LATTICE-BASED MEDIA ACCESS FOR ENERGY-EFFICIENT WIRELESS SENSOR NETWORKS

By

GHAYATHRI GARUDAPURAM

A thesis submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
in
Computer Science

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

AUGUST 2008

To the Faculty of Washington State University:

The members of the Committee appointed to examine the thesis of GHAYATHRI GARUDAPURAM find it satisfactory and recommend that it be accepted.

Chair

ACKNOWLEDGEMENT

First, I would like to acknowledge my advisor, Dr. Muralidhar Medidi. Without him, this thesis would not have been possible. I thank him for his patience and encouragement that carried me on through difficult times, and for his guidance that helped me to become a better researcher.

I would like to thank Dr. Sirisha Medidi and Dr. Dave Bakken for serving as committee members and evaluate my thesis.

I would like to thank all the members of SWAN group, especially Monique Kohagura for their encouragement and support.

LATTICE-BASED MEDIA ACCESS FOR ENERGY-EFFICIENT WIRELESS SENSOR NETWORKS

Abstract

by Ghayathri Garudapuram, M.S.
Washington State University
August 2008

Chair: Muralidhar Medidi

Media access in wireless sensor networks has distinct requirements including energy-efficiency, low latency, simplicity, scalability, etc. Further, constraining efficient media access is the spatially-correlated contention: several close-by nodes detecting an event send packets to the base station simultaneously. Such contention adversely affects delay, degrades throughput and wastes energy. Primarily motivated to overcome this contention, we propose a distributed topology control to construct a lattice communication backbone. This collaborative lattice naturally provides two forwarders for any backbone node to diffuse traffic, to employ low duty cycles, to allow staggered wakeup scheduling, and to reduce set-up latencies. Further, the controlled topology of a lattice allows a straightforward collision avoidance to overcome contention. We implemented our lattice construction and associated media access using ns-2 simulator for evaluating its performance. Results shows that our lattice backbone provide significant energy savings, lower delay, and higher throughput.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

Page

Dedication

This thesis is dedicated to my family and teachers,
for without their support I would
never have come this far.

CHAPTER ONE

INTRODUCTION

Wireless Sensor Networks (WSN) are event-based systems that utilizes dense deployment of micro-sensor nodes. These micro-sensors are low-cost and small form-factor embedded systems that are capable of sensing, processing and wireless communication [34, 35]. The need for remote monitoring is the main motivation behind the deployment of such a sensing and communication network (sensor network) consisting of a large number of these battery-powered nodes. For example, such system could be used either outdoors in inhospitable habitats, disaster areas, or indoors for intrusion detection or equipment monitoring.

These nodes monitor the environment to gather information like temperature, humidity, etc. process them and forward the processed information to a user, or in general a base station. In general, the base station is assumed to have more processing capability and energy when compared to the rest of the network. This forwarding typically occurs via other nodes using a flat or clustered multi-hop path [36, 37]. Thus a node in the network typically performs two essential tasks: (1) sensing its environment and processing the information and, (2) forwarding the traffic as an intermediate node in the multi-hop path. In typical sensor applications like fire detection, the events occur rather infrequently. In these scenarios, the sensors spend considerable portion of their lifetime to monitor the environment, during which little communication is required and the sensors are said to be in *monitoring state*. During the monitoring state, the radio is on and the node is ready to receive packets. We denote this radio state as *idle listening*. Once the events are detected, the sensors leave their monitoring state and actively participate in communication receiving and forwarding packets.

As sensor nodes are battery-powered and often it is not possible to replace or recharge their batteries, energy efficiency and extending the network lifetime are primary design objective in wireless sensor network protocols. As radio circuitry consumes a majority of energy in the sensor node, even in idle state [25, 27], it is desirable to turn off the radio when there is no need for communication without affecting the network connectivity. Because these networks are usually densely deployed to obtain desired coverage [38], this redundancy in nodes can be effectively exploited to achieve energy efficiency. Two approaches are popular for achieving energy efficiency utilizing dense deployment of sensor networks: the first uses topology control that utilizes the spatial redundancy to turn off nodes that are not needed for communication [28, 2]; and the other uses duty cycles, based on temporal redundancy, to turn off radio [14, 31, 32, 30, 25].

Topology control achieves energy-efficiency by turning off nodes that are redundant with respect to communication. As soon as a node powers down its radio, it is essentially disconnected from the rest of the network and therefore can no longer participate in packet forwarding activity. For simplicity, we refer to this state as the node being asleep. During this sleep state, the node's sensor and processor are still active and monitoring the environment. The goal of the topology control technique is to coordinate the sleep transitions of nodes while ensuring that the data can still be forwarded from source to base station. Topology control achieves energy-efficiency while preserving connectivity by selecting nodes to form a backbone network which is responsible for maintaining the network connectivity. Other nodes which are not part of the backbone can turn-off their radio and go to sleep. Protocols like GAF [28] and SPAN [2] are typical examples. GAF divides the deployment area into virtual grids and a node is elected from each grid to be backbone node. SPAN elects coordinators such that every node is either a coordinator itself or within one hop from a coordinator node. These protocols require backbone nodes to be active all the time to preserve connectivity. In order to avoid energy depletion of the backbone nodes, nodes take turns to be a part of the backbone. This distributes work load and prolongs the network lifetime.

On the other hand, a node need not be active all the time to preserve network connectivity. Duty cycles utilize this temporal redundancy of nodes to achieve energy efficiency. It allows each node to be active for only a small portion of the cycle time and sleep (or turn off radio) the rest of the cycle. If a node wakes up for a small amount of time T_{active} and sleeps for the rest of the time

T_{Sleep} in the cycle, duty cycle is defined as T_{active} /(T_{active}+ T_{Sleep}). The energy conserved will be proportional to this ratio. Though duty cycle increases the energy efficiency, it creates additional *setup latency*: at each hop, a node has to wait till its forwarder wakes up to pass on the packet. Many protocols were introduced to minimize this setup latency. Protocols like STEM [24], PTW [29] and LEEM [3] use dual radio to reduce setup latency. The additional radio is used to wakeup the forwarder and make it ready to receive the data packet without incurring higher delay. Wakeup scheduling schemes, on the other hand, were introduced to minimize this setup latency by using predetermined patterns like staggering in DMAC [14] to achieve better delay. In DMAC, the nodes along the path wakeup sequentially like a chain reaction, to allow continuous packet forwarding to the base station. Other protocols like ECR-MAC [31], MS-MAC [32] and Multi-parent approach [9] utilize multiple forwarders to reduce setup latency. Instead of a traditional tree topology, where a node is always associated with a single forwarder and the packets are always forwarded through the same fixed path, here each node is associated with multiple forwarders with different wakeup scheme. Depending on when a packet arrives at a node, it always chooses the fastest path to reach the base station. Most of these techniques rely on network-wide tight synchronization which is expensive.

Traffic pattern plays a critical role in designing an energy efficient protocol. Convergecast is the major traffic pattern in wireless sensor network [14, 11] which comprises of several nodes sending reports to the base station. WSN also allows base station to send command/queries to nodes [9]. However, all-to-all traffic pattern is not common in a typical sensor network application [14]. So, it is critical to optimize the traffic from sensor to base station. Worse, due to the dense deployment aspect of WSN, all the near-by nodes sensing an event try to send reports to the base station at the same time which leads to *spatially-correlated contention*[26, 8]. This contention, unique to sensor networks, is a fundamental challenge which can significantly impair the network performance and affect energy efficiency. Most of the proposed sensor networking protocols do not address this contention.

The two energy-efficiency approaches, based on spatial- and temporal-redundancy, are orthogonal and they can be combined to achieve better network performance and energy efficiency. Motivated primarily by the challenge of spatially-correlated contention and energy constraints characteristic to WSN, we propose to construct a communication backbone. The backbone addresses contention by (1) reducing the number of active contenders for the channel; (2) facilitating straightforward collision avoidance between known contenders of the backbone; and (3) allowing multiple paths (forwarders) between source and base station to diffuse traffic. Naturally, non-backbone nodes conserve energy by not participating in the communication aspects of the network. Further, to improve the energy-efficiency, backbone nodes themselves can employ a low duty cycle and a wakeup schedule to take advantage of multiple forwarders. We chose a *lattice* or *mesh*, a well-studied interconnection structure, as the backbone as it naturally provides many of the desirable features to address spatially-correlated contention and to provide better energy-efficiency, lower delay and better throughput.

1.1 Organization of thesis

The rest of this thesis is organized as follows. Chapter 2 describes the motivation and requirements. Chapter 3 reviews the related work. Chapter 4 describes our lattice-based topology control and media access protocol. Chapter 5 presents the performance evaluation of the lattice and associated media access. Finally, we conclude in Chapter 6.

CHAPTER TWO

MOTIVATION AND REQUIREMENTS

Many protocols have been proposed to handle various issues in wireless sensor networks to effectively support sensor network applications. Among them, media access is very important to ensure good performance. As sensor networks use wireless channel for communication, we need some media access mechanism to ensure that no two nodes are interfering with each other's transmission, and to deal with situations when they do. As wireless sensor networks are fundamentally different from traditional wireless network, media access designed for these networks must be tailored to its distinct requirements. Some of the distinct requirements of these networks are described below.

1. *Energy-efficiency*

   Battery operated wireless sensor nodes which are deployed in harsh environments, are not expected to be replaced of their batteries. Hence, these networks are expected to operate for a longer time period before failing. In such a scenario, it is expected that the media access mechanism has to be energy efficient and extend the lifetime of the network by a considerable amount of time. A good protocol for wireless sensor network should address this primary requirement.

   These networks, which are event driven, are idle most of the time as events occur infrequently. Hence, nodes spend most of the life time in idle listening, while spending little time in reporting events to the base station. Also, it has been well established that radio circuitry of the node is the major consumer of the energy. Hence, it is desirable to turn off the nodes' radio as much as possible to conserve energy. When a node turn's off its radio, it is disconnected from the network and cannot participate in packet forwarding. This poses a challenge for designing media access for sensor network which should preserve network connectivity and at the same time conserve energy by turning off nodes' radio.

2. *Spatially Correlated Contention*

   Sensor nodes are deployed in high density for obtaining reliability in event sensing [26]. Any event that occurs will naturally be sensed by large number of close-by nodes which will try to send packets simultaneously to the base station to report the event. This tendency to report simultaneously causes contention for the channel. This contention causes the nodes to back-off causing packet backlog and wasting energy. Since this contention is caused by nodes that are spatially related to the event, it is termed as *spatially correlated contention*. This contention characteristic to sensor network, is a fundamental problem that needs to be addressed by any media access protocol for sensor network. Unless it is taken care of, it will substantially degrade the network performance of the network in terms of delay and throughput.

3. *End-to-end Delay*

   As described earlier, wireless sensor network is event driven. When a phenomenon occurs, the nodes sensing the event generates packet to be sent to the base station to notify about the event. Depending on the type of application, the node may send packets at regular intervals of time like temperature monitoring or the node notifies the base station on sensing some rare event like in forest fires. In case of extreme deployment, the network might not be even functional after the event like in forest fires. This brings in the necessity to hurl the packet to the base station as soon as possible, so that appropriate action can be taken. For such applications, the end-to-end delay of packets from nodes to base station is very critical and media access protocols designed for sensor network should be designed to achieve small end-to-end delay.

4. *Fairness*

   The main philosophy behind the deployment of the sensors is to report to base station if anything abnormal is sensed. This abnormal event will be reported by not just one by many near-by sensor nodes due to sense deployment. As long as some of these packets reach the base station, the purpose of the network is served. So, inherently the concept of fairness for all nodes does not apply in wireless sensor network. This is different from a traditional network where each node in the network is considered independent and treated fairly. So, the protocol designed for sensor network should aim at delivering more packets to the base station rather than considering the fairness for the individual nodes in the network.

5. *Nature of Traffic*

   It is very essential to know the nature and the type of traffic in the network to design an efficient protocol that can be tailored to the network. The two predominant types of traffic in sensor networks are [14, 9]: nodes to base station (reverse direction) and base station to nodes (forward direction). Generally , the base station sends a query/command message to either all nodes or a subset of nodes in the network. Usually, this traffic from is not delay sensitive, where as the traffic from source to base station is very critical and its only for the timely delivery of these packets that the network itself is deployed. The protocol should be designed to optimize the traffic from source to base station.

   Also, it is very essential to know how much of traffic will flow in the network to design an efficient medium access protocol for any network. Due to the event driven nature of these networks, as long as no event occurs, the network is going to be in idle state just monitoring the environment. In most of the practical applications, occurrence of these events are rare and infrequent. The protocols designed for such networks can take advantage and have minimum functionality when the network is in idle state. Once an event occurs, nodes sense and generates packets for the base station. At this stage when there are packets to be delivered for base station, the network must operate to deliver packets with minimum delay. This is fundamentally different from traditional networks, which is designed for constant traffic flow. So, a media access for wireless sensor network should have minimum functionality when the network is idle saving energy and work at full efficiency when there are packets to be delivered.

In addition, the source to base station traffic is a *convergecast traffic* - traffic originating at any part of the network ultimately converges at base station . This kind of traffic pattern is very difficult to handle as they are prone to more contention and collision. Any protocol designed for wireless sensor network should be able to address such converge cast traffic and help in alleviating this contention to improve performance.

CHAPTER THREE

RELATED WORK

3.1 General Overview

In wireless sensor networks, the sensor nodes are battery operated and it is often very difficult to change or recharge their batteries for these nodes. In fact, some day we can expect these nodes to be so cheap that we can discard them instead of recharging them. Hence, energy-efficiency and prolonging the network lifetime are critical to any protocol designed for wireless sensor networks [34, 35, 36]. It is important to view the problem of energy-efficiency as the one of extending the lifetime of the network, rather than that of the individual nodes. Thus, in addition to improving the efficiency of nodes, techniques to tackle the problem on the level of entire network is necessary.

Many protocols have been proposed for achieving energy-efficiency in wireless sensor network. They can be broadly categorized as :

1. Topology Control Techniques.
2. Wakeup scheduling schemes.
3. Hybrid between Topology control and Wakeup scheduling schemes.

3.2 Topology Control Techniques

Topology control refers to the process of controlling the network connectivity graph in order to achieve desirable features like high energy efficiency, lower delay and higher throughput. The main goal behind the topology control mechanism is to reduce nodes energy consumption in order to increase the network longevity. Many topology control protocols have been proposed for increasing the network longevity and they can be broadly categorized into three types [10]:

1. Mobile node based.

2. Transmission power adjusting-based.
3. Sleep based techniques.

3.2.1 Mobile node based

Mobile node based techniques involve the use of mobile nodes and network topology is adjusted by moving the mobile nodes appropriately. Mobile nodes are generally deployed along with static nodes. The mobile nodes in the network will enhance the network capabilities - they could be used to physically collect and transport data or to recharge and repair the static nodes in the network. A key step towards realizing such networks is to develop techniques for network nodes to self-deploy and reconfigure. Further, for successful operation of the network, the deployment should result in configurations that not only provide good 'sensor coverage' but also satisfy certain local (e.g. node degree) and global (e.g. network connectivity) constraints. Poduri *et al.* considered the problem of self deployment of mobile sensor network [20] such that deployment strategy maximizes the area coverage of the network such that each node has at least k neighbors, where k is a user-specified parameter. The algorithm is based on artificial potential fields to move the mobile sensor and it is distributed, scalable and does not require previous map of the environment. Problem of coverage and exploring an unknown dynamic environment is considered in [1]. It is achieved using an efficient minimalistic algorithm that does not depend on global positioning system (GPS) and usage of beacons to assist robots in coverage. Since we deal with static deployment, we skip indepth discussion on mobile node based topology control. Interested reader can refer to [10] for a good review.

