CONGESTION AVOIDANCE ENERGY EFFICIENT MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS

by

Alexander Sundling

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Computer Science

Boise State University

December 2012

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Alexander Sundling

Thesis Title: Congestion Avoidance Energy Efficient MAC Protocol for Wireless Sensor Networks

Date of Final Oral Examination: 30 July 2012

The following individuals read and discussed the thesis submitted by student Alexander Sundling, and they evaluated his presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Sirisha Medidi, Ph.D.	Chair, Supervisory Committee
Murali Medidi, Ph.D.	Member, Supervisory Committee
Amit Jain, Ph.D.	Member, Supervisory Committee

The final reading approval of the thesis was granted by Sirisha Medidi, Ph.D., Chair of the Supervisory Committee. The thesis was approved for the Graduate College by John R. Pelton, Ph.D., Dean of the Graduate College.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Sirisha Medidi not only for her guidance and support in helping me to complete my thesis but for everything she has taught me during the whole duration of the Master′s program, her teachings will stick with me forever.

I would like to also thank Dr. Murali Medidi for his help throughout the years, his valuable inputs, and for serving as a committee member to evaluate my thesis.

I gratefully thank Dr. Amit Jain for his informative lectures, and for serving as a committee member to evaluate my thesis.

I wish to thank all members of SWAN Lab for their continuous support and encouragement.

I especially want to thank my family and friends for their love and support they have shown me during this thesis process.

ABSTRACT

Wireless Sensor Network (WSNs) are generally energy-constrained and resource-constrained. When multiple simultaneous events occur in densely deployed WSNs, nodes near the base station can become congested, decreasing the network performance. Additionally, multiple nodes may sense an event leading to spatially-correlated contention, further increasing congestion. In order to mitigate the effects of congestion near the base station, an energy-efficient Media Access Control (MAC) protocol that can handle multiple simultaneous events and spatially-correlated contention is needed. Energy efficiency is important and can be achieved using duty cycles but they could degrade the network performance in terms of latency. Existing protocols either provide support for congestion near the base station or for managing spatially-correlated contention. To provide energy-efficiency while maintaining the networks performance under higher traffic load, we propose an energy-efficient congestion-aware MAC protocol. This protocol provides support for congestion near the base station and spatially-correlated contention by employing a traffic shaping approach to manage the arrival times of packets to the layers close to the base station. We implemented our protocol using the ns-2 simulator for evaluating its performance. Results show that our protocol has an improvement in the number of packets received at the base station while consuming less energy.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WSN – Wireless Sensor Network

MAC – Medium Access Control

TDMA – Time division multiple access

RTS – Ready to Send

CTS – Clear to Send

CHAPTER 1

INTRODUCTION

With advances made in microelectronic fabrication and wireless communication technologies, low-cost, low-power, and multi-functional sensor nodes have emerged, which enables the construction of large-scale wireless sensor networks (WSN). A WSN is a network that consists of hundreds to thousands of sensor nodes. These sensor nodes are low-power devices with a battery as the main power source and have one or more sensors equipped. They are also equipped with wireless interfaces that allow them to communicate with one another to form a network so that once data is collected it can be transferred back to the base station [10]. The sensor nodes can be deployed indoors and outdoors and are typically used for the purpose of monitoring an area that is inaccessible or inhospitable [4]. Sensor nodes can be specifically tailored for many different applications such as monitoring the environment (detecting fires, pollution, radiation, etc.), equipment monitoring [1, 2], smart home/smart space [3], battlefield surveillance, detecting intruders, etc.

In event-driven WSNs, data is generated when an important event is triggered or detected and a burst of traffic will be sent to the base station. Data will continue to be sent to the base station as long as the event is still taking place. For example, in a fire detection system that is located in a forest, once a fire is detected the detecting nodes will generate the data and send it to the sink; these nodes will continue to do this until the fire is extinguished or burns out. If a fire is first detected in an small area that is only detected by a handful of nodes, this will not cause too many problems with delivering the data to the base station, but fires rarely stay in a small area and can spread very quickly. So it could easily go from only a few nodes reporting data to having several hundred nodes reporting data on the fire after it has spread and they will keep reporting on the fire until its gone. This will generate a lot of traffic and will cause two big problems. Firstly it will lead to spatially-correlated contention, which is when many nodes in the same area are trying to transmit data but collisions keep occurring between multiple nodes trying to send data simultaneously. Secondly, since all of the packets are heading to the same destination, there will be a funneling affect causing the nodes closer to the base station to become far more congested. Due to the congestion near the base station, there will be far more contention for the channel, causing an increase in delay and decrease in throughput, which will keep the area more congested for a longer period of time.

Energy efficiency is an important aspect of a WSN because it is often not feasible to replace or recharge batteries for sensor nodes [32]. Due to this constraint, it is extremely critical that a protocol be energy-efficient in order to prolong the lifetime of a network and allow more data to be collected. An energy efficient protocol should be a staple in all WSN protocols. In order to mitigate the effects of congestion near the base station, an energy-efficient Media Access Control (MAC) protocol that can handle multiple simultaneous events and spatially-correlated contention is needed.

1.1 Organization of Thesis

The rest of this thesis is organized as follows. Chapter 2 reviews the related work. Chapter 3 discusses our motivation and our approach for the MAC protocol that we have created. Chapter 4 presents the performance evaluation of our MAC protocol. Finally, we conclude in Chapter 5.

CHAPTER 2

RELATED WORK

2.1 General Overview

In WSNs, many protocols have been proposed to deal with different aspects found in a WSN. In order for a large deployment of sensor nodes to be considered cost-effective, nodes have to be resourced-constrained in terms of energy capacity, radio transmissions, processing capabilities, and memory storage. In most cases, it is important for a wireless sensor network to be energy efficient. Also, in many scenarios, a WSN will be vulnerable to contention and congestion due to the predominant traffic pattern known as *convergecast* found in WSN. Many protocols have been proposed to provide a network to be energy efficient, and protocols have been proposed to help reduce the contention and congestion found in a WSN. These protocols are broadly classified as

1. Protocols that provide support for contention and congestion
2. Protocols that provide energy efficiency

2.2 Contention and Congestion

A problem that occurs naturally in WSNs due to their design is contention and congestion. Contention will occur anytime multiple nodes try to gain channel access to the same forwarder. This happens very frequently, especially when an event is sensed by multiple nodes, which leads to a type of contention that is specifically found in WSN called spatially-correlated contention. Congestion also happens quite frequently in WSNs because a WSN has a predominant many-to-one traffic pattern known as *convergecast* [30]. All of the events that are sensed in a network have to be sent to one or more base stations. This traffic pattern causes the nodes closer to the base station to have to carry a heavier traffic load, which in turn makes the areas surrounding the base stations more congested. Due to the possibility of contention and congestion occurring in all WSNs, many methods have been proposed to detect and control contention and congestion or to try and avoid it [12, 15, 22, 23, 30, 31, 40].

ECR-MAC [31] employs the use of the Dynamic Forwarder Selection (DFS) mechanism, which is a form of *anycasting*. The DFS mechanism sends out a message to all potential forwarders and the first one to respond becomes the forwarder. This gives the node more flexibility in finding a suitable forwarder. This mechanism is a congestion avoidance mechanism because it helps avoid contention by giving nodes more options of where to send their data. By avoiding contention and helping to send as quickly as possible, it reduces the amount of backup that can occur in an area. ECR-MAC treats all parts of the network equally even though the nodes close to the base station will face far more contention then some of the outer nodes. Other papers have been published that deal with *anycasting*. Particularly in [29], it shows the tradeoffs of how long a node should wait for replies from potential forwarders before choosing a suitable forwarder. Another protocol that uses a congestion avoidance mechanism is Pump Slow, Fetch Quickly [23] (PSFQ). PSFQ is a scalable and reliable transport protocol that distributes data from a source node by slowing down the data speed so that any nodes that experience data loss are able to fetch the missing pieces from the intermediate nodes that have the pieces of data. Its objective is to minimize the number of retransmissions for loss detection. PSFQ consists of three operations: the pump operations, relay-initiated error recovery, and report operation. The pump operation injects the data into the network at a scheduled rate to help reduce the amount of congestion. A node will then cache the data that it receives so that is can forward it on to nodes that are trying to fetch any data that they didn′t receive. Finally, it has a reporting function that helps for the source to know what receivers received which information. When a node receives a piece of data it will check its cache to see if it has already received it, if it has it will discard the duplicate piece of data, otherwise it will store the data fragment. Once a gap is seen by the node, it will go into the FETCH mode trying to retrieve the missing data fragment. When a node requests many pieces of data together, the responding node tries to batch together the reply in order to help reduce the amount of traffic and congestion in the network. A disadvantage with PSFQ is that its pump operation can be very slow and leads to a large delay.

Event-to-sink reliable transport [15] (ESRT) was proposed to achieve event-to-sink reliability. ESRT tries to achieve reliable event detection using energy-efficient techniques and congestion control. ESRT tries to achieve reliability by staying in a comfortable congestion zone. ESRT defines five states of congestion: (No Congestion, Low Reliability); (No Congestion, High Reliability); (Congestion, High Reliability); (Congestion, Low Reliability); and (Optimal Operating Region). ESRT tries to stay in the Optimal Operating Region state. If it falls out of this state, it will adjust the reporting rate of the sensor nodes with the hope that it will return to the Optimal Operating Region state. In order to change the reporting frequency it is broadcasted to all the nodes in the network. In order to detect the current state of the network, the sink must be able to detect congestion in the network. This is done by the sensor nodes that detect congestion using the buffer sizes and set the congestion notification bit. When one of these packets with the notification bit set is received at the base station, it knows that the network is congested and will then update the reporting frequency accordingly. CODA [12] uses a combination of the present and past channel loading conditions and the current buffer occupancy to infer accurate detection of congestion at each receiver with low cost. As long as a node detects congestion, it sends “backpressure″ messages to upstream nodes for controlling reporting rate hop-by-hop. It is also capable of asserting congestion control over multiple sources from a single sink in the event of persistent congestion.

