CONGESTION MITIGATION BY TRAFFIC DISPERSION IN WIRELESS SENSOR NETWORKS

by

Sreekanth Yalamanchili

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

Sreekanth Yalamanchili

Thesis Title: Congestion Mitigation by Traffic Dispersion in Wireless Sensor Networks

Date of Final Oral Examination: 17 October 2012

The following individuals read and discussed the thesis submitted by student Sreekanth Yalamanchili, 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, Supervisory Committee. The thesis was approved for the Graduate College by John R. Pelton, Ph.D., Dean of the Graduate College.

Dedicated to my parents, Vijaya and Suresh

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. Sirisha Medidi, for her invaluable support, guidance, and motivation throughout my thesis.

My sincere thanks to Dr. Murali Medidi for being on my thesis committee and inspiring me with his lectures and words of wisdom.

I thank Dr. Jyh-haw Yeh for his inputs, for serving as a committee member, and for evaluating my thesis.

I would like to thank my fellow students (Vamsi, Ramya, Rahul, Divya, Haritha, and Mahesh) at Swanlab for their support.

My special thanks to my family, especially my Dad for believing in me and providing me an opportunity to pursue a Master's degree program, and to Nazia.

ABSTRACT

Wireless sensor networks (WSNs) are event-based systems that rely on the collective effort of several sensor nodes. When all nodes in an area sense an event and transmit that data, it causes sudden traffic bursts, which are spatially-correlated and lead to network congestion. Congestion can cause an increase in the amount of data loss, energy consumption, delay data transmission, and hinder network performance. To improve performance of event-driven applications, there arises a need for protocols that can reduce congestion and energy consumption. Existing protocols for sensing multiple events either handle congestion control or spatially-correlated contention, but not both, which can degrade network performance in terms of packet delivery ratio, latency, and energy consumption. Motivated primarily by the challenge to improve the performance of event-driven applications, we propose an energy efficient protocol to mitigate congestion that improves data delivery and reduces latency. This protocol mitigates congestion by dispersing network traffic using a forwarder selection mechanism that forces event reports from different nodes to disperse along different paths to the base station. Our protocol also reduces spatially-related contention by partitioning the sensors into different groups. All the sensors in a particular group cover the region of interest together, and these groups are scheduled in such that only one group is active to transmit the data at any given time. We implemented our protocol using the NS2 simulator for evaluating its performance. Results show that our protocol has significant improvement in the packet-delivery ratio, latency, and energy savings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Rapid development in technology in wireless communications has motivated the development of wireless sensor networks (WSNs). A WSN consists of autonomous sensors that are distributed spatially to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, or pollutants, and to cooperatively pass their data through the network to a main location [25]. A large number of sensor nodes are deployed in the field of interest to sense and report the event information to one or more main locations called base stations, and each sensor node is connected to one or more nodes.

Typically, a sensor node consists of a microcontroller, transceiver, external memory, power source, and one or more sensors [26]. The controller performs tasks, processes data, and controls the functionality of the other components. The transceiver transmits and receives the information. The memory used is the on-chip memory of a microcontroller and flash memory. Batteries are the main power source of any sensor node and they consume power for sensing, communicating, and data processing. The sensors measure the physical data of the parameter to be monitored. There can be many sensors attached to a single node to monitor different kinds of physical conditions.

These sensor networks can be used in applications such as military surveillance, environmental monitoring, industrial process monitoring, health care, and traffic control. The behavior of a WSN is characterized by the type of application which can be broadly classified into two types: event-driven applications and continuous-monitoring applications. Continuous-monitoring applications require periodic refreshed data at the base station. This requires the sensor nodes to continuously transmit the data periodically to the base station. Applications that monitor temperature and road traffic near a busy junction are some examples of continuous-monitoring applications. In event-driven applications, the sensor nodes transmit the information only when they sense the relevant data. Fire detection and military surveillance are examples of event-driven applications. As the nodes generate and transmit data only when an event of interest occurs, the traffic load is unpredictable, resulting in arbitrary traffic patterns. Hence, designing and analyzing traffic patterns and quality of service for event-driven models are more challenging.

There are many other challenges involved with WSNs because of their limited range, limited battery power, limited memory, and cost. Energy is the scarcest resource of the sensor nodes and these nodes determine the network lifetime. Because of this reason, the algorithms and the protocol designs should address issues like robustness, fault tolerance, low latency, and lifetime maximization. There are two kinds of traffic patterns in wireless sensor networks: the downstream traffic from the sink to the sensor nodes and the upstream traffic from sensor nodes to the sink. In the upstream traffic, once an event occurs in the WSN, a sudden surge of data traffic will be triggered by all sensor nodes in the event area, which can easily lead to network congestion. Network congestion occurs when a link or node carries so much data that its quality of service deteriorates [37].

Congestion usually exists inside WSNs due to some built-in characteristics inside

WSNs. First, in a multi-hop WSN, resources are limited. When a single event occurs, it could be detected by multiple nodes and several such events may occur simultaneously in the network. When all these nodes send the event reports at the same time to the base station, it leads to severe congestion, collisions, and spatially-correlated contention because of the limited resources. Second, sensors that detect an important event usually increase the data-generation rate to accurately report to the sink in time. For example, sensors used for temperature monitoring in forests generate a large number of alert packets in a short time to sinks when a fire is detected. Third, applications such as patient-health monitoring and image sensing require high throughput and low delay, which can further worsen the congestion inside WSNs. Therefore, congestion control is necessary and inevitable in a WSN [27].

Congestion can degrade the network performance and hinder the application objective. It can lead to packet losses, which means loss of information, increased packet delay, and severe energy consumption.

Several techniques such as rate control, queue management at the node level, and prioritization of the packets have been proposed to control the congestion in a WSN. However, when multiple events happen simultaneously, these techniques either control congestion or reduce spatially-correlated contention but not both. Motivated primarily by the challenge of applications that require a high packet delivery ratio and energy constraint characteristics of WSN, we propose a network design that extends the concept of grouping, which is energy efficient, while improving the packet-delivery ratio.

By adapting the Delaunay triangulation technique from [20], we divide the network into k mutually exclusive groups and introduce scheduling. We designed a forwarder-selection mechanism that forces the event reports from different nodes to disperse along different paths leading to the desired destination instead of a single path. Traffic dispersion makes it possible to alleviate the effects of bursty traffic and to balance the network traffic load. This combination of grouping, scheduling, and forwarder-selection mechanism helps to reduce spatially-correlated contention and mitigate congestion with a significant improvement in energy efficiency. The Delaunay triangulation and forwarder-selection mechanism are explained in Chapter 3.

1.1 Organization of Thesis

The rest of the thesis is organized as follows. Chapter 2 describes related work. Chapter 3 details the motivation and objective of the thesis and explains our protocol design. Chapter 4 analyzes the performance evaluation of our protocol design. Finally, we conclude in Chapter 5.

CHAPTER 2

RELATED WORK

Unlike the traditional networks, WSNs differ in several aspects. There are many challenges involved with WSNs because of their limited range, limited battery power, limited memory, and cost. Energy is the scarcest resource of the sensor nodes and these nodes determine the network lifetime. In event-driven applications, congestion control and energy efficiency play a very important role in WSNs research. Many protocols have been proposed to provide energy efficiency and congestion control in WSNs. These protocols are broadly classified as:

1. Protocols that provide congestion control.
2. Protocols that provide energy efficiency.

2.1 Congestion Control

A lot of research has been done to detect and control congestion. Most of the prior works of congestion-control mechanisms in WSNs are mainly embedded in the end-to-end controls, such as CODA [1], ESRT [2] and [33]. Though there are several advantages in end-to-end controls schemes, the need to propagate the onset of congestion between end-systems makes the approach slow. In general, a hop-by-hop control scheme [3, 5, 9, 12] reacts to congestion more quickly.

Congestion Detection and Avoidance in Sensor Networks (CODA) [1] uses both a hop-by-hop and an end-to-end congestion control scheme to react to the congestion by simply dropping packets at the node preceding the congestion area and employing an additive increase and multiplicative decrease (AIMD) scheme to control the generation rate of a source. Thus, CODA partially minimizes the effects of congestion, and as a result retransmissions still occur. Similar to CODA, Fusion [9] uses a static threshold value for detecting the onset of congestion even though it is normally difficult to determine a suitable threshold value that works in dynamic channel environments. Both CODA and Fusion detect congestion using the current buffer occupancy. Nodes then use a broadcast message to inform their neighboring nodes about the onset of congestion. A problem that both of these protocols face is that this broadcast message is not guaranteed to reach the sources because of the congestion.

