

**DYNAMIC TRANSMISSION SCHEDULING FOR
CONTENTION MITIGATION
IN WIRELESS SENSOR NETWORKS**

by

Suryanarayana Mahesh Tatapudi

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Computer Science

Boise State University

May 2013

© 2013
Suryanarayana Mahesh Tatapudi
ALL RIGHTS RESERVED

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Suryanarayana Mahesh Tatapudi

Thesis Title: Dynamic Transmission Scheduling for Contention Mitigation in
Wireless Sensor Networks

Date of Final Oral Examination: 14 March 2013

The following individuals read and discussed the thesis submitted by student Suryanarayana Mahesh Tatapudi, 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

Jyh-haw Yeh, 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.

Dedicated to the memory of my beloved father, who was my greatest supporter

ACKNOWLEDGMENTS

I would like to express deep gratitude to Dr. Sirisha Medidi and Dr. Murali Medidi, for helping me complete this thesis. This task would have been inconceivable without their support. Their encouragement and guidance throughout the duration of the Master's program played a key role in shaping my skill set.

I want to thank Dr. Jyh-haw Yeh for patiently evaluating this thesis while serving on my thesis committee.

Thanks to all the members of SWAN Lab, for their help and encouragement. I also wish to thank Surya and Ramya for their love and support throughout my schooling at Boise State University.

Special thanks to my family for their love, support, and unrelenting commitment towards my success.

ABSTRACT

Wireless sensor networks (WSNs) heavily rely on a dense deployment of sensor nodes in order to ease deployment, increase fault-tolerance and network coverage, so that events do not go undetected. A dense deployment however results in several sensor nodes close to each other detecting and transmitting event reports at almost the same time, resulting in severe contention for channel access. Channel contention is a serious problem in WSNs resulting in collisions, re-transmissions, energy depletion, and ultimately loss of event reports. TDMA-based protocols prevent contention, but require tight synchronization and may lead to severe wastage of bandwidth especially in event-based applications where the traffic is bursty in nature. Other approaches that handle spatially correlated contention are fairly complex and contradict the reason for dense deployment, by selecting only a subset of nodes that generate and transmit event reports, affecting the fault-tolerance and confidence of event detection.

Motivated by the challenge to reduce contention and improve performance, we propose a protocol that dynamically schedules transmissions in the network. The protocol exploits the broadcast nature of a wireless medium, which allows nodes to overhear transmissions of neighboring nodes and establish a cooperative transmission schedule dynamically, without the need for synchronization or explicit message exchange. To further mitigate contention, we propose a heuristic to reduce the number of active forwarder nodes in the network, by increasing the overlap of forwarder nodes used while routing packets. This forwarding mechanism can isolate the areas prone

to interference, within which the dynamic transmission scheduling mechanism works better to mitigate contention. We evaluate the performance of our protocol using the NS2 simulator. Results show that our protocol significantly reduces collisions and re-transmissions, thereby improving the performance of the network.

TABLE OF CONTENTS

ABSTRACT	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xii
1 Introduction	1
1.1 Wireless Sensor Networks (WSNs)	1
1.1.1 Organization of Thesis	4
2 Related Work	5
2.1 Collision Avoidance and Recovery	5
2.2 Selective Reporting based on Spatial Correlation	9
2.3 Contention-Window Management	10
2.4 Energy Efficient Medium Access Control	11
2.5 Topology Management	13
2.6 Congestion Control and Avoidance	14
3 Protocol for Contention Mitigation	17
3.1 Motivation and Design Considerations	17
3.1.1 Design Considerations	19

3.2	Assumptions	21
3.3	Contention Mitigation	21
3.3.1	Basic Approach	21
3.3.2	Dynamic Transmission Scheduling	23
3.3.3	Data Gathering with Increased Forwarder Set Overlap	30
4	Performance Evaluation	36
4.1	Performance Metrics	36
4.2	Comparison with SC-MAC	37
4.3	Protocol Evaluation	42
5	Conclusion	47
	BIBLIOGRAPHY	49

LIST OF TABLES

4.1	Simulation Parameters for Comparison with SC-MAC	38
4.2	Simulation Parameters for Protocol Evaluation	43

LIST OF FIGURES

3.1	Structure of Schedule Table	27
3.2	Initial Configuration for a Sample Network	28
3.3	Schedule establishment and dynamic maintenance	29
3.4	Comparison of Forwarder Configurations	32
4.1	Packet Delivery Ratio	39
4.2	Collision Rate	40
4.3	Energy Consumption	41
4.4	Average Delay	42
4.5	Packet Delivery Ratio	44
4.6	Packet Collision Rate	45
4.7	Energy Efficiency	46
4.8	Average Delay	46

LIST OF ABBREVIATIONS

WSN – Wireless Sensor Network

MAC – Medium Access Control

DTS-MAC – Dynamic Transmission Scheduling MAC

TDMA – Time Division Multiple Access

CSMA/CA – Carrier Sense Multiple Access with Collision Avoidance

RTS – Request to Send

CTS – Clear to Send

CHAPTER 1

INTRODUCTION

1.1 Wireless Sensor Networks (WSNs)

Recent developments in wireless communication and embedded computing technologies have led to the advent of wireless sensor network technology. Wireless sensor networks [4] typically consist of small, inexpensive, and low-powered distributed devices, capable of sensing various environmental parameters. These devices, called sensor nodes, are equipped with a low-power radio transmitter, various sensors, a small battery unit, limited memory, and a micro-controller. The sensor nodes process the sensed data and cooperatively disseminate information to a common sink or base station, in an ad-hoc manner, using wireless communication. The sink is a special node that has greater processing and communication capabilities over the other sensor nodes, as it must perform complex data processing of the sensed information. In some cases, it also must be able to reach all the sensor nodes in the network directly, for dispatching any information.

Wireless Sensor Networks can typically be used to achieve continuous monitoring or event-detection inside the supervised area. Applications using both continuous monitoring and event-driven reporting also exist. In continuous monitoring applications, each sensor node transmits its sensed data periodically to the sink node while

in event-driven applications, the sensor nodes do a small amount of pre-processing on the sensed data and only if an event of interest is found to occur, they generate reports relating to the event and transmit them to the base station. The base station then decides on an event, based on all the reports it got and performs necessary actions, depending on the application. Although WSNs were initially targeted for continuous monitoring applications, a lot of recent research has been focused on making WSNs more suitable for event detection.

WSNs meant for event detection are usually deployed in extreme temperatures, areas that are hostile or hard to access. Hence, a controlled deployment of sensor nodes is hard to achieve. Given these conditions, it is still important to ensure connectivity of each sensor node to the base station and ensure that the base station receives as many reports as possible from different sensor nodes, so that it can detect events with high confidence. Thus, WSNs heavily rely on a dense deployment of sensor nodes, which eases deployment, supports fault-tolerance, and helps achieve satisfactory coverage, so that events do not go undetected. However, in spite of its advantages, dense deployment introduces some serious issues, which have a deterrent effect on the performance of the network if not handled gracefully. The primary issue with dense deployment is that, when an event of interest occurs, several sensor nodes detect and attempt to report it at almost the same time. Several of these sensor nodes may also be spatially close to each other, resulting in severe contention for channel access [6]. Event-driven networks also generate and transmit data only when an event of interest occurs, making the traffic load unpredictable and bursty in nature. These characteristics of event-driven WSNs make them more challenging and harder to design, compared to the networks targeted for continuous monitoring.

This thesis focuses on event-detection applications, specifically ones where a single event can potentially be detected by a large number of sensor motes at the same time. For example, when sensors are deployed for detecting events such as volcanic activity, flash floods, seismic activity, etc., a single event is detected by a large group of sensors at the same time, since such event properties propagate through a large area in the network.

As mentioned above, channel contention is a serious issue in event-driven WSNs. Contention for channel access occurs when there are two or more nodes attempting to transmit data at the same time in a given transmission range. During this process, there is a likelihood that two or more nodes transmit data at the same time over the shared medium, resulting in a collision. The likelihood of a collision increases when the number of contending nodes is large, which usually is the case in event-driven networks due to the dense deployment. Collisions increase the number of packet drops and thus lower the throughput of the network. The cost of collision in wireless networks is more than just packet loss. Each collision triggers a re-transmission at the sender's end, resulting in greater energy consumption in the network. Moreover, there is always a time gap between when the collision occurs and when the data is re-transmitted, because a node only learns about the collision when the transmission is finished and the expected acknowledgment does not come back. Consequently, a collision will cause loss of an entire packet transmission duration, increasing latency in the network.

The wireless medium being inherently broadcast in nature is prone to interference and contention and thus requires highly optimized medium access control (MAC) [7],

especially in event-based WSNs, where there are several nodes trying to access the channel all at the same time during the occurrence of an event. The prime role of the MAC is to coordinate access to a medium common to several nodes such that the amount of contention and the number of collisions is minimal. However, it is very difficult to estimate the load conditions of the network and hence challenging to design effective media access protocols that reduce the number of collisions. In the wireless context, this is even more complicated by the fact that the common medium is the wireless channel, which is broadcast in nature; i.e., any ongoing transmission interferes with any other transmission within communication range. For this reason, more efficient contention resolution mechanisms are needed to overcome these challenges and reduce the collision probability among various contending nodes, which forms the primary motivation of this thesis.

1.1.1 Organization of Thesis

The rest of the thesis is organized as follows: Chapter 2 reviews the related work. Chapter 3 details the motivation and objective of the thesis and discusses the proposed protocol design. Chapter 4 presents the performance evaluation of the protocol. Finally, Chapter 5 highlights the conclusions.

CHAPTER 2

RELATED WORK

In recent years, a number of studies have been done on improving performance in wireless sensor networks. There are also specific contributions targeted towards medium access control protocols and on managing contention in wireless sensor networks with dense deployments. These have resulted in various approaches to handle contention and improve performance of the network.