Lightweight Medium Access Control (LMAC) [46] is a TDMA-based MAC protocol. LMAC reduces contention in the network by organizing time into slots, and assigning each slot to a node. When a node wants to send a packet, it needs to wait until its time-slot and then it will have to send a message that contains a data slot in which it can send data. By reducing the amount of contention in the network, it can help reduce the amount of congestion that will build up. The drawback to the LMAC protocol is that all of the nodes are always listening to all of the messages sent, this leads to energy waste. Rhee et al. proposed Z-MAC [47] that makes use of TDMA and carrier sense multiple access (CSMA). Z-MAC monitors the amount of contention in the network and when it is low will use CSMA, but when it is considered a high-contention network it will switch to TDMA. In Z-MAC, a time slot assignment is performed at the time of deployment. This causes a higher overhead at the beginning of the protocol but overhead is amortized over the lifetime of a network. In order to do the scheduling of the slots, Z-MAC uses an efficient scalable channel scheduling algorithm called DRAND. Each slot may have up to two nodes assigned to it. Unlike TDMA, any node can transmit during any slot, but the owner of the slot has priority for transmitting data during that slot. The setting up phase is expensive, but the results have shown that this protocol has done well in a network with medium to high contention. Z-MAC fails to address the spatially-correlated contention in the outer layers of the network. Another TDMA-based protocol is Funneling-MAC [40], which attempts to just deal with the congestion found near the base station by using TDMA around the base station. Funneling-MAC sends out a beacon that is used to select a subset of nodes close to the base station, which are known as f-nodes. After the f-nodes have been selected, the base station will create a TDMA schedule for them to follow. Each node will now have a time slot when it can send data to the base station. All of the nodes that aren′t a f-node will use CSMA to send data to its forwarders. Although this helps reduce congestion near the base station, it does not deal with the spatially-correlated contention in outer layers that can lead to congestion in other parts of the network.

2.3 Energy Efficiency

In wireless sensor networks because energy efficiency is deemed so important many different strategies have been created in order to try and prolong the lifetime of a network. There have been strategies used to increase the lifetime of the network and they can be broadly categorized as:

1. Duty Cycles
2. Topology Control
3. Clustering and Grouping

2.3.1 Duty Cycles

In previous works [32, 39, 41, 42, 43] it has been shown that idle listening is a major source of energy waste. Nodes spend most of their lives in idle listening because WSNs typically have sporadic events. So in order to try and avoid wasting this energy, they turn off the nodes when they are not needed. Many protocols have nodes turn on and off for a certain amount of time; this is known as a duty cycle and it is used to help nodes stop wasting energy. Based on this, a duty cycle is defined as the following:

$$dutycycle = \frac{T_{Active}}{T_{Active} + T_{Sleep}} \tag{2.1}$$

Although having a low duty cycle Medium Access Control (MAC) protocol is energy efficient, it still has three shortcomings. First, it increases the packet delivery latency because an intermediate node may have to wait until the receiver wakes up before it can forward a packet; this is called sleep latency in S-MAC [41]. The sleep latency increases proportionally with respect to the number of hops, with the constant of proportionality being the duration of a single cycle (active period plus sleep period). Secondly, a fixed duty cycle does not adapt to the traffic variation in sensor network. A fixed duty cycle for the highest traffic load results in significant energy waste when traffic is low while a duty cycle for low traffic load results in low message delivery and long queuing delay. Thirdly, a fixed synchronous duty cycle may increase the possibility of collisions. If neighboring nodes turn to the active state at the same time, they all may contend for the channel, making a collision very likely [32]. There have been many proposed schemes that try and tackle one or all of these shortcomings. In S-MAC [41], each node will use a fixed active/sleep duty cycle to reduce the amount of energy that is consumed from idle listening. Each node is able to choose its own sleep schedule but in order to save energy S-MAC makes it so that all nodes will broadcast their sleep schedules to their neighboring nodes so they can form virtual clusters that know each other′s sleep schedules. By doing this when a node needs to send a message to a node in its virtual cluster, it will not have to wait for it to wake up to send it because the nodes will stay synchronized in the cluster. S-MAC makes use of ready-to-send (RTS) and clear-to-send (CTS) control packets so they are able to coordinate two nodes that attempt to gain channel access at the same time. S-MAC causes unnecessary energy waste for handling low traffic and it will also have a low throughput when sporadic events happen. T-MAC [42] was proposed as an improvement to S-MAC′s energy efficiency. T-MAC uses adaptable duty cycles that change their listening period based on hearing no activity for a certain period of time. A node will stay awake if it overhears a transmission from one of its neighbors because it could possibly be used as a forwarder to the base station. This helps in reducing the sleep latency caused by duty cycles, but it only helps for a couple of hops before a node has to wait for its forwarder to wake up. D-MAC [32] was designed for the *convergecast* traffic pattern. The *convergecast* traffic pattern is when many nodes are sending to one node and this is the predominant traffic pattern found in wireless sensor networks. D-MAC uses a staggered wakeup scheduling mechanism that will schedule nodes based on their hop level to wake up in sequential reverse hop level order. This will help reduce the setup latency seen in S-MAC and T-MAC while still maintaining the same energy efficiency. D-MAC uses its own contention handling techniques such as data prediction and a more-to-send (MTS) packet to help reduce the effect of contention on the network. D-MAC creates a data gathering tree so that it can report events to the base station. Although widely used, the data gathering tree is not very flexible when node failures occur and nodes are likely to have collisions with one another when trying to gain channel access to a common forwarder. In D-MAC, collisions can occur even more frequently using the data gathering tree because nodes in the same data gathering tree located at the same level will wake up at the same time and could try and gain channel access for the same forwarder. This will have a negative impact on the network because the throughput will decrease and more energy will be consumed from extra retransmissions.

Another method that was proposed that made use of duty cycles is ECR-MAC [31]. Zhou and Medidi designed a protocol to improve both energy efficiency and delay. ECR-MAC employs a mechanism called Dynamic Forwarder Selection (DFS). DFS allows more flexibility for forwarding packets, which leads to an improvement in energy efficiency and delay. DFS allows a node to have multiple forwarders so that multiple paths can be taken to the base station. A node will no longer have to wait for a particular node, instead it can use the first available potential forwarder. This works particularly well in a dense network where a node is likely to have a sufficient amount of potential forwarders. ECR-MAC makes use of duty cycles to make the protocol more energy efficient. Each node has the same duty cycle but will wake up at random times, which requires no synchronization. When a node wakes up and it has a packet to send, it will wait a period of time to listen in on any ongoing communication; if nothing is heard, then the node will start the process to send its packet. First the node will send a WAKEUP message periodically until one of its potential forwarders is awake and replies back will a REPLY message. When the first reply message is received from the potential forwarder, the sender will continue the process by sending data to the potential forwarder. On a successful reception of the data at the potential forwarders end, the potential forwarder will respond with an ACK message. The process will continue in the same fashion all the way to the base station.

All of the previous methods that were talked about that use duty cycles were for nodes with one radio, but a handful of protocols have been proposed that make use of nodes with dual radios [9, 11, 39]. Sparse Topology and Energy Management (STEM) [11] was one of the first CSMA-based MAC protocols for WSN that made use of duty cycles. STEM is used for sensor nodes that have two radios on different channels. One channel is used to wake up neighboring nodes and the other is used to send data to its forwarder. The wake-up radio uses duty cycles while the data radio is always sleeping until woken up. When a node has a packet that it needs to forward, it will send a beacon packet to its forwarder. Once the forwarder receives the packet, it will turn on its data radio to receive the data. Since this process has to happen at each step on the way to the base station, the setup latency will increase the overall latency. This leads to a trade-off between energy-efficiency and delay. When the network needs to be more energy efficient, it will have a higher delay and vice versa. If the network needs to have a smaller delay, then it won′t be as energy efficient. PTW [39] was proposed to deal with the trade-off between energy efficiency and end-to-end delay. PTW uses a pipelining transmission technique to help shorten the amount of time it takes to wake up the nodes for data forwarding while still maintaining energy efficiency. Just like in STEM, PTW makes use of a dual radio setup with one radio being used to coordinate duty cycles and the other being used to transmit and receive data. With two radios, they saw that they could coordinate a duty cycle for its forwarder while receiving data. So the idea behind PTW is that one of the radios will wake up the nodes next forwarder while it is in the process of receiving data. This helps cut down on the setup delay, but its effectiveness depends on many factors, such as the data packet size and the data radio bandwidth. In PTW, in order for a node to wake up its forwarder, it must transmit a wake up tone. When a node hears this wake up tone, it cannot distinguish from which node it is being transmitted from so all it can do is wake up. Due to only one node being needed to act as forwarder, all of the nodes that woke up that will not act as forwarder have just consumed more energy. This can lead to a quick increase in energy consumption as more events happen in a dense WSN.