3.2.2 Transmission power adjusting-based Topology control

Many topology control techniques attempt to manage the network topology by adjusting the nodes' transmission power. Transmission power adjustment based techniques [21, 12, 17] aim at keeping the transmission power low, which improves energy efficiency and reduces node degree [12] and interference [17] while still maintaining the network connectivity, energy efficient routes [23], shortest paths [6] etc. Most of these techniques were designed for traditional ad-hoc networks where arbitrary communication occur and minimizing transmission power provides reduction in energy consumption. By eliminating longer links that incur more transmission power without affecting the network connectivity, the transmission power required to communicate with the neighboring nodes is reduced and hence energy is conserved.

A localized algorithm to construct Minimum-Energy Communication Network (MECN) [23] was proposed by Rodoplu *et al.* for stationary ad-hoc networks. MECN preserves minimum-energy path between each pair of nodes. A improved version SMECN (Small Minimum-Energy Communication Network) [39] of MECN was achieved by constructing a smaller network that incurs lower link maintenance cost and lower energy consumption. They proved that the necessary and sufficient condition for a subgraph G' to retain minimum-energy path is to let G' contain the minimum-energy subgraph G_{min} that removes all the redundant edges. As G_{min} requires great deal of communication to distant nodes, they designed SMECN to form the suboptimal subgraph G' to contain G_{min}. SMECN performs significantly better than MECN while being computationally simple. *Strong Minimum Energy Topology* (SMET) problem : “ adjusting each node’s transmission power such that there exists at least one bidirectional link between any pair of sensors (strongly-connected) and the sum of all the transmit power is minimized ” was studied in [40]. The problem is proved to be NP-Complete and two heuristics were proposed to provide near optimal solutions. Wang *et al.* proposed a localized Shortest Path Tree (SPT) based algorithm to construct an energy efficient topology for wireless ad-hoc networks [41]. The links are assigned weights bases on the required transmission energy and retain the links that contribute to minimum energy consumption between nodes. The generated topology achieves lower total energy consumption than that of SMECN.

In addition to keeping the minimum-energy path to achieve energy-efficiency, there are other metrics that have been considered in topology control techniques. For example, it is desired to have a bound on the node degrees, so as to prevent communication and energy bottlenecks. Li *et* *al.* [12] proposed a minimum spanning tree based topology control algorithm, denoted as LMST, in which each node collects information about the nodes within its maximum transmission range and then build a local minimum spanning tree independently. The generated topology preserves network connectivity and bounds the node degree to 6 and also can be transformed to one with only bidirectional links. However, bounding the node degree will not retain the minimum energy path anymore [42]. This issue can be addressed by finding a good balancing point. The idea of power spanner was proposed where the energy consumption of the routes can be at most a constant factor away from the energy-optimal routes. Wang and Li proposed the first energy bounded power spanner [43]. However, the node degree in the worst case can go up to 25 and topology construction requires a higher message overhead.

While most of the earlier techniques only attempted to address the energy-efficiency issues by minimizing the total energy consumption of the entire network, recently there have been works targeted toward the energy balance issues in topology control. Baek *et al.* [44] proposed a power-aware topology control algorithm in which each node determines the communication links based on the neighbor's location and residual energy. The resulting topology achieves better lifetime when compared to LMST and SPT.

There are also other topology control techniques that achieves other objectives. Gao *et al.* proposed a Restricted Delaunay Graph (RDG) based routing for wireless ad-hoc networks [45]. The most important feature of this graph is that between any two nodes in the graph , there exists a path in the RDG whose length, in terms of both hops or Euclidian distance, is only a constant times the optimal length possible. This feature will help to control the end-to-end delay in the packet transmission and achieve better energy efficiency.

3.2.3 Sleep-based Topology Control

All the previously mentioned topology control techniques focus on reducing the transmission power to conserver energy. In wireless sensor network, a node spend most of its lifetime in idle listening monitoring the environment, while spending only small portion of time reporting any event. Under this scenario, the most effective way to conserve energy is by turning-off nodes' radio as much as possible [28, 2, 24] , which is the basis of sleep based topology control. Once a node is turned off, it can no longer participate in communication and help in packet forwarding. One of the major challenge in sleep-based topology control technique is how to turn off more nodes without affecting the network connectivity. As sensor network are densely deployed to achieve desirable coverage [26], it is not necessary for all the nodes in the network to help in packet forwarding to maintain network connectivity. Sleep-based techniques exploit this dense deployment to construct a backbone which will maintain the network connectivity.

GAF [28] is one of the first sleep topology control protocol proposed to increase the network lifetime by exploiting node redundancy to conserve energy. GAF conserves energy by identifying nodes that are equivalent from a routing perspective and then turning off unnecessary nodes, keeping a constant level of routing *fidelity*. GAF moderates this policy using application- and system-level information; nodes that source or sink data remain on and intermediate nodes monitor and balance energy use. GAF is independent of the underlying ad hoc routing protocol. It addresses this problem of finding nodes equivalent in routing perspective by dividing the whole area where nodes are distributed into small *virtual grids*. The virtual grid is defined such that, for two adjacent grids A and B, all nodes in A can communicate with all nodes in B and vice versa. Thus all nodes in each grid are equivalent for routing. The deployment area is divided into virtual grids and only one node in a grid remains active at a time and other are put to sleep. Each node uses location information to associate itself with a virtual grid, where all nodes in a particular grid are equivalent with respect to forwarding packets. The nodes in a grid coordinate with each other to determine who will sleep and for how much time. Initially nodes start out in the *discovery state*. When in discovery state, a node turns on its radio and exchange discovery messages to find the other nodes within the same grid. When in discovery state a node sets a timer after which it broadcasts its discovery message and enters the *active state*. A node in the active state can go to sleep state when it knows that there is some other node as representative for the grid. Nodes then periodically wake up to trade places and hence accomplishing load balancing. Analysis and simulation studies of GAF show that it can consume 40% to 60% less energy than an un-modified ad hoc routing protocol. Also, GAF was able to improve the life time by four-fold without significantly affecting the delay and throughput. The disadvantage with GAF is that the radio range is underutilized, which means more hops are needed to cover a given distance. Also, it requires all the backbone nodes to remain active to provide network connectivity.

Span [2] proposed by Chen *et al.* is another sleep based topology control technique. It is a power saving technique for multi-hop wireless networks that reduces the energy consumption without significantly diminishing the capacity or connectivity. Span [2] is based on the observation that when a region of a shared-channel wireless network has a sufficient density of nodes, only a small number of them need to be awake at anytime to forward traffic for maintaining network connectivity. It is a distributed, randomized algorithm where a node decides whether to sleep or join *forwarding backbone* as a *coordinator*. Span adaptively elects coordinators from all nodes in the network. Span coordinators stay awake continuously and perform multi-hop packet forwarding within the ad-hoc network, while other nodes remain in the power-saving mode and periodically check if they should wakeup and become a coordinator. Span achieves four goals. First, it ensures that enough coordinators are elected so that every node is in the radio range of at least one coordinator. Second, it rotates the coordinators in order to ensure that all nodes share the task of providing global connectivity roughly equally. Third, it attempts to minimize the number of nodes elected as coordinators, there by increasing network lifetime, but without suffering a significant loss of capacity or an increase in latency. Fourth, it elects coordinators using only local information in a decentralized manner - each node only consults state stored in local routing tables during the election process. Span is proactive - each node periodically broadcasts HELLO messages that contain the node's status (whether or not the node is a coordinator), its current coordinators and its current neighbors. A span node switches from time to time between being a coordinator and being a non-coordinator. A simple election algorithm is implemented to decide who will be the next coordinator. Span was able to extend the network lifetime by a factor of two still preserving the total network capacity. It also requires all the backbone nodes to be active to provide network connectivity.

3.3 Wakeup scheduling schemes

In contrast to sleep based topology control techniques, wakeup scheduling schemes utilize the fact that a node need not be awake all the time. Sensor networks are essentially deployed over very vast area and when there are events occurring in one part of the network, the nodes in the other region do not participate in communication and need not be active. Also, when there are no traffic in the network, the nodes need not be active all the time wasting energy in idle listening [30, 14, 29]. Spatial redundancy utilize this idea and turn off the nodes' radio when there is no need for communication and provide some mechanism to wake the radio up when communication is necessary. In this scheme, a node wakes up for a small period of time T_{active} and sleeps for the rest of the time T_{Sleep} in the cycle. Based on this, duty cycle is defined as follows:

$$dutycycle = \frac{T_{active}}{T_{active}+T_{Sleep}}$$

In sleep based topology control techniques, since there is a active backbone, it will be able to maintain a network connectivity and sustain performance. This might not be true in the case of wakeup scheduling schemes. As the nodes wakeup only for a small period of time, the wakeup scheduling schemes may not sustain network performance anymore and this poses a challenge of optimizing the energy efficiency while maintaining network performance.

S-MAC [30] proposed by Ye *et al.* is the first MAC of its kind to employ duty cycling for achieving energy efficiency in sensor networks. Collision,overhearing, control packet overhead and idle listening are identified as major source of energy wastage. The main goal in S-MAC protocol design is to reduce energy consumption, while supporting good scalability and collision avoidance. The protocol tries to reduce energy consumption from all the sources that have been identified to cause energy waste, i.e., idle listening, collision, overhearing and control overhead. To achieve the design goal, the SMAC consists of three major components: periodic listen and sleep, collision and overhearing avoidance, and message passing. S-MAC employs periodic sleep wakeup mechanism to turn-off radio to reduce energy wastage caused by idle listening. Each node in a cycle sleep for 50% of the time and wakes up for the rest. This decreases the energy consumption by half. Also to employ such a kind of schedule periodic synchronization between nodes is required. Also, synchronization is used to form virtual clusters of nodes on the same sleep schedule. Each node maintains a *schedule table* which has the information on the sleep schedule of its neighbors. They build this table by exchanging messages before starting the duty cycle. This enable a node to determine when its forwarder wakesup to send the packet. Due to this duty cycling, at every hop a node has to wait for its forwarder to wakeup and hence increase the end-to-end delay. Collision avoidance is a basic task of MAC protocols. SMAC adopts a contention-based scheme. Since multiple senders may want to send to a receiver at the same time, they need to contend for the medium to avoid collisions. S-MAC follows 802.11 procedures, including both virtual and physical carrier sense and RTS/CTS exchange to address hidden terminal problem. In 802.11 each node keeps listening to all transmissions from its neighbors in order to perform effective virtual carrier sensing. As a result, each node overhears a lot of packets that are not directed to itself. This is a significant waste of energy, especially when node density is high and traffic load is heavy. S-MAC tries to avoid overhearing by letting interfering nodes go to sleep after they hear an RTS or CTS packet. Since DATA packets are normally much longer than control packets, the approach prevents neighboring nodes from overhearing long DATA packets and the following ACKs. S-MAC fragments a long message into many small fragments, and transmit them in burst. Only one RTS packet and one CTS packet are used. They reserve the medium for transmitting all the fragments. Every time a data fragment is transmitted, the sender waits for an ACK from the receiver. If it fails to receive the ACK, it will extend the reserved transmission time for one more fragment, and re-transmit the current fragment immediately. Though S-MAC was able to significantly reduce energy consumption when compared to 802.11, the duty cycle employed creates additional latency.

Lu *et al.* proposed DMAC [14] , an energy efficient and low latency MAC that is designed and optimized for unidirectional data gathering trees. In order to reduce *sleep latency* like in SMAC, DMAC uses the idea of *staggering wakeup schedule*, where the sleep/active schedule offset depends upon its depth on the tree. This avoids the data forwarding interruption problem and helps in continuous packet forwarding to base station. A staggered wake-up schedule has four advantages. First, since nodes on the path wake up sequentially to forward a packet to next hop, sleep delay is eliminated if there is no packet loss due to channel error or collision. Second, a request for longer active period can be propagated all the way down to the sink, so that all nodes on the multi-hop path can increase their duty cycle promptly to avoid data stuck in intermediate nodes. Third, since the active periods are now separated, contention is reduced. Fourth, only nodes on the multi-hop path need to increase their duty cycle, while the other nodes can still operate on the basic low duty cycle to save energy. When a node has multiple packets to send at a sending slot, it needs to increase its own duty cycle and requests other nodes on the multi-hop path to increase their duty cycles too. D-MAC employs a slot-by-slot renewal mechanism. It piggyback a more data flag in the MAC header to indicate the request for an additional active periods. The overhead for this is very small. Before a node in its sending state transmits a packet , it will set the packets more data flag if either its buffer is not empty or it received a packet from previous hop with more data flag set. The receiver check the more data flag of the packet it received, and if the flag is set, it also sets the more data flag of its ACK packet to the sender. With the slot-by-slot mechanism and the policy to set more data flag when buffer is not empty, DMAC can react quickly to traffic rate variation to be both energy efficient and maintain low data delivery latency. DMAC uses a data prediction scheme to improve latency and throughput. If a node in receiving state receives a packet, it predicts that its children still have packets waiting for transmission. It then sleeps only 3μ after its sending slot and switches back to receiving state. All following nodes on the path also receive this packet, and schedule an additional receiving slot. In this additional data prediction receiving slot, if no packet is received, the node will go to sleep directly without a sending slot. If a packet is received during this receiving slot, the node will wake up again 3μ later after this sending slot. To improve network performance under collision, D-MAC uses *More to Send* (MTS) packets. A node sends a request MTS to its parent if either of the two conditions is true: (1) It can not send a packet because channel is busy. After the nodes back-off timer fires, it finds there is not enough time for it to send a packet and it does not overhear its parents ACK packet. It then assume it lost the channel because of interference from other nodes. (2) It received a request MTS from its children. This is aimed to propagate the request MTS to all nodes on the path. DMAC was able to perform better when compared to SMAC in terms of energy and latency. However, DMAC uses data gathering tree which increase the probability of collision since multiple nodes at the same level will share common forwarders. Also, the nodes at the same level in the tree waking up at the same time competes for the channel which increases the probability of collision and impairs the throughput.