To achieve event-to-sink reliability, event-to-sink reliable transport in WSNs (ESRT) [2] has been proposed. ESRT seeks to achieve reliable event detection with minimum energy expenditure and congestion control. It has been tailored for use in sensor networks with adaptability to dynamic topology, collective identification, energy conservation, and biased implementation at the sink. Reliability is measured by the number of data packets received at the sink. To measure reliability, the concept of observed event-level reliability and desired event-level reliability have been introduced. Observed event reliability (r_i) is the number of packets received in a decision interval i at the sink. Desired event reliability R is the number of data packets required for reliable event detection. If the observed event reliability is greater than the desired reliability, the event is deemed to be reliably detected. Otherwise, appropriate actions have to be taken to achieve this reliability. The reporting period of a sensor node is defined as the number of packets sent out per unit of time by that node. ESRT configures the reporting frequency f such that the desired event-detection accuracy is achieved with minimum expenditure. Five different characteristic regions have been identified based on reporting frequency f, r, and R. The five states are (No Congestion, Low Reliability), (No Congestion, High Reliability), (Congestion, High Reliability), (Congestion, Low Reliability), and (Optimal Operating Region). The goal of ESRT is to maintain operation in the Optimal Operating Region. The network can reside in any one of these states. Depending on the current state (S_i) of the network, ESRT finds the updated reporting frequency ($F_i + 1$) and broadcasts it to all the source nodes. If the observed event reliability is less than the desired reliability, ESRT increases the reporting frequency of the nodes, otherwise if the observed reliability is more than the desired level, the reporting frequency is decreased to avoid congestion and reduce energy consumption. To detect the current state of the network, the sink must be able to detect congestion in the network. The sensor nodes detect congestion using the buffer sizes and set the congestion notification bit. Once the sink receives a packet with this bit set, it knows that congestion will take place and will update the reporting frequency accordingly. ESRT does not support end-to-end reliable data delivery and it is impractical to vary the transmission rates of the nodes depending on the applications.

Chen et al. proposed a light-weight opportunistic forwarding scheme (LWOF) [42] to provide reliable data delivery for wireless sensor networks with an asynchronous duty cycle. To exploit the non-deterministic characteristic of opportunistic forwarding, an energy-efficient MAC protocol was also proposed. LWOF scheme uses the preamble in Low Power Listening MAC to dynamically select the forwarder during data transmission, thus reducing the overhead of maintaining historical network information or contention process. A preamble that lasts at least as long as the sleep period of the receiver is transmitted by the sender. The receiver node when awake detects the preamble and stays awake to receive the data. The sequential detection of preamble and busy tone signals help in reliably forwarding the data to a unique downstream node dynamically. LWOF scheme uses two channels with lower and higher data rates for transmitting the busy tone and sensor data respectively. Although the proposed protocol achieves a higher packet-delivery ratio in the networks without congestion, performance of the protocol under congested network scenarios were not explained.

Congestion can also be detected by exploiting packet inter-arrival time and packet service time [5] [34], packet inter-arrival rate and packet service rate [33], channel busy time (CBT) [8] and packet service ratio [23]. Unlike CODA and Fusion, nodes in protocols [5], [7], [23], [33] and [34] use implicit congestion notification to avoid transmission of additional control messages and therefore help improve energy efficiency.

The interference-aware fair rate control (IFRC) protocol [4] uses static queue thresholds to determine the congestion level, whereas CODA exercises congestion control by adjusting the out-going rate on each link based on the AIMD scheme. Consequently, IFRC reduces the number of dropped packets by reducing the throughput. IFRC supports fair bandwidth allocation among the flows. However, IFRC requires nodes to collect rate information from their neighboring nodes, thus increasing processing overhead and energy consumption. Available protocols [1, 4, 6, 9] do not consider congestion due to fading channels in dynamic environments.

Apart from rate control, WSN protocols such as [10], [22], [23], [24], [33] and [34] try to alleviate congestion and reduce the spatially-correlated contention by exploiting available network resources to transiently accommodate the traffic surge. CC-

MAC [13] relieves contention and improves performance by reducing the redundant nodes. ECR-MAC [22] relieves contention by dispersing the paths the senders will take. They propose a dynamic forwarder selection mechanism that relieves contention by allowing senders to deploy independent routes that detour the congested network area, and hence improve the network throughput. CADA [24] avoids the hotspot by detouring the traffic on a path located at least two hops away from the original intersecting traffic. However, there are many control messages involved in establishing this path. In [33], when a node experiences congestion, its immediate child node splits the real-time traffic on its alternate parent (route) in proportion to its weight factor w_i. In [10], a “bias” is inserted in each packet, which determines the curvature of the path followed by the packet towards the destination. The bias is a measure of how far the trajectory will deviate from the greedy route and also indicates the side of the deviation. PCCP [34] maintains the load-balancing for all paths by employing the rate-adjustment algorithm independently to calculate the rate for each path.

2.2 Energy Efficiency

Energy is the scarcest resource of WSNs. Due to the energy constraints, it is important to dynamically configure WSNs by using sleep/wakeup scheduling, thus extending the network lifetime. Ammari and Das proposed two k-coverage protocols using different scheduling approaches, *self-scheduling driven k-coverage* and *triggered-scheduling driven k-coverage* [30]. However the complexity of the protocols adds to overhead and energy consumption. In [20], the group-based technique involves separating all sensors into K-mutually exclusive groups. The connectivity-based partition approach [35] is a distributed iterative process. It starts from the initial partition where each node forms a unique group. CPA continuously merges two groups into a larger one until further merging will break the constraints of the problem. But here, the head node in each group is expected to maintain some additional information, which means additional overhead. In [36], sensor nodes are divided into clusters with at least K sensor nodes in each cluster. Then, the M nodes are divided into groups of K nodes each and turned ON in a queue-like manner using the queue-scheduling scheme. Here also the cluster head needs to perform additional tasks, which adds overhead. If only a few nodes are active in an event area, it can help reduce the spatially-correlated contention as well.

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [38] is the protocol where only one node is chosen as a head node that sends the fused data to the base station per round. PEGASIS protocol requires the formation of a chain, which is achieved in two steps. During the chain-construction phase, the farthest node from the base station is considered first and a greedy approach is followed to construct the chain. During the data-gathering phase, a leader of each round is selected randomly. Randomly selecting the head node also provides a benefit as it is more likely for nodes to die at random locations, thus providing a robust network. When a node dies, the chain is reconstructed by bypassing the dead node. After the leader is selected, it passes the token to initiate the data-gathering process. Passing the token also requires energy consumption but the cost of passing the token is very small because the token size is very small. In PEGASIS, the transmitting distance reduces for the sensor nodes. Since each node gets selected once, energy dissipation is also less compared to LEACH. Experimental results show that PEGASIS provides improvement by a factor of two more than LEACH.

To reduce energy consumption without affecting the connectivity of the network, an energy-efficient coordination algorithm for topology maintenance in ad-hoc wireless networks (SPAN) [39] has been proposed. SPAN is based on the observation that a dense sensor network can work with only a part of its nodes being active. It is 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 as a coordinator. SPAN elects coordinators from all nodes in the network. SPAN coordinators 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 wakeup 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. It tries to minimize the number of nodes acting as coordinators to reduce the latency and increase the network lifetime. Also it elects coordinators using only local information in a decentralized manner. A non-coordinator node becomes a coordinator if it discovers, using only information gathered from local-broadcast messages, that two of its neighbors cannot reach each other either directly or via one or two coordinators. The intent to become a coordinator is announced with a HELLO message. Span resolves contention by delaying a coordinator announcement with a randomized back-off delay. In order to ensure fairness, after a node has been a coordinator for some time, it withdraws if every pair of neighbor nodes can reach each other via their neighbors or other coordinators. This gives a fair chance to all nodes that are eligible for being coordinators.