The Latency minimized Energy Efficient MAC protocol (LEEM) [9] is another dual radio protocol that was proposed to reduce the setup latency without sacrificing energy efficiency. LEEM requires that a system-wide synchronization takes place so that the nodes along a route can be scheduled to wake up sequentially. When a sensor node senses an event and is in sleep mode, it will wait for the next hop node to become active and then it will send a request packet asking the node to turn its data channel radio on. The next hop node will reply with an ACK and it will continue the process by sending a request packet to its next hop node. This will continue to be passed on until it reaches the base station. Now that the path is ready to become active to receive the data, the data transmission starts. The data will be propagated down the path until it reaches the base station. This mechanism helps eliminate the setup times of the intermediate nodes between the node that senses the event and the base station. This process comes at the expense of setting up the entire network schedule.

Geographic Random Forwarding (GeRaF) [45] uses geographical location information to create a wake-up schedule. GeRaF makes an assumption that all of the nodes have the means to determine their own location, and that they know the locations of any of the nodes to which messages need to be sent. When a node needs to send information to another node, it will broadcast a message and a node will volunteer to be its forwarder based on the location′s final destination node for the message. Due to this mechanism of forwarding messages, nodes can go to sleep and the message will attempt to take the best possible route to the destination node with the nodes that are still awake. To handle collisions, GeRaF must be used on a dual radio node. One radio is used for transmitting data and the second radio is used to transmit a busy tone to let other nodes know that a message is being sent.

STEM, PTW, LEEM, and GeRaF all make use of dual radios, which are more expensive than a single radio node. Though this money might not be very much when comparing one dual radio node to a single radio node, it adds up quickly when a network consists of hundreds to thousands of nodes. It is important to keep the wireless sensor network cost-effective.

2.3.2 Topology Control

Due to the large number of nodes that are deployed in a WSN, there are a redundant number of nodes at any point in an area of interest. Due to this quality in a WSN, another popular way of trying to reduce energy is by controlling the topology of the network [5, 7, 8, 25, 37]. In [37], they choose a subset of nodes that constructs a square or equilateral-triangular mesh that will fully cover the region and allow energy balancing. SPAN [7] was one of the first sleep-based topology control techniques for wireless ad-hoc networks. SPAN tries to reduce the amount of energy that is consumed without affecting the connectivity of the network. SPAN works when a network is dense and makes use of the fact that not all of the nodes are needed to be active in a dense network to keep connectivity. In SPAN, nodes make local decisions on whether to join the forwarding backbone as a coordinator or to go to sleep. The coordinators are awake all the time and are used to forward packets to the base station. The backbone is created in such a way that it ensures that all of the nodes in the network are within transmission range of at least one of the nodes on the backbone. In order to stop the coordinators from dying first, the coordinators keep changing so that the load is spread out across the network. When creating the coordinator backbone, SPAN tries to minimize the number of coordinators to help reduce the latency. All the decisions that a node will make when deciding to become a coordinator is done through local information gathered from local broadcast messages. In order to help reduce the contention in the setup process of becoming coordinators, a node will wait a random amount of time before sending its message. In [5, 8, 25], they propose different techniques that can be used on networks that consist of variable sensing nodes. They choose a small subset of nodes and through various techniques decide to which sensing radius each node should be set in order to completely cover the region of interest. Although these are good techniques, not every application deals with sensors that can have their sensing radius changed, many sensing radii will be a fixed length.

2.3.3 Cluster/Grouping Techniques

Another strategy that has been used often for creating an energy-efficient protocol is by grouping/clustering nodes together. This can be seen in [35, 36, 38]. In LEACH [38], clusters are created among nodes that are close together and then a cluster head is chosen for that cluster. The cluster head now acts as a local base station for that cluster, so anything received by the cluster head from the rest of the members in the cluster is then forwarded directly to the base station. Since this would drain the energy of the cluster head quite quickly, they implemented a random rotation of the cluster head so that energy spent is dispersed more evenly. PEGASIS [35], much like LEACH, makes use of cluster heads, but instead of having the cluster heads talk directly with the base station, they will form a path out of cluster heads to the base station. The chain that is formed of cluster heads is created in two steps. The first step is a greedy approach and is used to create the chain starting at the furthest node from the base station. After the nodes have been selected to form the chain, the next step is to choose a leader for the data gathering phase. During each round of this phase, a new leader will be selected randomly. When a node dies that is part of the chain, the chain is reconstructed in order to bypass the dead node and keep the connectivity intact. The leader of the data gathering process is in charge of passing the token to start the data gathering phase. The amount of energy that is consumed for passing the token is very small because of the small size of the token. In order to save more energy, the transmitting distance is reduced in the sensor nodes. Experimental results have shown that PEGASIS improves upon LEACH by a factor of two or more.

Geographical Adaptive Fidelity (GAF) [36] was proposed to help reduce energy consumption in ad-hoc wireless networks. GAF makes use of the nodes location information and determines which nodes to keep on while still maintaining connectivity throughout the network. GAF chooses nodes to keep on by looking at their location and whether or not another node already has the area covered. Each node will assign itself to a virtual grid that contains other nodes. Each node can be in one of three states: active, discovery, or inactive. Each node starts in the discovery state and from there makes a decision whether it should become active or inactive. Only one node at a time per virtual grid will be active, after a period of time it will go into discovery state and allow other nodes a chance to become the active node in the virtual cluster, this will allow all nodes a chance to take the responsibility of being the active node and the energy consumed will be spread across more nodes.

As one can see, a great deal of time has been put into finding energy-efficient techniques due to its importance in a WSN. All protocols have ways of saving a great amount of energy but with conceding another important metric to be worse off. The trick to creating a great energy-efficient technique is to create a protocol that finds a balance between saving energy while not completely sacrificing one of the other important design features of a WSN.

CHAPTER 3

CONGESTION MANAGEMENT MAC PROTOCOL

3.1 Motivation and Design Considerations

Due to the convergecast traffic pattern found in WSNs, two main issues occur. First, congestion becomes much greater the closer a node is to the base station. This makes it far more difficult for a node to gain channel access and send data to the base station. Secondly, when an event occurs, multiple nodes may sense it due to the density of a WSN. When the nodes attempt to send packets about the sensed event, it could lead to spatially-correlated contention and may have harmful effects on the network. When multiple events are simultaneously detected on all sides of the base station, such as a network monitoring forest fires having fire all around the base station, the detecting nodes forward their packets to the base station where they could all potentially arrive close to the same time. Having all of the packets arrive at the base station so close together may lead to collisions and congestion near the base station. This in turn would have a negative impact on the latency, throughput, and packet delivery ratio. In an application such as the fire detection example it is important that we receive as much information as we can regarding the fire during the peak times of congestion because every bit of information could make a huge difference when devising a plan to put out the fire. There is a need for a protocol that can help manage congestion in order to reduce it near the base station to improve the network performance while still being energy efficient. There has been no approach proposed that reduces the congestion near the base station and handles spatially-correlated contention in the network. ECR-MAC was proposed to deal with spatially-correlated contention by using anycasting to diffuse the traffic flows but does not specifically address the congestion near the base station. In the following section, we provide design considerations for our protocol.

3.1.1 Design Challenges

Congestion Near the Base Station

Due to the funneling effect caused by the convergecast traffic pattern, nodes that are closer to the base station are more likely to become congested. Once nodes near the base station become congested, the outer layers may soon follow. For this reason, it is important to design a protocol that helps to reduce the amount of congestion close to the base station to improve the overall health and efficiency of the network as a whole.

Spatially-Correlated Contention

In a WSN, sensor nodes are typically deployed in high density for obtaining reliability in event sensing. When an event occurs, there may be a large number of nodes that may detect it. The normal response of a sensor node that has just detected an event is to report it immediately. Having all of these nodes report simultaneously causes contention in the channel. This contention could lead to back-off causing packets to pile up in a node and it would also lead to wasted energy. This is a fundamental problem found in all WSNs and it should be addressed by the MAC protocol. Not taking care of this issue would have a negative effect on the network.

Energy-Efficiency

As WSNs are energy constrained, it is very important to reduce the energy consumed by the nodes. Sensor nodes in event-driven WSNs are idle most of the time due to events happening infrequently. Sensor nodes therefore spend a small portion of their lifetime reporting events to the base station and the rest of their lifetime is spent in an idle listening state. It is well known that idle listening is a major source of energy waste. Due to this energy constraint, it is important that energy efficiency be considered when designing a protocol.

Scalability

Due to sensor networks having different levels of congestion, a protocol should be designed to handle varying loads of data.

3.1.2 Metrics

Considering the design challenges, a congestion management energy efficient protocol is proposed in this thesis. In order to measure the performance of the protocol, the following standard metrics were chosen.

Number of Packets Received at the Base Station

It is important to calculate the amount of packets that are received at the base station because it shows a direct correlation to the effectiveness of the protocol.

End-to-End Delay

We will measure our protocol on how fast the first 10% of the packets reach the base station because the first packets are always the most important to let the base station know that an event is happening.

Energy Consumed

To identify the energy efficiency of the proposed protocol, the total energy consumed in the network will be calculated. The lower the energy consumption value, the better the energy efficiency of the protocol.

3.1.3 Assumptions

We make the following assumptions about the WSN, which motivate us to design Congestion Avoidance MAC (CA-MAC) and make it suit a typical wireless sensor network application. We assume that the WSN is typically over-provisioned, i.e., deploying a large number of nodes with a high density. We assume that there is only one base station that collects data from sensor nodes. We also make the assumption that each node has the capability of determining its own position.

3.1.4 Basic Approach

This thesis provides an approach that could improve the performance of a network by helping to reduce the amount of contention and congestion near the base station by coordinating traffic flows and by giving nodes alternate data paths to the base station. Using this two-part approach increases the amount of packets that are received at the base station and reduces the end-to-end delay while consuming a reasonable amount of energy. We have designed a protocol that utilizes anycasting and scheduling that could reduce the amount of congestion near the base station. Nodes are split up into sectors and layers, and forwarders are selected before using our protocol.