2.1 Collision Avoidance and Recovery

There are several protocols designed to avoid and/or recover from collisions due to contention in wireless networks. These can primarily be classified into contention-free and contention-based approaches. Contention-free MAC protocols are predominantly based on TDMA [25] [21], where each node has a preset slot only during which is it allowed to transmit. Even though they completely solve the problem of contention, such protocols perform very poorly, especially in event based WSNs due the bursty nature of traffic involved. Each node has to wait until its turn, even if other nodes have nothing to transmit, adding a lot of delay, unacceptable for critical-event reporting. TDMA is more appropriate where there is a continuous data flow by almost all the nodes in the network, and the amount of data each node has to send is comparable. Additionally, TDMA protocol requires tight network-wide synchronization, which

brings extra overhead of clock synchronization, which involves frequent exchange of control packets between nodes. Other contention-free approaches like FDMA and CDMA face similar drawbacks.

Due to the drawbacks of contention-free protocols discussed above, they are usually not used in an event-based WSN. The other type of protocols that are predominantly used in WSN are contention-based protocols, usually based on ALOHA and CSMA [8]. This approach is fairly simple compared to reservation-based protocols, mainly because neither global synchronization nor topology knowledge is required. In a contention-based approach, nodes compete for the use of the wireless medium and only the winner of this competition is allowed to access the channel and transmit.

ALOHA is again of two types: Pure and Slotted ALOHA. In Pure ALOHA, if a node has data to send, it sends it without sensing the channel for activity. If there is a collision, the node tries sending after some time, depending on the back off mechanism chosen. An improvement to the original ALOHA protocol was Slotted ALOHA, in which a node can only send during the start of a discrete time frame. This reduces collisions, compared to the pure version, because now a collision can only happen if two nodes happen to have data to transmit at the beginning of the same time slot. All other data transmission is postponed until the start of the next discrete time slot. Although Slotted ALOHA has better throughput compared to the traditional ALOHA, it suffers the disadvantage of needing synchronization between nodes.

In CSMA, a node having a packet to transmit first senses the channel before actually transmitting. In the case that the node finds the channel busy, it postpones

its transmission to avoid interfering with the ongoing transmission. If the node finds the channel clear, it starts transmitting. This protocol can efficiently reduce collisions using clear channel assessment and random back-off, however cannot completely eliminate collisions, especially in event-based WSN scenarios, where the traffic is bursty in nature with multiple nodes sensing the medium free at the same time [35].

ZMAC [22] combines the strengths of TDMA and CSMA while managing 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 collisions among two-hop neighbors at a low cost. However, traffic conditions are hard to estimate due to the sporadic and bursty nature of traffic in event-based WSNs. Moreover, it takes time to adapt to the changes in traffic patterns, which may not be acceptable.

There are interesting protocols that improvise either on CSMA-CA or TDMA or work across network layers for additional benefits [15] [23] [2] [21]. ECR-MAC [34] proposes a cross-layered MAC protocol, which employs a Dynamic Forwarder Selection (DFS) technique, to improve network performance. The core idea is to have multiple potential forwarders for each node and use an any-cast mechanism to forward data. ECR-MAC deliberately avoids synchronized wake-up of nodes which not only reduces overhead but also relieves contention, since synchronization, which usually lets senders and/or forwarders wake up at the same time, can intensify the contention. Each node also maintains multiple potential forwarders that choose their own wake-up schedule randomly and independently. So a sender, as soon as its report is ready to be forwarded, can usually identify a forwarder quickly without competing with other

senders whose reports may be ready at a later time. Even if two senders send reports at the exact same time, ECR-MAC can also relieve contention by dispersing the paths the senders will take. ECR-MAC depends only on randomness in wakeup schedules to avoid contention among nodes that are in the transmission range of each other. However, in dense deployments, it is very likely that there are several nodes within the transmission range of each other and active at the same.

Alert MAC [19] proposes a protocol for collecting event-triggered urgent messages from a group of sensor nodes. Alert minimizes contention among nodes by using a combination of time and frequency division multiplexing. Multiple frequency channels are used within time slots and contention is minimized by controlling the selection probability of each channel by the nodes. The authors however use a single hop test bed and assume that the base station is in the communication range of all the nodes in the network. Thus, the performance does not truly reflect a real-world, event-based scenario where there may be multiple events in the network at different regions and reports from all regions converging to the base station.

ACR [28] takes a different approach by focusing on how to efficiently recover collided packets in dense WSNs with heavy collisions, without emphasis on avoiding collisions. The key idea is to use the Partial Corrupted Pattern (PCP), where only some bits are actually corrupted in a lost packet and use mechanisms such as Forward Error Correction to recover the corrupted bits. However, not all except LS-Collisions produce a regular PCP in collided packets. Given the unpredictable nature of events and traffic patterns in event-based WSNs, probability that a collision is an LS-Collision is not very high, limiting the benefits of such an approach.

2.2 Selective Reporting based on Spatial Correlation

Event-detection applications usually rely on dense network deployments for increased confidence on event reports and network coverage. However, there are some who argue that event-based applications need only a subset of the reports generated by all source nodes detecting an event. Specifically, since source nodes spatially close to each other have highly correlated data, it may not be required for all of the reports from those nodes to reach the base station. It also makes sense to avoid contention between spatially close-by nodes, also called spatially correlated contention.

CC-MAC [26] exploits spatially-correlated contention, by choosing only a subset of nodes that act as representative nodes in the network. Whenever an event occurs, only the representative nodes send reports, thereby reducing the number of redundant reports and hence relieving contention. Based on a theoretical framework, a distributed, spatial Correlation based Collaborative Medium Access Control (CC-MAC) protocol was designed that has two components: Event MAC (E-MAC) and Network MAC (N-MAC). E-MAC filters out the correlation in sensor records while N-MAC prioritizes the transmission of route-thru packets. However, the way representative nodes are selected is based on complex algorithms requiring node distribution statistics. This may not be possible due to the highly random deployments with varying node densities and unpredictable events.

SCMAC [33] proposes an improvement over CC-MAC by choosing the subset of nodes that transmit based on signal strength. The protocol adopts priority node selection algorithm, based on signal strength, while prioritizing route-thru packets.

However, it suffers the same disadvantages of CC-MAC.

Both these approaches contradict the rationale behind a dense deployment, which is to improve fault-tolerance and network coverage. Spatially correlated contention is indeed bad and should be avoided. However, performing a selective reporting only reduces confidence of the event reports and leads to false positives, as a random deployment can contain several faulty and malicious nodes. Moreover, with selective reporting, the base station may fall short on the number of event reports. Thus, I believe spatially correlated contention should be handled not by deeming only certain nodes out of the available nodes to report, but by gracefully managing how and when the source nodes access the medium.

2.3 Contention-Window Management

CSMA/MACAW aim at reducing collisions by equally trying to protect all packets. In sensor networks, however, packets may have different importance. For example, in surveillance applications, all nodes detecting an intrusion send an alert to the sink that generates peak traffic around the intrusion region. In this case, it is more important to protect the first packets so that they reach the sink rapidly.

Sift [13] considers this issue by exploiting the spatio-temporal correlation that exists between events generated by nodes. Sift's goal is to reduce collisions of the first R of N potential reports from nodes to the sink. This is done using a fixed size contention window, and nodes pick slots from a probability distribution instead of a uniform distribution, such that only a few nodes contend for the first few slots.

Sift chooses a fixed contention-window size, but does not justify the rationale behind the specific choice of the window size. Other authors [10] [20] [5] argue that choosing a fixed contention-window size requires an engineering optimization and should be done by taking into account several factors such as the number of contending nodes, size of data packets etc. The papers arrive at a relation between contention-window size, delay, and energy consumption. However, it is hard to estimate the number of contending nodes, specially in event-based WSNs, where the deployment is random and dense. Also, when there are multiple events occurring at different event areas in a sensor network, it is hard to establish the urgency of a report or tell if it belongs in the first 'R' of the 'N' reports generated for that event, at other levels above the level the event occurred.

2.4 Energy Efficient Medium Access Control

There are a lot of energy saving MAC protocols [12] [16] [14] [17] [31] [24] proposed. The core idea behind these protocols is to use active/sleep duty cycles, so that nodes periodically switch off their radio to save energy and avoid idle listening. However, a node that has data to transmit has to wait until its forwarder is awake, causing additional transmission latency. Most proposed protocols addressed this trade-off by using synchronized wake up/sleep cycles and/or dual radio setup, which increases hardware cost and the protocol overhead considerably. More importantly, though employing synchronization is effective to shorten the delay in the scenario without contention, it can impair energy efficiency and throughput under spatially-correlated contention, since multiple senders that are synchronized to be activated at the same

time contend for forwarders causing collisions.

DMAC [17] considers the situation where many sources send data to a sink through a unidirectional tree, called converge-cast communication. Nodes exploit this tree to determine their active schedules. A node determines its active schedules according to the traffic load and to its depth in the tree. However, since the duty cycles are not random for each node, there is an overhead involved in synchronizing the wake-up schedules.

SMAC [31] copes with idle listening by repeatedly putting nodes in active and sleep periods in a random manner. Nodes turn off their radios in sleep periods to save energy and they turn them on in active periods to exchange packets. Applications may want to send packets while nodes are in sleep periods. SMAC postpones these transmissions until the next active period. As nodes sleep most of the time, strong contention for the channel is expected in the active periods. Specifically, nodes are implicitly synchronized at the beginning of active periods and there is a significant chance they simultaneously access the channel at the beginning of an active period.

