A NEW CONTENTION BASED MULTI CHANNEL MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree

of

INTEGRATED DUAL DEGREE

in

COMPUTER SCIENCE AND ENGINEERING

(With Specialization in Information Technology)

By

NAMAN GOEL

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

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work is being presented in the disseration work entitled "CONTENTION BASED MULTI CHANNEL MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS" towards the partial fulfilment of the requirement for the award of the degree of **Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology)** submitted to the **Department of Electronics and Computer Engineering, Indian Institue of Technology Roorkee, India** is an autentic record of my own work carried out during the period from May, 2010 to June, 2011 under the guidance and provision of **Dr. Manoj Misra, Professor, Department of Electronics and Computer Engineering, IIT Roorkee.**

I have not submitted the matter embodied in this dissertation work for the award of any other degree and diploma.

Date: June, 2011

Place: Roorkee

(NAMAN GOEL)

CERTIFICATE

This to certify that the declaration made by the candidate is correct to the best of my knowledge and beleif.

Date: June,2011

Place: Roorkee

Dr. Manoj Misra
Professor
E&CE Department
IIT Roorkee, India

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor **Dr. Manoj Misra**, Professor, Department of Electronics and Computer Engineering, Indian Institue of Technology Roorkee, for his trust in my work, his able guidance, regular source of encouragement and assistance throughout this dissertaion work. I would state that the dissertation work would not have been in the present shape without his inspirational support and I consider myself fortunate to have done my dissertaiton under him.

I also extend my sincere thanks to **Dr. S.N. Sinha**, Professor and Head of the Department of Electronics and Computer Enginnering for providing facilities for the work.

I would like to thank all my friends who supported and encouraged me to finsih this work.

Finally, I would like to say that I am indebted to my parents for everything that they have given to me. I thank them for sacrifices they made so that I could grow up in a learning environment. They have always stood by me in everything I have done, providing constant support, encouragement, and love.

NAMAN GOEL

List of Figures

Abstract

Wireless Sensor Networks is an emerging technology that has potential usage in a large variety of applications. These networks consist of a large number of sensor nodes. Each of these sensor node is equipped with sensor, a small microprocessor and a low power radio. These nodes are generally battery operated and communicate wirelessly. In most of the scenarios it is not feasible to replenish the battery once expired, hence energy efficiency is an important goal for all the sensor network protocols. Also, the emerging advanced applications of wireless sensor networks require high throughput and low latency, which demands effective use of the available bandwidth. This is the responsibility of a MAC protocol.

Most of the MAC protocols proposed till now consider only a single channel of operation. But low power low cost radios with abilities to work on multiple channels are already available. These radios cannot transmit and receive at the same time but can switch their frequency/channel dynamically. These channels are orthogonal to each other. This provides one more degree of freedom for wireless communication which is yet unexploited.

In this work we have proposed a contention-based MAC protocol using channel switching. In this protocol we divide the time frame into slots. The nodes then negotiate a slot and a channel value for communication and then multiple communications are carried out in parallel. This protocol is implemented in Castalia simulator and simulation results have been compared with the traditional single channel SMAC and TMAC protocols. Our protocol clearly indicates the improvement in latency and energy consumption by effective use of the channels.

Table of Contents

Chapter 1 - Introduction

1.1 Introduction

Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications, and digital electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate untethered in short distances [1]. These tiny sensor nodes consist of sensing, data processing, and communicating components. The sensing electronics measure ambient conditions related to the environment surrounding the sensor and transforms them into an electric signal. Processing such a signal reveals some properties about object located and/or events happening in the vicinity of the sensor. For example, these nodes can collect information about the temperature, humidity, vibration, pressure etc. These sensors communicate with each other over wireless channels. A large number of these disposable sensor nodes can be networked in many applications that require unattended operations and such a network is called wireless sensor network.

Wireless sensor network is an emerging technology that has potential usage in environment monitoring, defense, smart spaces, scientific application, medical systems and robotic exploration, target tracking, intrusion detection, wildlife habitat monitoring, climate control and disaster management etc. The sensor nodes are battery operated and are equipped with a low power radio. Low power capacities of sensor node results in limited coverage and communication range for sensor nodes compared to other mobile devices. Hence, to successfully cover the target area, sensor networks are composed of large number of nodes, which coordinate to perform a common task.

In general, it is not possible to recharge or replace the exhausted batteries for sensor nodes in WSNs, which is the primary objective of maximizing node or network lifetime, leaving the other performance metrics as secondary objectives. Energy efficiency is a critical issue in wireless sensor networks since batteries are the only energy source to power the sensor nodes. In a sensor node, there are three activities that are main sources of energy consumption. Those activities are –

(i) sensing,

(ii) computation, and

(iii) radio operations.

Out of those three sources, energy loss due to radio operation is the maximum one. Not only transmitting costs energy; receiving, or merely scanning the wireless channel for communication, can use up to half as much, depending on the type of radio. In wireless sensor networks, the radio operations are controlled by the MAC layer and thus it is a good area of research for obtaining energy efficiency. In particular MAC protocols must minimize the radio energy costs in sensor nodes keeping the other parameters like latency and throughput within acceptable limits.

Emerging applications such as intruder detection, structural health monitoring, etc. require data transfer at a higher rate by utilizing the use of the limited bandwidth. Moreover, it becomes more common to use sensor nodes that run multiple concurrent applications, which also results in higher data rate requirements. The common use of WSN will further result in overlapping and co-existing networks will make the bandwidth an important concern for WSN [11]. The fundamental limitation on the achievable throughput is the limited reuse and/or the wastage of bandwidth due to interference and the half-duplex operation of the radios on the sensor nodes. Nowadays, multiple channels have been provisioned to mitigate the effects of interference by assigning different frequency channels to interfering links. But most of the proposed MAC protocols are designed for single channel radios only. In this work we propose a new MAC protocol that makes use of multiple channels effectively.

1.2 Problem Statement

"To design a new contention based MAC protocol for wireless sensor networks making the use of multiple channels". The following objectives were set for the design of the protocol –

- Making effective use of multiple channels
- Minimize the number of packet collisions
- Reduce the latency of the network

1.3 Organization of thesis

This thesis includes six chapters including this chapter that introduces the topic and states the problem. The rest of the thesis is organized as follows.

Chapter 2 provides an overview of wireless sensor networks and taxonomy various MAC protocols used for wireless sensor networks. It also discusses the related work and research gaps.

Chapter 3 presents the motivation and detailed description of the proposed protocol.

Chapter 4 describes the simulation model including some code snippets of implementation.

Chapter 5 discusses the simulation results obtained for the proposed scheme and gives a comparison with other previously available protocols.

Chapter 6 concludes the work and gives some suggestions for future work.

Material used (eg. research papers, manuals etc.) is given in references.

Chapter 2
Background and literature review

2.1 Introduction to wireless sensor networks

A sensor network is composed of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it. Sensor nodes coordinate among themselves to produce high quality information about the physical environment [2]. These sensor nodes communicate among each other or directly to an external sink or base-station (BS). Each sensor node has a limited computational power, radio transmission range, storage size and battery power. Sink nodes are distinguishing devices with powerful computational capacity, large memory size, high power and large transmission range so as to collect data from sensor nodes. Sink nodes act as a gateway between WSNs and end user.

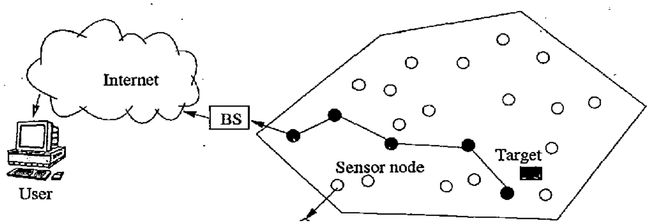


Figure 2.1 : A typical wireless sensor network

2.1.1 Comparison between WSNs and traditional Ad hoc Networks

Realization of sensor network applications requires wireless ad hoc networking techniques. Although many protocols and algorithms have been proposed for traditional wireless ad hoc networks, they are not well suited for the unique features and application requirements of sensor networks. To illustrate this point, the differences between sensor networks and ad hoc networks are outlined below:

- The number of sensor nodes in a sensor network can be several orders of magnitude higher than the nodes in an ad hoc network.
- Sensor nodes are densely deployed.
- Sensor nodes are prone to failures.
- The topology of a sensor network changes very frequently.
- Sensor nodes are limited in power, computational capacities, and memory.
- Sensor nodes may not have global identification (ID) because of the large amount of overhead and large number of sensors.