3.1.5 Initial Setup

Before the nodes are ready to collect data and send to the base station, they must perform initial setup in order to be able to be scheduled properly:

- Determine the location of the base station
- Determine its layer number and sector number
- Determine the potential forwarders

After all the nodes have been deployed, the base station will start broadcasting a BASESTATION_LOCATION message that contains the coordinates of the base station. Each node that hears this broadcast stores the location of the base station and further propagates the message and keeps track of how many times it has broadcasted the BASESTATION_LOCATION message. To help reduce the amount of collisions between nodes, a node waits a random amount of time before broadcasting the message. This continues to go on until every node that heard the message with the location of the base station has broadcasted the message three times. Most of the time, every node in the field will hear the message and will obtain the location of the base station. If a node did not hear the message, then it will sit idle until it does.

Layering and Sectoring Nodes

Each node, after receiving the location of the base station, is now able to determine its layer number and sector number. Each node has knowledge of its location information. Using this information along with the global position of the base station, a node is able

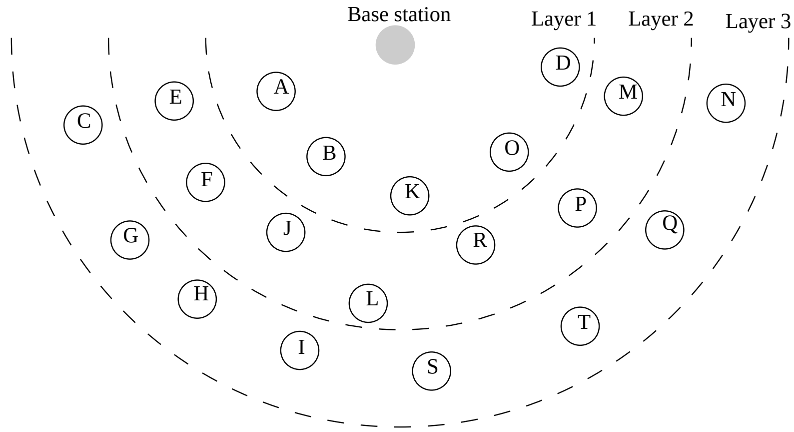


Figure 3.1: Layering

to determine the distance between itself and the base station. We use the following equation to determine the distance between the two nodes:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.1}$$

where (x_1, x_2) are the coordinates to the base station and (y_1, y_2) are the coordinates to the node that is trying to determine in which layer it is located. Once the node knows how far away it is from the base station, it now simply determines its layer number (Figure 3.1) by using this equation:

$$layer\ number = \lceil \frac{d}{layer\ length} \rceil \tag{3.2}$$

After multiple tests, we came to the conclusion that the best layer length to use is just under 50% of the transmission radius.

After determining the layer number of the node, it will go on to determine its sector number.

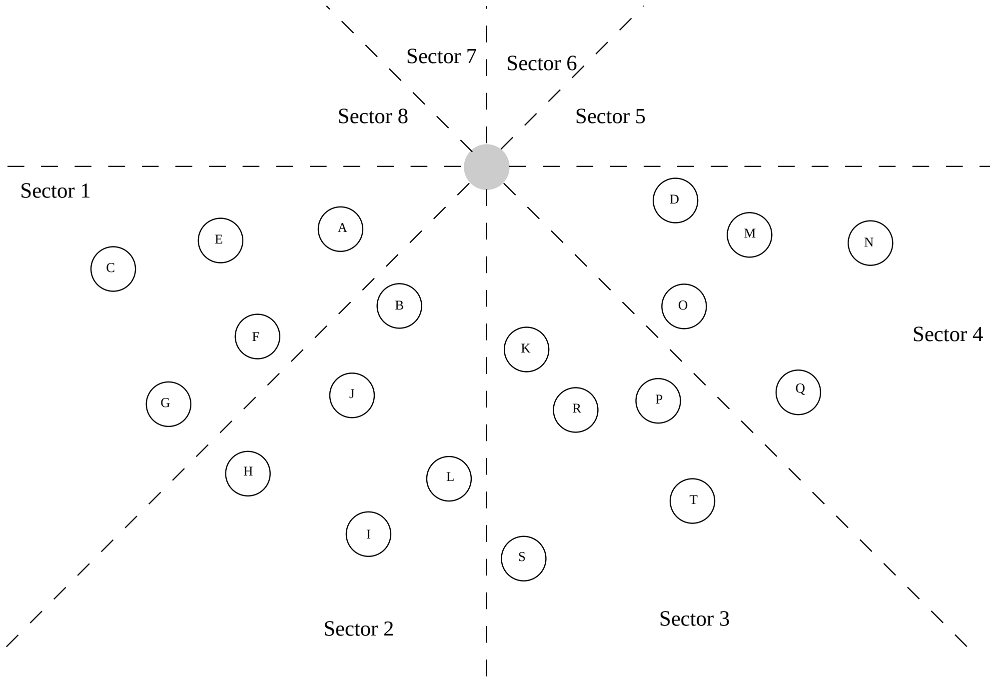


Figure 3.2: Sectoring

A sector is a slice of the network; an example of this can be seen in Figure 3.2 where a network is split up into 8 sectors. In order to determine which sector a node is located in, it must determine what the angle is between a line drawn from the node to the base station and an imaginary horizontal line that starts at the base station. This can be accomplished by using the following equation:

$$\theta = cos^{-1}(\frac{A * B}{||A||||B||}) \tag{3.3}$$

A and B are vectors. A is the vector between the node and the base station and B is the imaginary vector. $A * B$ is the dot-product, $||A||$ and $||B||$ are magnitudes. Using that angle, we plug it into one of the following equations:

$$\text{If } \theta < 180 : Sector\ Number = \lceil \frac{\theta}{45} \rceil \tag{3.4}$$

$$\text{If } \theta >= 180 : Sector\ Number = \lceil \frac{\theta - 180}{45} \rceil + 4 \tag{3.5}$$

We use 45 degrees because we′ve determined through tests that 8 sectors is the most efficient number of sectors to have and $360/8 = 45$.

3.1.6 Forwarder Selection

After each node has determined its layer and sector number, it will broadcast a NODE_INFORMATION message that contains the address of the node and its layer number. This message will be broadcasted three times so that its neighbors have a better chance of receiving it. When a node receives a NODE_INFORMATION message, it will first check what layer number the transmitting node is in. If the node is located in a layer that is further away than the receiving node′s layer, then it will store the address of the node in a list of nodes for which it can act as forwarder.

3.1.7 Congestion Reduction

Congestion is a problem that will affect any wireless sensor network and it could have negative effects that will degrade the network performance. Although congestion is a problem wherever it is happening in the network, having congestion near the base station is the most likely place due to the convergecast traffic pattern that is found in WSNs. Having congestion near the base station is also the worst place to have it because it may have the biggest effect on the network performance. To help reduce the amount of congestion near the base station, we wanted to design a protocol that manages the traffic flow at the first two layers. In order to achieve this, we set up a

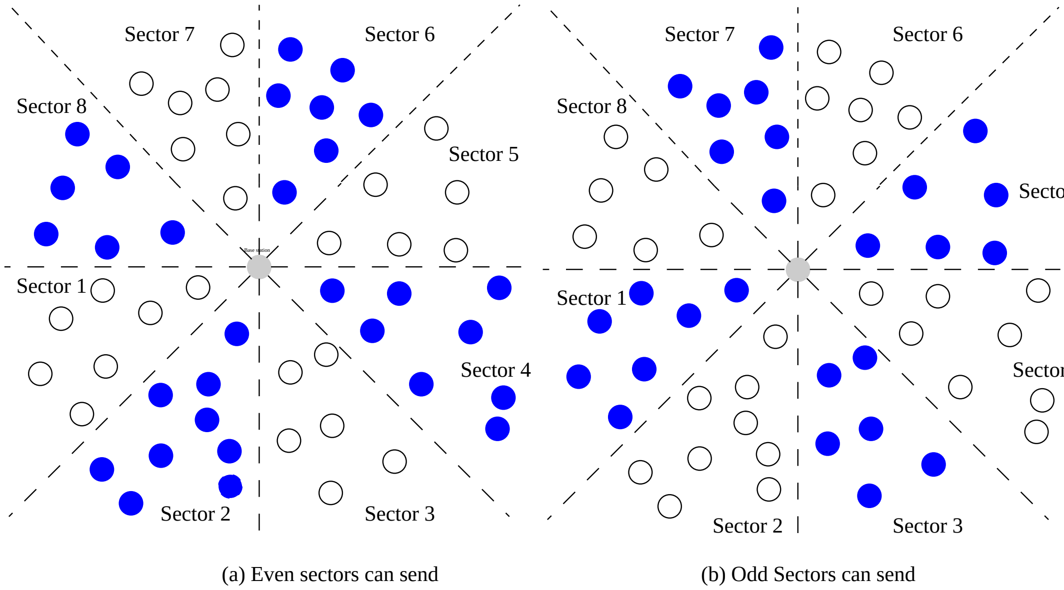


Figure 3.3: Sector sending schedule

schedule of sending times for each node based on their sector number. All the nodes in the even sectors will have a period of time that they can send. when that period is up, all of the nodes in the odd sectors will be able to send data to their forwarder. An example of this can be seen in Figure 3.3. This process will keep alternating between the sets of sectors can send. What this process achieves is that it will limit the amount of nodes that can send data, which may in turn help reduce the contention and congestion that would follow. By reducing the amount of contention, it could allow the base station to receive more packets. To provide even more flexibility at the base station, the nodes in the first two layers will be able to send whenever they have a packet that they need to send; the network will be split up like it is shown in Figure 3.4. This helps reduce the amount of packets that build up in nodes by allowing nodes to send their packets as soon as possible.