S-MAC protocol trades energy for latency and throughput. T-MAC [25] was proposed by Dam *et al.* to improve latency and energy-efficiency. To handle load variations in time and location T-MAC introduces an adaptive duty cycle in a novel way: by dynamically ending the active part of it. This reduces the amount of energy wasted on idle listening, in which nodes wait for potentially incoming messages, while still maintaining a reasonable throughput. T-MAC achieves energy-efficiency by minimizing the idle listening. The basic T-MAC protocol can be described as follows: Every node periodically wakes up to communicate with its neighbors, and then goes to sleep again until the next frame. Meanwhile, new messages are queued. Nodes communicate with each other using a Request-To-Send (RTS), Clear- To-Send (CTS), Data, Acknowledgment (ACK) scheme, which provides both collision avoidance and reliable transmission. A node will keep listening and potentially transmitting, as long as it is in an active period. An active period ends when no activation event has occurred for a time TA. An activation event is: the firing of a periodic frame timer; the reception of any data on the radio; the sensing of communication on the radio, e.g. during a collision; the end-of-transmission of a node's own data packet or acknowledgment; the knowledge, through overhearing prior RTS and CTS packets, that a data exchange of a neighbor has ended. A node will sleep if it is not in an active period. Consequently, TA determines the minimal amount of idle listening per frame. The described timeout scheme moves all communication to a burst at the beginning of the frame. Since messages between active times must be buffered, the buffer capacity determines an upper bound on the maximum frame time. To facilitate the implementation of duty cycle, synchronization is used and nodes that have the same schedule form a virtual cluster. T-MAC also implements overhearing avoidance. One of its side effect, collision overhead becomes higher: a node may miss other RTS and CTS packets while sleeping and disturb some communication when it wakes up. Consequently, the maximum throughput decreases; for short packets by as much as 25%. T-MAC suffers from *early sleeping* problem - node goes to sleep even if a neighbor has message for it, which affects its throughput when compared to S-MAC. One solution to overcome this problem is to use *future-request-to-send* which informs the forwarder that it has more packets to send. T-MAC outperforms S-MAC in terms of latency and energy efficiency.

Rhee *et al.* proposed a hybrid MAC protocol, called Z-MAC, for wireless sensor networks that combines the strengths of TDMA and CSMA while offsetting their weaknesses. Like CSMA, Z-MAC achieves high channel utilization and low-latency under low contention and like TDMA, achieves high channel utilization under high contention and reduces collision among two-hop neighbors at a low cost. A distinctive feature of Z-MAC is that its performance is robust to synchronization errors, slot assignment failures and time varying channel conditions; in the worst case, its performance always falls back to that of CSMA. Z-MAC uses CSMA as the baseline MAC scheme, but uses a TDMA schedule as a hint to enhance contention resolution. In Z-MAC, a time slot assignment is performed at the time of deployment - higher overhead is incurred at the beginning. Its design philosophy is that the high initial overhead is amortized over a long period of network operation, eventually compensated by improved throughput and energy efficiency. DRAND [47], an efficient scalable channel scheduling algorithm. DRAND is a distributed implementation of RAND [46], a centralized channel reuse scheduling algorithm. After the slot assignment, each node reuses its assigned slot periodically in every predetermined period, called frame. A node assigned to a time slot is called an owner of that slot and the others the non-owners of that slot. There can be more than one owner per slot because DRAND allows any two nodes beyond their two-hop neighborhoods to own the same time slot. Unlike TDMA, a node may transmit during any time slot in Z-MAC. Before a node transmits during a slot (not necessarily at the beginning of the slot), it always performs carrier-sensing and transmits a packet when the channel is clear. However, an owner of that slot always has higher priority over its non-owners in accessing the channel. The priority is implemented by adjusting the initial contention window size in such a way that the owners are always given earlier chances to transmit than non-owners. The goal is that during the slots where owners have data to transmit, Z-MAC reduces the chance of collision since owners are given earlier chances to transmit and their slots are scheduled a priori to avoid collision, but when a slot is not in use by its owners, non-owners can steal the slot. This priority scheme has an effect of implicitly switching between CSMA and TDMA depending on the level of contention. An important feature of this priority scheme is that the probability of owners accessing the channel can be adjusted independently from that of non-owners. This feature contributes to increasing the robustness of the protocol to synchronization and slot assignment failures while enhancing its scalability to contention.This protocol has the setup phase in which it runs the following operations in sequence : *neighbor discovery, slot assignment, local frame exchange and global time synchronization.* The results show that it has advantages in network with medium to high contention.

Schurgers *et al.* [24] approached the problem of energy-efficiency by exploiting two degrees of freedom in topology management: the path setup latency and the network density. They proposed Sparse Topology and Energy Management (STEM) [24], which aggressively puts nodes to sleep. STEM provides a method to wake up nodes only when they need to forward data, where latency is traded off for energy savings. It integrates efficiently with existing approaches that leverage the fact that nearby nodes can be equivalent for traffic forwarding. As a result, an increased network density brings in more energy savings. STEM exploits the time dimension rather than the density dimension. Strictly speaking, nodes only need to be awake when there is data to forward. This is referred to as the network being in the *transfer state*, and in many practical scenarios, this is a rather infrequent event. Most of the time, the sensor network is only monitoring its environment, waiting for an event to happen, and nodes can be asleep. For a large subset of sensor net applications, no data needs to be forwarded to the data sink in this *monitoring state*. In the monitoring state no communication capacity is required. So nodes can save energy by turning off their radio. However, if the nodes turn off their radio, they will not know when their neighbors require them to forward the data. A solution would be to periodically make the nodes to turn on their radio for a short interval of time to see if someone wants to communicate. The node that wants to communicate can send out wakeup messages to the target node. In STEM, this wakeup message is sent in a separate radio channel. The communication that happens in this frequency is called as *wakeup plane*. Once both the sender and receiver has agreed successfully, the real data transmission takes place in a separate frequency called *data plane*. The interval between sending wakeup messages by initiator to the time when both nodes turn on their data radio is denoted as *setup latency*. Clearly, there is a trade-off between setup latency and energy efficiency - higher energy efficiency incurs more setup latency. If the setup latency is high, it means the sender needs to send out more beacons which will waste more energy. Also, this techniques relies on dual radio setup which is expensive.

Pipelined Tone Wakeup (PTW) scheme [29] was proposed by Yang and Vaidya to achieve a balance between energy savings and end-to-end delay. The wakeup scheme uses the benefit of separate wakeup tone to achieve balance between energy savings and delay. This scheme uses an additional wakeup tone channel in addition to regular data channel. In PTW, each node will be awake for T_{dtone} duration and asleep for T_{sleep} duration periodically, where $T = T_{dtone} + T_{sleep}$. When a node has packet to be sent out, it sends a wakeup tone in the wakeup channel which will last for duration T_p. Once a node detects a wakeup tone, it will stay active to receive the data packet. As the wakeup tone does not have any receiver's identity, any node within the transmission range will be awakened. As the nodes are unsynchronized, the wakeup tone from the sender should be atleast T_p so that each neighboring node has at least one entire duration of T_{dtone} to detect the wakeup tone. T_p should satisfy:

$$T_p = 2 \times T_{dtone} + T_{sleep} = T + T_{dtone}$$

In this way, even if the worst case that the node starts its duty cycle just before the sender starts to send the wakeup tone, the former node still has the next entire duration of T_{dtone} to detect the node. After sending the wakeup tone, the sender knows that all its neighbors have been awake and it proceeds to send packets on its data channel. *Wakeup delay* may be defined as the elapsed duration from the time a packet arrives at node's wakeup module, to time the node passes the packet to MAC layer, knowing that the target node is awake. PTW pipelines the wakeup of nodes in order to reduce the setup delay. When a node has packet to send, it sends a tone in the wakeup tone channel to wakeup its neighbors. Once a neighbor is selected, it wakes up its neighbors in the wakeup tone channel simultaneously when the data is being sent in the data channel. Thus, PTW reduces the setup latency and minimize the end-to-end delay. Using wakeup tone also avoids the synchronization requirement for implementing the wakeup pipeline. Results show that PTW is much efficient than STEM in terms of energy-efficiency and end-to-end delay. PTW requires a dual radio setup, and effectiveness depends on the data packet size and network bandwidth.

Dhanaraj *et al*, proposed a reservation scheme, *Latency minimized Energy Efficient MAC protocol* (LEEM) [3], which is a novel hop-ahead reservation scheme in a dual frequency radio to minimize the latency in the multi-hop path data transmission by reserving the next hops channel a priori. Using LEEM, in a multi-hop sensor network, a packet can be forwarded to the next hop, as soon as it is received by a sensor node, which helps in eliminating the delay incurred for setting up the path. For the purpose of activating the data channel, the control channel radio is made active periodically and the sensor node checks whether any data is to be transmitted. When the active time durations of the control channel radios of the nodes in the sensor network are not synchronized, the wakeup signal is transmitted continuously until it receives the acknowledgment. This control channels active period is fixed to a minimum time period to save energy. Assuming R1 and R2 to denote the time to send the wakeup signal and acknowledgment, respectively, time for which the control channel is required to remain active, T_{active} is fixed at $(2R1 + R2)$. This ensures that the nodes can receive the wakeup signal even while they are not synchronized. At the time of network setup, each node uses a proactive or table-driven routing protocol to make an entry about the next hop node. Hence, every node maintains the information about the next hop node and the data packet is forwarded to the data sink via the next hop node, using this information. In addition, T_{active} period of each node is synchronized with its next hop node. In LEEM, a time period of $(R1 + R2)$ is sufficient to ensure proper working of the wakeup process. The synchronization helps to make reservations for multiple-hops and reduces the end-to-end latency. LEEM reduces the setup latency by reserving the channel for next hop ahead. When an event occurs, the same procedure of waiting for the control channel to get activated and sending the wakeup signal for acquiring the data channel is carried out at the source sensor node. However, from the second hop node onwards the waiting time becomes zero, as the channel reservation is done a priori. This is because, whenever a data transfer takes place, the receiver of the nodes reserves the channel for K hops ahead. If the value of K is one, it is an 1-Hop Ahead Reservation scheme. Otherwise, it is an N-Hop Ahead Reservation scheme. Hence, when the current transmission gets completed at the receiver, the next hop channel becomes ready. Since the reservation is done before the actual data transmission takes place in the multihop network, the delay for acquiring the next channel is avoided, thereby reducing the multi-hop path setup delay to zero, except for the setup delay at the source node. LEEM performs better in terms of energy-efficiency and end-to-end delay when compared to PTW and STEM.

Geographic Random Forwarding GeRaF [33] is a forwarding technique for ad-hoc and sensor networks based on geographical location. It enables nodes to be put to sleep and wake up without coordination and to integrate routing, MAC and topology management into single layer. GeRaF is based on the assumption that sensor nodes have a means to determine their location, and that the positions of the final destination and of the transmitting node are explicitly included in each message. In this scheme, a node which hears a message is able (based on its own position toward the final destination) to access its own priority in acting as a relay for that message. All nodes who receive a message may volunteer to act as relays, and do so according to their own priority. This mechanism tries to choose the best positioned nodes as relays. In addition, since the section of relays is done a posteriori, no topological knowledge or routing tables are needed at each node, but the position information is enough. Geographic routing is used here to enable nodes to be put to sleep and wake up without coordination, and to integrate routing, MAC and topology management into single layer. To handle collision, GeRaF uses busy tone on a separate radio. The availability of separate channels for the data traffic and the wakeup signaling is useful to facilitate protocol operation, in particular to avoid that prolonged beacon periods interfere with data traffic. In this case, there are no prolonged periods and therefore could use the second radio to let the receiving node issue a busy tone, which is a way to effectively prevent collisions at the receiver. More precisely, the first radio is used for data communication and second radio is used to issue busy tone. For collision handling, GeRaF requires extra hardware and does not address the problem of correlated contention characteristic to sensor networks.

B-MAC [19] proposed by Polastre *et al.* is a configurable MAC for sensor network that provides a flexible interface to obtain ultra low power operation, effective collision avoidance, and high channel utilization. To achieve low power operation, B-MAC employs an adaptive preamble sampling scheme to reduce duty cycle and minimize idle listening. B-MAC supports on-the-fly reconfiguration and provides bidirectional interfaces for system services to optimize performance, whether it be for throughput, latency, or power conservation. B-MAC protocol contains a small core of media access functionality. B-MAC uses clear channel assessment (CCA) and packet back-offs for channel arbitration, link layer acknowledgments for reliability, and low power listening (LPL) for low power communication. B-MAC is only a link protocol, with network services like organization, synchronization, and routing built above its implementation. Although B-MAC neither provides multi-packet mechanisms like hidden terminal support or message fragmentation nor enforces a particular low power policy, B-MAC has a set of interfaces that allow services to tune its operation (shown in Figure 1) in addition to the standard message interfaces. These interfaces allow network services to adjust BMACs mechanisms, including CCA, acknowledgments, back-offs, and LPL. By exposing a set of configurable mechanisms, protocols built on B-MAC make local policy decisions to optimize power consumption, latency, throughput, fairness or reliability. For effective collision avoidance, a MAC protocol must be able to accurately determine if the channel is clear, referred to as Clear Channel Assessment (CCA). Since the ambient noise changes depending on the environment, B-MAC employs software automatic gain control for estimating the noise floor. Signal strength samples are taken at times when the channel is assumed to be freesuch as immediately after transmitting a packet or when the data path of the radio stack is not receiving valid data. B-MAC provides optional link-layer acknowledgment support. If acknowledgments are enabled, B-MAC immediately transfers an acknowledgment code after receiving a unicast packet. If the transmitting node receives the acknowledgment, an acknowledge bit is set in the senders transmission message buffer. 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. Though B-MAC is light weight and able to provide good performance, it does not handle contention and hidden terminal problems.

CC-MAC [26] is the first MAC of its kind to exploit the spatially correlated contention to its advantage. It is designed based on the principle that a sensor node can act as representative node for several other sensor nodes observing the correlated data. A subset of nodes that send messages to sink are called *representative nodes* and CC-MAC employs a distributed algorithm that identifies these nodes such that distortion is minimized. When a specific source node, transmits its event record to the sink, all of its correlation neighbors have redundant information with respect to the distortion constraint. This redundant information, if sent, increases the overall latency and contention within the correlation region, as well as wasting scarce WSN energy resources. CC-MAC protocol aims to prevent the transmission of such redundant information and in addition, prioritize the forwarding of filtered data to the sink. In WSN, the sensor nodes have the dual functionality of being both data originators and data routers. Hence, the medium access is performed for two reasons: (1) Source Function: Source nodes with event information perform medium access in order to transmit their packets to the sink. (2) Router Function: Sensor nodes perform medium access in order to forward the packets received from other nodes to the next destination in the multi-hop path to the sink. According to the spatial correlation between observations in WSN, the medium access attempts due to the source function of the sensor nodes should be coordinated such that the transmission of the redundant information to the sink is collaboratively prevented. However, once a packet is injected into the network it has to be reliably transmitted to the sink since the correlation has now been filtered out. Hence, the route-thru packet is more valuable at an intermediate node than its own generated data packet. In order to address these two different contention attempts in

WSN, CC-MAC protocol contains two components corresponding to the source and router functionalities. Event MAC (E-MAC), filters out the correlated records and Network MAC (N-MAC) ensures prioritization of route-thru packets. More specifically, a node performs E-MAC when it has to transmit its sensor reading to the sink, while N-MAC is performed when a node receives a packet and tries to forward it to the next hop. Though CC-MAC shows a promising performance when compared to S-MAC and T-MAC, it still does the address the problem of correlated contention from multiple sources after employing CC-MAC.