TMAC [24] is an improvement on SMAC and proposes an adaptive duty cycle in which the duration of active periods is no longer fixed but varies according to the traffic. The key idea of TMAC consists in making a node predict channel activity during an active period so that it can switch its radio off before the active period ends, in case it does not expect any traffic. By downsizing active period lengths, TMAC saves more energy than SMAC.

PTW [30] and LEEM [11] use dual radios 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. 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.

LEEM requires that a system-wide synchronization takes place so that the nodes along a route can be scheduled to wake up sequentially. So the idea behind these protocols is that a node uses one of the radios to wake up its 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 packet size and the data radio bandwidth. Moreover, these mechanisms need some kind of synchronization and come in handy at the cost of having additional dual radio hardware.

2.5 Topology Management

There are also protocols that improve energy efficiency and overall performance of the network by managing the topology of the network. SPAN [9] and GAF [29] are the most popular protocols in this class.

SPAN is an energy-efficient coordination algorithm for topology maintenance in ad-hoc wireless networks. It is based on the observation that a dense sensor network can work with only a part of its nodes being active. This makes it possible to prolong

the network lifetime while maintaining its functionality by carefully choosing the active nodes. SPAN is a distributed, randomized algorithm where nodes make local decisions on whether to sleep or to join a forwarding backbone. It elects coordinators from all nodes in the network, which stay awake continuously and perform multi-hop packet routing within the network while other nodes remain active in power-saving mode and check if they should wake up and become a coordinator. SPAN ensures that sufficient numbers of coordinators are chosen so that every node falls under the range of at least one coordinator. The coordinators keep changing to ensure that all nodes share the task of providing connectivity equally.

GAF also works on a similar principle, that when there is significant redundancy in an ad-hoc network, multiple paths exist between nodes. Hence, a few nodes can be powered off while still maintaining connectivity. GAF identifies equivalent nodes for routing based on location information and turns off unnecessary nodes. Both these approaches however suffer similar disadvantages as the protocols that perform selective reporting based on spatial-correlation. Moreover, they fail to utilize the network's full potential.

2.6 Congestion Control and Avoidance

In WSNs, congestion and contention are very closely related to each other. Each one of these problems could potentially trigger the other. For example, nodes not being able to access the channel due to contention will cause packets to rapidly queue up at each node and hence can lead to congestion. Similarly, increased congestion in the network can trigger channel access issues because each node now has a greater number of

packets to transmit and hence attempts to access the channel more frequently. Thus, techniques used for congestion control are very relevant and worth considering while designing protocols that aim to reduce contention. However, it is important to note that congestion does not necessarily increase contention in a network, although the converse is true. For example, if a network operates in TDMA, there is no contention in the network, irrespective of the network load.

Event-to-sink reliable transport (ESRT) [3] tries to achieve reliable event detection using energy-efficient techniques and congestion control. It defines five states of congestion: (No Congestion, Low Reliability); (No Congestion, High Reliability); (Congestion, High Reliability); (Congestion, Low Reliability); and (Optimal Operating Region) and then tries to stay in the Optimal Operating Region state. If it falls out of this state, the reporting rate of the sensor nodes is adjusted so that it will return to the Optimal Operating Region state. The sensor node detects the current state of the network by using the buffer sizes and sets a congestion notification bit when congestion is detected. 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 [27] mainly uses 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 a back-pressure message to upstream nodes so that they can adjust their reporting rate accordingly.

Pump Slow, Fetch Quickly (PSFQ) [32] is a scalable and reliable transport protocol that distributes data from a source node by slowing down the data speed so that nodes in the network do not get overwhelmed with data at once. Also, any nodes

that experience data loss are able to fetch the missing pieces from the intermediate nodes that have the pieces of data. 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 it can forward it on to nodes that are trying to fetch any data that they did not receive. Once a gap is seen by a node it will go into the FETCH mode trying to retrieve the missing data fragment. A disadvantage with PSFQ is that its pump operation can be very slow and leads to a large delay.

CHAPTER 3

PROTOCOL FOR CONTENTION MITIGATION

3.1 Motivation and Design Considerations

The characteristics of event-driven WSNs combined with the nature of wireless medium pose unique challenges: Since the wireless medium is inherently broadcast in nature, any ongoing transmission interferes with any other transmission within communication range. This interference is intensified in event-driven WSNs, where the traffic is bursty in nature. Additionally, since event-driven WSNs are typically densely deployed for increased fault-tolerance, coverage and connectivity, several nodes try to access the channel at the same time during the occurrence of an event, resulting in severe contention for channel access. Contention increases collisions, which not only result in packet loss and decreased throughput, but also increase energy consumption and latency in the network. Collisions lead to back-off and piling up of packets in a node's transmissions queue, hampering its ability to transmit data. This results in network congestion and ultimately loss of event reports. Each collision also triggers a re-transmission, causing energy wastage and delayed event reporting.

The seriousness of the problem lies in the fact that contention could lead to a lot of other problems like loss of data, energy wastage, congestion, and delayed reporting, although the converse is not always true. For example, when operating in TDMA

mode, the nodes do not contend for channel access irrespective of the amount of congestion in the network. Given these challenges, it is still important for the base station to receive all the event reports, for confidently determining the occurrence and severity of the events. Thus, it is very important to manage the contention in the network, so that events can be reported reliably with as few collisions and re-transmissions as possible.

Although MAC protocols based on TDMA completely prevent contention, they perform very poorly in event-based WSNs. TDMA requires tight network-wide synchronization and is more appropriate where there is a continuous data flow by almost all the nodes in the network, and the amount of data each node has to send is comparable. On the contrary, event-based WSNs have multiple event centers and the traffic pattern is bursty in nature, with only a subset of nodes in the network having data to send at a time. Using TDMA or any such fixed scheduling mechanisms would not only waste bandwidth but are also hard to maintain due to problems such as clock drifts, control message overhead etc. On the other hand, pure contention-based protocols are known to fail, given the number of nodes that could potentially sense and try to report the events at the same time. It is hence essential to find an intermediate approach that captures the benefits of both contention-based protocols and TDMA-based protocols. The objective of this thesis is thus to avoid contention in the network thereby reducing collisions, improving packet delivery and energy efficiency. The proposed approach to mitigate contention is expected to improve the network performance of the WSN by gracefully managing how and when nodes access a shared medium, using a dynamic and cooperative transmission schedule among the sensor nodes.

3.1.1 Design Considerations

- **Spatially Correlated Contention:**

As discussed previously, event-driven WSNs have sensor nodes typically deployed in high density for obtaining high reliability in event sensing. When an event occurs, there may be a large number of nodes that 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 severe contention for channel access. This could lead to collisions, packet loss, energy wastage, congestion, and delayed reporting. This is a fundamental problem typically found in all event-driven WSNs and should be addressed gracefully. Not taking care of this issue would have a deterrent effect on the network's performance.

- **Node Failures:**

In sensor networks, nodes can fail for many different reasons, such as obstacles, hardware defects in the node, and harsh environmental conditions in which the node operates. Also, a drop in energy levels or any other unforeseen event may cause node failures. If a node fails, the packets that are to be sent through that node will be lost or corrupted. Additionally, if a failed node happens to be a source node, it may either not be able to generate event reports at all, or will generate faulty reports, resulting in false positives at the base station. Node failures are so typical that support for handling them should be an inherent part of the design: a network should be in a position to function reliably in spite of some failed nodes. Thus, as opposed to protocols like CCMAC [26] and SCMAC [33], which select only a subset

of the available nodes to report events, it is important that the network design uses the full potential of dense deployment in the network so that even if a few nodes fail or have faulty data, the base station can still reliably detect events.

- **Energy Constraint:**

WSNs work on low powered batteries and hence are energy constrained. It is thus important to reduce control messages and unnecessary transmissions in the network, so as to prolong the lifetime of the network. Event-driven WSNs typically experience severe contention for channel access, increasing the number of collisions. Each collision in-turn triggers a re-transmission to compensate for the loss of data due to the collision. Moreover, energy consumed by sensor nodes during transmission of messages is very high. Thus, increased number of re-transmissions leads to a lot of energy wastage in the network. In designing an energy-efficient protocol, the energy wastage due to collisions and re-transmissions must be considered and reduced. Also, the protocol design should use minimal or no control messages in order to further reduce interference and energy consumption in the network.

- **Scalability:**

As event-driven WSNs contain a very large number of sensor nodes, networks should be scalable enough to provide a high packet-delivery ratio. Additionally, the traffic pattern in these kind of networks is bursty and unpredictable in nature with several event centers. A protocol should thus be designed to dynamically adjust to varying levels of node densities, traffic load and contention in the network, without depending on prior knowledge or guess work of node distribution and traffic levels.

3.2 Assumptions

For simplifying the explanation, we make the following assumptions:

- The network is densely deployed and its outer boundary is identified.
- All the nodes know the status of their one-hop neighborhood nodes by local broadcast mechanisms.

3.3 Contention Mitigation

3.3.1 Basic Approach

The proposed approach exploits the broadcast nature of the wireless medium to benefit the network. The broadcast nature of the medium empowers nodes to hear the transmissions of all nodes in its vicinity. Nodes can use this property to dynamically overhear transmissions from other competing nodes in their transmission range and thus know who has data to send. The nodes then utilize the order of neighborhood transmissions to establish a dynamic schedule for themselves without any synchronization or explicit message exchange. Since the broadcast medium makes overhearing mutual in nature, the schedule established at each node does not conflict with the schedule at other nodes. Nodes that are not participating in the data transmission are implicitly not included in the schedule and hence do not waste any bandwidth as opposed to TDMA-based approaches, where all nodes get their share of bandwidth, whether or not they have data to send. Moreover, nodes operating with this protocol can co-exist with other nodes following a CSMA-only approach. This captures the benefits of both TDMA and CSMA-based protocols, while mitigating

their disadvantages.