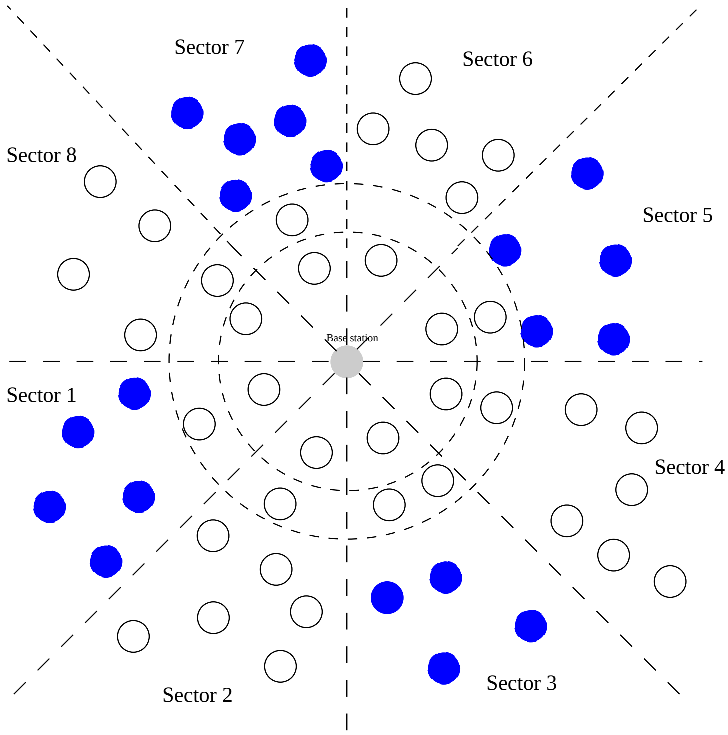


Figure 3.4: Sectors are not used in the first two layers

3.1.8 Spatially-Correlated Contention

Gaining channel access when a node only has one forwarder can lead to a bottle neck and create more congestion. To combat this effect, we employ anycasting when trying to gain channel access so that a node can potentially have more possible routes to take to the base station which helps diffuse the traffic. To gain channel access, we use the same four way handshake that is used in the 802.11 MAC with a few differences. Instead of unicasting a RTS message, we broadcast a RTS_LITE message. When a node receives a RTS_LITE, it will determine if it is a potential forwarder for the sender of the message by consulting its list of nodes for which it can act as a forwarder (that was created at setup). If the node is a potential forwarder, then it will reply with a CTS message. Having all potential forwarders sending at the same time would

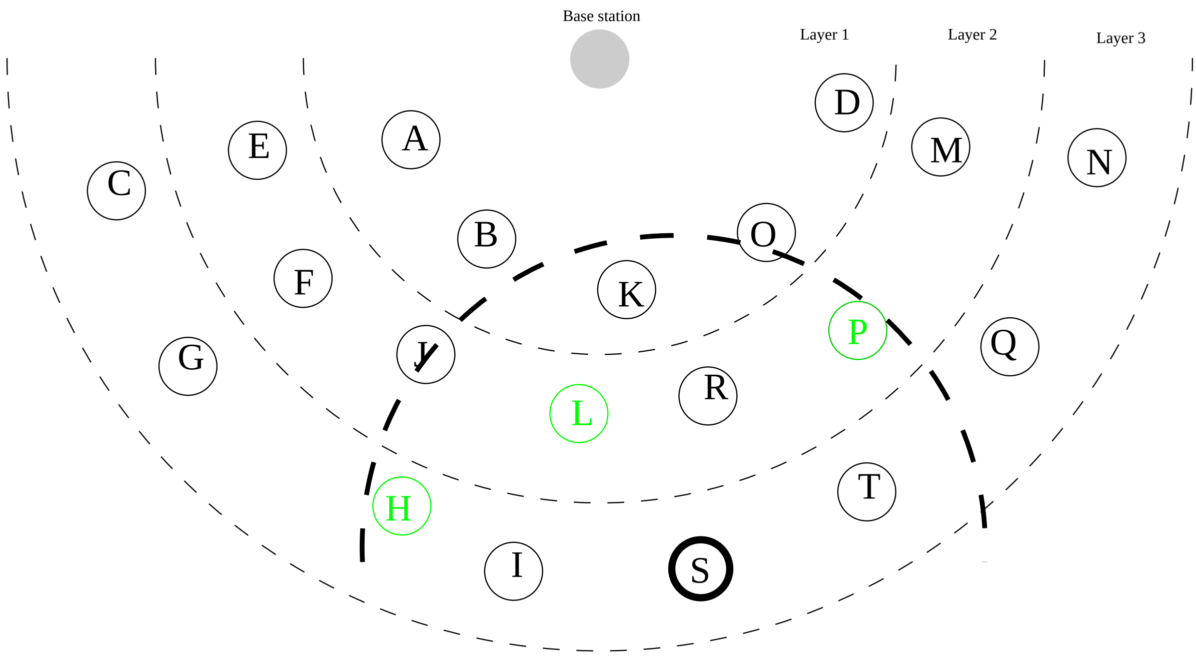


Figure 3.5: Anycasting is used to forward data

increase the chances of collisions, so in order to help reduce the chance of multiple nodes sending CTS messages at the same time, a node will wait a random amount of time before sending a CTS message. An example of this can be seen in Figure 3.5. If S is sending an RTS_LITE, all of the nodes that hear it that are at least one layer closer to the base station (K, L, R, P) will respond with a CTS after some random period of time. This helps in two major ways: it will reduce the amount of potential collisions from CTS messages, and it makes the network far more flexible, allowing multiple nodes to have the chance to be a forwarder for a single node. After receiving a CTS message, the four way handshake will continue as normal.

3.1.9 Energy Efficiency

In order to reduce energy consumption, protocols use a mechanism called a duty cycle that schedules a node to sleep and wakeup periodically for transmitting and receiving data. By having active and sleep cycles, nodes may save energy that would have been wasted on idle listening. All nodes in the network will have a fixed duty cycle and will start at random times. This helps reduce the amount of energy spent. An added benefit to using duty cycles is that it also helps reduce the amount of contention in the network. This can be seen when a node is attempting to gain channel access. When a node sends a RTS_LITE, multiple nodes may try and send a CTS message back, but when duty cycles are being used there is a chance that less nodes reply because not all of the nodes may be on to receive the RTS_LITE. An example of this can be seen in Figure 3.5. When S is sending only nodes H, L, and P are on and of those three nodes only P and L will respond to an RTS_LITE from S. Having only two nodes respond decreases the chances of a collision happening.

CHAPTER 4

PERFORMANCE EVALUATION

In order to evaluate the performance of the proposed protocol, we implemented it in the ns-2 simulator [44]. The performance of the approach was thoroughly tested through extensive experiments. As the proposed protocol extends upon the anycasting concept that is mentioned in ECR-MAC [31], we use it to provide comparison. Our comparison consists of the same metrics that ECR-MAC used to evaluate its approach: packets received at the base station, end-to-end delay of the first 10% of the packets that arrive at the base station, and the amount of energy consumed. Secondly, we compare our approach to two different variations of our protocol. In our comparison, we use the same standard metrics that we use to compare against ECR-MAC.

4.1 Simulation Setup

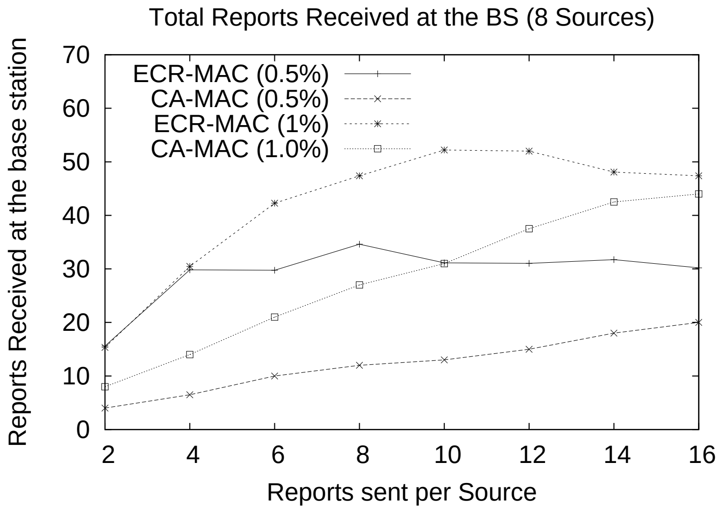
The simulations were run with the parameters shown in Table 4.1. We have 8 sources that report on an event when it occurs. Each source will generate Constant Bit Rate (CBR) traffic for the data packets. To evaluate the scalability of the protocol under different levels of congestion, we vary the number of packets sent per source and measure its performance. Each data point is taken from an average of 20 independent runs.