Convergent MAC (CMAC) [13] was proposed to support low duty cycle while avoiding synchronization. CMAC avoids synchronization overhead while supporting low latency. By using zero communication when there is no traffic, CMAC allows operation at very low duty cycles. When carrying traffic, CMAC first uses anycast to wake up forwarding nodes, and then converges from route-suboptimal anycast with unsynchronized duty cycling to route-optimal unicast with synchronized scheduling. CMAC has three main components: *Aggressive* RTS equipped with *double channel check* for channel assessment, *anycast* to quickly discover forwarder, and *convergent packet forwarding* to reduce the anycast overhead. The long preamble mechanism of BMAC incurs high latency in order to ensure that the receiver is awake before sending the data. However, the receiver may wake up much earlier than the end of the preamble, which makes part of the preamble transmission wasteful. CMAC replaces this preamble with aggressive RTS, which breaks up the long preamble into multiple RTS packets also called as RTS burst. The RTS packets will be separated by fixed short gaps each of which allows receivers to send back CTS packets. Once the transmitter receives a CTS, it sends the data packet immediately. Each gap need not accommodate the entire packet as long as the sender can detect the preamble and cancel the next RTS transmission. To allow the nodes to work at a very low duty cycle, the nodes must access the channel very quickly each time they wake up. In order to avoid a receiver waking up between RTS transmission to miss the RTS burst, double channel check is used. Under this mechanism a node will assess the channel twice with a fixed short separation between them each time a node wakes up. When an node does aggressive RTS, then nodes in the neighborhood can reply with CTS. The neighbor nodes which are closer to destination are defined as forwarding set. There are many nodes which can reply to RTS burst and hence CMAC prioritizes them to select the best forwarder. CMAC achieves this by sub dividing the region into three parts and then nodes in each region can send their CTS packet in their respective CTS slots. Further, each CTS slot is divided into mini CTS slots and each node in a region select a mini slot randomly. When other nodes here the CTS transmission, they will cancel their pending CTS transmissions. A node can use anycast or unicast to forward a data packet. The nodes can make local decisions depending on the given parameter to decide between them. When a node after a cycle of RTS burst, cannot find a better node in that region, it converges from anycast to unicast to make greater progress. Though CMAC provides better delay and energy efficiency, the RTS burst increase the energy consumption and wastes the channel. Also, CMAC does not address spatially-correlated contention.

Energy-efficient Contention Resilient MAC (ECR-MAC) [31] was proposed by Zhou and Medidi to address the energy-efficiency and delay without the overhead of synchronization or additional hardware. ECR-MAC uses *Dynamic Forwarder Selection*(DFS) to add flexibility to packet forwarding to improve energy-efficiency and delay. DFS uses a lightweight topology organization procedure that assigns multiple potential forwarders to collaboratively serve for a sender, and each forwarder employs independent wakeup schedules without synchronization to reduce protocol overhead. A sender dynamically chooses the first activated forwarder for its packet forwarding, so ECR-MAC can save energy by employing a low duty cycle and also achieve a short delay. ECR-MAC uses active/sleep duty cycles and operates as follows: each node will be periodically activated for T_{active} to detect any packet-forwarding requirements. Before transmitting packets, a sender x will sense for a long enough duration T_{cs} to detect on-going communication. If none is detected, x sends WAKEUP messages periodically to see if one of its potential forwarders is awake. Any xs potential forwarder that receives a WAKEUP message will send a REPLY message after sensing for a short random time R_{sense}, which effectively reduces REPLY collisions from neighboring potential forwarders. Upon receiving the first REPLY from a potential forwarder y, x sends data to y. y then becomes xs real forwarder and replies an ACK message. T_{active} should be at least $2T_{wakeup} + R_{sense} + T_{reply}$ long to ensure an active potential forwarder to receive WAKEUP messages. ECR-MAC handles reply collisions by letting the sender after receiving colliding messages, resend a WAKEUP message to notify the potential forwarder that causes collision. At this time, the corresponding potential forwarders randomly backoff a time period with in the contention window to send a second REPLY and also randomly adjust their wakeup time slot to avoid future collision.

3.4 Hybrid Schemes

Topology control techniques use the dense deployment of sensors to achieve energy-efficiency either by reducing the transmission power or by making redundant node go to sleep. These techniques still maintain network connectivity and are able to provide good network performance. On the other hand, wakeup schemes utilize the temporal redundancy to make nodes sleep when they are not needed for communication. Since these does not maintain network performance anymore, achieving energy-efficiency together with performance is a challenge in these schemes. There are new class of protocols that combine both topology control and wakeup schemes to achieve better energy efficiency. Under hybrid technique, the topology control is used to assign efficient wakeup schedule to nodes to enjoy the advantage of both topology control and wakeup schemes.

A new class of wakeup method called *multi-parent* [9] was introduced by Keshavarzian *et al.* which assigns multiple forwarder with different wakeup schedules to each node in the network. In many application scenarios and network deployments, the network is dense and therefore most of the nodes at higher levels have many neighbors and they can communicate with many lower level nodes. Multi-parent approach take advantage of this fact in the multi-parent idea and exploit the full connectivity of the network. Instead of using a tree network topology where a single parent is assigned to each node in the network and the messages are always forwarded through the same fixed path, multiple paths and multiple parents with different wakeup schedules are associated with each node in the network. Basically, in the multi-parent idea when a message arrives to a node in the network, depending on its arrival time it chooses the fastest path in the network to get to its destination. For example, if the node has two parents it forwards the message to the the parent which will wake up earlier. Another message that comes at a later time may find the other parent/path to be optimal at that moment. The main assumption for the multi-parent method is that we can divide the nodes in the network into multiple disjoint groups such that at least one parent from each group can be assigned to any node in the network. For example, nodes in a graph are divided into two groups, namely red and blue group. Then each node in the network has one red parent (mother) and one blue parent (father). The base station is a special node which belongs to both groups and can act as both parents. Unfortunately, this problem of dividing the nodes into groups even for two forwarders is *NP-Complete*. Although a centralized heuristic was proposed, its applicability in wireless sensor nodes with limited capability is questionable.

Zhou and Medidi proposed a distributed sleep based topology control[32] to schedule nodes' wakeup time slots, and designed a MAC protocol to benefit from this topology control for improving energy-efficiency and delay, and handle spatially-correlated contention. This technique aims at achieving very low duty cycle with low bounded end-to-end delay. In this topology control mechanism, the deployment area is divided into concentric circles and each node in a circle selects forwarder from nodes in neighboring circle closer to the base station. The width of the circles decides the path length to the base station. Each node in a ring will have equal number of forwarders and their wakeup times are evenly distributed. This gives the bound on the end-to-end delay. Also, staggering is employed by making nodes wake up like a chain to improve the delay. But this topology control mechanism assumes a tight global synchronization which is very expensive to achieve.

CHAPTER FOUR

LATTICE-BASED MEDIA ACCESS

The protocol design for wireless sensor network is steered by certain fundamental constraints characteristic to these networks like energy and spatially-correlated contention. The proposed technique is designed to improve the energy-efficiency and address the problem of spatially-correlated contention in wireless sensor network. The key idea behind the proposed technique is to construct a communication backbone across the network to address energy-efficiency and contention by reducing the number of active contenders for the channel. Also, the backbone provides a controlled topology which facilitates more than one forwarder for each backbone node to diffuse traffic as opposed to a data gathering tree which only increases contention in the converge-cast traffic inherent in sensor networks. Further, the constructed topology should allow a simple collision avoidance mechanism to reduce contention, a low duty cycle without increasing the setup latency and staggering to lower the delay to achieve significant energy savings, higher throughput and lower delay. The design choices considered while selecting backbone and construction of the lattice along with media access is described in this chapter.

4.1 Need for backbone

In wireless sensor network, when a phenomenon occurs, many near-by node sense the event. These nodes which sense the event try to send packets to the base station simultaneously. Obviously, in wireless network, when many nodes try to access the channel at the same time, it causes contention for the channel. Thus, among the nodes which compete for the channel only one node will win and can send the data. Other nodes back-off to try at a later time. This creates backlog of packets in nodes causing increase in delay and reducing the throughput. Also, energy is wasted in retrying to send packets. This contention is caused by nodes that are spatially correlated i.e, they are close-by nodes deployed in the same area. And hence, this contention from near-by nodes is termed as *spatially correlated contention* [26]. This contention is a fundamental problem characteristic to sensor network which degrades performance like increasing energy consumption, prolongs the delay and affects the throughput. If spatially-correlated contention is not taken care, it will definitely degrade the performance of the network. In some cases, it may even affect the performance of the network so much that the network will not serve its deployment purpose. This contention is not confined to the event area and will persist at every hop to the base station. In traditional *converge-cast traffic* - where traffic originating at any node ultimately reaches the base station, many nodes at a hop compete for a single forwarder at next hop. The nodes sensing the event might forward the data to different forwarder which could compete for forwarder at next hop. This continues in the converge-cast tree till the packet reaches the base station. This increases the delay at every hop as it take more time for the node to forward packets to its forwarder in the presence of contention.

As spatially-correlated contention is detrimental to the network performance, it must be addressed by protocols designed for sensor network. The root cause of this contention is many nodes trying to access the channel simultaneously. Restricting the number of nodes that can access the channel reduces this contention. Based on this observation, our protocol handles this contention by selecting a subset of nodes from the network that can access the channel. Since the access to the channel is exclusively given to this selected subset of nodes, it is the responsibility of these nodes to maintain the network connectivity. Hence, this node subset collaborate to provide a communication backbone infrastructure to the network. The non-backbone nodes in the network can use this backbone infrastructure to send the data to the base station. The non-backbone nodes do not participate in any communication activity to maintain network connectivity; hence they can turn off their radio and go to sleep. Using such a node subset to assist in communication makes the packet flow without incurring much delay due to reduction in contention. This technique of selecting node subset to form the backbone restricts the communication pattern in the network and can be categorized a topology control mechanism.

Protocols like GAF [28] and SPAN [2] also implement topology control mechanism and construct a backbone. GAF divides the entire deployment area into virtual grid and selects a representative from each grid to be a backbone node. SPAN uses a distributive algorithm to select coordinator nodes to be a part of backbone. The motivation behind their backbone construction is to make more nodes turn-off their radio and goto sleep. The proposed backbone construction aims at selecting nodes so as to minimize the spatially-correlated contention which is not addressed by other protocols. In addition to minimizing contention, this protocol also makes all non-backbone nodes goto sleep which brings the additional advantage of being energy-efficient. By selecting the subset of nodes to form backbone, the proposed technique minimize contention as well as achieve energy efficiency which substantiates the need for a backbone in sensor networks.

4.2 Backbone design - Lattice

Our communication backbone is comprised of nodes that have been selected specifically to minimize contention. Such a backbone can be made efficient by designing them based on certain characteristic specific to sensor networks. This section explores some of the design choices considered during backbone structure selection and reason for choosing lattice as a suitable backbone to minimize contention and improve energy-efficiency.

Topology control mechanism like GAF [28] and SPAN [2] which constructs backbone for energy-efficiency, make their backbone nodes active throughout. Since idle listening consumes substantial amount of energy, their backbone nodes tend to lose their battery power quickly and die early. This decreases the life time of the backbone and hence the network lifetime. In order to improve the backbone lifetime, nodes take turn to be a backbone node. This allows load sharing among nodes resulting in uniform energy depletion among them. In order to decide the next backbone node, these protocols use the residual energy level of nodes. For this, they exchange messages about their energy level and other parameters which are inputs to an election algorithm which decides the next backbone node. Always preference is given to node having more residual energy. This process of sharing information about the energy level and selecting next backbone node wastes energy unnecessarily.

As the traffic in sensor network is infrequent, the backbone nodes need not be active all the time to provide connectivity. It is a waste of energy to make backbone nodes active all the time when the network is going to be idle most of the time. Instead the backbone nodes can just wakeup at regular intervals of time to see if it needs to help in packet forwarding activity or else goto sleep. In other words, backbone nodes can have duty cycle and follow a wakeup schedule scheme to conserve energy without affecting the performance of the network. By making the backbone nodes sleep, more energy is saved which increases the life time of the backbone as well as the network. Thus the proposed technique uses both topology control and wakeup schedule; hence achieving higher energy-efficiency.

Although, duty cycles help in conserving energy, it comes with a price. At each hop, a node has to wait for its forwarder to wakeup before it can send the packet. The delay between the time a node receives a packet to the time at which it can forward the data to its next hop forwarder is termed as *setup latency*. Having duty cycles on backbone nodes will incur setup latency at each hop. This would increase the end-to-end delay of the packet. Other protocols [14, 25, 31, 32] which use duty cycle for energy-efficiency also suffer from setup latency. Some of them use additional hardware to reduce this latency. Additional hardware increases the cost of the network which is undesirable. Others use a technique named as *staggering* - aligning the wake up of nodes along a path sequentially like a chain reaction, to allow continuous packet forwarding to the base station. A example of such staggering in a data-gathering tree is shown in Figure 4.1. It was first introduced in DMAC [14] and later used by many other protocols like MS-MAC [32] to minimize the delay. Staggering significantly reduces the time required to wait for the forwarder to wakeup and hence reduces the overall end-to-end delay. As duty cycles are desired in the proposed technique to achieve better energy-efficiency, allowing staggering on the backbone will reduce the setup latency improving the end-to-end delay.

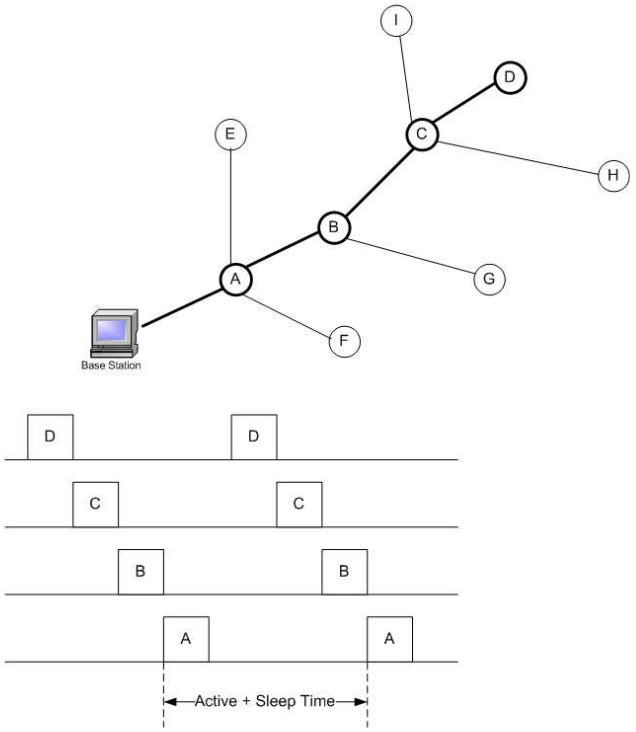

Figure 4.1: Staggering in Data-gathering Tree

The proposed technique improves energy-efficiency and handles spatially-correlated contention by selecting a subset of nodes to form a communication backbone. To further improve energy efficiency and delay it is desirable to have duty cycle and staggering on the backbone nodes. Following this, the structure of the backbone needs to be decided.