2.1.2 Applications of WSN

The small size and wireless nature of the sensors in a WSN allows it to be used in locations where either wired sensors are too costly to deploy, or the size of a traditional wireless sensor does not allow it to be placed within the confines of the environment. Typical applications of WSNs include monitoring, tracking, and controlling. There are some applications, which require a continuous data updating from the network, like pressure reading, video monitoring etc. On the contrary, there are some applications where the network is inactive for a long period of time. As soon as an event occurs, the sensor nodes become active and data transfer is initiated, like fire detection, object tracking etc. Different domains where WSNs find real applications are [1]:

Military: Since sensor networks are based on the dense deployment of disposable and low-cost sensor nodes, destruction of some nodes by hostile actions does not affect a military operation as much as the destruction of a traditional sensor, which makes sensor networks concept a better approach for battlefields. Sensor networks are able to monitor friendly forces, equipment and ammunition; battlefield surveillance; reconnaissance of opposing forces and terrain; targeting; battle damage assessment; and nuclear, biological and chemical (NBC) attack detection and reconnaissance.

Environmental: Some environmental applications of sensor networks include tracking the movements of birds, small animals, and insects; monitoring environmental conditions that affect crops and livestock; irrigation; macro-instruments for large-scale Earth monitoring and planetary exploration; chemical/ biological detection; precision agriculture; Earth, and environmental monitoring in marine, soil, and atmospheric contexts; forest fire detection; meteorological or geophysical research; flood detection; bio-complexity mapping of the environment; and pollution study.

Healthcare: Some of the health applications for sensor networks are providing interfaces for the disabled; integrated patient monitoring; diagnostics; drug administration in hospitals; monitoring the movements and internal processes of insects or other small animals; telemonitoring of human physiological data; and tracking and monitoring doctors and patients inside a hospital

Smart Home: Tiny sensor nodes can be embedded into home appliances, such as television, air conditioner, washing machine, microwave oven, refrigerator, and main power supply etc. These are able to communicate with each other and the room server to monitor and control various activities like automatic switching of air conditioner, refrigerator, microwave over and television etc; fire alarm; theft detection, remotely controlling home appliances and switches etc. These room servers and sensor nodes can be integrated with existing embedded devices to become self-organizing, self-regulated and adaptive systems to form a smart environment

Other applications: Some of the commercial applications are monitoring material fatigue; building virtual keyboards; managing inventory; monitoring product quality; constructing smart office spaces; environmental control in office buildings; robot control and guidance in automatic manufacturing environments; interactive toys; interactive museums; factory process control and automation; monitoring disaster area; smart structures with sensor nodes embedded inside; machine diagnosis; transportation; factory instrumentation; local control of actuators; detecting and monitoring car thefts; vehicle tracking and detection; and instrumentation of semiconductor processing chambers, rotating machinery, wind tunnels, and anechoic chambers

2.2 Medium Access Control Protocols for Wireless Sensor Networks

A medium access control (MAC) protocol decides when competing nodes may access the shared medium, i.e. the radio channel, and tries to ensure that no two nodes are interfering with each other's transmissions. MAC protocols for WSN usually trade off performance (latency, throughput, fairness) for cost (energy efficiency, reduced algorithmic complexity), while providing a good scalability and adaptability for topology changes. Major sources of energy waste in wireless sensor network are basically of four types [3][4] -

i. **Collision**: When a transmitted packet is corrupted due to interference, it has to be discarded and the follow on retransmissions increase energy consumption. Collision increases latency also.
ii. **Overhearing**: Picking up packets that are destined to other nodes.
iii. **Packet Overhead**: Sending and receiving control packets consumes energy too and less useful data packets can be transmitted.
iv. **Idle listening**: Listening to receive possible traffic that is not sent. This is especially true in many sensor network applications. If nothing is sensed, the sensor node will be in idle state for most of the time.
v. **Overemitting**: Transmitting a packet when the destination is not ready to receive.

The main goal of any MAC protocol for sensor network is to minimize the energy waste due to idle listening, overhearing and collision. There are many MAC protocols that have been proposed which aim to minimize the energy consumption by some or all of the above mentioned sources. All of such protocols can be broadly classified into two categories:- (i) TDMA based protocols and (ii) Contention based protocols.

2.2.1 TDMA based protocols

In the TDMA-based protocols nodes are often required to form a cluster. The system time is divided into time slots. Each of the nodes has assigned its own time slot, and may access the shared medium only in this time slot [5]. This allows avoiding collisions, idle listening, and schedules sleep of the transceiver, without additional overhead. However, such an approach provides a number of drawbacks. The difficulty for the cluster to dynamically change its frame length and time slot assignments, in the event of node changes or node inclusions, contributes to poor scalability and poor mobility. In addition, effective slot assignment in multi-hop networks is also challenging. Furthermore, the TDMA-based protocol requires high quality time synchronization since the clock drift may lead to disastrous consequences.

Wise-MAC

WiseMAC [6] protocol uses non-persistent CSMA (np-CSMA) with preamble sampling as in to decrease idle listening. In the preamble sampling technique, a preamble precedes each data packet for alerting the receiving node. All nodes in a network sample the medium with a common period, but their relative schedule offsets are independent. If a node finds the medium busy after it wakes up and samples the medium, it continues to listen until it receives a data packet or the medium becomes idle again. The size of the preamble is initially set to be equal to the sampling period. However, the receiver may not be ready at the end of the preamble, due to factors such as interference, which causes the possibility of overemitting-type energy waste. Moreover, overemitting is increased with the length of the preamble and the data packet, since no handshake is done with the intended receiver. To reduce the power consumption incurred by the predetermined fixed-length preamble, WiseMAC offers a method to dynamically determine the length of the preamble. That method uses the knowledge of the sleep schedules of the transmitter node's neighbors. The nodes learn and refresh their neighbor's sleep schedule during every data exchange as part of the acknowledgment message. In that way, every node keeps a table of the sleep schedules of its neighbors. Based on the neighbor's sleep schedule tables, WiseMAC schedules transmissions so that the destination node's sampling time corresponds to the middle of the sender's preamble. To counter drifting clocks, the potential clock drift between the sender and the receiver, which is a multiple of the time since the last

ACK packet arrived, is also taken into consideration while calculating the preamble length.

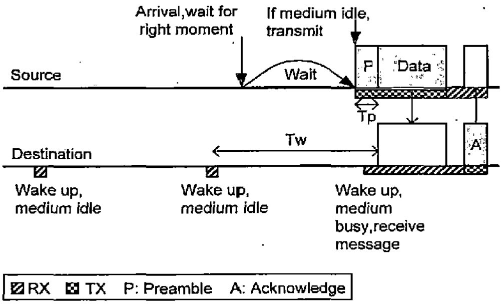


Figure 2.2: Concept used in WiseMAC

Simulation results show that WiseMAC performs better than one of the S-MAC variants [6]. Besides, its dynamic preamble length adjustment results in better performance under variable traffic conditions. In addition, clock drifts are handled in the protocol definition, which eliminates the requirement of external time synchronization.

Main drawback of WiseMAC is that decentralized sleep-listen scheduling results in different sleep and wake-up times for each neighbor of a node. This is especially an important problem for broadcast type of communication, since broadcasted packet will be buffered for neighbors in sleep mode and delivered many times as each neighbor wakes up. However, this redundant transmission will result in higher latency and power consumption. The hidden terminal problem is also present with WiseMAC model as in the Spatial TDMA and CSMA with Preamble Sampling algorithm. That is because WiseMAC is also based on non-persistent CSMA. This problem will result in collisions when one node starts to transmit the preamble to a node that is already receiving another node's transmission where the preamble sender is not within the range.

DMAC

The Data–Gathering Medium Access Control (D-MAC) [7] is a schedule based MAC protocol which has been designed and optimized for tree based data gathering (converge cast communication) in wireless sensor network. The main objective of this MAC protocol is to achieve low latency and still maintaining the energy efficiency. In this protocol the time is divided in small slots and runs carrier sensing multiple access (CSMA) with acknowledgement within each slot to transmit/receive one packet. The sensor node periodically executes the basic sequence of '1' transmit, '1' receive and 'n' sleep slots. In this approach a single packet from a source node at depth 'k' in the tree reaches the sink node with a delay of just 'k' time slots. This delay is very small and it is in the order of tens of milliseconds. A data-gathering (converge cast) tree with staggered DMAC slots is shown in the Fig. 2.3. D-MAC includes an overflow mechanism to handle the problem when each single source node has low traffic rate but the aggregate rate at intermediate node is larger than the basic duty cycle. In this mechanism the sensor node will remain awake for one extra time slot after forwarding the packet. Therefore, if two children were contending for parents receive slot, the loosing child will get a second chance to send its packet. The simulation results shows that the D-MAC protocol outperforms the Sensor S-MAC protocol in terms of energy efficiency, latency and throughput in both multi-hop chain topology and random data gathering tree topology.

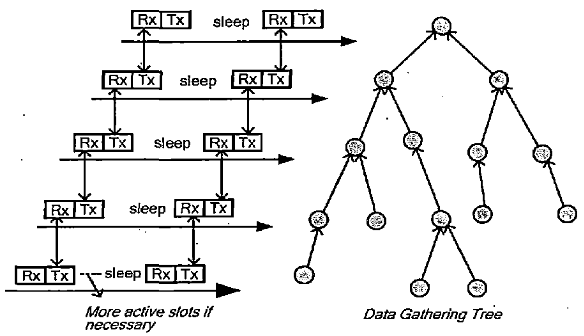


Figure 2.3: Data Gathering tree in DMAC

2.2.2 Contention based protocols

The contention-based protocols allow nodes for the independent access to the shared wireless medium [5]. Nodes are not required to form a cluster. These protocols inherit good scalability, and support node changes and new node inclusions. However, a node is not able to know when to switch its radio to a proper state. Sleeping mechanism becomes rather complex, and to avoid unnecessary energy consumption, while preserving desired latency and throughput, it requires control overhead to keep neighbor nodes synchronized. That is, idle listening, collisions, overhearing, and control packet overheads are the major sources of energy inefficiency.