The other important aspect of the approach is about data being forwarded to the base station in a way that compliments the transmission schedules and mitigates the contention among the forwarder nodes located on route to the base station from the event center. In a densely deployed network, there are several nodes within transmission range of each other that contend to gain channel access. As the number of active forwarder nodes increase, the contention for channel access between them also increases, especially if the forwarders happen to be in transmission range of each other. Additionally, the transmission scheduling mechanism described above is more efficient when the number of neighboring nodes that a particular node has to wait for before it can access the channel is reduced. An increase in the nodes participating in forwarding data will have a negative impact on the scheduling mechanism, by increasing the period a node will have to wait before it can access the channel. Thus, a data forwarding mechanism that supports the transmission schedules should have fewer nodes close to each other participate in forwarding data.

One can reduce the number of forwarders taking part in data forwarding by dedicating a single forwarder (parent) node for multiple nodes (children). Although this reduces the number of forwarding nodes, it results in congested nodes/bottle necks in the network. The energy at these bottle-neck nodes also depletes very quickly, as each of these forwarder nodes need to handle more transmissions. Thus, the proposed protocol takes an intermediate approach; the potential bottle necks are taken care of by allowing nodes to have multiple forwarders (if available), but at the same time, the number of forwarders participating in data transmission are reduced

by restricting the number of forwarders each node can actually use to forward data. Thus, the proposed approach implements a data forwarding mechanism where each child node has the property of using only a subset of the total number of parent nodes available to it. This subset will be chosen such that there is increased overlap in the parent nodes chosen by all child nodes at each hop level in the network. This helps reduce the contention between forwarder nodes and the waiting period of a node using a transmission schedule, while avoiding potential congested spots in the network.

3.3.2 Dynamic Transmission Scheduling

In event-driven WSNs, due to the unpredictability of the occurrence of events, traffic is usually bursty with a sudden surge of packets flowing across the network when an event occurs. Unlike networks targeted towards continuous-monitoring applications, where all nodes in the network have comparable amount of data to send at all times, only a specific set of nodes that detect an event or participate in forwarding event reports have data to send in an event-driven network. Strict scheduling mechanisms like TDMA inherently give all nodes in the network an equal share of bandwidth whether or not they have data to send. Hence, if a node does not have any data to send during its turn, the bandwidth allocated to it during its time slot is wasted. This increases the latency incurred at nodes that have data to send. Additionally, TDMA-like mechanisms require tight synchronization among all the nodes and suffer from issues like clock-drifts and frequent control message overhead. Hence, any strict scheduling mechanism like TDMA does not work well when applied to event-driven WSNs due to the high density of nodes in multiple event centers across the network, not all of which having event reports to send.

On the other hand, although pure contention-based protocols using CSMA overcome the disadvantages of TDMA-like protocols, they are still not suitable for event-driven WSNs due to a couple of reasons: Firstly, these protocols depend on a back-off mechanism for collision-avoidance, which is known to fail specially in densely deployed networks, where there are several nodes within transmission range of each other attempting to access the channel at the same time. Secondly, in an event-detection application, event reports from all nodes in the network are equally important and the order in which nodes gain access to the channel to transmit these reports does not matter. What matters most is that all the event reports ultimately reach the base station reliably with as few collisions and re-transmissions as possible. Thus, at a more fundamental level, it makes a lot more sense for nodes to cooperatively access the channel, giving each other chance to transmit, rather than merely contending or competing with the other nodes for gaining access to the channel. Thus, there is a need for a mechanism that is capable of dynamically scheduling transmissions of only those nodes that have data to send, without any synchronization or control message overhead, thereby reducing the latency incurred due to waiting for nodes that have no data to send in TDMA-like protocols. The proposed mechanism achieves this goal by exploiting the broadcast nature of wireless medium combined with CSMA principles, capturing the benefits of both TDMA and CSMA approaches and mitigating their cons.

Schedule Establishment

The proposed dynamic transmission scheduling mechanism works by establishing a transmission schedule at each node in the network. Once a schedule is established at a node, it also dynamically updates the schedule, as new neighboring nodes begin

to transmit or the transmission needs of the nodes existing in the schedule evolve. Additionally, depending on the transmissions overheard, it determines if the node is free to transmit or has to wait. Initially, the nodes contend for channel access using traditional CSMA, with RTS/CTS. During this process, if a node overhears a transmission, it records the source MAC address of the overheard transmission in a schedule table. Once a node successfully gains access to the channel for the first time, we call it a “scheduled” node. The property of a scheduled node is that after each transmission, it waits for all nodes in its own schedule table to access the channel once, before it attempts to transmit again. To achieve this, when a node overhears from another node that happens to already exist in its schedule table, it decrements the total number of nodes it still needs to wait for (*waitCounter*) before it is free to transmit. Once the *waitCounter* reaches 0, the node is free to transmit.

Since overhearing is mutual in nature, if say node *A* and node *B* are scheduled, i.e, they have both gained access to the channel at least once, and if a node *A* has node *B* in its schedule table, it is implicit that *B* also has *A* in its schedule table. Having node *A* in node *B*’s schedule table implies that *A* cooperates with *B*. This way, scheduled nodes cooperate with other scheduled nodes. Hence, at any given instance of time, a node only contends with nodes that are not “scheduled” yet, mitigating contention. As time progresses, the network operation moves closer to a stable state where each node cooperates with all nodes having data to send.

Dynamic Schedule Maintenance

As discussed above, a node is scheduled when it gains access to the channel for the first time. Once scheduled, it waits for all the nodes that are in its schedule table to

transmit once, before it attempts to transmit again. However, if a node hears from new neighboring nodes that are not currently in its schedule table, then its schedule table should be updated, so that it waits for these newly learned nodes starting from the next time it gets to transmit. Thus, while a node is waiting, if it happens to hear from a new node not in its schedule table, it adds the new node's MAC address to the table, so that next time, it is deemed to wait for these new nodes along with the nodes that already existed in the table.

We now know that once a node gets to transmit, it waits to hear from all the nodes in its schedule table before transmitting again. However, there might be a case where one or more nodes in its schedule table has no more data to transmit at that moment, leading to infinite waiting. We do not want to wait until a timeout is triggered for these cases, as they may occur frequently. The dynamic scheduling mechanism at a node should thus be in a position to detect such nodes that do not have any data to send and avoid waiting for them. To achieve this, we modify the structure of the schedule table by adding a column containing the number of packets a particular node has queued at the moment. To keep this value up-to-date in each node's schedule table, we add a new field in the 802.11 MAC packet header that reflects the number of packets a particular node has queued at the moment, and have all nodes initialize this field with their latest queue sizes every time they transmit a packet. The queue size is calculated by each node as the sum of the number of packets the node has generated (applicable only for source nodes) and the number of packets that it has received from other nodes for forwarding. Thus, each time a node overhears a transmission from another node, it precisely knows how many more packets the node has left to send at that moment. If at any point of time this value

is 0 for a specific node, instead of waiting for it, we decrease the *waitCounter* as if that node has finished transmitting. The modified structure of the schedule table is shown in Figure 3.1.

Source MAC Address	Current Queue Size

Figure 3.1: Structure of Schedule Table

The following illustrates how the dynamic transmission schedule establishment and maintenance work in detail. For ease of explanation, we consider a simple network with 3 nodes *A*, *B* and *C*, which are all in transmission range of each other; each node can overhear the transmissions of all the other nodes in the network. Additionally, suppose that each node has 2 packets to send. Node *A* and *B* use *C* as their forwarder and *C* transmits to the base station. Initially each of the node's schedule table is empty as shown in Figure 3.2. As nodes start to transmit event reports, Figure 3.3 shows how the schedule tables at nodes *A*, *B* and *C* respectively get updated so as to achieve a dynamic transmission schedule.

Suppose that node *A* happens to be the first node to gain access to the channel and transmit. Since node *B* and *C* hear this transmission, they add *A* to their corresponding tables, along with the current queue size of *A*, which is extracted from the MAC header of the packet transmitted by *A*. Since *A* has already transmitted a packet, its queue size is now 1. Now suppose that *B* gains access to the channel and transmits its first packet. This is heard by *A* and *C* and they update their

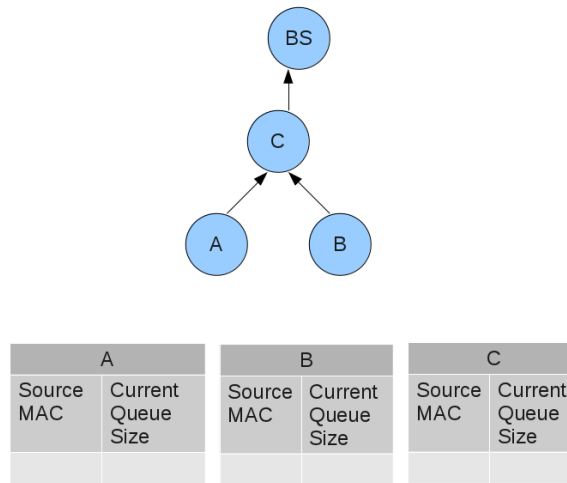


Figure 3.2: Initial Configuration for a Sample Network

transmission tables accordingly. Note that since B 's schedule table contains A , it will wait for A to transmit before it attempts to transmit again. Lets say C happens to be the next node to transmit its first packet. Since C acts as forwarder for A and B , the total number of packets it still needs to send are 3. This is reflected at the schedule tables of A and B . Also, since C has both A and B in its schedule table, it will now need to hear from both of them before it is clear to transmit again.