Traditionally, many protocols suggest using shortest-path tree for data-gatherings in sensor network. Since traffic originating at any node will ultimately converge at the base station, a tree structure naturally captures this converge-cast traffic. In a shortest-path tree, each node in the tree is associated with a single forwarder which is one hop closer to the destination. So it takes minimum number of hops to reach the destination using a shortest path tree. Although shortest-path tree is an ideal solution, as many nodes in a tree share a single forwarder, it gives rise to contention. This contention for forwarder worsens as the traffic rate increases and degrades the network performance in terms of delay, energy-efficiency and throughput. This contention for forwarder occurs because each node is associated with only one forwarder. If a node has multiple forwarder, then this contention would decrease as there are more forwarder for a node to choose from. Multi-parent approach [9] is based on this idea and allows each node to have more than one forwarder. The wakeup times of these forwarders are uniformly distributed. Hence depending on when a packet arrives at a node, it takes the forwarder who wakes up first. This allows a node to choose the forwarder based on packet arrival leading to multiple paths between source and destination. As opposed to traditional shortest path tree, this diffuses the traffic in the network relieving contention and improving performance. ECR-MAC [31] uses *dynamic forwarder selection* to select a forwarder among multiple forwarders. Here, the forwarders have randomly selected wakeup times and a nodes uses a forwarder that wakes up first. Similarly, MS-MAC [32] uses topology control to select multiple forwarders whose wakeup times are evenly distributed. All these protocols were able to benefit from having multiple forwarders and achieve better performance. Having multiple forwarder also allows the network to have a lower duty cycle and still maintain the performance. This motivates the proposed protocol to have multiple forwarders which would reduce contention by diffusing the traffic improving the performance.

From the above discussion, it can be summarized that the proposed technique requires the backbone to support duty cycle, staggering and multiple forwarder to achieve better energy-efficiency and address contention.

Lattice or mesh, a well-studied interconnection structure, can be considered for backbone topology. An example of lattice is shown in Figure 4.2. Lattice structure has inbuilt regularity where every node (except ones in boundary) has four neighbors. Nodes in the lattice backbone structure can implement duty cycles following some wakeup schedule scheme. As lattice is a controlled structure, the lattice nodes can be assigned a level number which will reflect their lattice hop distance from the base station in the lattice. Using this level number, the wakeup time of the lattice node can be offset in a cycle time to create staggering to minimize the delay. This indicates that lattice structure inherently provides a framework for implementing staggering. If base

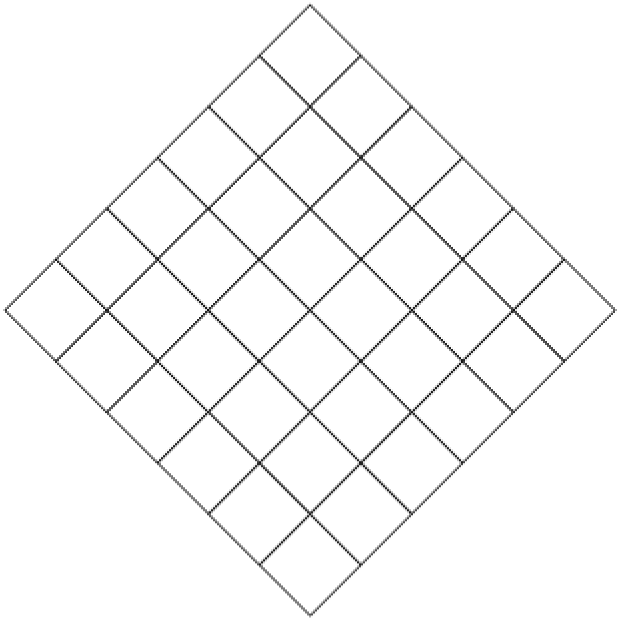

Figure 4.2: Lattice Structure

station is assumed to be one of the lattice nodes, then every lattice node has two lattice neighbors which are closer to the base station and hence can be regarded as forwarders (parents). The other two lattice neighbors are the ones from which the given node can receive data and are denoted as senders (children). Thus, lattice provides two forwarders for each backbone node supporting multiple paths between source and base station. Since each lattice node has just four neighbors, a node needs to just synchronize with its neighbors for implementing staggering and hence local synchronization is sufficient.

4.3 Assumptions

Similar to various other topology control techniques [28, 32], it is assumed that each node is aware of its own location, which can be obtained using GPS or by other localization techniques [15]. Also it is assumed that each node knows its two hop information which can be easily obtained by two stages of local broadcasts. Further, for simplifying the description, it is assumed that the network deployment does not have any physical holes and the outer boundary identified.

4.4 Lattice Construction

To construct the lattice, nodes that form the lattice need to be chosen and tagged that they will be part of the backbone. In identifying such nodes which will form the backbone, the link length between the adjacent nodes in the lattice is an obvious design metric and must be decided. Intuitively, selecting this link length to be the maximum radio transmission range, r, of the sensor nodes seems obvious. A lattice where link length between adjacent nodes is r is denoted to be an *ideal lattice*: in the sense that it requires the minimum number of nodes to encompass the network and also minimizes the path length in the backbone among possible lattices.

As most of the time, sensor nodes are sprinkled over the sensing area in an uncontrolled fashion, deployments typically suffer from irregularities and identifying an ideal mesh is difficult and requires global information. Moreover, the proposed technique is relying on the links in the lattice heavily to accomplish all communication in the WSN: with a link length of r, there will be more wireless losses at the receiver. In particular, after about 70% of the maximum radio transmission range, the signal-to-noise ratio starts deteriorating significantly. So, instead the link length is chosen to be around 70% of the transmission range. This provides additional flexibility by allowing more choices for node selection while constructing a mesh. Obviously, this is not as good as the ideal lattice and requires more nodes to encompass the network at the same time inflating the path lengths in the backbone.

An obvious technique to identify the lattice, where the link length is around $0.7r$, is by identifying physical coordinates relative to the base station and using the node closest to these coordinates as the lattice nodes. However such an obvious technique, in the experimentation, did not lead to controlled lattices (unless the deployment is very dense) making some of the links needed too long and, sometimes, even artificially creating holes in the lattice. Another approach is to progressively add nodes to form lattice in a greedy fashion. As this method requires only local information to make local greedy decisions, its implementation can also be distributed which is suitable for implementation in sensor networks. Naturally, such a greedy technique may not necessarily provide the best possible lattice.

Our greedy construction of the lattice starts out from the base station BS. The BS selects four nodes X_1, X_2, X_3 and X_4 from its one hop neighborhood as shown in Figure 4.3. From the pairs (X_1, X_2), (X_2, X_3), (X_3, X_4) and (X_4, X_1), BS selects common neighbors Y_1, Y_2, Y_3, and Y_4 respectively so as to form four quadrilaterals and extends the lattice from the available two-hop information. While there are many node pairs which satisfy this condition, the BS selects the best pair depending on how close they are to provide the desired link lengths. Once the BS selects these nodes, it tags these nodes as participants in the lattice by sending appropriate messages.

Nodes Y_1 and Y_3 are further charged to grow the chain in opposite directions. Node Y_1 selects two nodes A_1 and A_2 first and then their common neighbor Y_1'. Among all the pairs that are available, Y_1 selects the best pair again based on their positions and desired link lengths. These new nodes are inducted into the lattice and further Y_1' is tagged responsible for continue growing the chain (in the vertical direction, in the figure). Similarly, in the other direction, Y_3 selects B_1 and B_2 along with their common neighbor Y_3' to keep growing the chain.

As this process of adding quadrilaterals for chain construction progresses, node Y_1 has two neighbor A_1 and X_1 from which it can add a quadrilateral by finding a common neighbor C_1, again using the two hop information, to continue lattice construction in the horizontal direction. Adding lattice neighbor C_1 allows node X_1 to add another quadrilateral by identifying an appropriate common neighbor of nodes Y_4 and C_1.

This process continues till it reaches the boundary. At that point, lattice nodes try to grow along the boundary; if they cannot, their neighbors are checked for the possibility of continuing the boundary expansion.

Figure 4.4 shows a lattice obtained with this greedy technique. Clearly, this final lattice does not look anything similar to the one expected out of the obvious approach utilizing physical coordinates

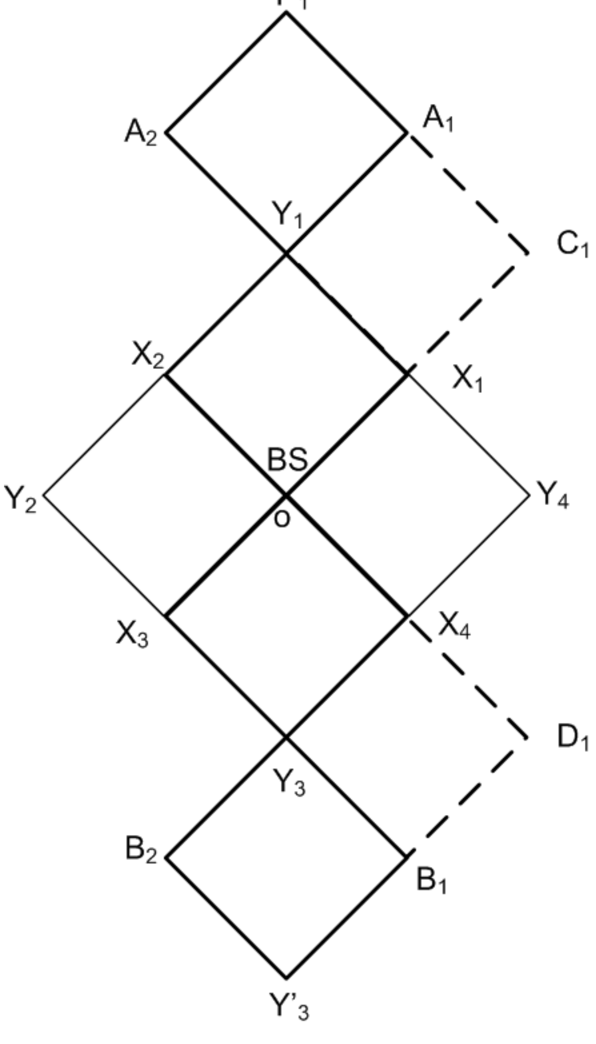

Figure 4.3: Lattice Construction

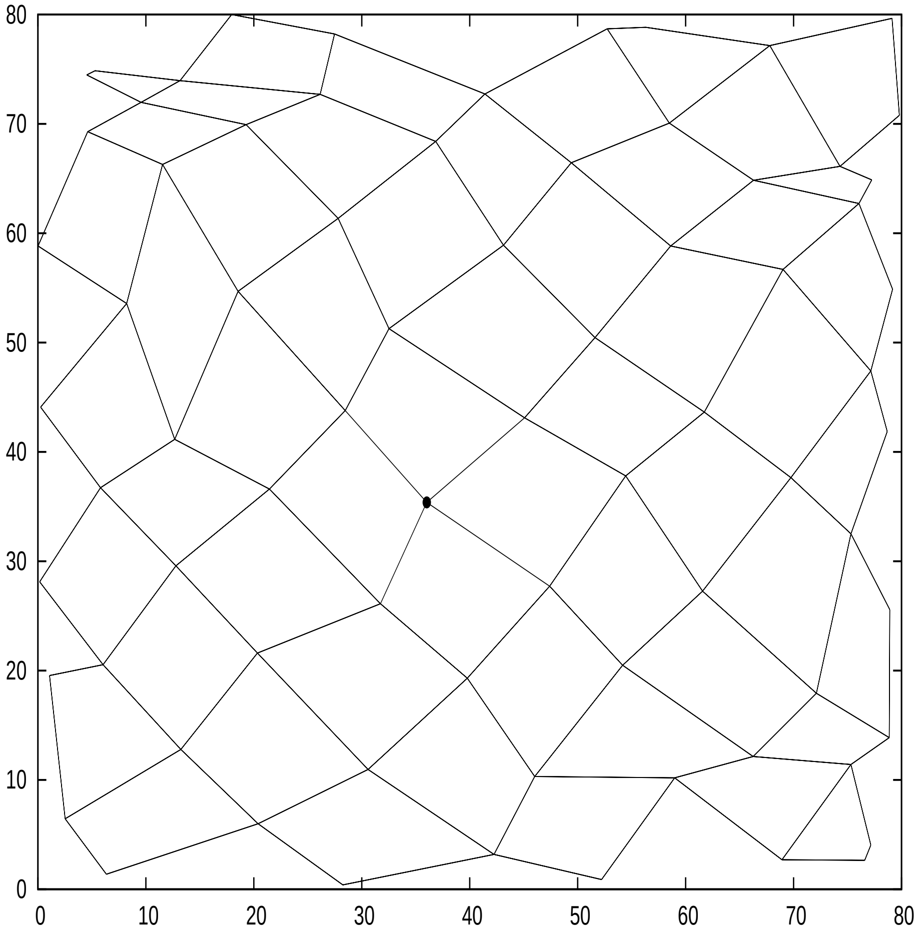

Figure 4.4: Resultant Lattice

but the links in the obtained lattice have lengths closer to the desired value and overcomes spatial irregularities in the deployment. More critically, the desired lattice can be constructed quickly in a distributed fashion using only local information.

4.5 Duty cycle with Staggering and Collision Avoidance

The previous section described a greedy, distributed lattice construction technique. While formulating the design choices for the backbone, the backbone is required to support duty cycle, staggering and multiple forwarders. The bare lattice that is obtained from construction does not have all the properties that is desired in a backbone. Hence, lattice by itself, once constructed, does not deliver the desired performance. Its properties need to be exploited, in the context of wireless sensor networks, to design a tailored media access protocol to achieve the properties desired in a backbone. Also, in lattice, each node in the lattice will have two children (or nodes which will use it as a forwarder) from which it can receive data and could lead to contention. To take advantage of this lattice structure to achieve better performance, the wake-up scheduling and collision resolution needs to be fine-tuned to achieve energy-efficiency.

Traffic in wireless sensor network is infrequent enough that the backbone need not be active always. The proposed protocol can implement duty cycle over the lattice nodes also, providing an edge over other protocols: not only non-lattice nodes can turn off their radio circuitry, but also lattice nodes conserve energy by implementing a duty cycle thus extending the network lifetime.

To implement duty cycle, an obvious design metric is to determine the duration for which a node will be active in a cycle time. It is desirable to have a smaller active period as it results in higher energy-efficiency. During this active period in a cycle, the node has to receive data from its senders. The node has to be active long enough to receive the data packet. If during this time, the node has to receive a longer data packet, the active period has to long enough to accommodate the reception of a data packet. As the traffic is infrequent, the node wastes energy by being in active state for longer duration even when there is no traffic. In order to reduce this active period, the sender can send a small beacon indicating to the receiver that, receiver needs to be active to receive the data packet that is going to be sent next. This reduces the active time of the nodes as the beacon messages are small. When the network is idle, the backbone nodes will be active only for a small duration of time to receive a beacon. When there are traffic to be served, the nodes are indicated by beacon to stay awake longer to help in packet forwarding. In the proposed protocol, the beacon messages are denoted as REQUEST packets.

Naturally, the size of the REQUEST packet will determine the time period of the active slot. The amount of energy conserved is determined by the duty cycle which is ratio of $\frac{active\ time}{active+sleep\ time}$.

As the active period decreases, proportionally there is a better energy saving. In lattice, the regularity of the structure allows only two nodes which can forward data to any node and thus enabling a very small REQUEST packet which allows shrinking of the active time further. In MS-MAC [32] and ECR MAC [31], the active period has to be long enough to resolve contention from multiple senders and is 2.2 ms whereas in lattice it can be reduced up to 1 ms which provides better energy-efficiency for the same network parameters.