Sensor-MAC (SMAC)

Sensor S-MAC [8] a contention based MAC protocol is modification of IEEE 802.11 protocol specially designed for the wireless sensor network in 2002. In this medium access control protocol sensor node periodically goes to the fixed listen/sleep cycle. The basic idea behind SMAC protocol is locally managed synchronizations and periodic sleep–listen schedules based on these synchronizations. Neighboring nodes form virtual clusters so as to set up a common sleep schedule. If two neighboring nodes reside in two different virtual clusters, they wake up at the listen periods of both clusters. A time frame in SMAC is divided into to parts: one for a listening session and the other for a sleeping session. Only for a listen period, sensor nodes are able to communicate with other nodes and send some control packets such as SYNC, RTS (Request to Send), CTS (Clear to Send) and ACK (Acknowledgement). By a SYNC packet exchange all neighboring nodes can synchronize together and using RTS/CTS exchange the two nodes can communicate with each other. Further the listen period of each node is divided into two parts SYNC and DATA communication period. Schedule exchanges are accomplished by periodic SYNC packet broadcasts to immediate neighbors during the SYNC part of the listen interval. Unicast data packets are exchanged with the RTS/CTS mechanism which is used for both collision and overhearing avoidance. Periodic sleep scheme followed by SMAC may result in high latency, especially for multihop routing algorithms, since all intermediate nodes have their own sleep schedules.

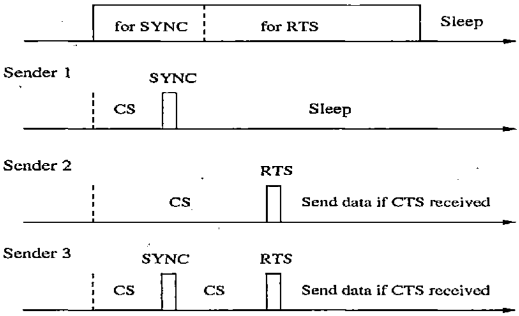


Figure 2.4: Timing relationship in SMAC

Timeout-MAC (TMAC)

TMAC [9] is the protocol based on the SMAC which was proposed to enhance the poor results of SMAC protocol under variable traffic loads. TMAC improves on S-MAC's energy usage by using a very short listening window of time 'Ta' at the beginning of each active period. The interval Ta > Tci + Trt + Tta + Tct where Tci is the length of the contention interval, Trt is the length of an RTS packet, Tta is the turn-around time (time between the end of the RTS packet and the beginning of the CTS packet) and Tct is the length of the CTS packet. After the SYNC section of the active period, there is a short window to send or receive RTS and CTS packets. If no activity occurs in that period, the node returns to sleep. By changing the protocol to have an adaptive duty cycle, TMAC saves power at a cost of reduced throughput and additional latency. TMAC, in variable workloads, uses one fifth the power of SMAC [9]. In homogeneous workloads, TMAC and SMAC perform equally well. TMAC suffers from the same complexity and scaling problems of SMAC. Shortening the active window in T-MAC reduces the ability to snoop on surrounding traffic and adapt to changing network conditions.

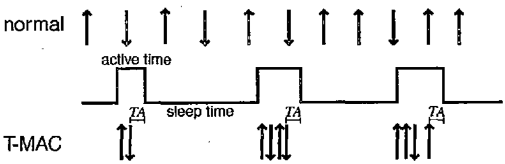


Figure 2.5: Basic TMAC scheme with adaptive active time. Arrows indicate transmitted and received messages.

BMAC

B-MAC [10] is the protocol that is implemented and used in TinyOS due to its small implementation and code, which reduces the RAM size requirements. B-MAC also has a set of interfaces that allow services to tune its operation in addition to the standard message interfaces. It uses Clear Channel Assessment (CCA) and packet backoffs for effective collision avoidance, and provides optional link layer acknowledgement support for reliability. B-MAC duty cycles the radio through periodic channel sampling that we call Low Power Listening (LPL). Each time the node wakes up, it turns on the radio and checks for activity. If activity is detected, the node powers up and stays awake for the time required to receive the incoming packet. After reception, the node returns to sleep. If no packet is received (a false positive), a timeout forces the node back to sleep.

The set of interfaces provided by BMAC allow network services to adjust its mechanisms, including CCA, acknowledgments, backoffs, and LPL. Because B-MAC is lightweight and configurable, many sensor network protocols may be implemented efficiently using its primitives.

2.3 Use of multiple channels in wireless sensor networks

2.3.1 Challenges of using multiple channels in WSNs

Following are some challenges faced by protocols using multiple channels in wireless sensor networks[12]

- Synchronization: If the channel assignment is done dynamically, i.e. the radios are switching between channels instead of being fixed on one channel, a detailed coordination of channel switching is required between the senders and receivers in order to be on the same channel at the same time.
- Partitions: If transceivers of two nearby nodes are fixed on different frequencies, they cannot communicate with each other.
- Joining the network: A new node joining the network may disrupt the channel organization or may be required to scan all the channels to find the suitable channel to transmit.
- Broadcast support: If the nodes are switching between channels dynamically, it might be problematic to support local broadcasts. However, local broadcasts are important for WSN traffic since sensor nodes may require in-network processing before they transmit the data towards the sink node.
- Channel switching: The radio cannot switch between the channels immediately but takes some time, for instance it is around 650μsec for Nordic Nrf905 radio. So MAC protocols must account for this time in their design

2.3.2 Protocols using multiple channels

Few protocols have been proposed using multiple channels in wireless sensor network. In this subsection we present some of them.

MMSN

Zhou et al. [13] introduced the MMSN multi-frequency MAC protocol especially designed for WSN in 2006 which was the first protocol in this domain. It is a slotted CSMA protocol and at the beginning of each timeslot nodes need to contend for the medium before they can transmit. MMSN assigns channels to the receivers. When a node intends to transmit a packet it has to listen for the incoming packets both on its own frequency and the destination's frequency. A snooping mechanism is used to detect the packets on different frequencies, which makes the nodes to switch between channels frequently. MMSN uses a special broadcast channel for the broadcast traffic and the beginning of each timeslot is reserved for broadcasts.

TFMAC

TFMAC[14] is a TDMA based multi channel MAC for wireless sensor networks. It requires Time Synchronization and it uses single half duplex transceiver. This protocol divides each channel into time slots and the slot scheduling has been done for the medium access. The frame has been divided into contention access period where the slot scheduling and channel allocation is done and contention free period where the data transfer is done. Initially each node choses its reception frequency and broadcasts it to all its neighbors. Each active node establishes a timetable during the activation time using the timetables and reception frequencies of all its neighboring nodes. The timetable for each node is set up such as each sender and receiver pair has a unique time and frequency usage combination. This protocol suffers from the limitations of general TDMA schemes i.e. adaptability to traffic, scalability and latency.

MCSMAC

MCSMAC[15] is a contention based protocol for wireless sensor networks which is based on the SMAC protocol. It has fixed periodic listen and sleep intervals. Handshaking is done using the RTS/CTS mechanism used in SMAC[8] and 802.11 All nodes are on the same channel at beginning of each listen interval and contend the medium for sending the RTS packet which offers a channel to the receiver. Receiver uses carrier sensing to determine if the channel is free and respond with CTS only if the channel is available. Then both nodes switch at the channel and transfer of data packet and acknowledgement is done. This protocol suffers from multi-channel hidden terminal problem as after a transfer both nodes loose the channel information. Also no mechanism has been provided to check whether channel switching is done. This may lead to overemitting type waste of energy.

2.4 Research Gaps

In the recent years, a large number of medium access control (MAC) protocols for the wireless sensor networks have been published by the researchers. Most of the work on the MAC focuses primarily on the energy efficiency in the sensor network and the use of a single channel. Nowadays multiple channel radios are commonly available and not much work has been done in this area. Some of the research gaps that have been identified are:

- Contention based multi channel protocols – There are a few protocols that have been proposed till now which use multiple channels and most of these protocols are TDMA based. Not much work has been done on contention based multi channel protocols.
- Adaptability – Most of the multi channel protocols that have been proposed are not adaptive to the changing traffic conditions.
- Fairness – Most of the protocols that have been developed so far assume a single application running on all the nodes and do not consider fairness. However, nowadays it is becoming common to use sensor nodes with multiple concurrent applications running on them. In such cases fairness becomes important.
- Nodes Mobility – Most protocols assume the nodes to be static and do not consider changes in network topology.
- Evaluations on Sensor Platforms – Most of the protocols for the wireless sensor networks have been evaluated through the simulations. However, the performance of the MAC protocol needs to be evaluated on the actual sensor systems on which results can vary significantly.