Geographical Adaptive Fidelity (GAF) [40] is introduced to reduce energy consumption in ad-hoc wireless networks. GAF identifies equivalent nodes for routing based on location information and turns off unnecessary nodes. In wireless networks, a lot of energy is spent in idle listening. Idle energy is almost equal to transmission energy or reception energy. Powering off the radio conserves energy both in over hearing due to data transfer and in idle state energy dissipation when no traffic exists. Hence, nodes that power down their radios are explored. It is also observed 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. Routing Fidelity is defined as uninterrupted connectivity between communicating nodes. Routing Fidelity can be maintained as long as any intermediate node is awake. Each GAF node uses location information to associate itself with a virtual grid, where all nodes in a particular grid square are equivalent with respect to forwarding packets. Nodes in the same grid will then coordinate who will sleep and how long. In GAF, nodes are in one of these three states: sleeping, discovery, or active. Initially nodes start out in state discovery. In this state, nodes turn their radio on and exchange discovery messages to find other nodes in the same grid. When a node enters the discovery state, it sets a timer, and when the time expires, it moves to the active state. When a node enters the active state, it sets a timer and moves to the discovery state when the timer expires. A node in the discovery or active state changes state to sleeping when it determines that some other equivalent node can handle routing. GAF employs a load balancing strategy so that all nodes remain up and running together for as long as possible. This ensures that all nodes are given equal chance and no one node is penalized more than the others. After a node remains in the active state for some time, it changes its state to discovery to give a chance to other nodes in the grid to become active. When the active node changes its state to discovery, it is more likely that it has less remaining energy than its neighbor nodes. In the ideal scenario, there is one active node at any point of time in one grid. When the nodes move, there is every chance that we might have a grid with no active nodes at all. To avoid this problem, each node estimates the time it expects to leave the grid and includes this information in the discovery message. When other nodes enter the sleeping state, they decide how long to stay in the sleeping state based on this information.

A Sparse Topology and Energy Management (STEM) was proposed in [41]. The main objective of this scheme is to reduce energy consumption in a monitoring state to minimum while ensuring satisfactory latency for transitioning to the transfer state. The majority of time, the network is only sensing its environment. This is referred to as the monitoring state. Once an event occurs, data has to be forwarded to the sink and the network transitions to the transfer state. STEM reduces the energy consumption of the nodes by putting them into a sleep state. In the monitoring state, instead of full sleep, a node goes into low power listen mode. However, in return for this energy reduction, a certain amount of latency is introduced to wake up the nodes. Nodes in this design have three states: sleep, active, or listening. The node that wants to communicate (initiator node) polls the node, which it has to wake up (target node) continuously. As soon as the target node hears the poll, the link between the nodes is activated. Once the link is activated, data transfer takes place using a MAC protocol. To wake up a node, a wake up message is sent to the node in the form of a beacon packet (STEM-B meaning beacon based) or a simple tone (STEM-T meaning tone based) resulting in two variants of STEM. The topology management in STEM is specifically geared toward those scenarios where the network spends most of its time waiting for events to happen without forwarding traffic. Simulation results show a considerable improvement over GAF in both scenarios. Though this scheme has many advantages, it suffers from the energy consumption due to continuous polling and requires extra radio on the sensor nodes.

The latency minimized energy efficient MAC protocol (LEEM) [43] is a hop ahead reservation scheme that minimizes latency and increases energy efficiency. It does this by reserving the channel of the next hop in advance. It is useful in time-critical applications where sensed events are to be reported immediately to the sink to take remedial or defensive actions. LEEM assists in sending the packets with minimum delay by using dual frequency radio set up. Since the channel of the next hop is reserved in advance, the intermediate nodes in the data path forward the packet as soon as it is received. In an event-driven sensor network, the nodes spend a lot of time sensing the event. In order to reduce this energy, the control channel radios are kept in a low power sleep mode. The control channel radio is made active periodically to check for any data transmissions and activate the data channel. In LEEM, nodes are resynchronized every hour. The synchronization helps in making reservations and reduces the delay. In a synchronized network, since each node knows the time at which its next hop node is active, it need not send continuous wake-up signals. This results in lower energy consumption. The reservation scheme in LEEM helps in eliminating the set-up latency at the intermediate nodes. When the control channel radio of a sensor node is in sleep mode and an event occurs, the sensor node waits for the next hop node to become active. It then requests the next hop node to activate its data channel radio by sending a request packet. The receiver agrees by sending back an ACK. This procedure continues throughout the data path until the packet reaches the data sink. Whenever the data transfer takes place, the receiver of the packet reserves the channel for K hops ahead. If the value of K is one, it is a 1-hop reservation scheme, otherwise it is an N-hop reservation scheme. When the current transmission gets completed at the receiver, the next hop channel becomes ready. Hence, the delay in setting up the next channel is avoided except at the first hop. Although LEEM shows a significant improvement over other protocols like STEM and PTW in terms of energy consumption and latency, it is not applicable for applications that have a continuous occurrence of events.

To address spatially-correlated contention, Zhou and Medidi proposed a distributed topology control [31] to schedule node wake-up slots and design a MAC protocol to benefit from this topology control for improving energy efficiency and reducing latency. Energy consumption in an idle listening state is as much as the transmission and reception energy. One way to save energy is to employ duty cycles. By employing duty cycles only, a subset of nodes remain active at any point of time. The remaining nodes turn off their radios and keep checking their eligibility to remain active periodically. However, a lower duty cycle can require each node to spend a longer set-up latency to wake up its forwarder, which increases the end-to-end delay. In order to have low delay while having low duty cycles and high-energy efficiency, the following sleep-based topology control was designed. To address spatially-correlated contention, the topology is controlled such that each node has multiple potential forwarders. Allowing each node to have multiple forwarders along with staggered scheduling not only reduces congestion but also achieves shorter delay since it significantly reduces the first hop set-up latency and eliminates the latencies in further hops.

CHAPTER 3

ENERGY EFFICIENT CONGESTION MITIGATION PROTOCOL

3.1 Motivation and Design Requirements

Wireless sensor networks (WSNs) can be widely used in medical, industrial, and military surveillance applications. A wireless sensor network consists of nodes, from a few to several hundreds or even thousands, where each node is connected to one or several sensors. These sensor nodes are typically energy constrained. In a WSN, data flows from both sensor nodes to sink (upstream) and sink to sensor nodes (downstream). The prominent traffic pattern is from sensor nodes to sink in which the sensor nodes forward the sensed data to the sink. In event-driven applications, sensor nodes transmit information only when they sense relevant data. Nodes generate a lot of data when simultaneous events occur, resulting in extra traffic, and it becomes unpredictable, resulting in arbitrary traffic patterns. This unpredictable data generated can bring the message rates beyond the expected capacity of the network, leading to congestion. Additionally, multiple nodes may sense the same event, leading to spatially-correlated contention, further increasing congestion. This degrades the network performance by increasing collisions, delay, and energy consumption. Congestion in wireless sensor networks not only causes loss of event reports that are being delivered to the base station but also lead to excessive energy consumption. So improving network performance by reducing congestion and extending network lifetime is usually the primary design objective in wireless sensor network protocols.

Many protocols either try to reduce congestion or spatially-correlated contention but not both. An energy-efficient collision-free MAC protocol (TRAMA) is presented in [14], which is based on a time-slotted structure. Each node determines its own time slot using a distributed election scheme based on traffic requirements of every two-hop neighbor. Although the protocol achieves a high delivery ratio with tolerable delay, the performance of the protocol depends on the two-hop neighborhood information in each node. Since this information is collected through signaling, in the case of high-density sensor networks, the signaling cost increases significantly, resulting in either incomplete neighbor information due to collisions or high energy consumption due to signaling costs. Vuran and Akyildiz proposed a spatial correlation-based collaborative MAC (CC-MAC) that relieves contention and improves performance by reducing the redundant node reports [13]. However, no congestion-control technique is considered in this work. Moreover, CC-MAC achieves a lower packet-delivery ratio since it filters the redundant data injected into the network. Although energy efficiency improves due to suppressed correlation neighbors, there is a huge amount of data loss, which is not desirable. The complicated nature of the Iterative Node Selection (INS) algorithm, which generates a correlation radius, may also limit the application of the protocol. As the number of sensing events increase, especially if the sensing conditions change with time, the overhead associated with computing the correlation radius and distributing throughout the network increases. For large networks, this overhead may become significant. Along with improving event reporting, if the protocols designed were not energy efficient, the lifetime of the network degrades, thus affecting the performance. The existing congestion reduction protocols for event-driven applications are not scalable and do not ensure energy efficiency. To fulfill these requirements, there is a strong need for a protocol that is energy efficient, scalable, and reduces congestion as well as spatially-correlated contention while reducing the cost of overhead.