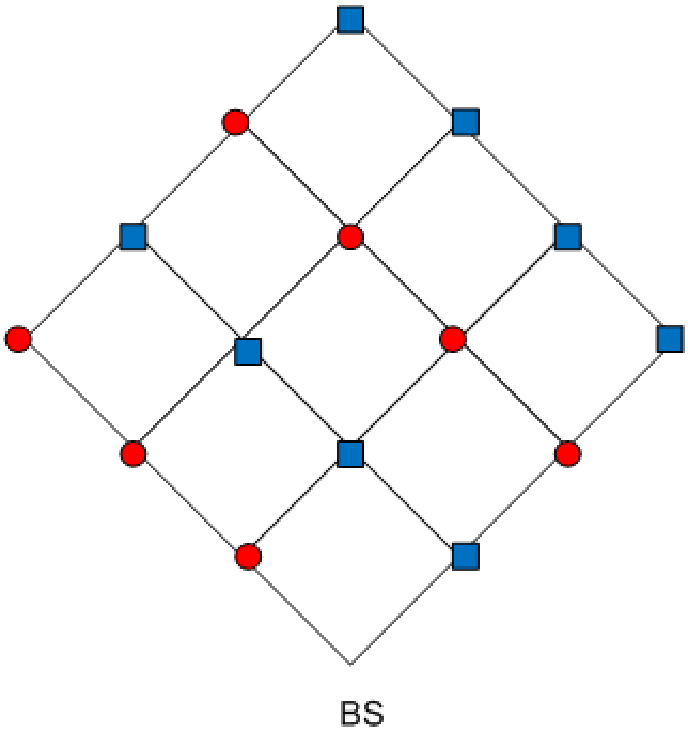

Figure 4.5: Lattice Color Assignment

In a lattice, each node has two forwarders through which it can send data to the base station. If these two forwarders wake up at the same time, then the node can choose one of them to forward data. But, this will not fully utilize the two forwarders that a node has. Instead, if these two forwarders wake up at different time, then the node will have two chances in a cycle time to forward the data to the forwarders. This will also reduce the delay at first hop since a node need not wait for a single slot in the cycle. If the wakeup slots of these two forwarders are evenly spaced in a cycle time, this naturally halves the delay at the first hop. Moreover, it is not always required to evenly space the forwarder wakeup times. This opens up an option of designing how the wakeup times needs to be distributed in a cycle. In order to select the wakeup time for each node, a simple mechanism of two color assignment (red and blue) for nodes can be used. The lattice nodes are colored in such a way that each node has a red and blue forwarder. All the red nodes will wakeup together at the same time and all the blue nodes will wakeup together at a different time. An example such a color assignment is shown in Figure 4.5. In the figure, circle represents red nodes and square represents blue nodes and every node has two forwarders - red and blue.

The proposed technique requires the lattice to support staggering to minimize the end-to-end delay. Thus all the red nodes at the same level wakeup at the same time but after an offset from all the red nodes at the next (farther from the base station) level. This minimizes the setup latency and provides better delay. But for each node we have two forwarders of different color (red and blue). It is possible that wakeup times of red and blue nodes at the same level can be next to each other producing a double pipe effect or they can be equally spaced in a cycle to produced two single pipe. They are shown in Figure 4.6 and 4.7. Altering the wakeup slot of two forwarders produces different effect on the performance, which we will discuss in the performance evaluation section.

Assigning colors to the nodes of the lattice and implementing staggering alone does not fully exploit the structure of the lattice. In lattice, a node can receive data only from two known senders (children). But these two senders can try to send the data to a forwarder at the same time. This causes contention and collision among the two senders which can be avoided by exploiting the regularity of the lattice structure where every node has two children. This regularity allows us to design a simple TDMA based collision avoidance mechanism to resolve contention from the two senders. It is achieved by simply and consistently giving preference to one sender over the other - for example, for a red colored node, its red colored (called left, for ease) sender over the blue (right) sender. To decrease the duration of active period in a cycle, we only send a REQUEST packet to make the node ready to receive the real data packet. Also, for each cycle in the proposed scheme,

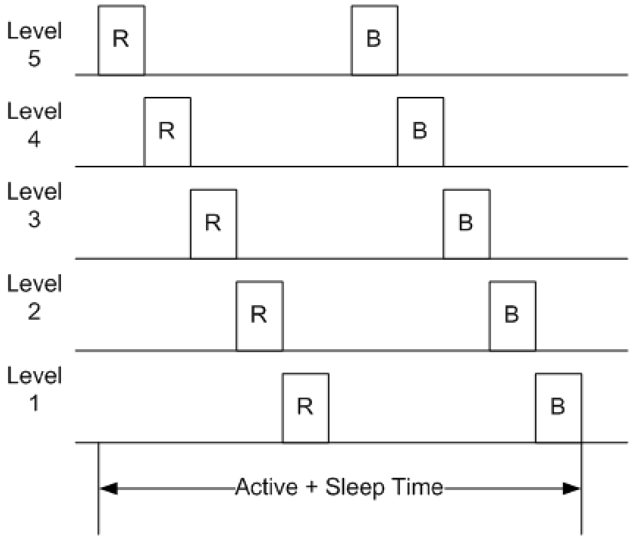

Figure 4.6: Staggering - Single Pipe

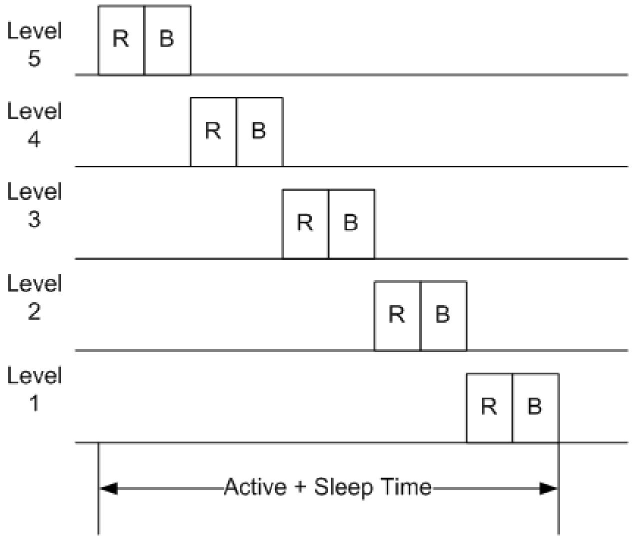

Figure 4.7: Staggering - Double Pipe

a forwarder can accept two data packets only. When a lattice node has a data to send, it follows the following step to achieves it. Each node left followed by right, sends a REQUEST packet. This will indicate the number of packets it needs to send. The forwarder node which receives these two packets decide how much each of its children needs to send. If they both have packets to send, then the forwarder decides that they should send one packet each. If there is only one sender requesting, it can send two packets. The forwarder conveys this information to the senders in a REPLY packet. This is shown in Figure 4.8. Then depending on the reply, if a sender can use all the two slot, then it sends one data packet followed by the other. If the sender can send one packet only, then if it is a left node it will send in the first slot followed by right node which sends in the second slot. Thus at any point of time, the senders to a lattice forwarder will never compete or contend for the channel. The number of data packets could have been adaptively decided based on the load. But this would lead to collisions when implementing staggering. Naturally, the staggering offset is slightly larger (increasing the end-to-end delay) than if the technique allowed only a single data packet to be forwarded: this design choice prioritizes coping with spatially-correlated contention as more critical over speeding up the packet forwarding.

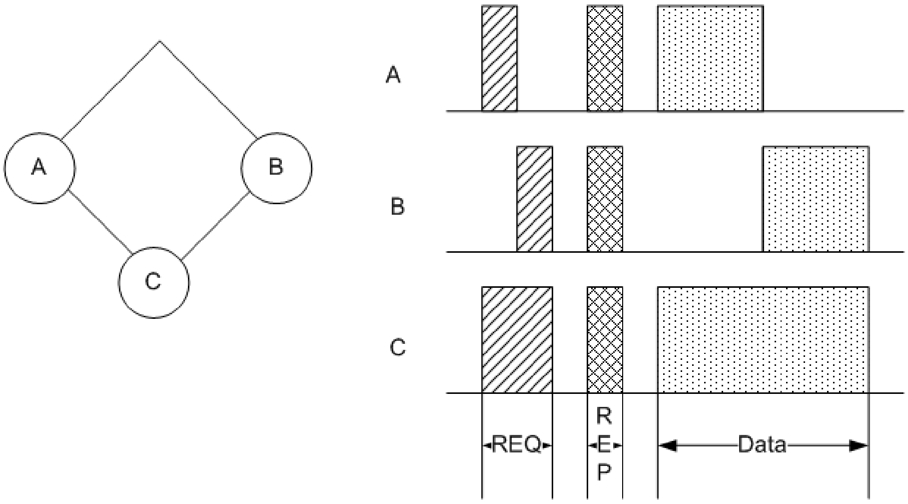

Figure 4.8: Collision Avoidance

4.6 Data aggregation

Lattice construction and designing the media access to take advantage of this backbone were described in the previous section. Lattice is only subset of nodes in the network. The nodes which are not a part of the backbone needs to send data to the lattice so that it can be forwarded to the base station. Since the backbone structure is sparse, many non-backbone nodes try to access the lattice causing contention and affecting the backbone communication. In order to avoid contention to access backbone and preserve the communication slots of the backbone to maintain network connectivity, the proposed technique uses a simple *packet collection mechanism* as follows.

During lattice construction, all nodes listen and collect information regarding their backbone neighbors. Using this information, a non-lattice node picks up the nearest lattice backbone neighbor and associates with it. When the non-lattice nodes sense an event, they will wait for the backbone node to wakeup and send the packet. In order to reduce contention, a small random wait time before sending packet to the backbone neighbor is included. All non-lattice nodes that need to send packets to a lattice neighbor will be active and listening to the channel. When one of the contenders succeed in sending packet to the lattice node, the others back-off and wait for a collection message from the lattice node. A lattice node which gets a packet from the non-lattice neighbor immediately switches to a *packet collection* mode, become active for an appropriate interval of time which does not affect the backbone communication, and sends the collection message. Since the nodes sending data to a lattice node are spatially-correlated, we can perform a meaningful *first-hop aggregation* to improve the performance of the network: we approximate an aggregated packet to be twice in length than original packet length to account for aggregation.

Lattice along with collision avoidance, staggering and data aggregation provides better performance in terms of delay, throughput and energy consumption. Since every node in the lattice can receive data only from two senders, it is possible to avoid collision with a simple collision avoidance mechanism. So, the nodes no longer need to compete for channel. Having more than one forwarder helps in diffusing the traffic. Staggering reduces the setup latency and improves the delay. Moreover, the energy saving can be attributed to regular nodes going to sleep and low duty cycles on the lattice nodes.

CHAPTER FIVE

PERFORMANCE EVALUATION

To evaluate the performance of the proposed protocol, the lattice construction and the associated MAC were implemented using $ns2$ [18] simulator. Extensive experiments were conducted to test the performance of lattice and its associated MAC. As lattice employs low duty cycle, staggering and multiple forwarders, it is compared against DMAC[14] (which utilizes staggering), ECR-MAC[31] (low duty cycle and multiple forwarders) and MS-MAC[32] (staggering, low duty cycle and multiple forwarders) in terms of end-to-end delay, throughput and energy consumption. These protocols have been shown to perform better than the other existing protocols. In addition, for baseline comparison, it is compared against IEEE 802.11 which is a fully-active CSMA/CA without any duty cycles.

5.1 Simulation Setup

The simulations were conducted on $80m \times 80m$ network with 500 nodes randomly distributed having transmission radius of $15m$ to mimic experiments reported in [14, 31, 32]. The base station, located at the left bottom corner, remains active at all times. For lattice backbone, results of both 1% and 0.5% duty cycles are reported. For DMAC, ECR-MAC and MS-MAC, 10%, 1% and 1% duty cycles are used as suggested in [14],[31] and [32]. The 0.5% duty cycle for lattice corresponds to a similar cycle time as that of ECR-MAC and MS-MAC. Moreover, for 1% lattice, we test the performance under two different types of staggering (single pipe and double pipe). Lattice 1% single pipe is denoted simply as Lattice 1% in the plots. Other parameters, which are the same as in [14, 31, 32], used in simulation are listed in Table 5.1. Experiments were conducted under single source scenario to test the basic performance against other protocols and in multi-source scenario to test the performance in the presence of spatially correlated contention. Each simulation was run for 30 sec, sufficiently long enough for event reporting activity. Each data point is an average of

Table 5.1: Simulation Parameters

Parameter	Value
Data Packet Size (bytes)	64
Bandwidth (Kbps)	100
Transmit Power (W)	0.66
Receive Power (W)	0.395
Idle Power (W)	0.35
Sleep Power (W)	0.00

20 independent runs, again similar to experimentation reported by the competing protocols.

5.2 Single-Source Scenario

In the single-source, the basic performance of lattice is compared against DMAC, ECR-MAC, MS-MAC and 802.11(labeled fully active in the plots). Similar to [3, 31, 32], a random source node (for lattice, the source is a backbone node) that is 6 hops away from the base station is selected to send reports. Figure 5.1 shows the end-to-end delay against the hops: the hops here represent the ideal hops and actual lattice path length is roughly double the ideal number of hops. In spite of the path inflation, the 1%-lattice outperforms other protocols and has the lowest delay, since it uses a smaller cycle time, employs staggering which almost eliminates the intermediate setup latency and has two forwarders with evenly distributed wakeup schedule. At 1% duty cycle, lattice has the lowest cycle time due to smaller active time which is possible because of existance of only two senders only for each lattice node. Whereas ECR-MAC and MS-MAC needs a longer active to resolve contention among the forwarders. The 1% lattice implementing double pipe staggering has higher delay when compared to the one implementing single pipe. In double pipe staggering, a node has to wait for its sibling to receive packets before it can forward the data to its forwarder. This increases the delay by almost twice when compared to single pipe staggering. Even the 0.5%-lattice has a delay comparable to that MS-MAC, which has the lowest delay among the competing protocols and which benefits from the multiplicity of forwarders in this single-source scenario.