Chapter 3
Slotted Multi Channel Medium Access Control Protocol (SMC-MAC)

3.1 Motivation

MAC protocols have an important role in energy efficiency and bandwidth utilization of a network. Most of traditional MAC protocols for WSNs are designed for an environment where sensor nodes use simple, low-cost transceivers that can operate on a single frequency (channel), only. On the other hand, the current low-cost, low-energy transceivers, such as CC2420 radio [16], already support multiple frequencies. Such transceiver cannot transmit and receive at the same time, but it can switch its frequency dynamically. Availability of multiple channels adds one more degree of freedom to wireless communications that can be exploited to increase the spatial reuse by providing more simultaneous transmissions than is possible in single-channel WSNs. Thus, network throughput can potentially be increased.

Although there have been some proposals of protocols using multiple channels but most of these are TDMA based which require external time synchronization and are not adaptive to the changing traffic. This leads to poor channel utilization. Thus we propose a contention based MAC protocol that makes use of availability of multiple channels.

3.2 System assumptions

The following are the assumptions that have been taken for designing the SMC-MAC protocol -

- All the nodes are symmetric and have same initial energy.
- There is no interference, other than the interference generated by node packet transmissions.
- The sensor nodes may/may not be mobile.

- Each of the sensor node is equipped with an omnidirectional radio with dynamic channel switching capabilities.
- Each node has its own independent clock and clocks of neighboring nodes may drift apart with a known maximum rate.
- All sensor nodes are running a common application and we do not need to consider node level fairness.

3.3 Proposed Protocol

3.3.1 Frame structure and the basic scheme

SMC-MAC divides the whole timeline into frames. Each frame has an active period, during which the nodes exchange messages, and an inactive period during which the nodes sleep. This basic scheme of periodic activity has been inspired from SMAC protocol. The basic frame structure of the SMC-MAC protocol has been shown in Fig 3.1.

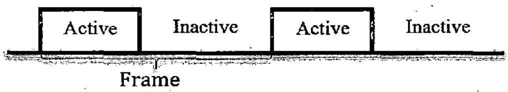


Figure 3.1: Basic scheme for SMC-MAC protocol

Each active period is further split into different slots. Different time slots in a frame are reserved for different purposes. A detailed splitting of a frame into time slots can be seen in Fig 3.2. First part of the active period is reserved for synchronization. Next part of the frame is control slot in which all the negotiations are done for data packet exchange which is later done in one of the data slots. During the synchronization and the control slot all the nodes are tuned to the same channel and this channel is called the control channel. The control slot is followed by 'n' number of data slots. During each data slot, a node may be involved in an exchange (transmit or receive) of data, otherwise it goes to the sleep state and turns its radio off. The duration of the data slot is based on the maximum size of the data packet that can be received from the network layer, the channel switching time and some relaxation for the clock drift.

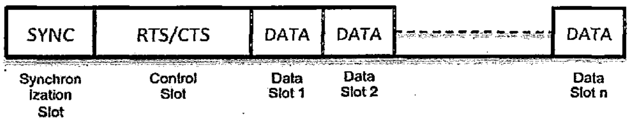


Figure 3.2: Splitting of active time into slots

3.3.2 Synchronization

Our scheme requires periodic synchronization among neighboring nodes to remedy their clock drift. All the neighboring nodes follow the same schedule. The sink node broadcasts a SYNC message to its neighbors after choosing a schedule. Each node that receives the schedule further broadcasts it to its neighbors. The SYNC message contains two things: -

- Next Frame Start – This is a relative time for the start of the next frame and is calculated as the absolute time for start of the next frame minus the current time. Using this relative time for the frame start removes the dependency on the current absolute time at each node.
- Sequence number – Each time the source node sends a new SYNC packet it increments its sequence number in order to control the flooding of the SYNC message in the network.

3.3.3 Negotiation

As the transmission of DATA packets is done during the data slots and also there are multiple channels available for the transmission of data, thus for every data packet to be transferred the two communicating nodes must negotiate a slot and a channel to transmit. This negotiation is carried out during the control slot. In this slot each node's radio is tuned to a fixed common channel. For keeping the information about the busy slots and channels each node maintains three data structures: -

- Busy Slot Vector – This data structure is group of 'n' bits, where 'n' is the number of slots and the ith bit indicates if the node has already scheduled a transmission in slot number 'i'. Only the data slots that are free can be used for scheduling new transmissions as a node can only participate in one data transmission at a time.
- Selected Channels List – This is list of all the channels which have been selected for communication during the slots during which this node will remain active.
- Busy Channel Vector – This data structure has 'n' groups of 16 bits each, where 'n' is the number of slots. Here for each slot 'i' the corresponding 16 bits indicate the status of the 16 channels during that slot. If channel 'j' is busy during the ith slot then the jth bit in the ith group would be set to 1.

Next we will discuss the working of the sending and the receiving node during the negotiation phase.

Working of sender

Firstly each node that has a data to send contends the common medium for sending an Request to Send (RTS) message to the receiving node. The node waits for a random amount of time and senses the carrier for activity during that time. If no activity is detected during that time the node sends an RTS packet to destination. Along with the source, destination and data packet sequence number information the RTS packet also contains the slot vector and the busy channel vector of the source node. The receiver then finds out a common slot and channel for transmission and replies with this information to the sending node in a Clear to Send (CTS) packet. After receiving a CTS packet the nodes makes entries in the slot vector and selected channels list. If there is no common slot channel available for transmission then the receiver doesn't reply with the CTS, in this case the sending node times out and again goes back to contending the medium.

Working of receiver

The receiving node is listening to the medium during the control slot. When it receives a RTS packet destined to it, it extracts the slot vector and receiver channel vector. Then it calculates a slot and channel combination that is free for both the nodes involved using the mechanism given in the following pseudo-code

```
findCommonSlotChannel(receivedSlotVector,receivedChannelVector[]) begin

    obtain temporarySlotVector as
            temporartSlotVector = ownSlotVector or receivedSlotVector

    for every ith bit that is unset in the temporarySlotVector begin

            obtain temporaryChannelVector as
            temporaryChannelVector=receivedChannelVector[i]orownChannelVector[i]

            if there exists a unset bit in temporaryChannelVector then begin
                    selectedChannel = first unset bit in the temporaryChannelVector
                    selectedSlot = i
                    return (selectedSlot,selectedChannel)
            end if

    end for

    return error : no common slot,channel available

end
```

Figure 3.3: Pseudocode for selection of a common slot and channel

If a common slot and channel combination is found these are sent to the sending node in the CTS packet and it also makes an entry into its own slot vector about the slot negotiated and an entry into the selected channels list.

The following figure shows the exchange of control messages

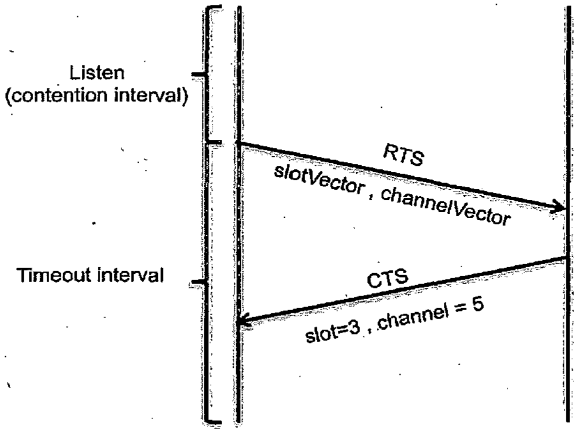


Figure 3.4: Exchange of messages between sender and the receiver during negotiation

3.3.4 Data Packet Transfer

If a node has negotiated a data transfer during the control slot some data slots it wakes up when the timer for those particular slots fire. After waking up a node immediately switches to the negotiated channel. As waking up and switching the channel may consume some time that may not be equal for both the nodes and also due to some clock drift the wake up times of both nodes may slightly vary. So the sending node needs to make sure that the receiving node is ready to accept the data before sending the data packet. So when it wakes up, it sends a CHECK_ALIVE message to the receiving node waits for some time for receiving node to reply. If no reply is heard then it sends the message again and this process goes on. When the receiving node wakes up and switches the channel it waits for the CHECK_ALIVE message from the other node and replies with a TELL_ALIVE message telling the sending node that it is ready to receive the data. Then sending node sends the data to the receiving node, which sends an ACK packet back to the sender to indicate successful transfer. If the sender does not receive an ACK packet it tries to send the packet again during the next frame.