At this point, Node B is still on hold because it is waiting for A . Hence, node A is the only node that is free to send and hence transmits, leaving it no more packets to send. Since node B and C hear this transmission, they learn that the current queue size of A is 0 and update their tables accordingly. Previously B was waiting for A , but after A 's recent transmission, it is time for B to send; whereas, C is still waiting for B . Hence, B transmits its final packet. When C overhears this transmission, it

A transmits. It still has 1 packet queued

Source MAC	Current Queue Size	Source MAC	Current Queue Size	Source MAC	Source MAC
		A	1	A	1

B transmits. It still has 1 packet queued

Source MAC	Current Queue Size	Source MAC	Current Queue Size	Source MAC	Current Queue Size
B	1	A	1	A	1
				B	1

B is waiting for A. C is waiting for A and B. A is the only node clear to transmit and hence transmits.

Source MAC	Current Queue Size	Source MAC	Current Queue Size	Source MAC	Current Queue Size
B	1	A	1	A	1
C	3	C	3	B	1

C transmits. It still has 3 packets queued

Source MAC	Current Queue Size	Source MAC	Current Queue Size	Source MAC	Current Queue Size
B	1	A	0	A	0
C	3	C	3	B	1

B is now clear to send. C is still waiting for B.

Source MAC	Current Queue Size	Source MAC	Current Queue Size	Source MAC	Current Queue Size
B	0	A	0	A	0
C	3	C	3	B	0

C is now clear to send the rest of the packets as the queue sizes of A and B reached 0

Figure 3.3: Schedule establishment and dynamic maintenance

knows that it is free to send now and sends its second packet.

According to the scheduling mechanism, *C* should now start waiting for *A* and *B* before it can transmit its third packet. However, from the current queue size values of *A* and *B*, it knows that they do not have any data to send at that moment. Hence, without any further waiting, it can transmit its remaining packets. This way, nodes

wait for other nodes only if they have any data to send, reducing the wait time. The procedures used for scheduled data transmission and dynamic maintenance of schedule tables at each node are presented below. These procedures are executed for each packet that is to be transmitted or is received.

Procedure 1 Dynamic Schedule Table Maintenance

```

pkt ← Incoming Packet
if pkt's src MAC exists in the schedule table then
  if waitCounter ≠ 0 then
    – – waitCounter
    update value of current queue size for pkt's source node in the schedule table
  else
    callback transmit function for the first packet in the backlog buffer
  end if
else
  Add src MAC address of pkt to the schedule table
  Initialize value of the current queue size for the new table entry
end if
if pkt is destined to self then
  update queue size of self to account for pkt
end if

```

Procedure 2 Scheduled data transmission

```

pkt ← Packet to be transmitted
if waitCounter = 0 then
  Send pkt to be sent immediately
  Reset waitCounter to current schedule table size
else
  Add pkt to backlog buffer and wait
end if

```

3.3.3 Data Gathering with Increased Forwarder Set Overlap

WSNs consist of a group of sensor nodes that collectively transmit data to a common gateway, also called base station. Typically, not all nodes in the network are within

transmission range to the base station. If a node cannot reach the base station directly, its data is forwarded recursively through a set of intermediate nodes, until it finally reaches a node that is 1 hop away from the base station, which then sends the data to the base station. Therefore, in any given WSN, every node that is not a single hop away from the base station needs to configure forwarder(s) that it can use to transmit its data.

In wireless sensor networks, data should be forwarded to the base station in a way that mitigates the contention among the forwarder nodes located on route to the base station from the event center. In general, at a given hop level, as the number of nodes being actively used as forwarders increase, the contention between them also increases. This is corroborated in event-driven WSNs by the fact that there are typically several nodes within transmission range of each other at any given hop level, contending for channel access. Even though the dynamic scheduling mechanism works on mitigating the contention between these forwarders, a good data forwarding mechanism that compliments the scheduling mechanism is beneficial.

At any given hop level, the proposed data forwarding mechanism aims to reduce the number of forwarder nodes (parents) being actively used by the nodes one hop below (children), thereby mitigating contention between the forwarder nodes. One can reduce the number of forwarders taking part in data forwarding by dedicating a single parent node for multiple child nodes. Although this may reduce the number of forwarding nodes, it results in congested nodes/bottle necks in the network. The energy at these bottle-neck nodes also depletes very quickly, as each of these forwarder nodes need to handle more transmissions. Thus, the proposed protocol takes

an intermediate approach, where nodes are allowed to have multiple forwarders (if available), but will only use a subset of the total available forwarders. This subset will be chosen such that there is increased overlap of parent nodes among all the child nodes, so as to reduce the effective number of forwarders used at each corresponding hop level. The rationale behind this approach to data gathering is illustrated below.

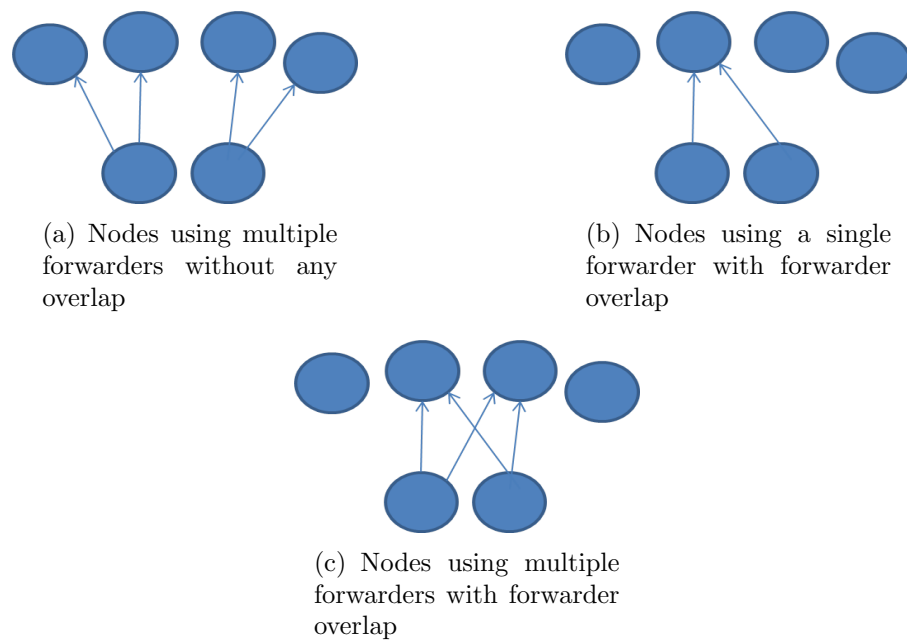


Figure 3.4: Comparison of Forwarder Configurations

Figure 3.4 shows nodes at two consecutive hop levels say ' i ' and ' $i-1$ ', in a sample WSN. For ease of explanation, let's assume again that all nodes shown are within transmission range of each other. Thus, the transmissions of each node can potentially interfere with that of the others. Let us consider the nodes in the i th hop level to be child nodes and the ones in the $(i-1)$ th hop level to be their potential parent nodes. Figure 3.4(a) shows the scenario where child nodes are allowed to have multiple

forwarders, but in the worst case, their forwarder sets have no overlap. In this case, since all nodes in the $(i-1)$ th hop level are being actively used for data forwarding, they will all need to contend among themselves for channel access. The amount of contention in the first scenario can be mitigated, by having the nodes use only a single overlapping forwarder, as shown in Figure 3.4(b). Although this effectively mitigates the contention at level $(i-1)$ compared to the first case where there were four nodes contending, the sole forwarder in this case potentially becomes a bottle neck and increases congestion in the network. Moreover, the energy of this forwarder depletes very quickly, since it needs to forward data from both its child nodes alone.

The proposed data forwarding approach uses an intermediate approach shown in Figure 3.4(c) where nodes are allowed to have multiple forwarders with increased overlap in their forwarder sets. This mitigates the contention at level $(i-1)$ because now there are only two active forwarders as opposed to the four active forwarders in the first case. Additionally, since nodes have multiple potential forwarders, the issue of a sole forwarder acting as a potential bottle neck is mitigated. By increasing the overlap in the forwarder sets of child nodes, the proposed data forwarding mechanism also helps restrict the area that is prone to interference, as opposed to increasing the radius of the zone that is prone to interference and contention. Within this contained zone, the dynamic transmission scheduling mechanism works more effectively to mitigate contention for channel access.

The following describes the heuristic algorithm used by each node in the network to configure its forwarder set, such that there is increased overlap in forwarder sets among all nodes in its hop level. The forwarder configuration happens in a hop-by-hop

fashion, with the base station initiating the process. At each hop level, starting with the base station, parent nodes broadcast a *PARENT – HELLO* packet to their down stream neighbors. Then, they start a *CHILD – ACK* timer during which they wait for *CHILD – ACK* messages from the child nodes. For each *CHILD – ACK* message received, a parent node increments its value of *numChildren*, indicating the number of children it could potentially support. Once the *CHILD – ACK* timer expires, a parent node broadcasts a *CHILD – COUNT* packet containing its latest value of *numChildren* to all the child nodes.

After sending a *CHILD – ACK* to its parents, a child node starts a *CHILD – COUNT* timer, during which it waits for all the parents to respond with their respective values of *numChildren* through a *CHILD – COUNT* packet. Out of all the forwarders it hears from, a child node only uses those forwarders with the top number of children. This way a child node achieves increased overlap with the forwarder sets of other child nodes at the same hop level. The total number of forwarders it selects is configured during initialization. Initially, the *finalForwarderCount* is set to this number for all nodes, assuming each node has those many forwarders available. However, based on how many parent nodes a child actually hears from and the upper limit on the number of forwarders a node is allowed to use, this value is updated. By the time forwarder configuration is complete, the value of *finalForwarderCount* at a particular node reflects the true number of forwarders it is allowed to use.