Table 4.1: Simulation Parameters

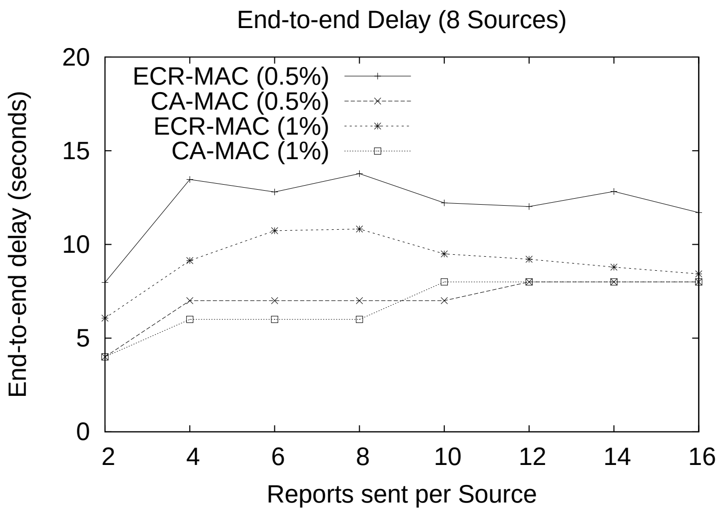
Parameter	Value
Area	80m x 80m
Deployment Strategy	Uniform Random
Transmission Radius	15m
Total number of nodes	500
Data Packet size (bytes)	64
Transmit power (mW)	14.88
Receive power (mW)	12.50
Idle power (mW)	12.36
Sleep power (mW)	0.016

4.2 Single Event Comparison

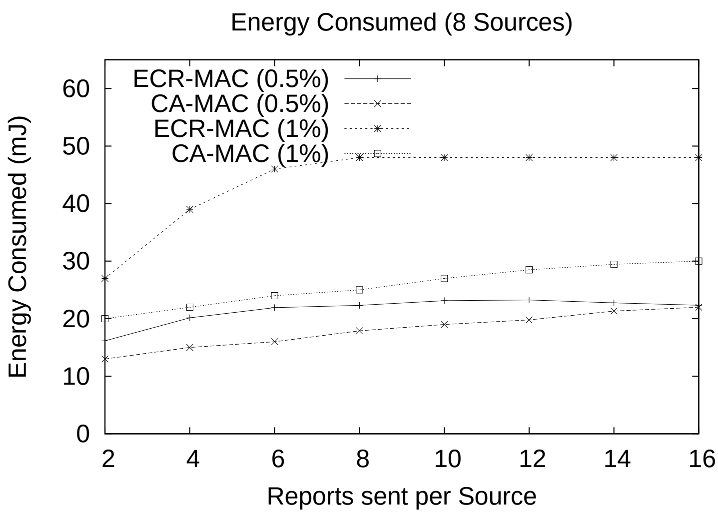
In our single event comparison, we have one event take place where 8 nodes within a 10 meter radius are randomly selected to report on the event. The proposed protocol is compared with a previous approach that is used to handle spatially-correlated contention. Figures 4.1(a), 4.1(b), and 4.1(c) show the performance comparison of both of protocols with a 0.5% and 1% duty cycle in terms of packets received at the base station, end-to-end delay of the first 10% of the packets received, and energy consumed. ECR-MAC receives more packets at the base station at all packet intervals in the simulation but as more packets are injected into the network the gap starts to close between CA-MAC and ECR-MAC. ECR-MAC has a greater end-to-end delay for the first 10% of packets received but that is attributed to more packets being received at the base station. CA-MAC consumes less energy than ECR-MAC. This happens because ECR-MAC has a very aggressive RTS mechanism that will keep transmitting RTS packets until the packet is sent to a forwarder. In CA-MAC, there is a cap on the number of RTS attempts a node can make before the packet is dropped. This cuts down on the amount of energy spent but it also leads to more packet drops, which has a greater effect on packets received at the lower packet intervals.



(a) Packets Received at the base station



(b) End-to-end delay of the first 10% of packets received



(c) Energy Consumed (mJ)

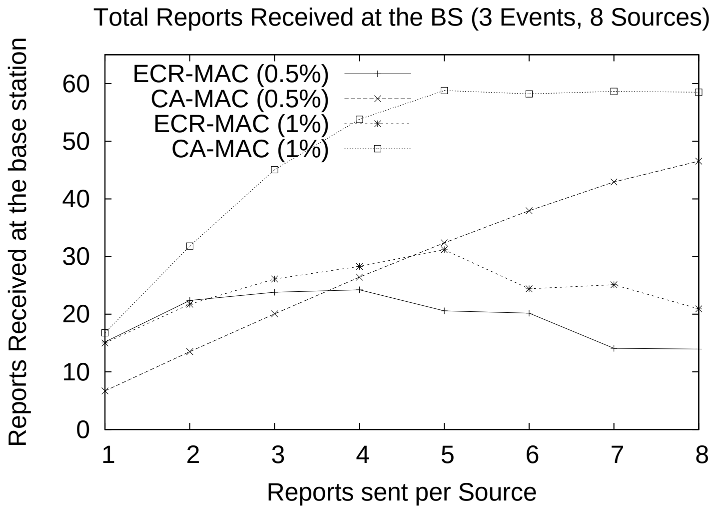
Figure 4.1: Single Event

4.3 Multiple Event Comparison

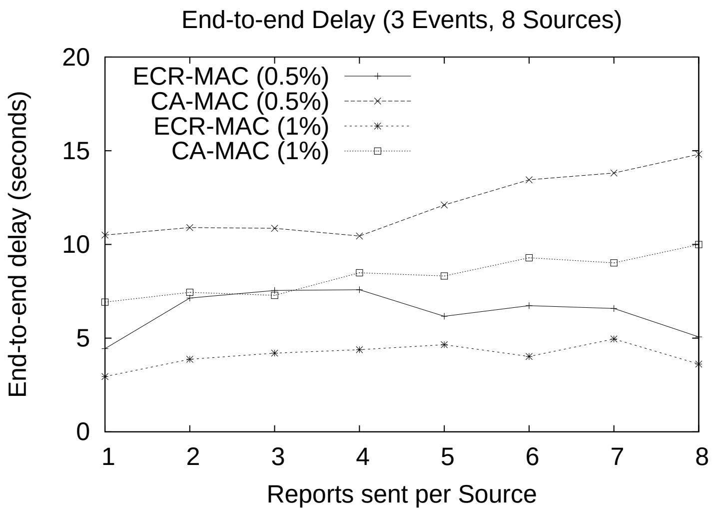
In an effort to make a more realistic scenario, we ran simulations where 3 concurrent events are detected around the network. Each event is sensed by 8 random nodes within a 10 meter radius. In Figure 4.2(a), CA-MAC (1%) receives more packets at each packet interval. CA-MAC (0.5%) starts to receive more packets than ECR-MAC (0.5%) at 4 reports sent per source. The improvement in the number of packets received at the base station can be attributed to the fact that in our design when we allow only certain sectors to send packets we are managing the amount of packets that can be sent to the first two layers at any point in time. By reducing the number of packets near the base station, the chances of contention are reduced and allow for more packets to make it to the base station. Though ECR-MAC (0.5%) performs better at the lower packet intervals, CA-MAC (0.5%) increases the gap between the two after a packet interval of 4. In Figure 4.2(b), it can be seen that CA-MAC consumes less energy than ECR-MAC due to less RTS transmissions being sent and received. In Figure 4.2(c), CA-MAC has a greater end-to-end delay because the protocol doesn′t allow nodes to send packets unless it is their sector′s turn to send. This extra time spent waiting leads to an increase in the end-to-end delay.

4.4 Protocol Comparison

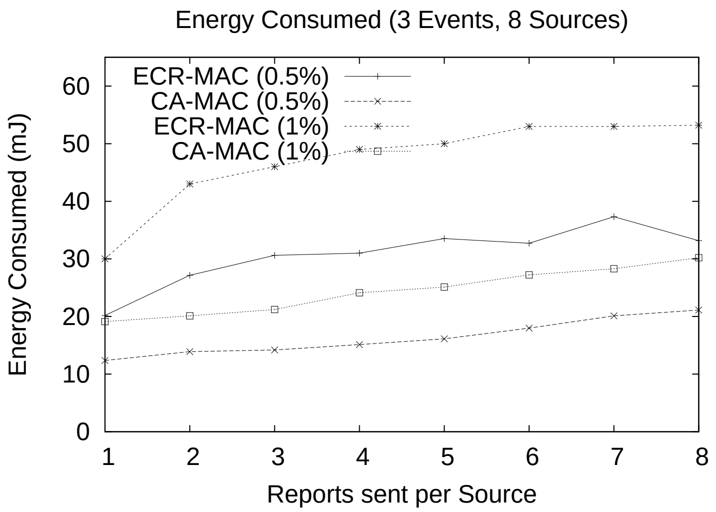
To evaluate the performance of our protocol, we compare our protocol to two other variations of our protocol. First, we compare it to a version of CA-MAC that has sectors all the way to the base station as opposed to CA-MAC where the sectoring stops at layer 2. Secondly, we compare it to a version of CA-MAC that uses TDMA with all of the nodes in the first layer. We do this because Funneling-MAC uses



(a) Packets Received at the base station



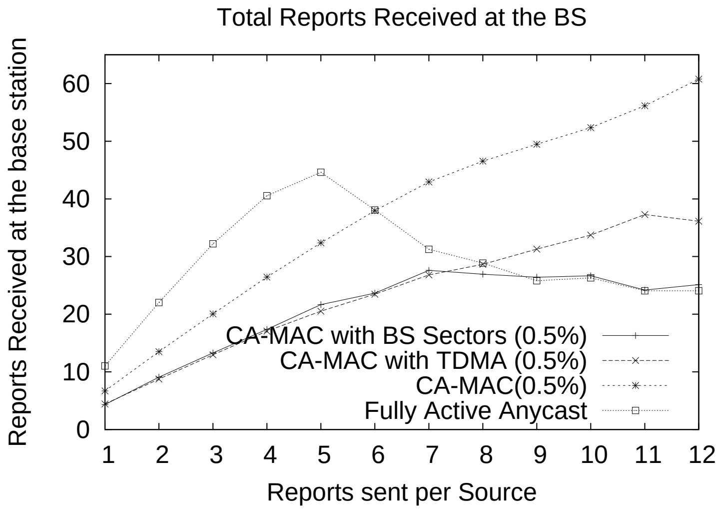
(b) End-to-end delay of the first 10% of packets received