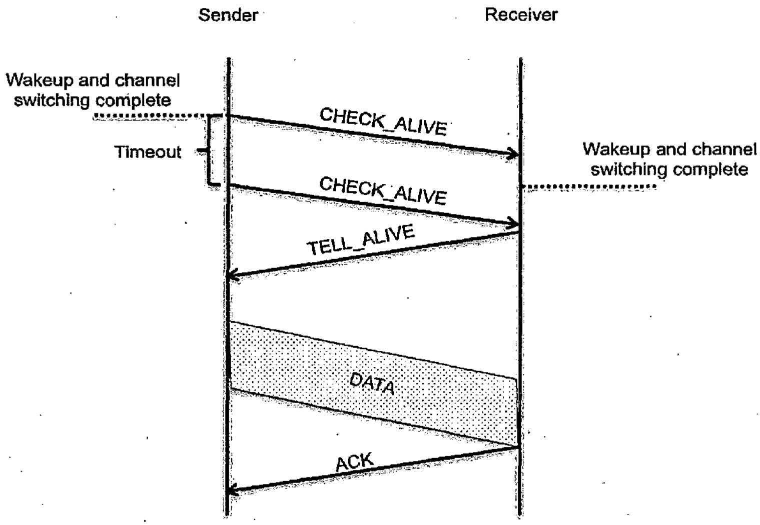


Figure 3.5: Exchange of messages between communicating nodes during data packet transfer in a data slot

3.3.5 Collision Avoidance

Collisions during the control slot are prevented by using a random value of time for which a node senses the medium. But collisions may also occur during a data slot if two neighboring nodes are using the same data channel during the same data slot for their transfers. This is taken care of at the time of negotiations using virtual carrier sensing. Virtual carrier sensing is different from physical carrier sensing as an actual sensing of the carrier is not required by the node. Each node gathers the information about the slot and channel pair used by its neighboring nodes by overhearing their control packets. CTS packets have the exact information about the selected slot and channel but the RTS packet does not have such information. So the neighboring nodes of the sending node cannot get the information about the slot and channel used by the node just by overhearing the RTS packet. For this purpose a special packet has been introduced which is called the CONFIRM packet. The sending node broadcasts this packet when the negotiation is complete. The confirm packet contains information about the selected slot and selected channel that have been negotiated by the two nodes.

Now when an overhearing node hears a RTS packet it waits for some time to hear a corresponding CONFIRM packet in case the negotiation is successful. It also starts a timer for timeout if CONFIRM packet does not arrive.

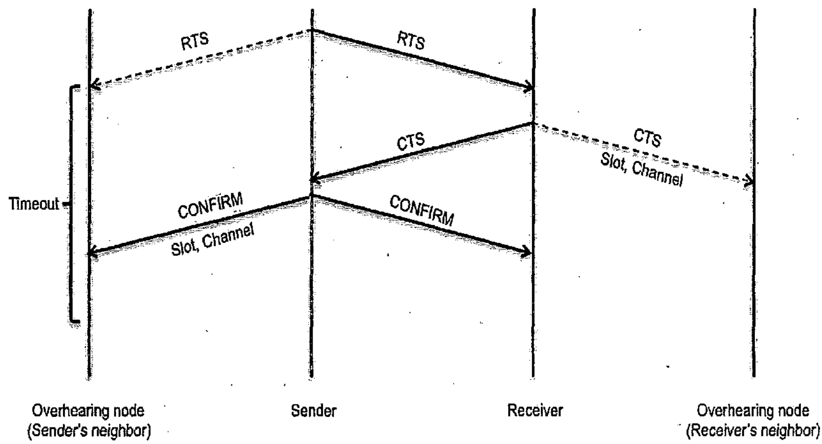


Figure 3.6: Slot/Channel information collection by overhearing nodes

Chapter 4
Simulation Model and Implementation

4.1 Introduction to Castalia

Simulator used for the purpose of all the simulations is Castalia 3.2. Castalia [17] is a simulator for Wireless Sensor Networks (WSN), Body Area Networks (BAN) and generally networks of low-power embedded devices. It is based on the OMNeT++ platform. Basic features that are available in Castalia are: -

- Advanced channel model based on empirically measured data supporting temporal variation of path loss and mobility of nodes.
- Advanced radio model based on real radios for low-power communication.
- Extended sensing modeling provisions providing highly flexible physical process model.
- Node clock drift
- Some popular protocols are available
- Designed for adaptation and expansion.

These features make Castalia a good choice for a simulator to be used by researchers and developers who want to test their distributed algorithms and/or protocols in realistic wireless channel and radio models, with a realistic node behavior especially relating to access of the radio. Castalia can also be used to evaluate different platform characteristics for specific applications, since it is highly parametric, and can simulate a wide range of platforms.

4.2 Simulation Model

4.2.1 Communication patterns used for testing

Kulkarni defines three types of communication patterns in wireless sensor networks [18]:

- Broadcast - A broadcast pattern is generally used by a base station (sink) to transmit some information to all the sensor nodes of the network. Broadcasted information may include queries of sensor query-processing architectures, program updates for sensor nodes, or control packets for the whole system.
- Local gossip – Local gossip pattern is used by neighboring nodes to exchange information about the state of the nodes or to exchange information about an event that some nodes may have detected.
- Convergecast – This pattern is used by the sensing nodes to report the perceived information to a central node. This central node may be a clusterhead, a data fusion center or a base-station.

Broadcasts are generally performed on the control channel during the control slot so this pattern is not used for comparison purpose. The other two patterns, i.e. Local Gossip and Convergecast, have been used for comparison. Different networks have been constructed for both patterns that we described later.

4.2.2 Network model

Features of the network model that we have used for simulations are as follows –

- The number of nodes used for simulation is in the range of 20 to 100 nodes.
- The area of the network is rectangular.
- The kind of deployment used for the nodes is uniform for the local gossip pattern and grid for the convergecast pattern.
- All nodes are static for all simulations.
- The transmission power of all the nodes is fixed and the transmission range of each node is approximately 25m.
- All nodes run a same single application that generates packets at a fixed rate, which is passed as a parameter at the runtime.

4.3 Modeling of a node

A node is Castalia is modeled as a composite module consisting of many other modules: Sensors Manager module; Application module; Communication module; Resource Manager module and Mobility Manager module. The Communication module is again a composite module consisting od Routing, MAC and Radio modules. The interaction among these modules and the physical process and wireless channel is shown in Fig 4.1.

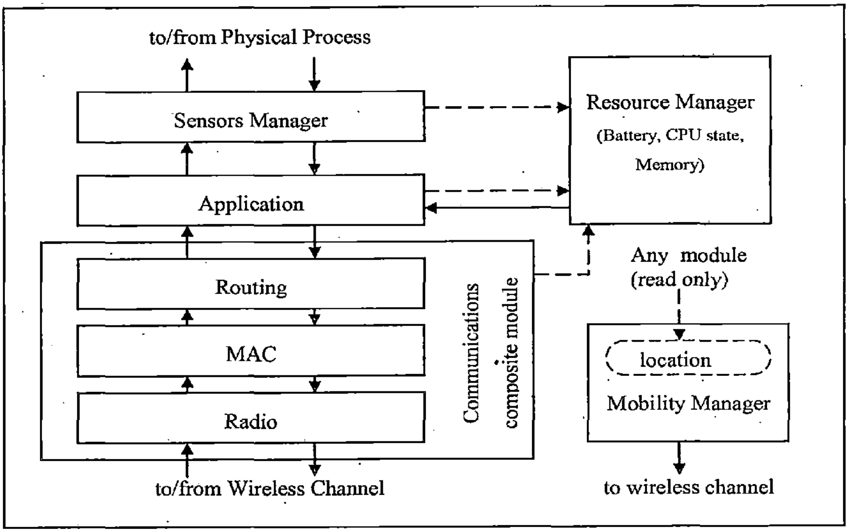


Figure 4.1: Composition of node module

In the above figure the solid arrows signify message passing and the dashed arrows signify simple function calling. Castalia provides some implementations of all the above given modules and allows customizations and provides facilities for new implementations of all of the modules.

The simulation work for this dissertation was divided into two parts

- Developing a new MAC module using the proposed scheme for implementation of out SMC-MAC protocol
- Developing some new Application modules suitable for the network and traffic patterns for which the performance comparison has been done

4.4 Implementation of SMC-MAC module

A new module implementation in Castalia has two components: a ned file and a C++ class. The ned file describes the parameters of the module which can be changed at load time of the network. It may provide default values for some of the parameters. It also provides some communication gates of the module through which other modules interact with this module. A part of the ned file for the SMC-MAC module is given as follows-

```
package node.communication.mac.nMac;

simple NMAC like node.communication.mac.iMac {
 parameters:
     int macPacketOverhead = default (11);
     int macBufferSize = default (32);
     int maxTxRetries = default (2);

     double resyncTime = default (6);
     double contentionPeriod = default (2);
     double listenTimeout = default (15);
     double waitTimeout = default (5);
     double dataWaitTimeout = default(5);
```

```
    int syncPeriod = default (5); //in msecs
    int controlPeriod = default (30);
    double frameTime = default (270);
    int slotTime = default(20);
    int nSlots = default(8);

gates:
    output toNetworkModule;
    output toRadioModule;
    input fromNetworkModule;
    input fromRadioModule;
    input fromCommModuleResourceMgr;
}
```

The C++ implementation of the class describes the functionality of a module. In Castalia every MAC module has to be derived from the 'VirtualMAC' class provided by the simulator. The prototype of our class is given as follows –

```
class SMCMAC: public VirtualMac;
```

A large number of functions have been used to simulate full SMC-MAC protocol. In this section we present and describe some of the important ones along with their prototypes –

- timerFiredCallback – This function is called each time a timer fires. There are many timers that are used in the implementation. Timers used are given on the next page. Based on the type of timer this function calls another function to take appropriate action.

```
enum SMCMACTimers {
    SYNC_RENEW = 1,
    FRAME_START = 2,
    CONTROL_START = 3,
    CARRIER_SENSE = 4,
    TRANSMISSION_TIMEOUT = 5,
    NAV_TIMEOUT = 6,
    NEGOTIATION_DONE = 7,
    CHECK_ALIVE = 8,
    FINISH_TRANSFER = 9,
    DATA_SLOT_START = 10,
    DATA_SLOT_END = 11,
};
```

- fromNetworkLayer – this function is called whenever a packet is received from the network layer of the node. This packet is considered as DATA packet and is buffered in the MAC layer buffer for transmission.

