy Efficient, Collision Free Medium Access Control for Wireless Sensor Networks", SenSys'03, pp. 181-192, November 2003, Los Angeles, USA.
17. S. C. Ergen and P. Varaiya, "PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks", IEEE Transactions On Mobile Computing, Volume. 5, Issue 7, pp. 920-930 July 2006.
18. B. H. Liu, N. Bulusu, H. Pham and S. Jha, "CSMAC: A Novel DS-CDMA Based MAC Protocol for Wireless Sensor Networks", IEEE GlobeCom Workshops, pp. 33-38, December 2004, NSW, Australia.
19. S. Singh and C. Raghavendra, "PAMAS: Power Aware Multi-access Protocol with Signaling for Ad Hoc networks", Volume 28, Issue 3, pp. 5-26, July 1999,.
20. T.V. Dam and K. Langendoen, "An Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks", SenSys'03, pp. 171-180, November 2003, Los Angeles, USA.
21. J.M. Rabaey, M.J. Ammer, J.L. da Silva, D. Patel, and S. Roundry, "PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking", Computer, Volume 33, Issue 7, pp. 42-48,July 2000, Berkeley, CA.
22. X. Shi, G. Stromberg, Y. Gsottberger, and T. Sturm, "Wake-Up- Frame Scheme for Ultra Low Power Wireless Transceivers", Proc. GLOBECOM, Volume 6, pp. 3619-3623, November 2004, Munich, Germany.
23. http://www.omnetpp.org/
24. Product specification, http://www.rfm.com/products/data/tr1001.pdf.
25. http://www.cs.berkeley.edu/ awoo/smartdust/.