Figure 5.2 show the energy consumption against reporting frequency. Separate plots to show

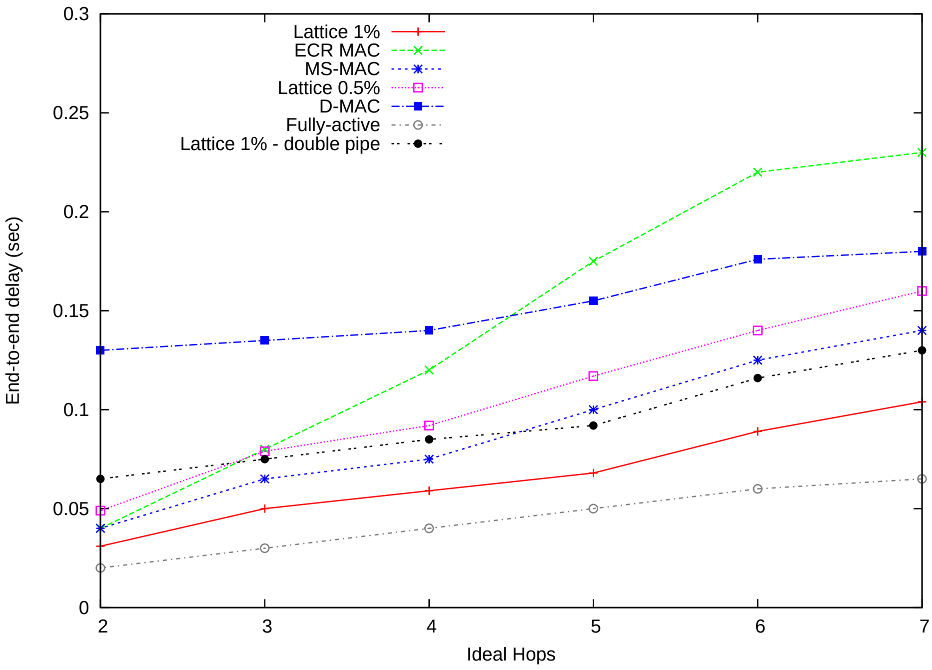

Figure 5.1: End-to-end delay vs. Ideal Hops

the average energy consumption of lattice nodes alone are provided. The fully-active protocol results in this plot is skipped since its energy consumption is too high. Clearly, lattice consumes least energy among all the competitor protocols as lattice employs two types of energy saving - all non-backbone nodes are made to sleep and also backbone nodes employ duty cycles. The average energy consumption of lattice nodes alone is lower as it employs the lowest active time which minimizes the idle listening still providing comparable, if not better delay. Although, the two different types of staggering shows a difference in terms of delay performance their energy energy consumption is nearly equal. The lattice energy consumption curve remains flat which indicates that lattice is able to handle increased traffic rate well. The 0.5% lattice has the lowest energy consumption as it employs the lowest duty cycle while maintaining a comparable peformace. D-MAC has the highest energy consumption as it employs the highest duty cycle of 10% and expends energy in idle listening.

Experiments to see the performance of lattice in terms of delay and throughput as the reporting

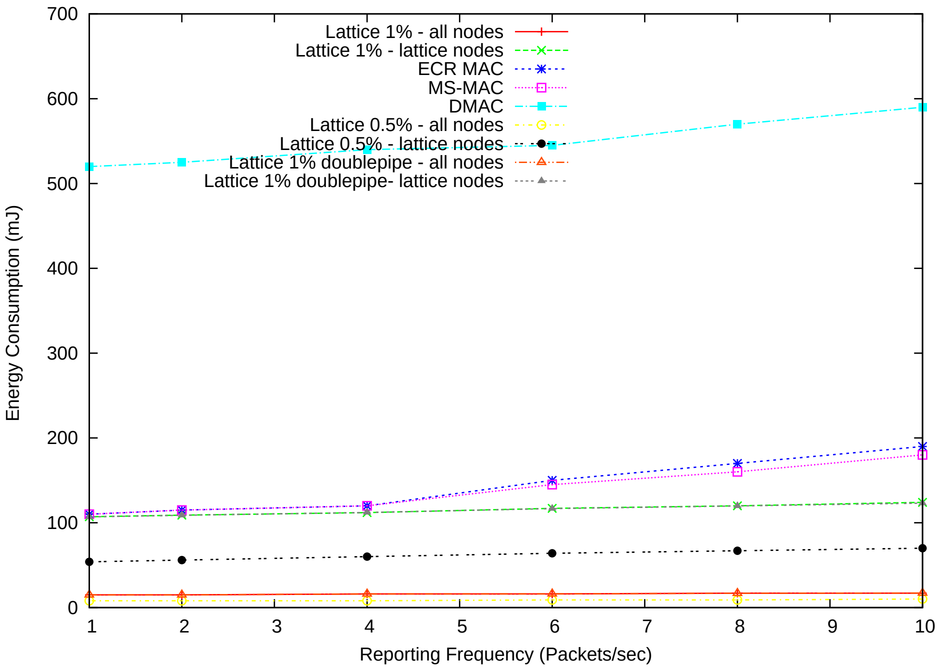

Figure 5.2: Energy Consumption vs. Reporting Frequency

frequency increases were conducted. Figure 5.3 shows the end-to-end delay in the lattice as reporting frequency increases. As the reporting frequency increases, as expected in lattice, the delay increases gradually but slowly since there are multiple paths available which diffuses the traffic. Delay curve of lattice 1% and 0.5% has a constant gap which is caused by the increased latency at the first hop for 0.5% lattice. Between lattice 1% double and single pipe curve there is a cross over point after which their delay characterstic changes. When the traffic rate is low, the single pipe performs better as it delivers packet with lower delay, where as in double pipe packets have to wait for extra slots unnecessarily increasing the delay. This trend changes as the traffic rate increases - double pipe performs better when compared to single pipe. When there are more packets to be sent, in double pipe since the wakeup of two forwarders are next to each other, the node can packets quickly without waiting for the forwarder like in the case of single pipe. This reduces the delay drastically when the traffic rate is high.

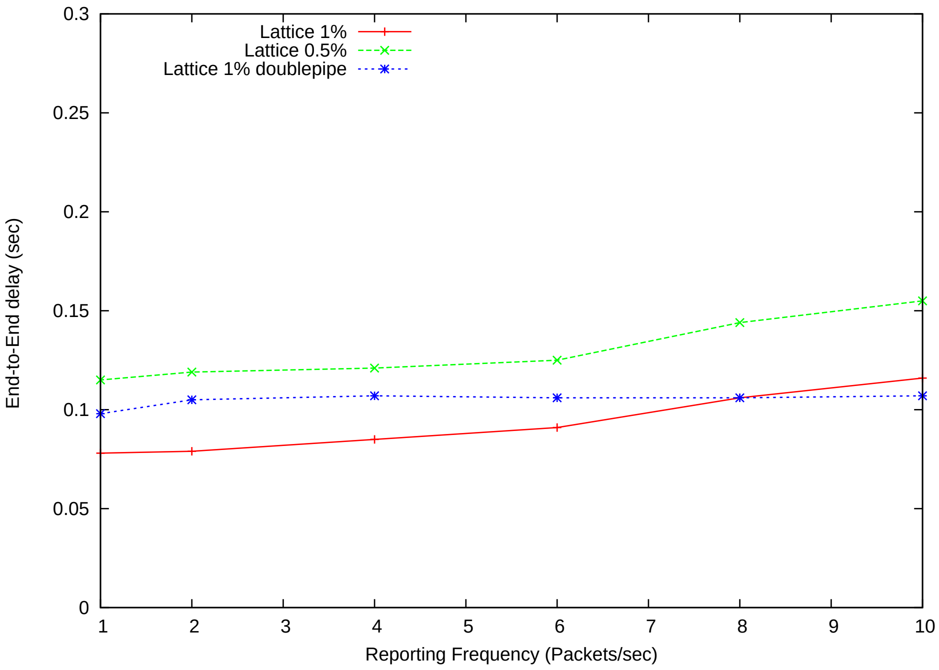

Figure 5.3: End-to-end delay vs. Reporting Frequency

The results of throughput against increase in frequency is reported in Figure 5.4. For throughput, almost all packets were received by the base station in the given reporting frequency range indicating that the network throughput was not affected. The type of staggering does not affect the throughput.

In summary, in a single-source scenario, the lattice which is a subset is able to perform equally well or better when compared to the competing protocols which uses all the nodes in the network. Using the smaller subset allows the proposed protocol to achieve significant energy savings. The controlled structure with two parents that allowed a natural collision avoidance coupled with staggering is able to provide a lower delay.

As the lattice constructed is a natural representation of the deployment area, if we consider each lattice node to be a representative of the nearby nodes as in CC-MAC[26], we can handle spatially-correlated contention in a straightforward fashion. However, since only a few nodes in a region report the event, there will be less event-level reliability.

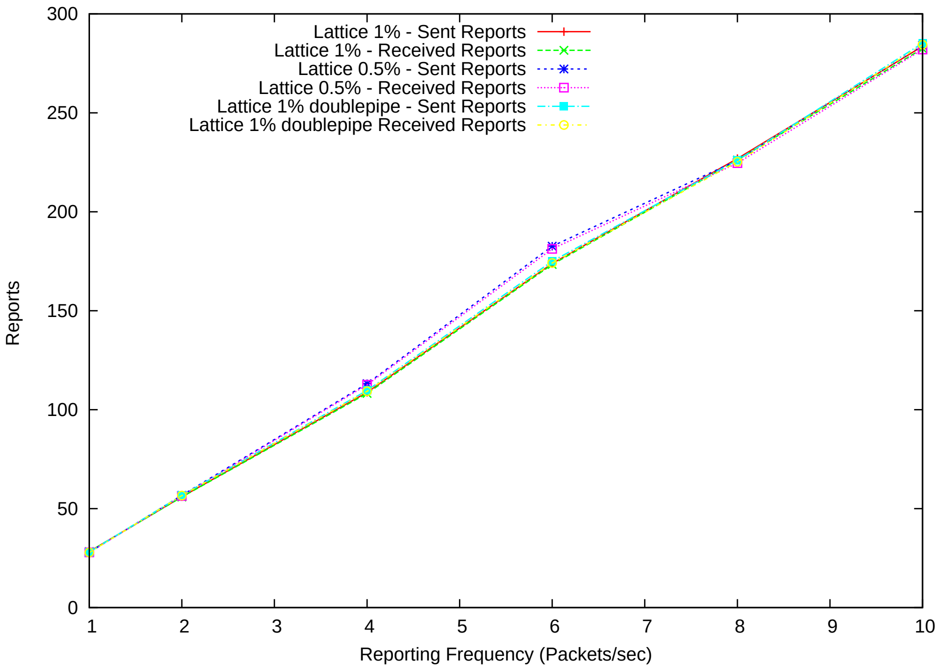

Figure 5.4: Number of Reports vs. Reporting Frequency

5.3 Multi-Source Scenario

Next the performance of the lattice is evaluated in multi-source scenario under spatially-correlated contentions by mimicking experiments reported in [31, 32]. An event randomly placed 6 hops away from the base station is selected: the sensor nodes within 10m of this event will report to the base station. Also, similar to [31, 32], 8 such sources are randomly choose to send the reports. Under this scenario, the performance of 0.5% and 1% lattice against DMAC, ECR-MAC and MS-MAC is evaluated, while increasing the reporting freqency.

For some sensor applications, it is desirable to allow first 10% of the packets to reach basestation quickly so that event response can be quick [14]. The end-to-end delay of the first 10% packets against reporting frequency is shown in Figure 5.5. Since aggregation is used to collect packets by a backbone node, we use the time difference between the creation of the oldest packet in an aggregated packet and the time the aggregated packet reaches the base station to represent the delay. Lattice outperforms its competitors and has the lowest delay in multi-source scenario.

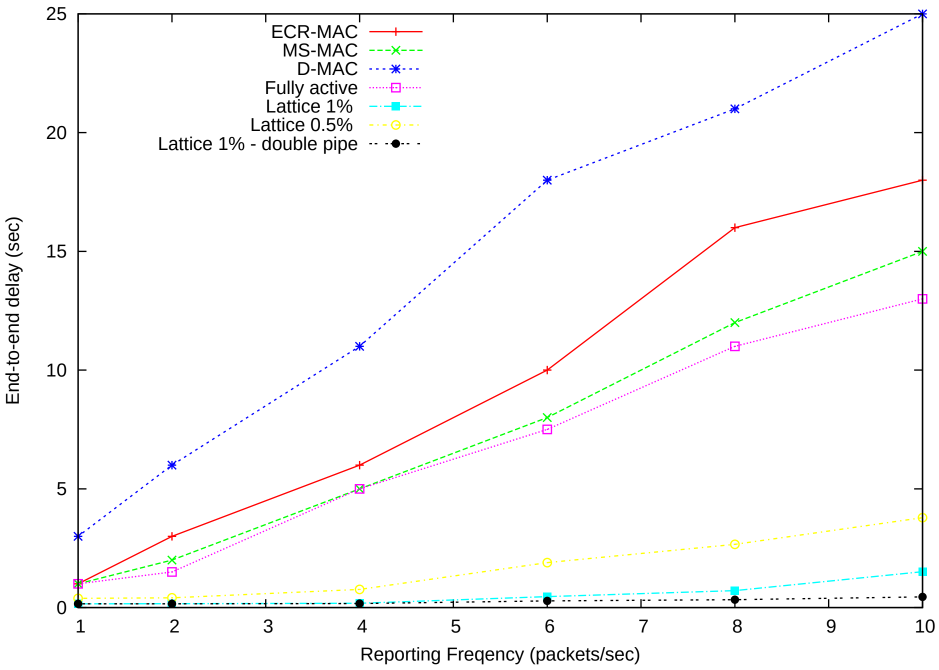

Figure 5.5: End-to-end delay vs. Reporting Frequency

In lattice when a non-backbone node senses an event and has packet to send, it tries to forward the packet to the nearest backbone node to which it is associated. When many nearby nodes sense an event and wants to sends packets to base station, they will only contend for the nearest backbone node. Since we have a simple packet collection mechanism where the backbone turns to fully-active mode to collect packets over a interval of time which will not affect the backbone communication, the regular nodes will be able to send packets to the backbone fairly quick. Once the packets are at the backbone node, they are aggregated with the packets that were collected during that cycle time. Once the packets are aggregated, they are transported to base station through the lattice backbone. As lattice utilizes staggering and multiple parents (hence multiple paths), the packets reach the base station quickly. The single-source scenario results supports this argument. Moreover the collision avoidance mechanism natural to the lattice structure gives us a handle over collisions, whereas other protocols like DMAC still suffer from contention at every hop. Though ECR-MAC and MS-MAC have the advantage of multiple forwarders, they will not be as beneficial as in single source scenario as there is high level of contention which degrades the performance. The 1% lattice with double pipe has lower delay than the lattice with single pipe. When there are more packets to send, the sending packets quickly rather than waiting for the next pipe as in single pipe case reduces the delay.

The energy consumption against the reporting frequency is shown in Figure 5.6. Again the plot for fully active 802.11 protocol is omitted as its too high. Clearly, lattice has the lowest energy consumption among all the competitor protocols. Lattice has the lowest energy consumption as it enjoys two types of energy saving: the non-backbone nodes go to sleep without participating in maintaining the network connectivity and the backbone nodes also employ duty cycle. The average energy consumption of lattice nodes alone is still lower than that of all nodes in the case of DMAC, ECR-MAC and MS-MAC due to smaller active time possible due to regularity of the lattice structure. The energy consumption trend almost remains flat with slight increase as the traffic increases which illustrates that lattice is able to handle the traffic load well. The lattice 0.5% has the lowest energy consumption due to smaller cycletime still providing a comparable performace. ECR-MAC and MS-MAC has lesser energy consumption when there is low traffic and increases as the traffic increases and stabilizes. D-MAC has the highest energy consumption due to increased cycle time and wastes energy in idle listening. Staggering (single and double pipe) does not change the energy consumption much as expected.