- fromRadioLayer – this function is called when radio of a node receives a packet while listening. Two things are checked here, the destination node of the packet and the type of the packet. If the packet is a DATA packet destined to self then it is passed to the network layer otherwise appropriate control action is taken. The type of packets used are –

```
enum SMCmacPacket_type {
    SYNC_SMCMAC_PACKET = 1;
    RTS_SMCMAC_PACKET = 2;
    CTS_SMCMAC_PACKET = 3;
    DATA_SMCMAC_PACKET = 4;
    ACK_SMCMAC_PACKET = 5;
    CONFIRM_SMCMAC_PACKET = 6;
    CHECK_ALIVE_SMCMAC_PACKET = 7;
    TELL_ALIVE_SMCMAC_PACKET = 8;
};
```

- switchChannel – this function switches the current channel of the Radio to the specified channel.
- findCommonSlotChannel – this function is called when a node receives a RTS packet. It takes the received slot vector and the received channel vector as parameters and uses the own slot and channel vectors to find out a common slot and channel for the data transfer
- finishTransfer – this function is called after a data transfer is compete. This switches the node back to the default control channel and then puts the node to sleep, as it has no remaining activity in this slot.

4.5 Implementation of the Application module

Similar to MAC layer each Application module class implemented in Castalia has to be derived from the 'VirtualApplication' class provided by the simulator. The working of the module varies with the changing network and traffic, but the basic class definition is the same.

```
class myApplication: public VirtualApplication {
 private:
     double packetInterval;
     int packetSize;
     string sinkNode;
     int sequenceNumber;
     int numNodes;
protected:
     void startup();
     void timerFiredCallback(int);
     void fromNetworkLayer(ApplicationPacket ,char*)
};
```

4.6 Construction of network

After the implementation of MAC and Application modules we need to construct the testing network scenario and use these modules out network. In Castalia this is done using initialization (ini) files. This file initializes the whole network, sets the number of nodes and topology, gives the name of the module implementations to be used at each layer and can change the parameters default values for each module used. Castalia gives us the flexibility of having different sensor network configurations in the same initialization file and then choose one or multiple of these configurations to be run at a time. A part of an initialization file used in one of the scenarios is given as

–

```
include ../Parameters/Castalia.ini

sim-time-limit = 100s

SN.field_x = 20
SN.field_y = 20
SN.numNodes = 17
SN.deployment = "uniform"

SN.wirelessChannel.sigma = 0
SN.wirelessChannel.bidirectionalSigma = 0

SN.node[*].ApplicationName = "RandomizedUniformTraffic"
SN.node[*].Application.packetInterval= ${interval=3.5,3,2.5,2}
#SN.node[*].Application.packetInterval = 1
SN.node[*].Application.collectTraceInfo = false


[Config SMCMAC]
SN.node[*].Communication.MACProtocolName = "NMAC"
SN.node[*].Communication.MAC.frameTime = ${frameTime
                                =450,500,550,600,650,700,750,800,850}


[Config sensorMAC]
SN.node[*].Communication.MACProtocolName = "MMAC"

[Config TMAC]
SN.node[*].Communication.MACProtocolName = "TMAC"
```

Chapter 5
Results and Discussion

We have compared our protocol with the single channel SMAC and TMAC protocols for different network scenarios varying the load on the network. We have chosen SMAC and TMAC protocols for comparison, as these two are the most widely used protocols in wireless sensor networks.

Two metrics are used for evaluating the performance of the protocol –

- Energy Consumed – This is the average energy consumed per node in the network over the duration of the simulation.
- Latency – This is the average end-to-end delay in the network.

For the SMC-MAC protocol the energy consumed and latency vary when we change the control slot time and the number of data slots for protocol. Hence we test each network under every traffic load varying the control slot time and the number of data slots and choose the best results according to the metric under consideration.

In SMAC protocol also the latency and energy consumed vary a lot when we change the duty cycle of the frame. The latency of the network is inversely related to the energy consumed by the nodes. So under a particular traffic load we test the network varying the duty cycle of the SMAC protocol and choose the best values of the metric under comparison.

For comparison with the SMAC protocol we construct two graphs for each network. –

- Constant energy graph – In this graph we compare the latencies obtained by the two protocols when they consume the same amount of energy.
- Constant latency graph – In this graph we compare the average energy consumed by a node in the network when the latency obtained is kept at a constant value

TMAC protocol is an adaptive protocol. It adjusts its energy consumed under varying traffic load. So for the purpose of comparison of performance with the TMAC protocol construct just one graph comparing the latencies of the TMAC and SMC-MAC protocol when the energy consumed by the SMC-MAC is equal to the energy consumed by the TMAC protocol under same traffic load

5.1 Simulation Parameters

Simulation parameters are mostly the same for all the simulations that are done. These are summarized in the following table

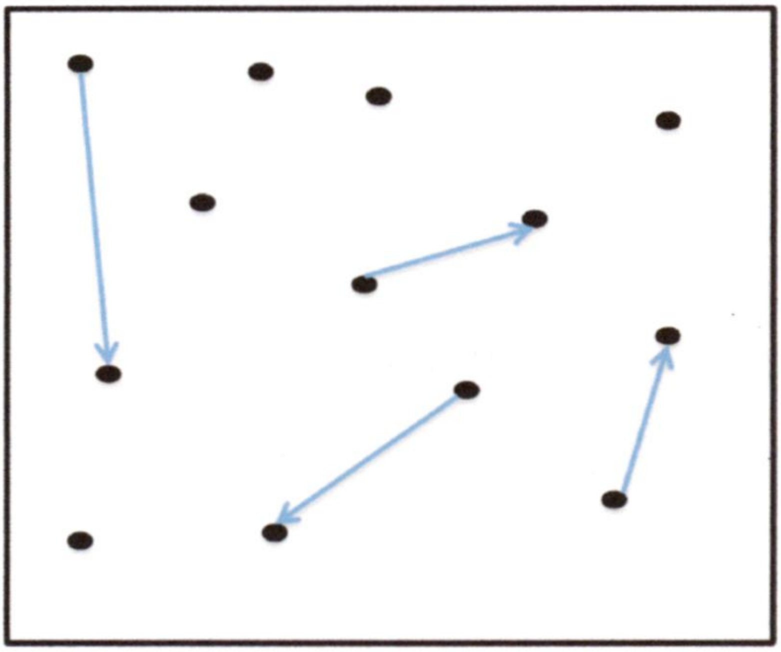
Number of channels available	16
Data rate/channel	250kbps
Tx/Rx power level	57.4mW
Interference model	Additive
Path loss model	Lognormal Shadow Model

The number of nodes and their deployment change according to the change in communication pattern under consideration and will be given with the results.

5.2 Simulation Results

5.2.1 Local Gossip

For simulating the local gossip pattern we construct the network as having a small number in a small region and having packets to transmit to their neighboring nodes. The network is shown in Fig 5.1.



Number of nodes	20
Area	$400m^2$
Deployment	Uniform

Figure5.1: Local Gossip communication pattern

Comparison graphs are obtained for this network for comparison with SMAC and TMAC as discussed earlier. These graphs are shown in Fig. 5.2 and 5.3 respectively.