Procedure 3 Forwarder Configuration with Increased Hop-by-hop Forwarder Set Overlap

Output: *baseStationID* = MAC address of the base station

numChildren \leftarrow 0

hopdistance \leftarrow -1

potentialForwarders \leftarrow 0

finalForwarderCount \leftarrow DESIRED-NUM-FORWARDERS

for $i = 1 \rightarrow$ *finalForwarderCount* **do**

forwarderSet[i].*saddr* \leftarrow -1

forwarderSet[i].*childcount* \leftarrow -1

end for{Initialization Complete}

if *baseStationID* = MAC address **then**

hopdistance \leftarrow 0

 Broadcast PARENT-HELLO msg to all downstream neighbors

 start CHILD-ACK timer

end if

loop

if CHILD-ACK timer expired **then**

 Broadcast CHILD-COUNT msg to all downstream neighbors with *childcount*
 set to *numChildren*

end if

if CHILD-COUNT timer expired **then**

finalForwarderCount \leftarrow *potentialForwarders*

 Broadcast PARENT-HELLO msg to all downstream neighbors

 start CHILD-ACK-timer

end if

msg \leftarrow *receivePacket*()

if *msg.Type* = PARENT-HELLO **then**

hopdistance = *msg.hopdistance* + 1

 Reply with CHILD-ACK

 re-start CHILD-COUNT timer

else if *msg.Type* = CHILD-ACK **then**

numChildren \leftarrow *numChildren* + 1

else if *msg.Type* = CHILD-COUNT **then**

potentialForwarders \leftarrow *potentialForwarders* + 1

 sort *forwarderSet*[] array based on *childcounts* of each forwarder in the array

if *forwarderSet*[1].*childcount* < *msg.childcount* **then**

forwarderSet[1].*childcount* = *msg.childcount*

forwarderSet[1].*saddr* = *msg.saddr*()

end if

end if

end loop

CHAPTER 4

PERFORMANCE EVALUATION

To evaluate the performance, the proposed Dynamic Transmission Scheduling MAC (DTS-MAC) is implemented in the NS2 simulator [1]. Extensive experiments were conducted in order to test the performance of the transmission scheduling based approach. First, we compare the performance of our protocol with SC-MAC [33], because it handles spatially-correlated contention and outperforms CC-MAC [26] and other protocols like S-MAC [31]. We provide a comparison of our approach with SC-MAC in terms of packet delivery ratio, energy consumption, collision rate, and average latency. Secondly, we investigate the overall performance of the proposed protocol under different network densities in order to prove that DTS-MAC is scalable and works in a realistic environment. To study the performance gains due to the dynamic transmission scheduling approach, we compare DTS-MAC with a base variant that does not schedule transmissions.

4.1 Performance Metrics

The following are the metrics used to compare and evaluate the performance of the protocol:

Packet Delivery Ratio

It is the ratio of the total number of packets generated to the total number of packets delivered to the base station. This shows the performance of the protocol and illustrates its level of reliability.

Energy Efficiency

To identify the energy efficiency of the proposed protocol, the average energy consumed by each node in the network is calculated. The lower this value, better is the energy efficiency and life time of the overall network.

Average Delay

Average delay is measured to identify the latency incurred while forwarding packets to the base station. It is calculated as the average time taken by each packet to reach from its source to the base station.

Collision Rate

It is the ratio of the number of packets dropped due to collisions during medium access and total number packets generated. This gives a good insight into the number of collisions and re-transmissions that took place in the network, with respect to changing traffic loads in the network.

4.2 Comparison with SC-MAC

In order to compare performance of our protocol with SC-MAC, simulations were run using the simulation parameters of SC-MAC mentioned in Table 4.1. All the

Table 4.1: Simulation Parameters for Comparison with SC-MAC

Parameter	Value
Area	500m x 500m
Transmission Radius	100m
Total Number of Nodes	50
Sources	15
Packet Size	100 <i>bytes</i>
Transmit Power	24.75mW
Receive Power	13.5mW
Idle Power	13.5mW
Sleeping Power	15uW

nodes are randomly deployed in a 500m x 500m sensor field, one of which is used as base station. Each packet has a fixed size of 100 bytes. We select 15 source nodes randomly all of which report event information to the sink with a 0.5s packet interval. SC-MAC investigates the effect of traffic load by varying the reporting period of the sensor nodes. The reporting period determines the period for which a source node generates packets for each occurrence of an event. We also vary the reporting period of the sensor nodes in order to compare our protocol with SC-MAC. For all the experiments, data is collected from an average of 20 independent runs.

Figure 4.1 shows a comparison of the performance of our protocol in terms of packet delivery ratio. As seen from the graph, DTS-MAC outperforms SC-MAC with about 23% improvement in the packet delivery ratio. SC-MAC reduces contention by selecting a subset of nodes that detect and generate event reports and thus has about 25% lower traffic volume compared to DTS-MAC. In spite of this higher volume of traffic in DTS-MAC compared to SC-MAC, we see a good improvement in the packet delivery ratio. The improvement can mainly be attributed to the significant reduction in the number of collisions when transmissions are dynamically scheduled

in DTS-MAC. Although SC-MAC reduces contention among the source nodes by allowing only a subset of source nodes to generate event reports, there is severe contention between the several intermediate nodes that actively help forward the event reports to the base station. SC-MAC also has a high number of packet drops during the First Contention Phase, where all source nodes with event information contend and representative nodes for each region are iteratively selected. DTS-MAC on the other hand uses a more holistic approach where each node in the network can co-operate with and establish a dynamic transmission schedule relative to its neighboring nodes, thereby reducing contention and the number of collisions throughout the network. Additionally, the packet delivery ratio is relatively stable in DTS-MAC with varying traffic loads, as opposed to SC-MAC, where there is noticeable drop at higher reporting periods.

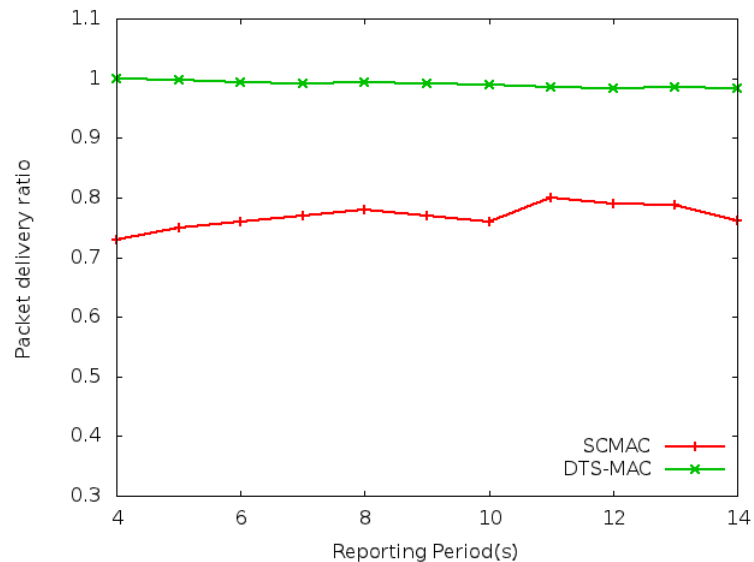


Figure 4.1: Packet Delivery Ratio

The effect of dynamically scheduling the transmissions as seen in Figure 4.2 depicts collision rate in the network. Since nodes using DTS-MAC cooperate with each other by scheduling their transmissions, the contention between them is significantly reduced in spite of higher traffic volumes and we observe a 42% decrease in the number of collisions compared to SC-MAC, even at the highest reporting period used.

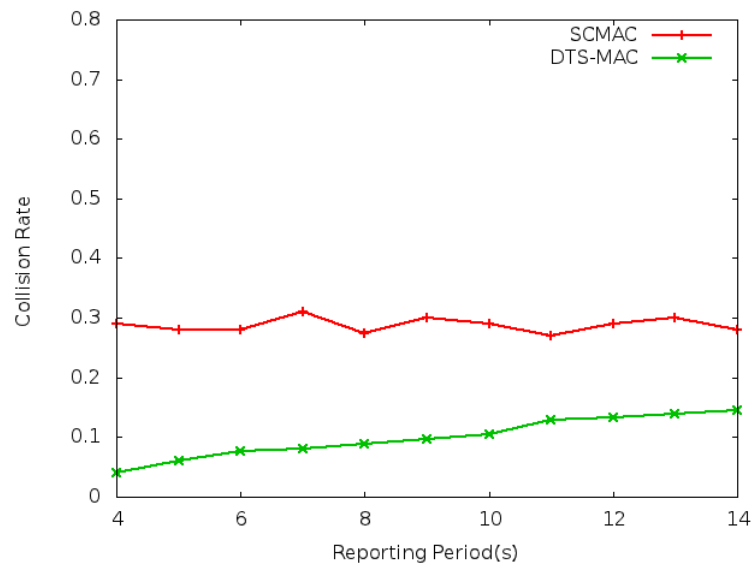


Figure 4.2: Collision Rate

As a result of a decrease in the number of collisions, the number of re-transmissions and hence the energy consumption per node in the network is reduced with DTS-MAC. Figure 4.3 shows a comparison of DTS-MAC and SC-MAC with respect to energy consumed per node in the network. Although SC-MAC conserves energy by selecting only a subset of source nodes that generate and transmit event reports, energy is depleted due to a fairly complex representative node selection process. DTS-MAC on the other hand reduces energy consumption by efficiently coordinating

access to the medium, reducing the number of re-transmissions in the network. DTS-MAC further reduces energy consumption of the idle forwarder nodes that result from increasing forwarder set overlap among child nodes during initial forwarder configuration, by turning off their radios.