In this thesis, we propose a protocol that reduces congestion and spatially-correlated contention and improves network lifetime. The proposed protocol reduces spatially-correlated contention and energy consumption by inducing less traffic into the network. This can be achieved through dividing the network into groups and incorporating scheduling among these groups in such a way that, at any given time, only one group is active to transmit the data. Scheduling ensures that nodes close to each other do not send or receive data at the same time, thus decreasing spatially-correlated contention. Furthermore, choosing only a subset of nodes to be active at any time in data forwarding reduces the energy consumption, thus improving network lifetime.

To reduce congestion, we propose a forwarder selection mechanism that forces a parent node to act as a forwarder for at most two child nodes. As it limits the number of child nodes that can send the event reports to the same parent node, it helps in reducing the collisions and mitigates the congestion. Also choosing a data forwarder with the minimum hop distance from the sink reduces latency. In WSNs, packet losses are due to congestion, spatially-correlated contention, collisions, link failures, node failures, and resource constraints. In the following sections, we identify the design challenges and provide solutions to the challenges using our protocol.

- High Network Traffic:

  Usually sensors are densely positioned at random to provide coverage for the geographical region. With more traffic in the network, and due to the event-driven sensor networks, several sensors detect and transmit data at the same time, thereby causing congestion, spatially-correlated contention, and higher energy consumption in the network. There is a need for protocols that reduce the network traffic by choosing a subset of nodes required for the region to be entirely covered and also the connectivity between nodes to be maintained.

- 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 causes node failures. If a node fails while transmitting/receiving a packet, all the packets that are sent from or intended for that node will be dropped. In order to achieve successful event detection and reporting, the protocols should be designed in such a way that events are detected and reported. This will help in reducing packet drops and increasing the quality of service of the network.

- Link Failures:

  Apart from node failures, link failures can also cause packet losses in wireless networks. Due to errors such as signal attenuation and noise interference, packets are not transmitted successfully over two nodes. Attenuation refers to any reduction in the strength of a signal and is caused by signal transmission over long distances. As a result, packets will be corrupted by the time they reach the receiver. Data loss could also occur when two nodes try to transmit data simultaneously. When two nodes try to send data at the same time, they may collide and packets from either of the nodes might get dropped. In order to improve event reporting, the designed protocols should have the ability to avoid collisions in case of link failures.

- Congestion:

  In a WSN, when a single event occurs, it could be detected by multiple nodes and several such events may occur simultaneously in the network. When all these nodes send the event reports at the same time to the base station, it could lead to severe congestion. The network will have more traffic and it becomes unpredictable, resulting in arbitrary traffic patterns. This unpredictable data generated can bring the message rates beyond the expected capacity of the network, leading to congestion. Also, as the medium around the sensor nodes is congested, more packet transmissions result in collisions thereby dropping the packets. The protocol design should provide a mechanism to handle the network in congested scenarios.

- Packet Loss Recovery:

  Packets get dropped due to congestion in the network, link failures, node failures, and etc. Mechanisms like TCP/IP in wired networks provide efficient packet-loss recovery. However, these mechanisms cannot be applied to wireless sensor networks as a lot of energy is consumed due to retransmissions. As most of the transmissions in WSNs are hop-by-hop, packet losses need to be handled at the link level. This requires protocols that ensure improved packet delivery at the base station for better event reporting.

- Energy Efficiency:

  As the sensor networks are energy constrained, in order to extend the network lifetime, it is very important to reduce energy consumed by the nodes. Energy consumed due to transmission and reception of messages is very high. Also, due to large deployments of nodes in the sensor network, energy consumption increases with a larger number of sensors sensing and reporting the events to the base station. Energy consumption also increases due to redundancies in the deployment of the sensor network. In designing an energy-efficient protocol, this energy waste must be considered and reduced.

- Scalability:

  As sensor networks contain a very large number of sensor nodes, networks should be scalable enough to provide a high packet-delivery ratio. Protocols need to be distributed in nature in order to reduce the overhead caused in the case of very large networks.

Considering the above challenges, we propose a congestion-mitigation protocol that provides event reporting with improved energy efficiency. In order to measure the performance of the protocol, we chose the following standard metrics:

- Packet-Delivery Ratio:

  It is the ratio of the total number of packets successfully delivered to the base station to the total number of packets generated. This packet delivery ratio shows the performance of the protocol and illustrates the level of successfully delivered data to the destination.

- Energy Efficiency:

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

- Delay:

  Average delay is measured to identify the latency in forwarding the packets to the base station. Depending on the nature of applications in the sensor networks, a delay in the network plays a crucial role.

3.2 Assumptions

For simplifying the explanation, we make the following assumptions for the proposed protocol:

- The network is densely populated to report any event to the base station.
- All the nodes know their one-hop neighborhood node's information (ID, hop-distance) by local broadcast mechanisms.
- The network deployment does not have any physical holes and the outer boundary is identified.

3.3 Congestion Mitigation

Before sensor nodes send any event reports to the base station, all sensor nodes should participate in an initial setup in order to make themselves available for data transmission and reception. The initial setup process consists of the following steps:

- Dividing the network into mutually exclusive groups
- Employing scheduling among groups
- Choosing a forwarder through the forwarder selection mechanism
- Establishing data paths with the base station.

3.3.1 Grouping

In a WSN, sensors are deployed randomly in harsh environments. Sensor nodes are energy constrained, and utilizing all the sensor nodes for sensing and communication would deplete the network resources as more energy is consumed. In a given region with over-provisioned sensors, nodes sense the event occurring at a location in the region and report to the sink. Multiple nodes may sense the same event, leading to spatially-correlated contention and causing packet drops. With all the sensor nodes utilized for sensing and communication operations, more transmission and reception of messages take place between sensor nodes, thereby reducing the energy levels in sensors. Messages transmitted by sensor nodes simultaneously increase congestion in the network, and packets are dropped, which reduces the quality of service provided by the sensors in the region. Dividing the network into groups and scheduling these groups reduces congestion and contention in the network, and also reduces the energy consumption of the nodes.

We adapt the Delaunay triangulation technique from [20] to divide the network into k-mutually exclusive groups. The Delaunay triangulation-based coverage technique locally optimizes the sensing radii in order to achieve (1) good global optimality in energy consumption and energy-balancing, and (2) complete coverage for reliable surveillance. This technique emphasizes simplicity and scalability and hence it can be adapted for large-scale deployments. It extends the lifetime of the network by balancing the energy throughout the network. It can be used with scheduling schemes to further reduce the redundant coverage within each mutually exclusive set of sensor nodes and hence contribute in reducing spatially-correlated contention.

Computational geometry is frequently used in WSN coverage-optimization. The most commonly used computational geometry approach are the Voronoi diagram and Delaunay triangulation. They have been very influential in solving the coverage problems of wireless sensor networks. The Voronoi diagram is partition of sites in such a way that points inside a polygon are closer to the site inside the polygon than any other sites, thus one of the vertices of the polygon is the farthest point of the polygon to the site inside it. One of the properties of Voronoi diagrams is that the adjacent polygons in a Voronoi diagram are equidistant from the edge dividing two neighboring sites in the construct. Figure 3.1(a) shows an example construct of a Voronoi diagram. Detailed explanation about Voronoi diagrams can be found in [44] and [45].