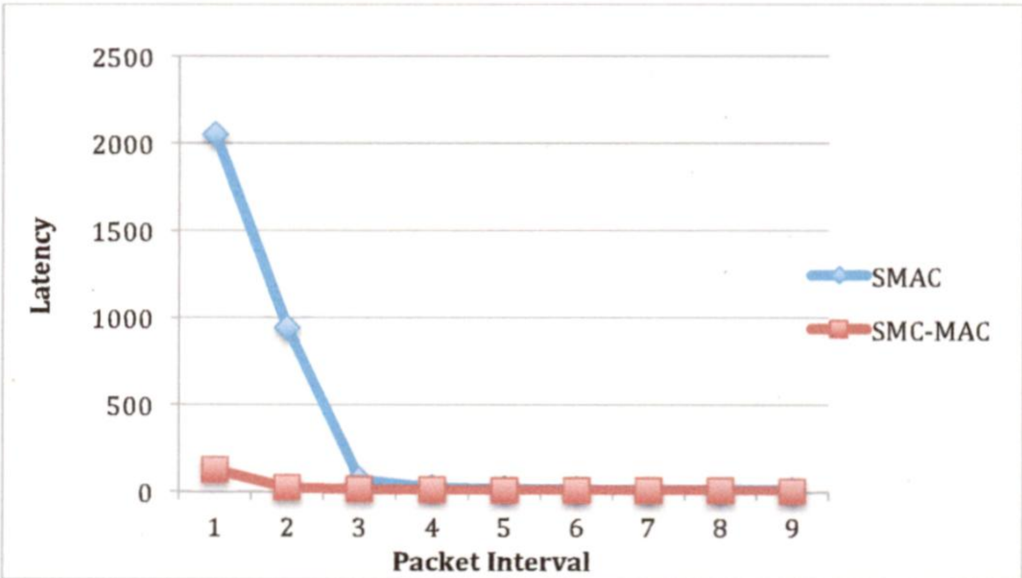


Figure 5.2(a): Constant Energy graph for local gossip

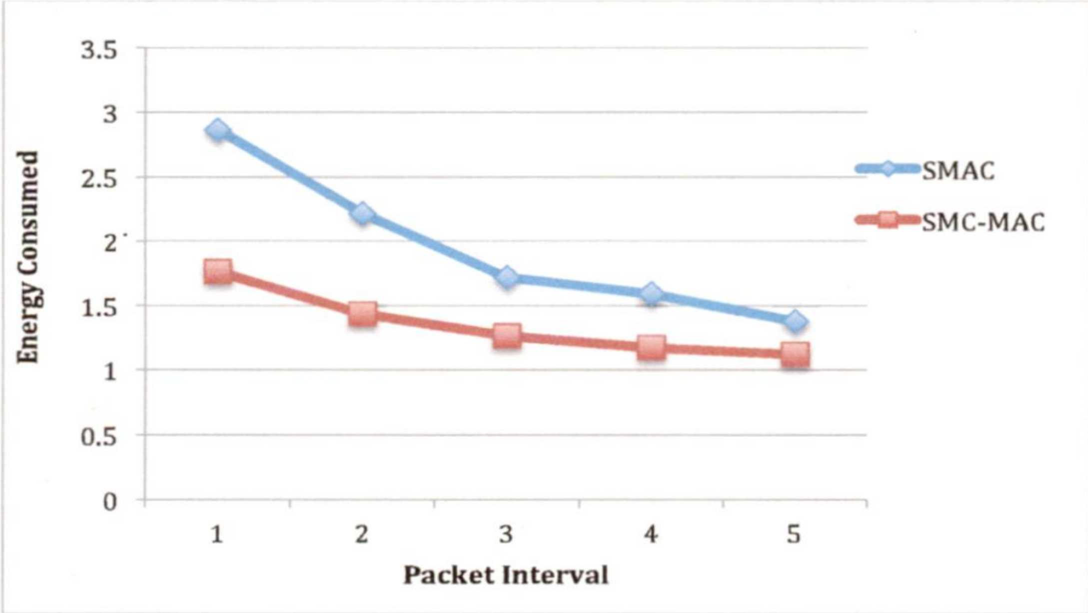


Figure 5.2(b): Constant latency graph for local gossip

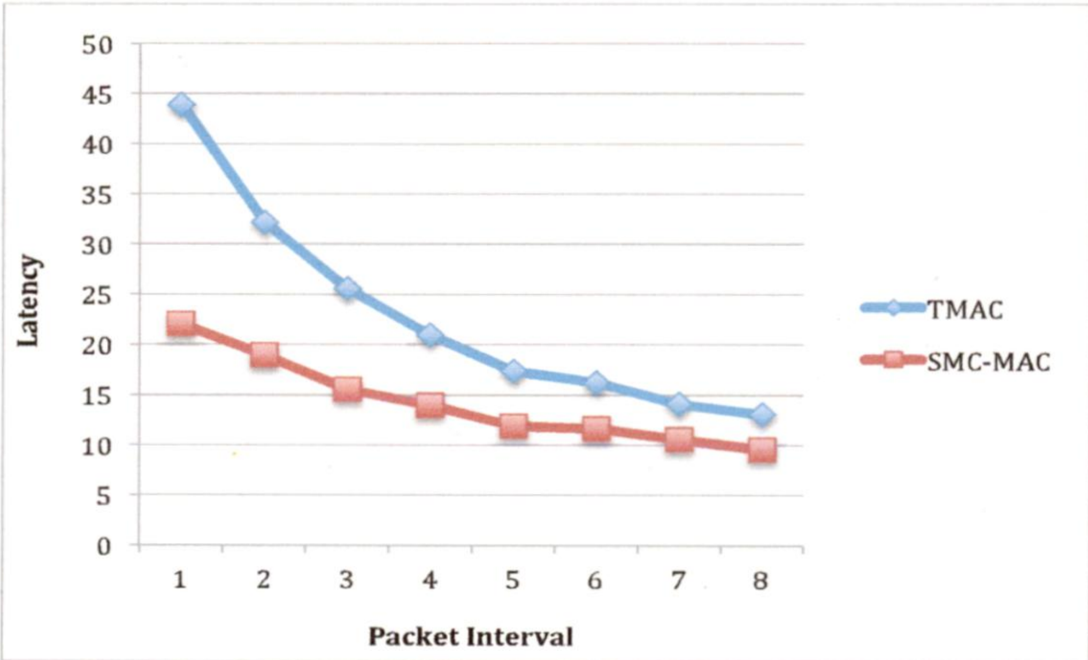
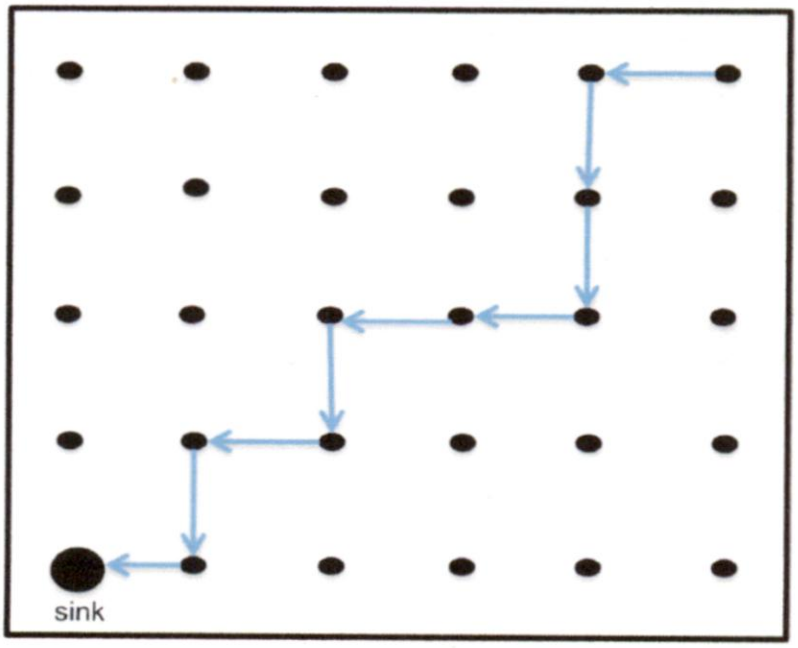


Figure 5.3: Comparison with TMAC protocol for local gossip

5.2.2 Convergecast

The next communication pattern we consider is convergecast. In this pattern all the nodes of the sensor network cooperate to deliver the perceived information by the sensor to the base station. In this pattern multiple nodes send their data packets to a single node. For simulating this traffic pattern we have constructed a gird. Data packets are generated by all nodes and are transferred to the base station that is located at the corner of the grid through multi-hop communication. We assume that no data aggregation has been done at the intermediate nodes. This is the worst-case scenario as applications generally incorporate some data aggregation methods to reduce the network traffic. The network is shown in Fig. 5.4.