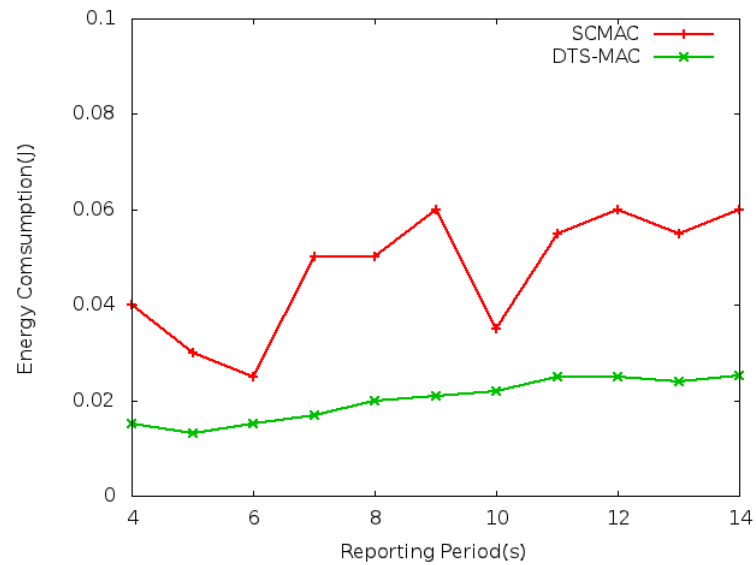


Figure 4.3: Energy Consumption

Figure 4.4 shows the average latency incurred in each protocol. With DTS-MAC, nodes inherently wait for each other to avoid collisions and contention in the network. Due to this overhead, the average latency of DTS-MAC increases with increase in reporting period used. However, due to higher number of collisions and re-transmissions in SC-MAC, combined with its repeated representative node selection, the latency incurred due to DTS-MAC turns out to be lower than that of SC-MAC under reasonable traffic loads. The difference in the average delay however decreases as the traffic increases, and the delay of DTS-MAC could potentially cross

over that of SC-MAC under sufficiently high traffic loads.

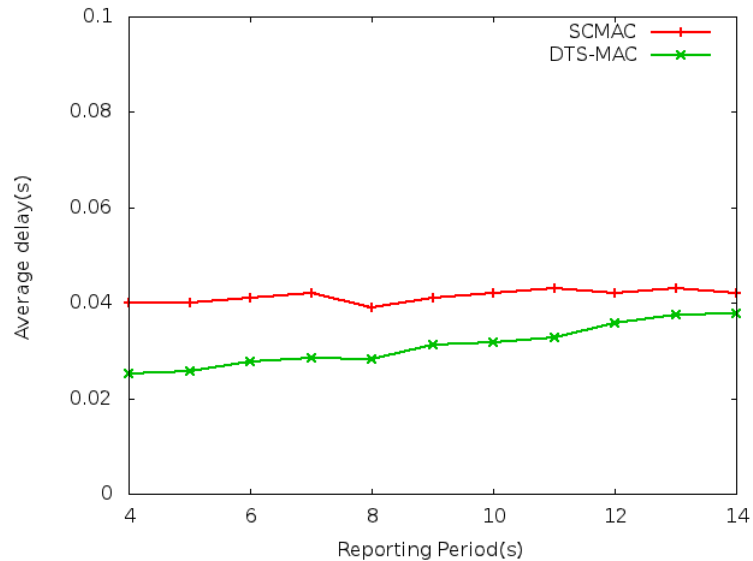


Figure 4.4: Average Delay

4.3 Protocol Evaluation

To investigate the scalability and performance of our protocol, we compare it with another variation of the protocol, which does not use dynamic scheduling. As the simulation parameters used for comparison with SC-MAC do not sufficiently stress the network to test for scalability and robustness, simulations were run using the a different set of parameters mentioned in Table 4.2 [18] that are commonly used for evaluating wireless sensor network protocols. For data packets, Constant Bit Rate (CBR) traffic is used. We evaluate the scalability of the protocol using two different network densities, 50 and 100 nodes with 15 source nodes each and a packet interval varying between 0.1s to 1.0s. The packet size used is 512 bytes. In all the experiments,

Table 4.2: Simulation Parameters for Protocol Evaluation

Parameter	Value
Area	1000m x 1000m
Transmission Radius	250m
Total Number of Nodes	50 – 100
Sources	15
Packet Size	512 <i>bytes</i>
Packet rate	0.1(0.5KB/s) – 1(5KB/s)
Transmit Power	0.01488 W
Receive Power	0.01250 W
Idle Power	0.01236 W
Sleep power	0.000016 W

each data point calculated is an average of 20 independent runs with a simulation time of 400s each, on random network deployments.

As seen in Figure 4.5, when the traffic load is increased, the packet delivery ratio drops significantly if no scheduling is done. In contrast, even at high traffic loads, we notice that the packet delivery is relatively stable with scheduling. Moreover, we see that increasing the network density does not affect packet delivery when scheduling is implemented, whereas there is noticeable drop in the packet delivery ratio with increased network density, if the transmissions are not scheduled. This is mainly because, when the transmissions are scheduled, the contention between the nodes and hence the number of collisions that occur is greatly reduced.

When collisions occur, there is a limit to the number of times a node re-transmits a packet before it gives up and drops the packet. Thus it is essential to reduce collisions in the network, so that nodes are able to transmit packets successfully with very few or no re-transmissions. Dynamic scheduling of the transmissions helps achieve this

very goal, as nodes cooperate with each other, reducing contention and the chance of collisions. This can be observed in Figure 4.6, which shows how the collision rate varies with and without scheduling. When the transmissions are dynamically scheduled, the packet collision rate clearly drops significantly.

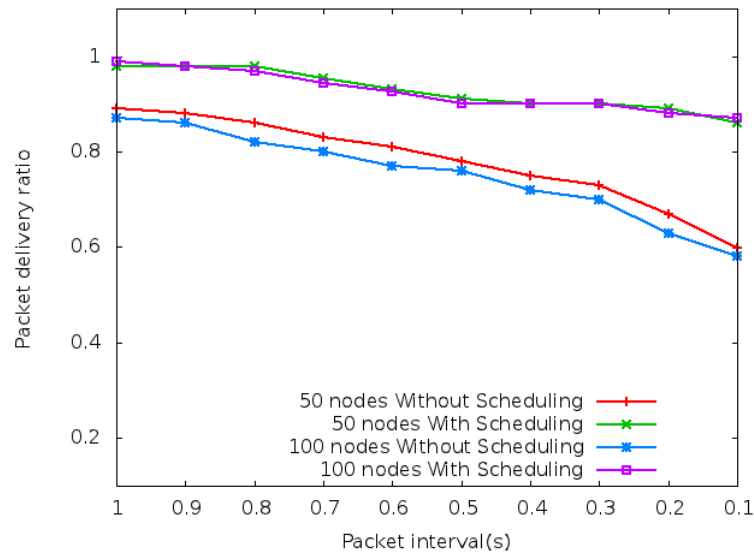


Figure 4.5: Packet Delivery Ratio

Figure 4.7 shows the energy efficiency of our protocol. With transmission scheduling enabled, we see that the power consumption of the network is reduced. This is because when transmissions are scheduled, the number of collisions and the associated re-transmissions, which are a major cause for energy wastage in the network, are greatly reduced. We also observe that when node density is increased for each variation of the protocol, the increase in power consumption is very small, because of the forwarder set overlap among the child nodes. The increased overlap in forwarder sets helps decrease the number of nodes that act as forwarders and actively participate

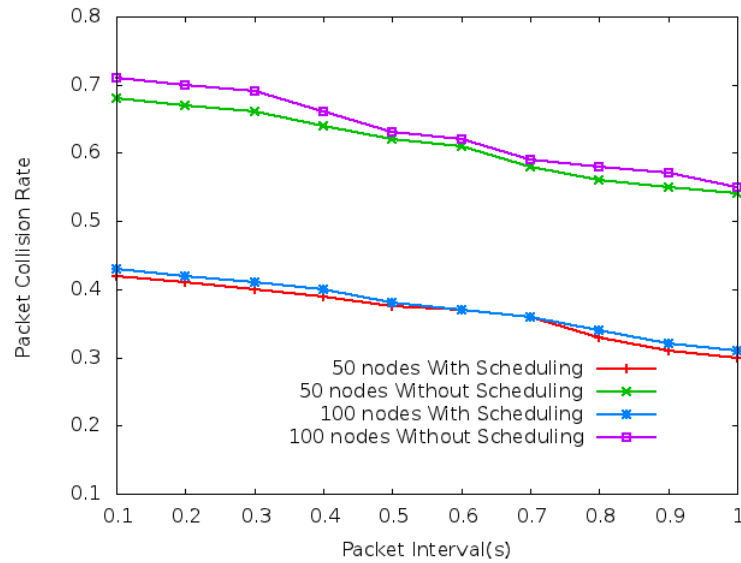


Figure 4.6: Packet Collision Rate

in data transmission. The radios of the resulting idle nodes can then be turned off to save power and maintain the longevity of the network even with higher node densities.

As seen in Figure 4.8, when transmissions are not scheduled, the delay shoots up at high traffic loads. With scheduling, since nodes wait for each other, the average delay calculated over all packet intervals is slightly higher than when no scheduling is done. However, the delay is relatively stable due to the reduction in the number of collisions. Moreover since nodes wait only for other nodes that have data to send, the increase in delay when transmissions are scheduled is counter-intuitively small.