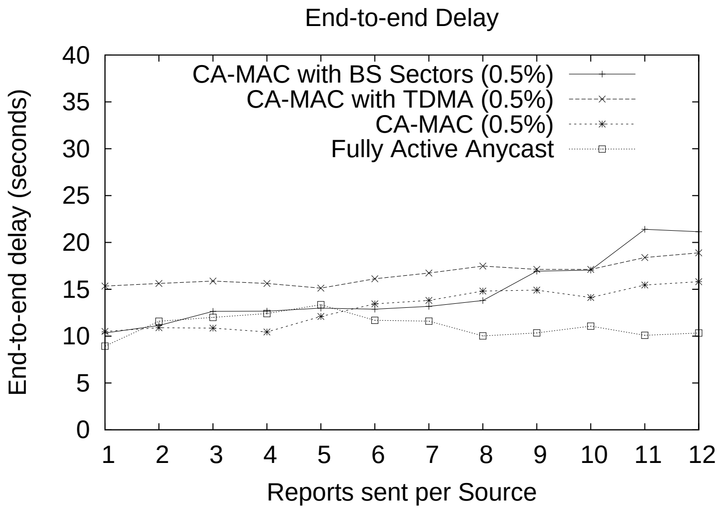
(c) Energy Consumed (mJ)

Figure 4.2: Multiple Events

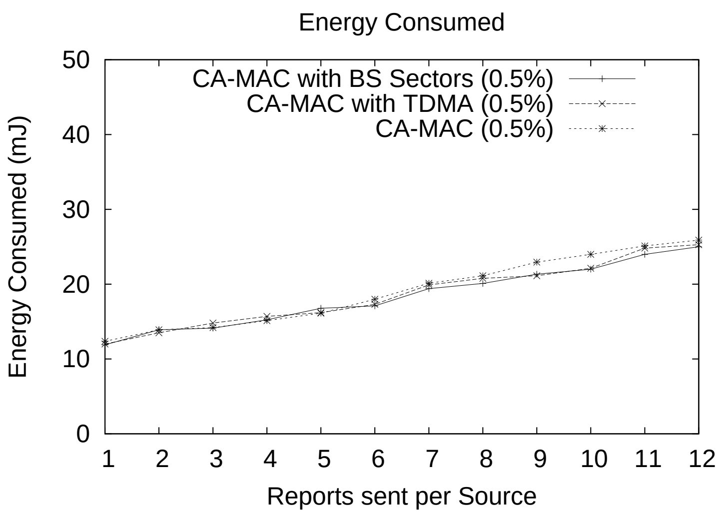
TDMA near the base station to reduce congestion near it. For the nodes that are not in range of the base station, a simple CSMA is used. We implemented TDMA on top of our CSMA so that it would be a fair comparison. To compare the protocols, we use the same standard metrics that we used in our comparison to ECR-MAC. In Figures 4.3(a), 4.3(b), and 4.3(c), it can be shown that CA-MAC receives more packets than the other two versions of CA-MAC for each packet interval while outperforming fully active anycast after a packet interval of 6. The reason that CA-MAC performs better than the CA-MAC version with sectors all the way to the base station is that by not having sectors used in the first two layers from the base station it gives the nodes with packets a better chance of sending the packets to the base station because they can send as soon as they get a packet to forward to the base station. CA-MAC outperforms CA-MAC with TDMA because each node close to the base station has a slot with which it can send data to the base station. This increases the amount of time a node must wait and it in turn backs up the nodes in the outer layers while waiting to send to the nodes in the closer layers. The fully active anycast does well when the network is not very congested, but after a packet interval of 5, the performance starts to decrease because there is too much contention in the network because every node is always on, which increases the chance of collisions occurring. CA-MAC has a smaller end-to-end delay while having received more packets at the base station. As expected, CA-MAC with TDMA has a greater end-to-end delay because it is giving a chance to every single node in the first layer to send data even if the node does not have any data to send. CA-MAC with BS sectors will have to wait longer too because each node in the first layer will only be able to send to the base station when it is the turn of the sector as opposed to CA-MAC where a node in the first two layers does not have to wait for its sector′s turn. Fully active anycast surprisingly



(a) Packets Received at the base station



(b) End-to-end delay of the first 10% of packets received



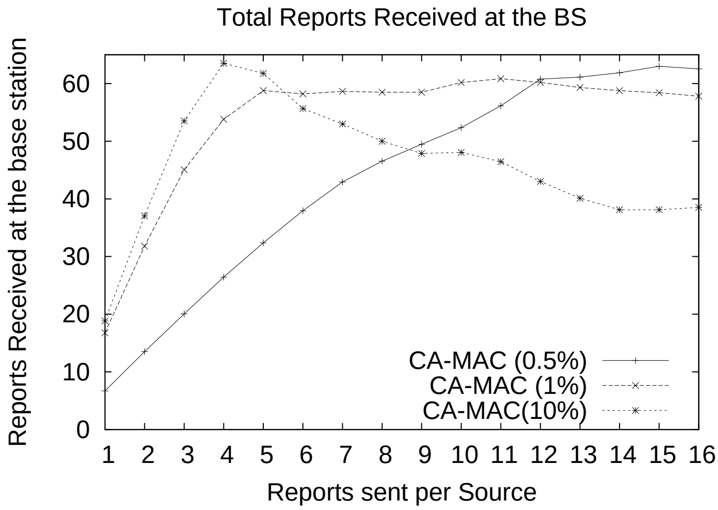
(c) Energy Consumed (mJ)

Figure 4.3: Protocol Comparison

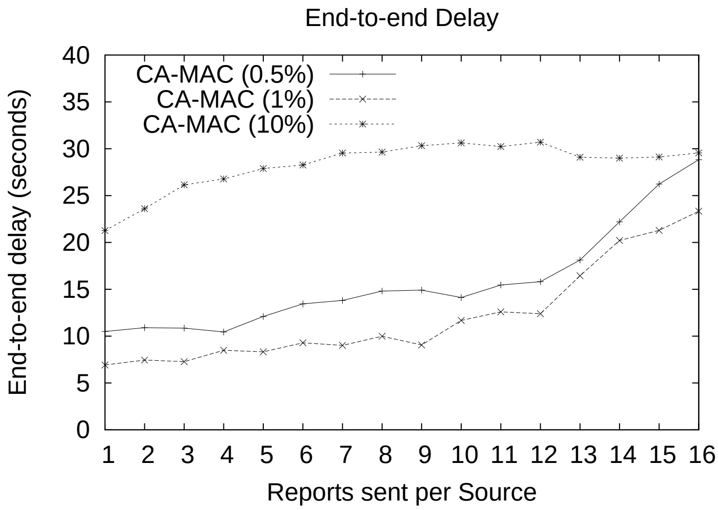
doesn′t have a smaller end-to-end delay. This is because the fully active anycast has far more contention in the network and this increases the delay. Energy for CA-MAC and the two other version of it are all right around the same, which is as expected because they are all using the same duty cycle, where the fully active anycast protocol consumes much more energy.

4.5 Duty Cycle Comparison

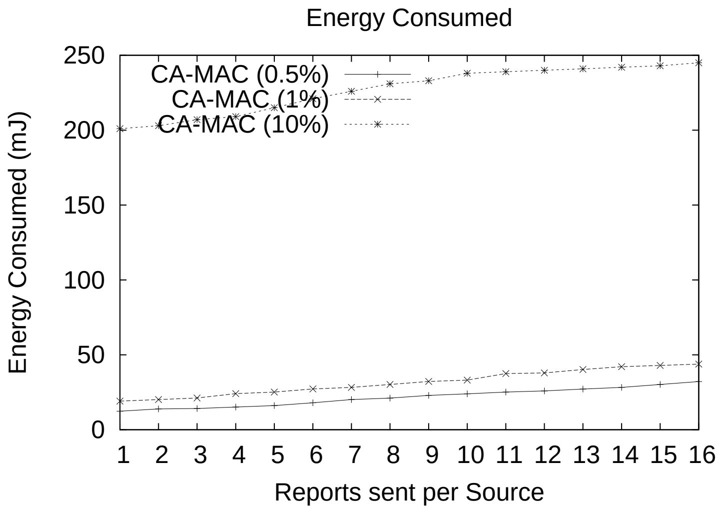
To see the effects of different duty cycles, we evaluated our protocol with duty cycles of 0.5%, 1%, and 10%. CA-MAC with duty cycles of 1% and 10% do better at the beginning compared to CA-MAC (0.5%) because the higher the duty cycle the greater the chance there will be more nodes on to be potential forwarders to a node sending a RTS. As the packet interval increases, CA-MAC (10%) begins to decrease after a packet interval of 4 because the network begins to become more congested and more collisions begin to occur because a greater number of nodes are on. CA-MAC (1%) rises quickly to approximately 60 received packets and stays there for each subsequent interval until a packet interval of 10 where it takes a slight dip. CA-MAC (0.5%) rises at a much slower pace than CA-MAC (1%) but ultimately ends up at approximately 60 received packets for the highest packet intervals even outperforming CA-MAC (1%) by a little bit. CA-MAC (1%) and especially CA-MAC (0.5%) do well the more congested the network becomes. CA-MAC (0.5%) has a greater end-to-end delay than CA-MAC (1%), which shows the well-known energy-to-delay tradeoff. Counterintuitively, CA-MAC (10%) has a greater end-to-end delay and this is because there is more contention in the network when more nodes are on. As expected, energy results are as they should be with CA-MAC (0.5%) consuming the least amount of energy and CA-MAC (10%) consuming the most.