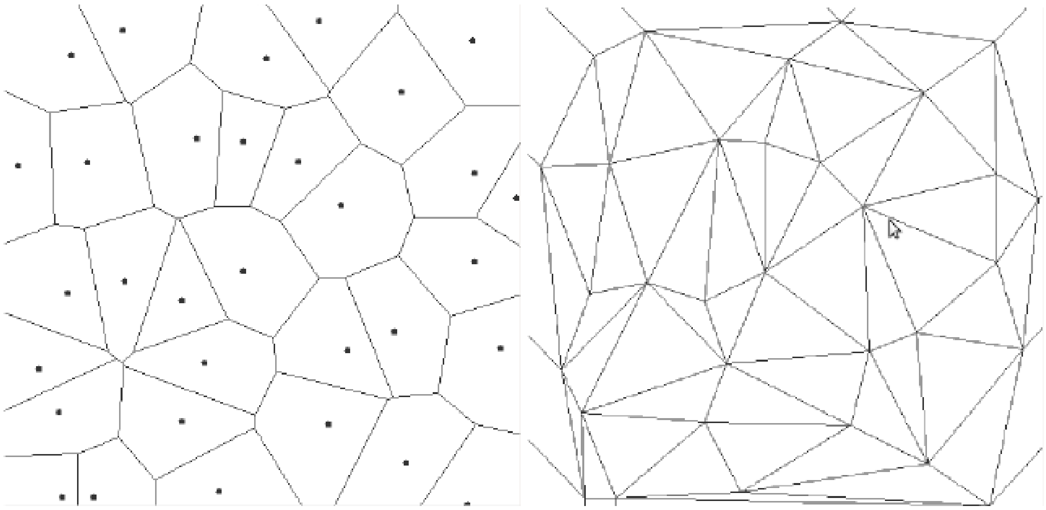

(a) Voronoi diagram
(b) Delaunay triangulation

Figure 3.1: Voronoi diagram and Delaunay triangulation of a random topology

Delaunay triangulation is another construct in computational geometry, which is a dual of Voronoi diagram. It can be generated by joining the vertices of neighboring sites of Voronoi diagrams that share a common edge between them. Delaunay triangulation of a set of P points in a 2D plane maximizes the smallest angle in the triangle and no point in set P is inside the circumcircle of any triangle in the triangulation. Figure 3.1(b) illustrates an example of a Delaunay triangulation of a set of P points in a 2D plane. Delaunay triangulation of a set of points can be produced in different methods like incremental, divide and conquer, sweepline, and flip algorithms. Delaunay triangulations have a major influence in WSNs as neighborhood information can be easily extracted by considering the neighboring sites and the shortest euclidean distance between two nodes of the triangulation. Since WSNs are energy constrained, it is necessary for the network to use local information to perform Delaunay triangulation. In this thesis, Delaunay triangulation is performed over the network using the one-hop or local neighborhood information [20] of each sensor node. Each node having the one-hop information incrementally adds every node, performs triangulation, and checks for the validity of the Delaunay properties. Edges of the triangles are flipped to maintain the validity if the properties are not satisfied. Delaunay triangulation and its properties are presented in [45].

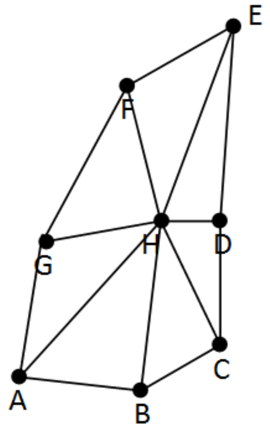
3.3.2 One-Hop Approximation of Delaunay Triangulation

One-hop approximation of Delaunay triangulation is explained in detail in [20]. Delaunay triangulation (DT), the dual of the Voronoi diagram, has the following characteristics:

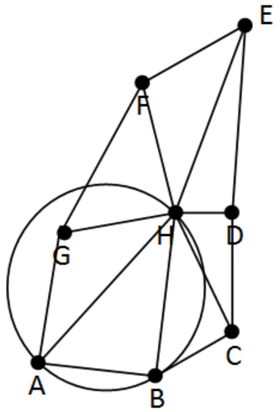
- Fat triangles, in the sense that the minimum angle of any Delaunay triangle is as large as possible.
- The Empty Circle Property, defined as a circle that runs through the vertices of any triangle with no other vertex inside the circle.

This algorithm is based on the centralized edge-flipping algorithm. In this algorithm, each node maintains a list of its one-hop neighbors *NeighborList*. After an arbitrary triangulation is constructed (Figure 3.2a), each node independently tests its adjacent triangles to determine whether they all satisfy the Empty Circle Property. If an adjacent triangle cannot satisfy the Empty Circle Property, the corresponding edge is flipped. For example, in Figure 3.2b, $\triangle ABH$ is, at first, a non-Delaunay triangle because point G lies inside of $\triangle ABH$'s circumcircle. Then, AH is flipped to BG, and point A is deleted from H's *NeighborList*. The result is the formation of the Delaunay triangle $\triangle$BGH as shown in Figure 3.2c. In Figure 3.2d, $\triangle DEH$ is identified as a non-Delaunay triangle and, similarly, to make the conversion, EH is flipped to DF and point E is eliminated from H's *NeighborList*. The final result is the creation of $\triangle DFH$, with no other points located inside its circumcircle as shown in Figure 3.2e. The edge-flipping process continues until H's adjacent triangles can all be classified as Delaunay triangles.

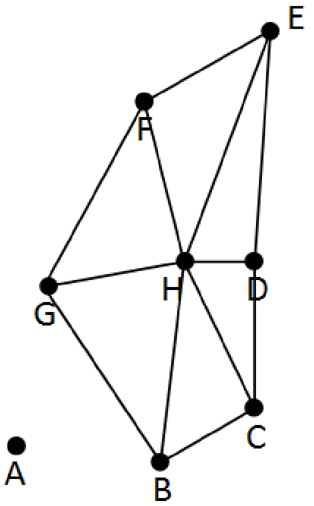
The one-hop approximation of DT can be implemented easily on sensors with low communication and computation overhead; however, with only one-hop information, the resulting triangulation may differ from the traditional DT. Local approximation of DT is equivalent to the traditional DT, provided that: (1) the area can be completely covered by the maximum sensing radius; and (2) the sensors satisfy $2R_s \leq R_x$, where R_s and R_x represent the maximum sensing radius and the maximum transmission radius, respectively. Condition (1) is a basic requirement for any reliable surveillance and condition (2) holds for most hardware. For example, MICA sensors have a sensing range of 2-6m and a transmission range of 30m. Furthermore, $R_x \geq 2R_s$ is commonly assumed to obtain connectivity with full coverage.



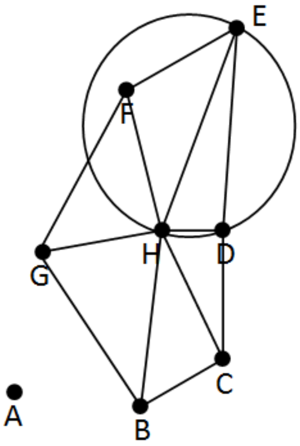
(a) Initial triangulation



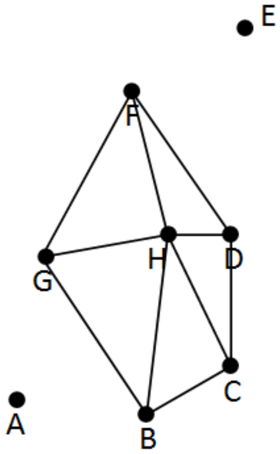
(b) $\triangle ABH$ is not a Delaunay triangle



(c) Flip AH to BG

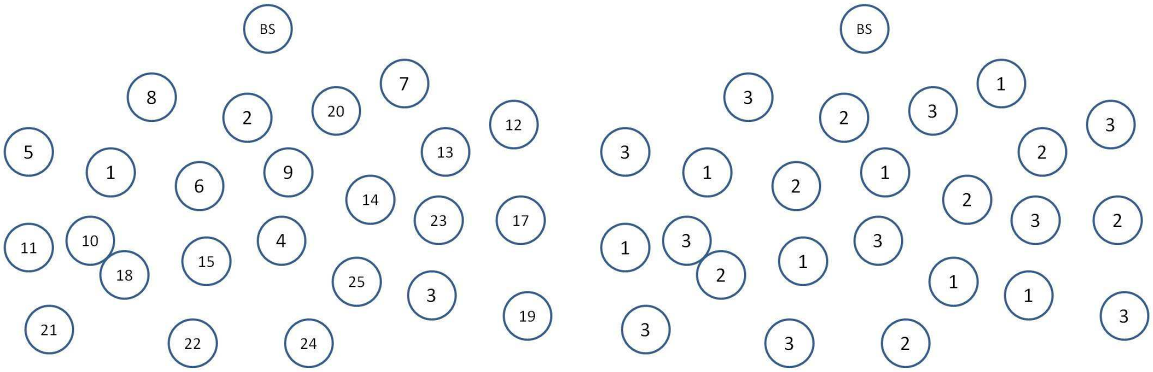


(d) $\triangle DEH$ is not a Delaunay triangle



(e) Flip EH to DF

Once the network is deployed, each node retrieves its list of one-hop neighbors. Using this local information, every node performs a Delaunay triangulation over the one-hop neighbor nodes. Each node incrementally adds every node from its list of one-hop neighbors, performs triangulation, and checks for the validity of the Delaunay properties. As described in the above algorithm, edges of the triangles are flipped to maintain the validity if the properties are not satisfied. The first set of nodes that come as a result after implementing this algorithm are listed as group 1 nodes. These nodes are turned off and the algorithm is implemented again on the rest of the nodes. This process is repeated until we obtain three mutually exclusive sets of nodes and these are listed as group 1, group 2, and group 3. Figure 3.3 shows an example of dividing the network into groups.