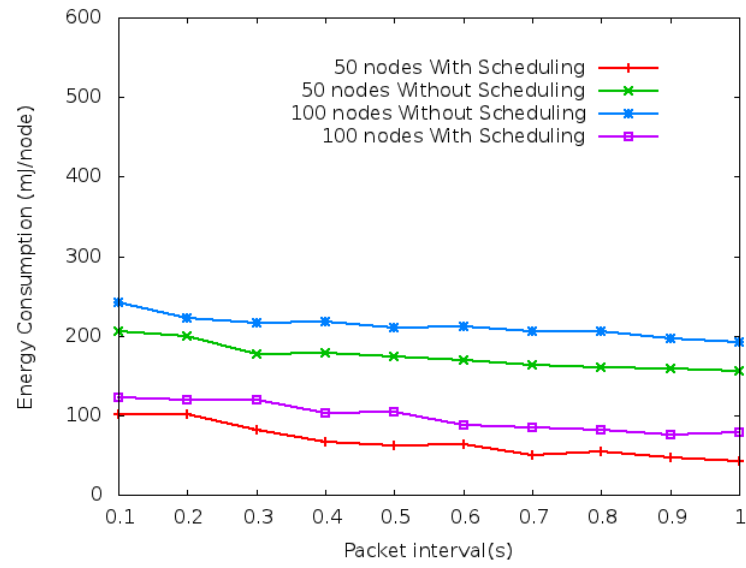


Figure 4.7: Energy Efficiency

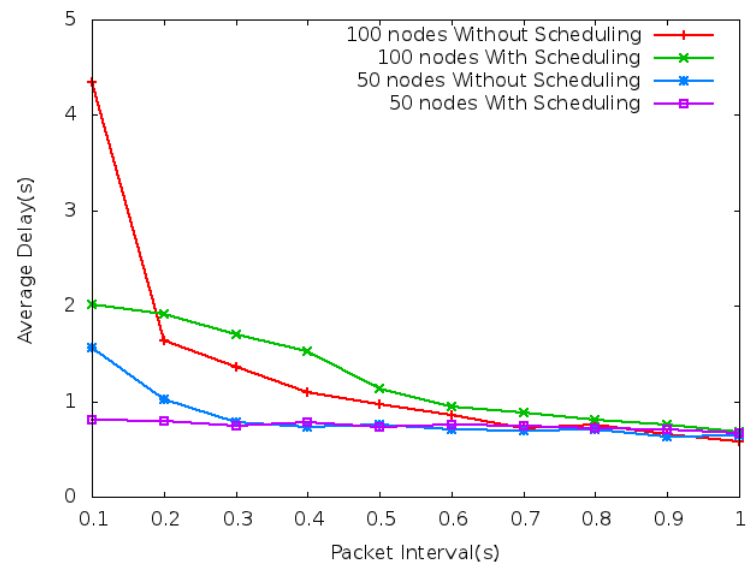


Figure 4.8: Average Delay

CHAPTER 5

CONCLUSION

Wireless sensor networks heavily rely on a dense deployment of sensor nodes in order to ease deployment, increase fault-tolerance and network coverage, so that events do not go undetected. A dense deployment however results in several sensor nodes close to each other detecting and transmitting event reports at almost the same time, resulting in severe contention for channel access. Channel contention is a serious problem in WSNs, which increases the likelihood of collisions in the network. Collisions in-turn lead to re-transmissions, energy depletion, and ultimately loss of event reports. Event-detection applications demand several event reports from ideally unique sensor nodes, in order to support fault tolerance and detect events with high confidence. Existing protocols like SC-MAC handle spatially correlated contention by selecting only a subset of source nodes that detect and report events, defeating the purpose of dense deployment. They also reduce the sink's ability to confidently detect events as it receives reports only from a subset of source nodes that detect an event. Hence, there is a need for a protocol that can reduce contention and improve performance of the network even under high traffic loads, when the network fully functional.

In this thesis, we propose DTS-MAC that reduces contention and improves packet delivery by dynamically scheduling the transmissions and by reducing the number

of active forwarders in the network. Unlike TDMA-based approaches that require tight network-wide synchronization, nodes using DTS-MAC exploit the broadcast nature of the wireless medium to maintain the schedules dynamically, without the need for synchronization and message exchange. By scheduling the transmissions, we reduce contention and the number of collisions in the network, thereby improving packet delivery ratio. As collisions reduce, the number of re-transmissions and hence the energy wastage also goes down. Simulation results show that the proposed protocol significantly reduces the collision rate and improves packet delivery and energy efficiency in the network.

BIBLIOGRAPHY

- [1] Ns2 - network simulator, <http://www.isi.edu/nsnam/ns>.
- [2] 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.
- [3] O.B. Akan and I.F. Akyildiz. Event-to-sink reliable transport in wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 13(5):1003 – 1016, oct. 2005.
- [4] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102 – 114, aug 2002.
- [5] M. Al-Mamun, G.C. Karmakar, and J. Kamruzzaman. Qos-centric collision window shaping for CSMA-CA mac protocol. In *Global Telecommunications Conference (GLOBECOM 2010)*, 2010 *IEEE*, pages 1 –6, dec. 2010.
- [6] Muneeb Ali, Umar Saif, Adam Dunkels, Thiemo Voigt, Kay Römer, Koen Langendoen, Joseph Polastre, and Zartash Afzal Uzmi. Medium access control issues in sensor networks. *SIGCOMM Comput. Commun. Rev.*, 36(2):33–36, April 2006.
- [7] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung. MAC essentials for wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 12(2):222 –248, quarter 2010.
- [8] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: a media access protocol for wireless lan's. In *Proceedings of the conference on Communications architectures, protocols and applications*, SIGCOMM '94, pages 212–225, New York, NY, USA, 1994. ACM.
- [9] 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.

- [10] Ilker Demirkol and Cem Ersoy. Energy and delay optimized contention for wireless sensor networks. *Comput. Netw.*, 53(12):2106–2119, August 2009.
- [11] 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 *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pages 117–126, march 2005.
- [12] C.C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Peiris. Wisenet: an ultralow-power wireless sensor network solution. *Computer*, 37(8):62–70, aug. 2004.
- [13] Kyle Jamieson, Hari Balakrishnan, and Y.C. Tay. Sift: a MAC Protocol for Event-Driven Wireless Sensor Networks. In *Third European Workshop on Wireless Sensor Networks (EWSN)*, Zurich, Switzerland, February 2006.
- [14] Abtin Keshavarzian, Huang Lee, and Lakshmi Venkatraman. Wakeup scheduling in wireless sensor networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '06*, pages 322–333, New York, NY, USA, 2006. ACM.
- [15] Anis Koubaa, Mário Alves, Bilel Nefzi, and Ye-Qiong Song. Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks. In *Proceedings of the Workshop of Real-Time Networks (RTN 2006), Satellite Workshop to ECRTS 2006*, Dresden, Germany, July 2006. Jean-Dominique Decotignie.
- [16] P. Lin, C. Qiao, and X. Wang. Medium access control with a dynamic duty cycle for sensor networks. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 3, pages 1534–1539 Vol.3, march 2004.
- [17] G. Lu, B. Krishnamachari, and C.S. Raghavendra. An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, page 224, april 2004.
- [18] Sirisha Medidi, Jin Ding, Ghayathri Garudapuram, Jiong Wang, and Muralidhar Medidi. An analytical model and performance evaluation of transport protocols for wireless ad hoc networks. *Simulation Symposium, Annual*, 0:131–138, 2008.
- [19] Vinod Namboodiri and Abtin Keshavarzian. Alert: An adaptive low-latency event-driven MAC protocol for wireless sensor networks. In *Proceedings of the 7th international conference on Information processing in sensor networks, IPSN '08*, pages 159–170, Washington, DC, USA, 2008. IEEE Computer Society.

- [20] K. Nguyen, T. Nguyen, Chua Kee Chaing, and Mehul Motani. A prioritized MAC protocol for multihop, event-driven wireless sensor networks. In *Communications and Electronics, 2006. ICCE '06. First International Conference on*, pages 47–52, oct. 2006.
- [21] Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 181–192, New York, NY, USA, 2003. ACM.
- [22] Injong Rhee, Ajit Warrier, Mahesh Aia, Jeongki Min, and Mihail L. Sichitiu. Z-MAC: A hybrid MAC for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 16(3):511–524, June 2008.
- [23] Y. C. Tay, K. Jamieson, and H. Balakrishnan. Collision-minimizing CSMA and its applications to wireless sensor networks. *IEEE J.Sel. A. Commun.*, 22(6):1048–1057, September 2006.
- [24] 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.
- [25] L.F.W. van Hoesel, T. Nieberg, H.J. Kip, and P.J.M. Havinga. Advantages of a TDMA based, energy-efficient, self-organizing MAC protocol for WSNs. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 3, pages 1598 – 1602 Vol.3, may 2004.
- [26] Mehmet C. Vuran and Ian F. Akyildiz. Spatial correlation-based collaborative medium access control in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 14(2):316–329, April 2006.
- [27] 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.
- [28] Yafeng Wu, Gang Zhou, and J.A. Stankovic. ACR: Active collision recovery in dense wireless sensor networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, march 2010.
- [29] 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.
- [30] Xue Yang and N.H. Vaidya. A wakeup scheme for sensor networks: achieving balance between energy saving and end-to-end delay. In *Real-Time and Embedded Technology and Applications Symposium, 2004. Proceedings. RTAS 2004. 10th IEEE*, pages 19 – 26, may 2004.
- [31] Wei Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1567 – 1576 vol.3, 2002.
- [32] Chieh yih Wan. PSFQ: A reliable transport protocol for wireless sensor networks. pages 1–11. ACM Press, 2002.
- [33] Guoqiang Zheng and Shengyu Tang. Spatial correlation-based MAC protocol for event-driven wireless sensor networks. *Journal of Networks*, 6(1), 2011.
- [34] Yuanyuan Zhou and M. Medidi. Energy-efficient contention-resilient medium access for wireless sensor networks. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 3178 –3183, june 2007.
- [35] Eustathia Ziouva and Theodore Antonakopoulos. CSMA/CA performance under high traffic conditions: throughput and delay analysis. *Computer Communications*, 25(3):313 – 321, 2002.