(a) Packets Received at the base station



(b) End-to-end delay of the first 10% of packets received



(c) Energy Consumed (mJ)

Figure 4.4: Duty Cycle Comparison

CHAPTER 5

CONCLUSIONS

Wireless Sensor Networks are mainly deployed for monitoring purposes in various fields. When an event occurs, due to the nature of a WSN, many nodes will sense the event and try and forward their data to the base station. When many events are happening around the network, many packets are being sent to the base station, which will cause contention and congestion near it. Having this congestion near the base station will only make the network performance worse. Congestion at the base station is a guarantee when high traffic loads are in the network, but a protocol should be used that can prolong the usefulness of a WSN. Due to the energy constraints of a sensor node, it is also important that energy efficiency be one of the main concerns when designing a protocol for the network.

To enhance the networks performance in highly congested scenarios, we designed CA-MAC to be an energy-efficient, congestion-avoidance MAC protocol by using anycasting to diffuse traffic from areas with high contention and by using sending sectors to manage the traffic flow arrivals at the first two layers of the network. Many protocols have been proposed that deal with spatially-correlated contention or congestion near the base station, but no protocol has been proposed that addresses both issues. Our simulation results show that this technique improves both the number of packets received at the base station and the amount of energy consumed in highly congested scenarios. However, our protocol has an increase in the end-to-end delay because we use low duty cycles.

REFERENCES

[1] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM.

[2] Yanmin Zhu. *Low-power event detection and wakeup scheduling in wireless sensor networks*. PhD thesis, 2007. AAI3298801.

[3] S. K. Das, D. J. Cook, A. Battacharya, E. O. Heierman, III, and Tze-Yun Lin. The role of prediction algorithms in the MavHome smart home architecture. *Wireless Commun.*, 9(6):77–84, December 2002.

[4] Wook Choi. *A novel framework for energy and application-aware data gathering in wireless sensor networks*. PhD thesis, Arlington, TX, USA, 2005. AAI3179118.

[5] Zongheng Zhou, Samir R. Das, and Himanshu Gupta. Variable radii connected sensor cover in sensor networks. *ACM Trans. Sen. Netw.*, 5(1):8:1–8:36, February 2009.

[6] F. Stann and J. Heidemann. RMST: reliable data transport in sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 102 – 112, may 2003.

[7] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wirel. Netw.*, 8(5):481–494, September 2002.

[8] Jiong Wang and Sirisha Medidi. Energy Efficient Coverage with Variable Sensing Radii in Wireless Sensor Networks. In *Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, WIMOB '07, pages 61–, Washington, DC, USA, 2007. IEEE Computer Society.

[9] M. Dhanaraj, B. S. Manoj, and C. Siva Ram Murthy. A New Energy Efficient Protocol for Minimizing Multi-Hop Latency in Wireless Sensor Networks. In *Proceedings of the Third IEEE International Conference on Pervasive Computing*

and Communications, PERCOM '05, pages 117–126, Washington, DC, USA, 2005. IEEE Computer Society.

[10] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52(12):2292–2330, August 2008.

[11] A Wakeup Scheme for Sensor Networks: Achieving Balance between Energy Saving and End-to-end Delay. In *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium*, RTAS '04, pages 19–, Washington, DC, USA, 2004. IEEE Computer Society.

[12] Chieh-Yih Wan, Shane B. Eisenman, and Andrew T. Campbell. CODA: congestion detection and avoidance in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 266–279, New York, NY, USA, 2003. ACM.

[13] Cheng Tien Ee and Ruzena Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 148–161, New York, NY, USA, 2004. ACM.

[14] Chonggang Wang, Mahmoud Daneshmand, Bo Li, and Kazem Sohraby. A survey of transport protocols for wireless sensor networks. *IEEE Network, May/June*, 20:34–40, 2006.

[15] Özgür B. Akan and Ian F. Akyildiz. Event-to-sink reliable transport in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 13(5):1003–1016, October 2005.

[16] Bret Hull, Kyle Jamieson, and Hari Balakrishnan. Mitigating congestion in wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 134–147, New York, NY, USA, 2004. ACM.

[17] Alec Woo and David E. Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, pages 221–235, New York, NY, USA, 2001. ACM.

[18] Peng Xie, Zhong Zhou, Zheng Peng, Jun-Hong Cui, and Zhijie Shi. SDRT: A reliable data transport protocol for underwater sensor networks. *Ad Hoc Netw.*, 8(7):708–722, September 2010.

[19] Seung-Jong Park, Ramanuja Vedantham, Raghupathy Sivakumar, and Ian F. Akyildiz. A scalable approach for reliable downstream data delivery in wireless

sensor networks. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '04, pages 78–89, New York, NY, USA, 2004. ACM.

[20] Shigang Chen and Na Yang. Congestion Avoidance Based on Lightweight Buffer Management in Sensor Networks. *IEEE Trans. Parallel Distrib. Syst.*, 17(9):934–946, September 2006.

[21] Hongwei Zhang, Anish Arora, Young-ri Choi, and Mohamed G. Gouda. Reliable bursty convergecast in wireless sensor networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '05, pages 266–276, New York, NY, USA, 2005. ACM.

[22] Jaewon Kang, Yanyong Zhang, and Badri Nath. TARA: Topology-Aware Resource Adaptation to Alleviate Congestion in Sensor Networks. *IEEE Trans. Parallel Distrib. Syst.*, 18(7):919–931, July 2007.

[23] Chieh-Yih Wan, Andrew T. Campbell, and Lakshman Krishnamurthy. PSFQ: a reliable transport protocol for wireless sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 1–11, New York, NY, USA, 2002. ACM.

[24] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, February 2003.

[25] Mihaela Cardei, Jie Wu, and Mingming Lu. Improving network lifetime using sensors with adjustable sensing ranges. *Int. J. Sen. Netw.*, 1(1/2):41–49, September 2006.

[26] Fikret Sivrikaya and Blent Yener. Time synchronization in sensor networks: A survey. *IEEE Network*, 18:45–50, 2004.

[27] Jeremy Eric Elson. *Time synchronization in wireless sensor networks.* PhD thesis, 2003. AAI3104330.

[28] Jeremy Elson and Kay Römer. Wireless sensor networks: a new regime for time synchronization. *SIGCOMM Comput. Commun. Rev.*, 33(1):149–154, January 2003.

[29] Y. Xue, M. Can Vuran, and Byrav Ramamurthy. Cost-Efficiency of Anycast-Based Forwarding in Duty-Cycled WSNs with Lossy Channel. In *IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9, 2010.

[30] Yuanyuan Zhou and Muralidhar Medidi. Sleep-based Topology Control for Wakeup Scheduling in Wireless Sensor Networks. In *SECON'07*, pages 304–313, 2007.

[31] Yuanyuan Zhou. *Energy-efficient protocols and topologies for sensor and personal-area networks*. PhD thesis, Pullman, WA, USA, 2007. AAI3277076.

[32] Gang Lu, Bhaskar Krishnamachari, and Cauligi S. Raghavendra. An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks. In , pages 224–231, 2004.

[33] Divyasheel Sharma, Vladimir I. Zadorozhny, and Panos K. Chrysanthis. Timely data delivery in sensor networks using whirlpool. In *Proceedings of the 2nd international workshop on Data management for sensor networks*, DMSN '05, pages 53–60, New York, NY, USA, 2005. ACM.

[34] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[35] Hüseyin Özgür Tan and Ibrahim Körpeoğlu. Power efficient data gathering and aggregation in wireless sensor networks. *SIGMOD Rec.*, 32(4):66–71, December 2003.

[36] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for Ad Hoc routing. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, pages 70–84, New York, NY, USA, 2001. ACM.

[37] Jiong Wang and Sirisha Medidi. Mesh-Based Coverage for Wireless Sensor Networks. In *GLOBECOM*, pages 362–366, 2008.

[38] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8*, HICSS '00, pages 8020–, Washington, DC, USA, 2000. IEEE Computer Society.

[39] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, and Mani Srivastava. Optimizing Sensor Networks in the Energy-Latency-Density Design Space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, January 2002.

[40] Gahng-Seop Ahn, Se Gi Hong, Emiliano Miluzzo, Andrew T. Campbell, and Francesca Cuomo. Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks. In *Proceedings of the 4th international conference on*

Embedded networked sensor systems, SenSys '06, pages 293–306, New York, NY, USA, 2006. ACM.

[41] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506, June 2004.

[42] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 171–180, New York, NY, USA, 2003. ACM.

[43] Chunlong Guo, Lizhi Charlie Zhong, Jan M. Rabaey, I. Low, Power Ad, and Hoc Radio Networks. Low power distributed MAC for ad hoc sensor radio networks. *IEEE Globecom*, 2001.

[44] Network Simulator. *http://www.isi.edu/nsnam/ns.*

[45] Michele Zorzi, Ramesh R. Rao, Senior Member, and Senior Member. Geographic random forwarding (geraf) for ad hoc and sensor networks: Multihop performance. *IEEE Transactions on Mobile Computing*, 2:337–348, 2003.

[46] L.F.W. Hoesel van and P.J.M. Havinga. A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches. In *1st International Workshop on Networked Sensing Systems, INSS 2004*, pages 205–208, Tokio, Japan, 2004. Society of Instrument and Control Engineers (SICE).

[47] Injong Rhee, Ajit Warrier, Mahesh Aia, and Jeongki Min. Z-mac: a hybrid mac for wireless sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, pages 90–101, New York, NY, USA, 2005. ACM.