(a) Without Groups

(b) With Groups

Figure 3.3: Dividing network into groups

Topology-control techniques and scheduling mechanisms reduce the energy consumed by the sensor network as well as spatially-correlated contention. In the proposed protocol, dividing the network into groups and choosing only a group of nodes at a given time for data forwarding reduces the energy consumed by the network. It also reduces spatially-correlated contention as close by nodes belonging to different groups do not send data at the same time. However, if nodes belonging to same group are close by, this contention still exists. In the below section, we discuss more about our scheduling mechanism.

3.3.3 Scheduling

The main purpose in designing a scheduling mechanism is to allocate time slots depending on the topology and the node packet-generation rates. A good schedule not only avoids collisions by avoiding the interferers of every receiver node in each time slot but also minimizes the number of time slots, hence the latency.

More than one node can transmit at the same time slot if their receivers are in non-conflicting parts of the network. There are two types of conflicts, namely, primary conflict and secondary conflict [21]. Primary conflict occurs when two nodes transmit the data to a node at the same time. The forwarder selection mechanism (explained in 3.3.4) and IEEE 802.11 RTS/CTS mechanism can handle this problem. Secondary conflict occurs when a node, an intended receiver of a particular transmission, is also within the transmission range of another transmission intended for other nodes. This conflict is also implicitly taken care by the IEEE 802.11 RTS/CTS mechanism.

In scheduling, all the nodes that belong to the same group have the same time slot for sending or receiving data. The scheduling consists of repeating fixed length slots of time X in which nodes of the same group participate in data transmissions/receptions. This group of nodes will be in sleep mode for time $2X$ as the other two groups of nodes use these time slots for data forwarding. These consecutive time slots are referred as G_i-slot (i=*1,2,3*), S_1-slot, and S_2-slot (where G_i, S_1, and S_2 denote group, sleep-1, and sleep-2, respectively). It is necessary for nodes of different groups to coordinate their schedules so that a group time slot does not coincide with the other two group time slots. The coordination is performed as follows:

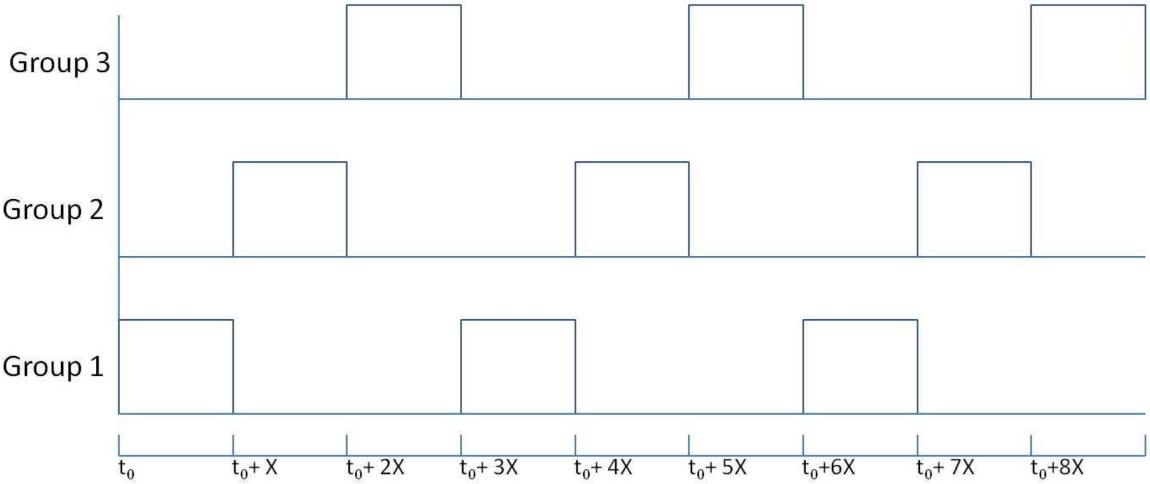


Figure 3.4: Schedules of different groups in CoMiT

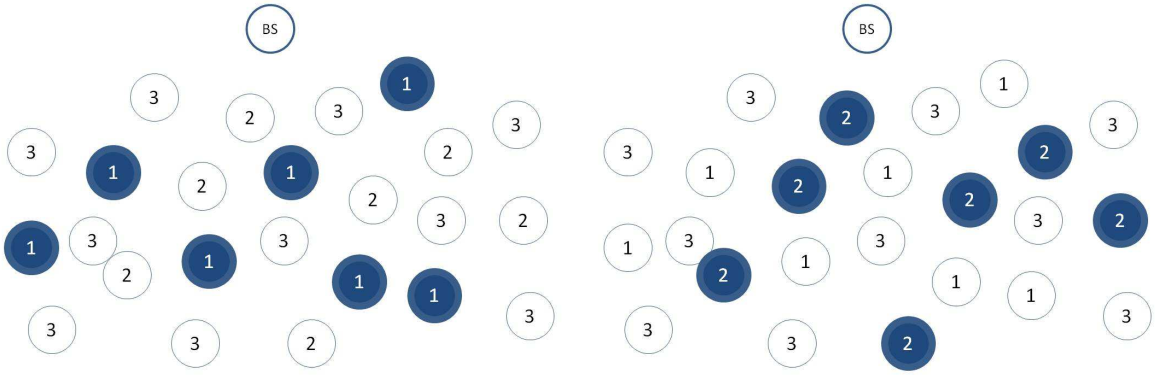
- Once the network is divided into groups, the base station broadcasts a G_i-slot message.

- All the group 1 nodes that receive this message mark its time slot as a G_1-slot and further relay this message to nodes that are away from the base station.

- Group 2 and group 3 nodes that receive the G_i-slot message will mark their time slot as S_1-slot and further relay this message.

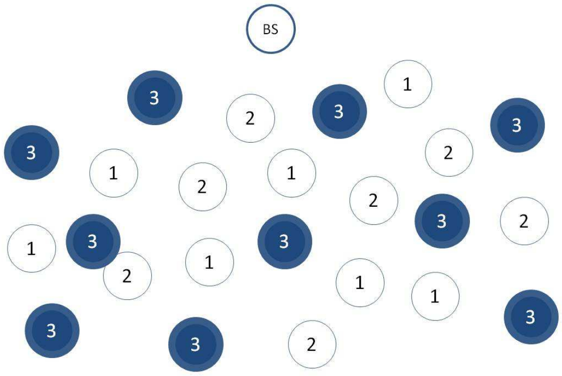
- In the next hop, group 1 nodes will make their slot as S_1-slot, group 2 nodes as G_2-slot, and group 3 nodes as S_2-slot.

- Subsequently, each node in the network follows this G_i-S_1-S_2 schedule. Figure 3.4 demonstrates the coordinated schedule of three groups (group 1, group 2, and group 3).



(a) Group 1 in transmit/receive mode

(b) Group 2 in transmit/receive mode



(c) Group 3 in transmit/receive mode

Figure 3.5: Different groups cover the network independently

Figure 3.4 shows that group 1, group 2, and group 3 nodes go to data transmission/reception mode sequentially. Figure 3.5 shows an example of how the network is divided into different groups. Each group works independently and provides coverage.