Number of nodes	100
Area	1000 m^2
Deployment	10x10 grid

Figure5.4: Convergecast communication pattern

The comparison graphs for this communication pattern is shown in Fig5.5 and Fig 5.6

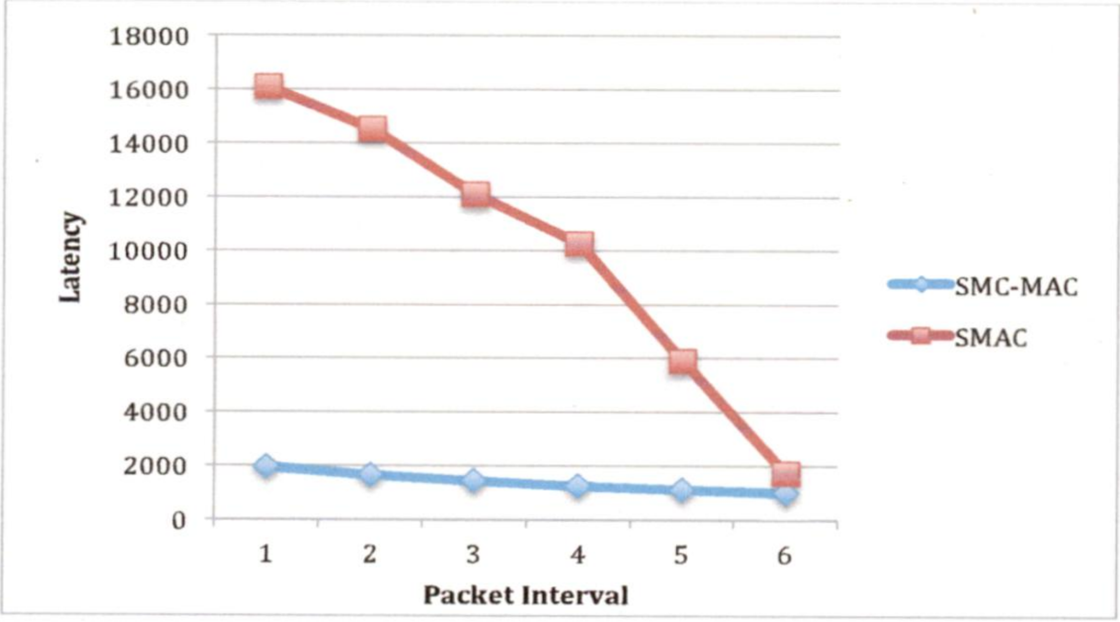


Figure 5.5(a): Constant energy graph for convergecast

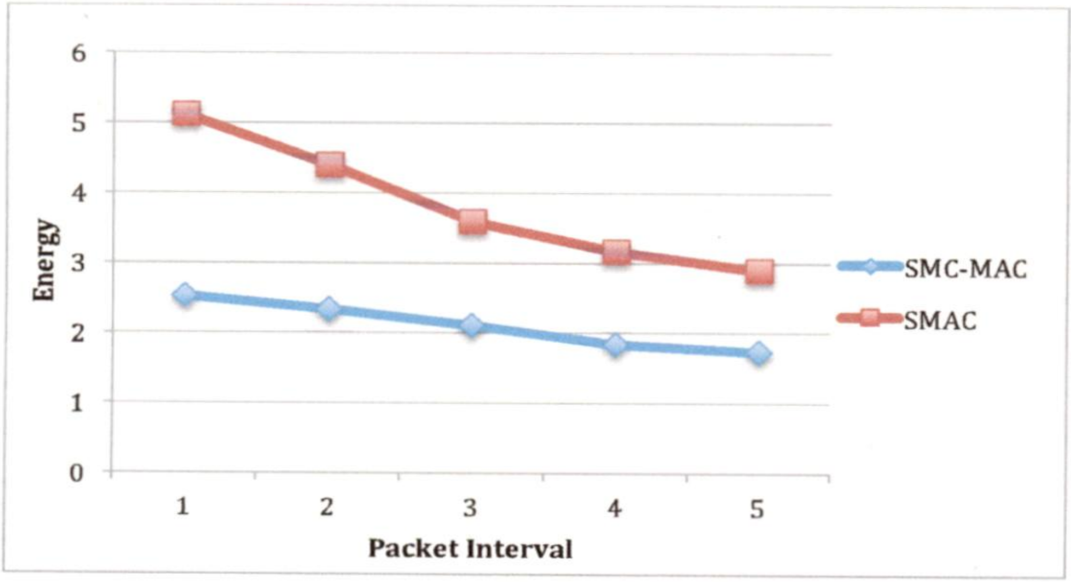


Figure 5.5(b): Constant Latency graph for convergecast

Chapter 6
Conclusions and Future work

The advancement of technology in wireless sensor networks have led to the development of low-cost, low power radios which can work on multiple channels and can switch their channel dynamically. These channels are orthogonal and are free from interference from one another. This increases the throughput that can be achieved from the network as multiple data transmissions can now go on in parallel. Also emerging applications have more bandwidth and latency requirements. This calls for effective use of the available channels to reduce the overall latency and increase the throughput of the network keeping the energy consumption low.

In this work, we have proposed a contention-based MAC protocol that serves the purpose of low latency with low energy consumption. This protocol makes use of the basic periodic listen and sleep scheme of SMAC protocol. It then divides the active period into slots of contention and data transmission. Slot and channel for transmission of data is negotiated between the nodes during the contention slot. During the data slots multiple transmissions can go on in a same neighborhood without any interference. Depending on the traffic a node may use one or more than one data slot during a frame. This makes the protocol somewhat adaptive to the traffic load on the network.

We have simulated the proposed protocol using Castalia and proposed scheme has shown better results than the SMAC and TMAC protocols.

Suggestions for future work

Although our protocol supports mobility, we have assumed static nodes for our simulation. Our protocol needs to be tested in mobile nodes scenario. Also we have used simulation for the performance evaluation of our protocol. Next step could be implementing the protocol on real sensor nodes.

We have assumed a single application running on all the nodes and thus have not considered fairness. In case of nodes sharing multiple applications fairness becomes an important issue and our protocol will need some modifications to ensure fairness amongst applications.

The protocol proposed in this work is adaptive in the sense that a node may use one or more data slots based on the number of data packets that need to be exchanged. But one can add a further level of adaptation by making use of the basic proposed by TMAC protocol of having a variable duty cycle. We can extend that idea to our protocol by using a variable length control slot. This will make our protocol highly adaptive.

Also our protocol does not work well when the node density decreases. In such a case there is no need for multichannel communication and the protocol performs worse due to increased overhead. This protocol can be made more flexible to adjust to such situations.

References

[1]. I.Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks".IEEE Communications Magazine, vol. 40 iss. 8, pp.102-114, August 2002.

[2]. Al-Karaki, J.N., Kamal, A.E. "Routing techniques in wireless sensor networks: a survey." IEEE Wireless Communications, vol. 11, no. 6, pp. 6–28 December 2004

[3]. Rajesh Yadav, Shirshu Varma, N Malaviya, "A Survey of MAC Protocols for Wireless Sensor Networks", UBICC Journal, vol. 4, no. 3, August 2009.

[4]. I. Demirkol, C. Ersoy, f. Alagoz, "Mac Protocols for Wireless Sensor Networks: A Survey", IEEE Communications Magazine, vol. 44, iss. 4, pp. 115-121, 2006.

[5]. P. Czapski, "A survey: Mac protocols for applications of wireless sensor networks," TENCON 2006. 2006 IEEE Region 10 Conference, pp. 1–4, Nov. 2006.

[6]. C. C. Enz, A. El-Hoiydi, J-D. Decotignie, V. Peiris, "WiseNET: An Ultralow-Power Wireless Sensor Network Solution", IEEE Computer, vol. 37, iss. 8, August 2004.

[7]. G. Lu, B. Krishnamachari, C. Raghavendra: An Adaptive Energy Efficient and Low Latency MAC for Data Gathering in Wireless Sensor Networks, Proceedings of 18thth International Parallel and Distributed Processing Symposium (April 2004).

[8]. W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," IEEE/ACM Trans. Net., vol. 12, no. 3, pp. 493–506,. June 2004.

[9]. T. V. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," 1st ACM Conf. Embedded Networked Sensor Sys., Los Angeles, CA, Nov. 2003.

[10]. Joseph Polastre, Jason Hill, and David Culler, "Versatile low power media access for wireless sensor networks," in SenSys'04, 2004.

[11]. G. Zhou, J. Stankovic, and S. Son, "Crowded spectrum in wireless sensor networks," in EmNets 2006: Proceedings of the Third Workshop on Embedded Networked Sensors, May 2006.

[12]. O. Durmaz Incel, P.G. Jansen, and S.J. Mullender "MC-LMAC: A multi-channel mac protocol for wireless sensor networks" Technical Report TR-CTIT-08-61, Centre for Telematics and Information Technology, University of Twente, Enschede, 2008.

[13]. G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, "MMSN: Multi-frequency media access control for wireless sensor networks," in Proceedings of IEEE Infocom, 2006.

[14]. Jovanovic D. M. and Djordjevic G. L., "TFMAC: Multi-channel MAC Protocol for Wireless Sensor Networks," in 8th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services - TELSIKS 2007

[15]. Khosrowshahi, A.C.; Dehghan, M.; Pedram, H.; Alizadeh, M., "A Novel Multi Channel Sensor Network MAC Protocol," Fourth International Conference on Systems and Networks Communications, 2009. ICSNC '09. pp.230-235, 20-25 Sept. 2009

[16]. CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, online available at http://www.chipcon.com

[17]. The Castalia simulator for Wireless Sensor Networks, online available at http://castalia.npc.nicta.com.au.

[18]. S.S., Kulkarni, "TDMA services for Sensor Networks", Proceedings of 24th International Conference on Distributed Computing Systems Workshops, pp. 604 – 609, 23-24 March 2004.