Figure 5.7 shows the number of reports that reach base station against the reporting frequency. Here the real (aggregated) packets that reach base station is denoted as physical packets . Each physical packet is some form of aggregation of individual packets sent by non-lattice nodes which is denoted as logical packets. Separate curves for physical and logical packets that reach base station is provided. As the reporting frequency increases, the number of packets aggregated into a single packet is higher, hence the gap between physical and logical packets increase as the rate increases which is to be expected. As the reporting rate increases, the number of physical packets reaching the base station is more in the case of double pipe as it is able to forward packets quickly

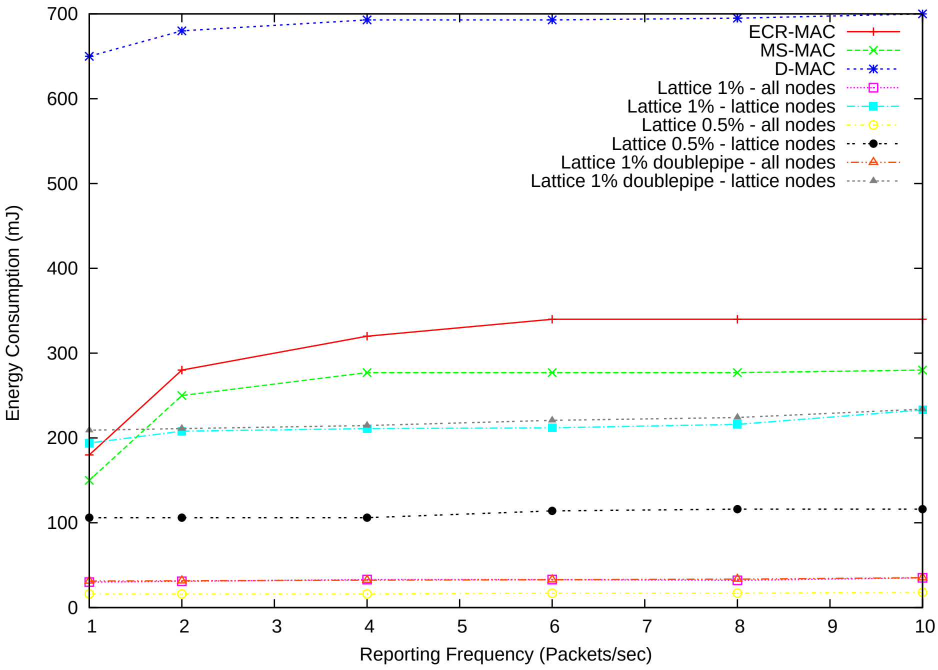

Figure 5.6: Energy Consumption vs. Reporting Frequency

and cope with contention. The single pipe provides good delay when the reporting rate is less where as double pipe provides better delay when the reporting rate is high.

Even though at lower rates all the protocols have comparable performance, at higher rates 1% lattice has higher throughput. Since lattice has a handle on contention even at higher rates and lower cycle time, it is able to deliver higher throughput. The 0.5% lattice has lower physical throughput since there is more aggregation. And, as the reporting rate increases, the logical packet throughput as expected surpasses the competitors.

In summary, a subset of nodes that form lattice is able to show much better performance in terms of energy efficiency, delay and throughput as it is able to handle contention better. For the simulation scenarios reported, the energy consumed for the one-time lattice construction was around 6 J. To put this in context, the capacity of 2/3-rds of two AA battery is around 10.8×10^3 J. The energy efficiency achieved with the lattice well compensates for this construction cost.

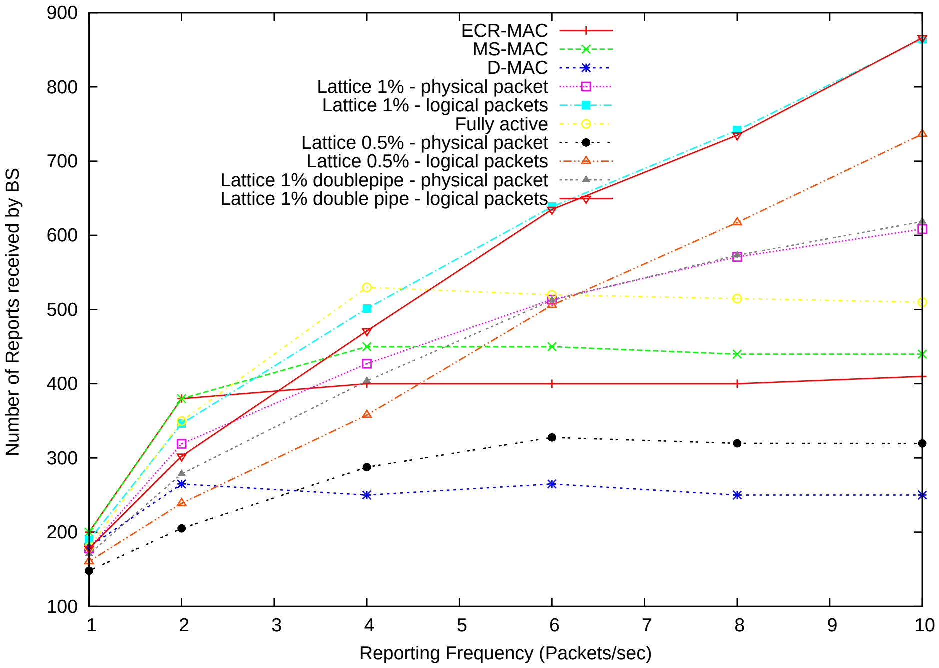

Figure 5.7: Throughput vs. Reporting Frequency

CHAPTER SIX

CONCLUSIONS

Wireless sensor network is an event based multi-hop wireless network which is used for remote monitoring like forest fire monitoring. These network consists of large scale deployment of sensor nodes which has sensing, processing and wireless communication capabilities. These sensor nodes are battery operated which introduces the energy constraint. Hence, the protocol designed for these network needs to be energy-efficient to increase the network longevity. Moreover, when an event happens not one node, by many close-by nodes sense the event and try to send packet to the base station simultaneously to report the event. This simultaneous access to the channel causes spatially correlated contention - a fundamental problem which adversely affects the delay, degrades throughput and wastes energy. Many protocols designed for sensor network does not address this contention. To address spatially-correlated contention and energy-efficiency in wireless sensor networks, we propose a distributed topology control to construct a lattice communication backbone. Lattice backbone addresses contention by reducing the active number of contenders for the channel, naturally providing a controlled structure for straightforward collision avoidance and two forwarders for each backbone node to diffuse traffic. The non-backbone nodes can use this backbone to send packets to the base station. Energy-efficiency is achieved by making non-backbone nodes sleep and having a low duty cycle on backbone nodes. Since the non-backbone that needs to send data to the base station contends only the nearest backbone neighbor, the contention is localized to the event area. Where as in other techniques, the contention exists at every hop to the base station increasing the delay. Also, at every backbone node, the packets from the nearest non-backbone nodes are aggregated before sending the data to the base station. This aggregation is meaningful as it aggregates data from spatially correlated nodes. The aggregation reduces the number of physical packets that needs to be transported into the lattice naturally reducing the traffic improving the energy-efficiency. Compared to other protocols that also have low duty cycles and/or staggering, the lattice backbone achieves shorter end-to-end delay, higher throughput and achieves better energy-efficiency both in single-source and multi-source scenario.

In lattice backbone, it is possible that a node may run out of energy and die. In this scenario, the lattice communication will get affected and hence we need a mechanism to tackle this situation. For future work, we are currently devising a local healing mechanism, that involves only the local nodes to elect a new backbone neighbor to preserve the backbone communication.

In sensor network, the predominant traffic pattern is converge-cast traffic - sensor nodes sending packets to the base station. This causes the *energy hole* problem, where nodes closer to the base station incur heavier workload and deplete energy quicker, impairing the network lifetime. We are currently investigating, similar to the approach [16] of assigning different duty cycles for nodes at different distances, a differential assignment to cope with the energy hole problem and obtain balanced energy consumption across the topology for increased network lifetime without sacrificing network performance. The lifetime of the lattice backbone can be also improved by having multiple such backbones and operating them one at a time alternatively. We are currently exploring an efficient lattice construction mechanism for further lattices, given a base lattice.

BIBLIOGRAPHY

[1] M. Batalin and G. S. Sukhatme, “Coverage, exploration and deployment by a mobile robot and communication network,” in *Telecomm. Systems Journal*, Special Issue on WSN, 2004.

[2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” in *Proc. of MobiCom*, 2001, pp. 85−96

[3] M. Dhanaraj, B. S. Manoj, and C. S. R. Murthy, “A new energy efficient protocol for minimizing multi-hop latency in wireless sensor networks,” in *IEEE PerCom*, 2005, pp. 117−126.

[4] “D-MAC,” http://ceng.usc.edu/ anrg/downloads.html.

[5] P.Floreen, P.Kaski and J.Suomela, “A distributed approximation scheme for sleep scheduling in sensor networks,” in *IEEE SECON*, 2007, pp. 152−161.

[6] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, “Geometric spanners for routing in mobile networks,” in *ACM MobiHoc*, 2001, pp. 45−55.

[7] “IEEE standard 802.11,” in Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[8] K. Jamieson, H. Balakrishnan, and Y. Tay, “Sift: A mac protocol for event-driven wireless sensor networks,” in *LCS-TR-894*, MIT, 2003.

[9] A. Keshavarzian, H. Lee, and L. Venkatraman, “Wakeup scheduling in wireless sensor networks,” in *ACM MobiHoc*, 2006, pp. 322−333.

[10] B. Krishnamachari, “*Networking Wireless Sensors*”, Cambridge University Press 2006.

[11] K. Langendoen and G. Halkes, “*Embedded Systems Handbook*”, CRC Press, 2005, ch. Energy-Efficient Medium Access Control.

[12] N. Li, J. Hou, and L. Sha, “Design and analysis of an mstbased topology control algorithm,” in *IEEE INFOCOM*, 2003, pp. 1702−1712.

[13] S. Liu, K.Fan, and P.Sinha, “CMAC: An Energy Efficient MAC Layer Protocol Using Convergent Packet Forwarding for Wireless Sensor Networks”, in *IEEE SECON*, 2007, pp. 11−20.

[14] G. Lu, B. Krishnamachari, and C. Raghavendra, “An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks,” in *IEEE IPDPS*, 2004.

[15] M. Medidi, R. Slaaen, Y. Zhou, C. Mallery, and S. Medidi, “Scalable localization in wireless sensor networks,” in *IEEE HiPC*, vol. LNCS 4297, 2006, pp. 522−533.

[16] M. Medidi and Y. Zhou, “Extending Lifetime with Differential Duty Cycles in Wireless Sensor Networks ”, in IEEE GlobeCom, November, 2007, pp. 1033−1037.

[17] K. Moaveni-Nejad and X. Li, “Low-interference topology control for wireless ad hoc networks,” in *Ad Hoc & Sensor Wireless Networks: An International Journal*, 2005, pp. 41−64.

[18] “ns-2 network simulator 2”,http://www.isi.edu/nsnam/ns.

[19] J. Polastre, “J. Hill, and D. Culler, Versatile low power media access for wireless sensor networks,” in *ACM SenSys*, 2004, pp. 95−107.

[20] S. Poduri and G. S. Sukhatme,“Constrained coverage for mobile sensor networks,” in *IEEE ICRA*, 2004, pp. 165−171.

[21] R. Ramanathan and R. Roasles-Hain, “Topology control of multihop wireless networks using transmit power adjustment,” in *IEEE INFOCOM*, 2000, pp. 404−413.

[22] I. Rhee, A. Warrier, M. Aia, and J. Min, “Z-mac: a hybrid mac for wireless sensor networks,” in *ACM SenSys*, 2005, pp. 90−101.

[23] V. Rodoplu and T. H. Meng, “Minimum energy mobile wireless networks,” in *IEEE JSAC*, 1999, pp. 1333−1344.

[24] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Optimizing sensor networks in the energy latency-density design space,” in *IEEE Trans. on Mobile Computing*, 2002, pp. 70−80.

[25] T. van Dam and K. Langendoen, “An adaptive energy-efficient mac protocol for wireless sensor networks,” in *ACM SenSys*, 2003, pp. 171−180.

[26] M. Vuran and I. Akyildiz, “Spatial correlation-based collaborative medium access control in sensor networks,” in *IEEE/ACM Trans. on Networking*, vol. 14, 2006, pp. 316−329.

[27] W. Ye, J. Heidemann, and D.Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” in *IEEE/ACM Trans. on Networking*, 2004, pp. 493−506.

[28] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in *Proc. of MobiCom*, 2001, pp. 70−84.

[29] X. Yang and N. Vaidya, “A wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay,” in *IEEE RTAS*, 2004, pp. 19−26.

[30] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient mac protocol for wireless sensor networks,” in *IEEE INFOCOM*, 2002, pp. 1567−1576.

[31] Y. Zhou, M. Medidi, “Energy-efficient Contention-Resilient Medium Access for Wireless Sensor Networks,” in *IEEE ICC*, 2007, pp. 3178−3183.

[32] Y. Zhou, M. Medidi, “Sleep-based Topology Control for Wakeup Scheduling in Wireless Sensor Networks,” in *IEEE SECON* , 2007, pp. 304−313.

[33] M. Zorzi and R. Rao, “Geographic random forwarding (geraf) for ad hoc and sensor networks: energy and latency performance,” in *IEEE Transactions on Mobile Computing*, vol. 2, 2003, pp. 349−365.

[34] K. Sohrabi, J. Gao, V. Ailawadhi, G. Pottie, “Protocols for self-organization of a wireless sensor network,” in *IEEE Personal Communications Magazine*, Vol.7, 2000, pp. 16−27.

[35] L. Clare, G. Pottie, J. Agre, “Self-organizing distributed sensor networks,” in *SPIE - The International Society for Optical Engineering*, 1999, pp. 229−237.

[36] D. Estrin, R. Govindan, “Next century challenges: scalable coordination in sensor networks,” in *MobiCom* 1999, Seattle, WA, 1999, pp. 263−270.

[37] W. Rabiner Heinzelman, A. Chandrakasan, H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *HICSS*, 2000.

[38] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, “Coverage problems in wireless ad hoc sensor networks,” in *Proc. IEEE INFOCOM*, 2001, pp. 1380−1387.

[39] L. Li and J. Halpern, “Minimum energy mobile wireless networks revised,” in *IEEE ICC*, 2001, pp. 278−283.

[40] X. Cheng, B. Narahari, R. Simha, M. Cheng, and D. Liu, “Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics,” in *IEEE Trans. on mobile computing*, vol. 2, 2003, pp. 248−255.

[41] S. Wang, D. Wei, and S. Kuo, “Spt-based power-efficient topology control for wireless ad hoc networks,” in *IEEE MILCOM*, 2004, pp. 1483−1490.

[42] Y.Wang and X. Li, “Distributed spanners with bounded degree for wireless ad hoc networks,” in *IEEE IPDPS*, 2002, pp. 194−201.

[43] Y.Wang and X. Li, “Efficient construction of bounded degree and planar spanner for wireless networks,” in *ACM DIALM-POMC Joint Workshop on Foundations of Mobile Computing*, 2003, pp. 374−384.

[44] W. Baek, D. Wei, and C. Kuo, “Power-aware topology control for wireless ad-hoc networks,” in *IEEE WCNC*, 2006, pp. 406−412.

[45] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, “Geometric spanners for routing in mobile networks,” in *ACM MobiHoc*, 2001, pp. 45−55.

[46] S. Ramanathan, “A unified framework and algorithms for (T/F/C)DMA channel assignment in wireless networks,” In *IEEE INFOCOM*, 1997, pp. 900−907.

[47] I. Rhee, A. Warrier, and L. Xu, “Randomized dining philosophers to TDMA scheduling in wireless sensor networks”, *Technical report, Computer Science Department, North Carolina State University*, 2004.