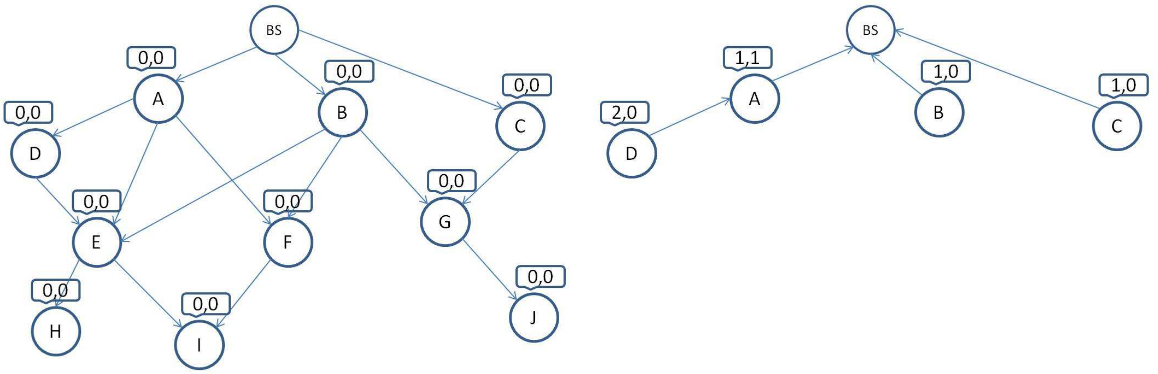
3.3.4 Forwarder-Selection Mechanism

Once scheduling is completed, the next step is to form a data path for all the nodes within a group. As the network is divided into groups, each group will follow this procedure independently. Each node has to select a forwarder for sending its data. We try to reduce the number of nodes choosing a node as a forwarder based on *Child count* and *Hop count*, which are discussed below. The following are the steps involved for a node in choosing its forwarder.

Algorithm 1 Algorithm for Forwarder-Selection Mechanism: Functionality at each node

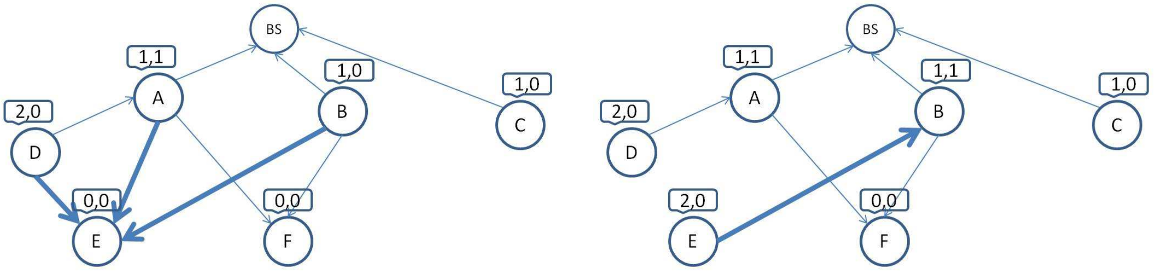
```
Let ChildCount be the number of nodes for which a node is acting as a forwarder.
Let HopCount be the number of hops away from the base station.
while $Node \neq LeafNode$ do
  Broadcast FORWARDER-REQ-MESG to all one hop neighbors
end while
if $SourceAddress = BaseStation$ then
  $Forwarder \leftarrow BaseStation$
  $HopCount \leftarrow HopCount + 1$
else
  Check for total number of FORWARDER-REQ-MESGs received.
  if $total < 2$ then
    $Forwarder \leftarrow N$
    Reply to node $N$ with a FORWARDER-REPLY-MESG
    For node $N$:
    $ChildCount \leftarrow ChildCount + 1$
    Broadcast the updated ChildCount to all its children.
  end if
else
  Sort the potential forwarders according to minimum $HopCount$
  Select a node $N$ with minimum $ChildCount$
  $Forwarder \leftarrow N$
  Reply to node $N$ with a FORWARDER-REPLY-MESG
  For node $N$:
  $ChildCount \leftarrow ChildCount + 1$
  Broadcast the updated ChildCount to all its children.
end if
```

Each node maintains two variables: *Child count* - The number of nodes for which it is acting as a forwarder and *Hop count* - The number of hops away from the base station. A node receives *FORWARDER-REQ-MESG* from its one-hop neighbors and stores this information. The forwarder selection mechanism is initiated by the base station. The base station broadcasts a *FORWARDER-REQ-MESG* message. All the child nodes that receive this message will set the base station as their forwarder and will also set their *Hop count* as one. These child nodes will include their *Hop count* and further relay this message to nodes that are away from the base station. This continues until the message reaches the leaf nodes. If a node receives *FORWARDER-REQ-MESG* from only one node n, it sets n as its forwarder. It replies to n with a *FORWARDER-REPLY-MESG* saying that it has chosen n as its forwarder. Node n increments its *Child count* to one and relays the updated *Child count* to all of its children. If a node receives multiple *FORWARDER-REQ-MESG*s, first it will select only those nodes with minimum hop distance. Then, among those nodes, it will select a node n with minimum *Child count*. If two or more nodes have the same *Child count*, it will choose a node among those nodes randomly as its forwarder and reply with a *FORWARDER-REPLY-MESG*. This process is repeated until each node in the group has at least one forwarder node.



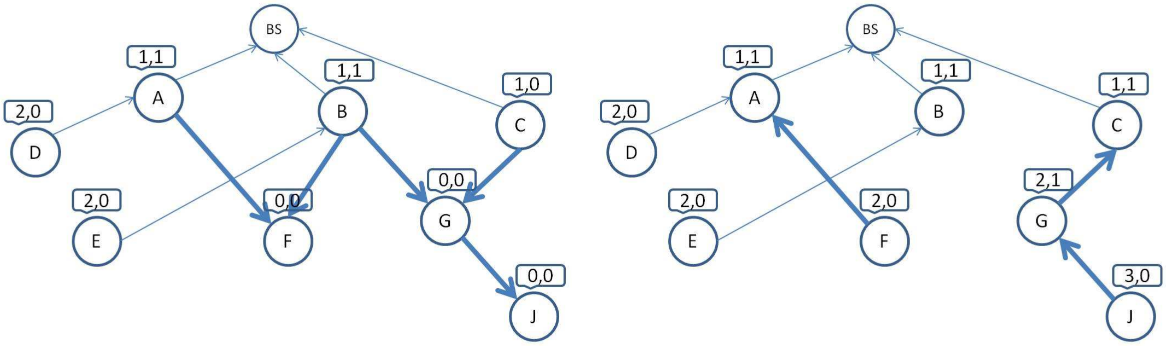
(a) Basestation initiates FORWARDER-REQ-MESG

(b) D selects A as its forwarder



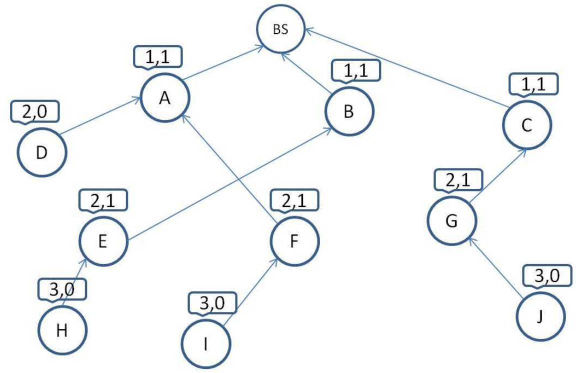
(c) E receives multiple FORWARDER-REQ-MESG's

(d) E chooses B as its forwarder



(e) F and G receive multiple FORWARDER-REQ-MESG's

(f) F chooses A, G chooses C as their forwarders



(g) Final data paths

Figure 3.5: Forwarder-Selection Mechanism

To better illustrate the forwarder selection mechanism, consider the Figure 3.5. We will explain the forwarder-selection mechanism using group 3 nodes from Figure 3.5(c) as an example. Node BS represents the base station. Nodes *A, B, C, D, E, F, G, H, I,* and *J* form group 3 nodes. The two variables in a small rectangle above each node represent *Hop count* and *Child count*, respectively.

To form the data path, the base station initiates and broadcasts a *FORWARDER-REQ-MESG*. Nodes *A, B,* and *C,* which receive this message will store and relay this message further away from the base station until it reaches the leaf nodes *H, I, J.* Figure 3.5(a) shows *FORWARDER-REQ-MESG*s propagating until the leaf nodes, where arrows indicate the direction messages, are sent. To prevent collisions amongst these broadcasts, a node will randomly chose a time to broadcast the packet within a reasonable time frame. In order to restrict the broadcast messages from going in a loop, each node broadcasts these messages only three times. This is to make sure that even if some of the broadcast packets are dropped as a result of collisions another broadcast message would possibly make it to the receiving node. After a certain period of time, nodes start to configure their final forwarder nodes.

Nodes *A, B,* and *C* set *BS* as their forwarder and also update their *Hop count* as one. As Node *D* received a *FORWARDER-REQ-MESG* from node *A* only, it chooses *A* as its forwarder and it replies to *A* with a *FORWARDER-REPLY-MESG* (Figure 3.5(b)). Node *A* updates its *Child count* to one and relays the message to all of its one-hop neighbors.

As seen in Figure 3.5(c), Node *E* has received multiple *FORWARDER-REQ-MESG*s from *D, A,* and *B*. It discards the request from *D* as *D*'s *Hop count* is greater than *A* and *B*, and chooses *B* as its forwarder as *B* has s lower *Child count* than *A* (Figure 3.5(d)). Note that if *A* and *B* have the same *Child count*, a node will be chosen randomly. For example, node *F* in Figure 3.5(f) chooses *A* as its forwarder. Node *E* replies to *B* and the process is repeated until every node in the group has at least one forwarder. The final data paths are shown in Figure 3.5(g).

CHAPTER 4

PERFORMANCE EVALUATION

To evaluate the performance, the proposed protocol (CoMiT) is implemented in the NS2 simulator [28]. Extensive experiments were conducted in order to test the performance of the grouping-based approach. We compared the performance of our protocol with existing protocols such as CC-MAC [13] and TRAMA [14] because they handle spatially-correlated contention. First, we provide a comparison of our approach with the previous approaches in [13] and [14] in terms of packet delivery ratio, average latency, and energy consumption. Second, we compare our approach to the one without the support of grouping and the forwarder-selection mechanism in terms of standard metrics, such as packet-delivery ratio, average latency and energy consumption. Additionally, we tried to observe the overall performance under different network densities. These evaluations prove that CoMiT is scalable and works in a realistic environment.

4.1 Simulation Setup

The simulations were run with the simulation parameters mentioned in Table 4.1. For data packets, Constant Bit Rate CBR traffic is generated. To evaluate the performance and scalability under a congested scenario, network density is varied from 100 to 300 nodes with 15 source nodes for a varied packet interval rate from

$5KB/s$ to $0.5KB/s$ with a packet size of 512 *bytes* to load the network with heavy traffic. In all the experiments, each data point taken is an average of 20 independent runs.

Table 4.1: Simulation parameters for protocol evaluation

Parameters	Value
Area	$1000m$ x $1000m$
Total number of nodes	100 to 300
Sources	15
Transmission range	$250m$
Data packet size	512 *bytes*
Packet interval rate	$0.1(5KB/s)$ to $1(0.5KB/s)$
Transmit power	$0.01488\,W$
Receive power	$0.01250\,W$
Idle power	$0.01236\,W$
Sleep power	$0.000016\,W$

4.2 Comparison with CC-MAC and TRAMA

Table 4.2: Simulation parameters for protocol comparison

Parameters	Value
Area	$500m$ x $500m$
Total number of nodes	50
Sources	16
Transmission range	$100m$
Data packet size	128 *bytes*
Packet interval rate	0.5 ($0.0625KB/s$)
Transmit power	$24.75mW$
Receive power	$13.5mW$
Idle power	$13.5mW$
Sleep power	$15\mu W$

The simulations were run with the simulation parameters of CC-MAC [13] and TRAMA [14] and are mentioned in Table 4.2. All the nodes are randomly deployed in an area of a 500m x 500m sensor field. Each packet has a fixed size of 128 *bytes*. We assume that one of the nodes is a sink and we select 16 source nodes randomly. Each source node reports its event information to the sink. CC-MAC investigates the effect of the traffic load by varying the reporting period of the sensor nodes. The reporting period determines the period each node creates packets about the event information. We also vary the reporting period of the sensor nodes in order to compare our protocol with CC-MAC and TRAMA. For all the experiments, data are collected from an average of 20 independent runs, keeping the base station in the center.

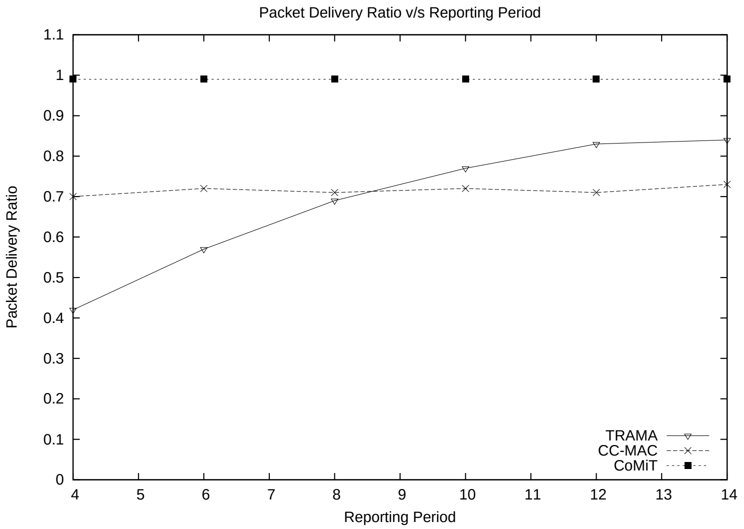


Figure 4.1: Packet-Delivery Ratio v/s Reporting Period

Figure 4.1 shows the performance comparison of our protocol in terms of packet-delivery ratio. CC-MAC has a packet-delivery ratio of 70%, while TRAMA achieves a packet-delivery ratio between 40% and 80%. CoMiT outperforms CC-MAC and TRAMA for high traffic load. Note that the packet-delivery ratio is insensitive to reporting time in the case of CoMiT and CC-MAC. It depends on the number of

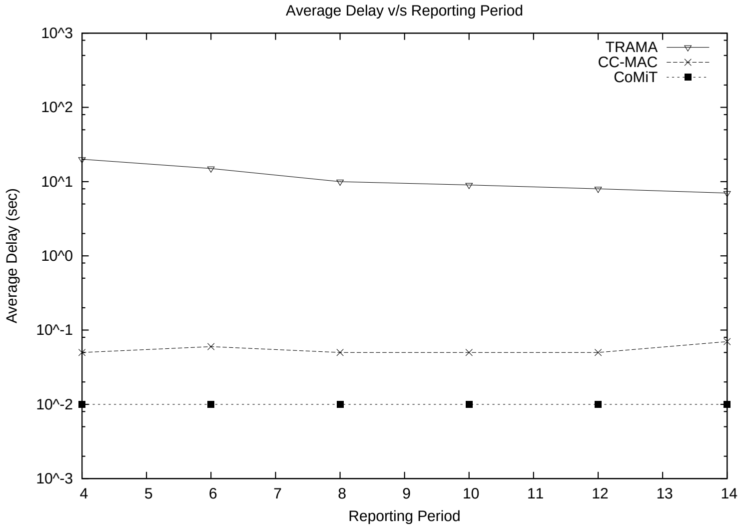


Figure 4.2: Average Delay v/s Reporting Period

nodes trying to access the medium. As CoMiT prevents spatially-correlated contention through grouping and scheduling, and disperses traffic through the forwarder selection mechanism, the packet-delivery ratio is significantly higher. CC-MAC has a high packet drop rate because all the nodes with event information contend for the medium for the first time in *First Contention Phase* to become a representative node for that region. If there are many nodes with event information, this leads to a greater potential for packets to be dropped. TRAMA has a varying packet-delivery ratio according to the traffic load. The scheduling approach of the protocol reduces collisions but as the load increases, the packet-delivery ratio decreases since the packets cannot be accommodated in the transmission slots.

Figure 4.2 shows the average delay achieved by each protocol. Delay performance of the three protocols is relatively constant with variable reporting time. CoMiT achieves the lowest delay of $10ms$ because the traffic is dispersed through the nodes that have minimal hop distances. CC-MAC has a delay of $50ms$ because of the initial

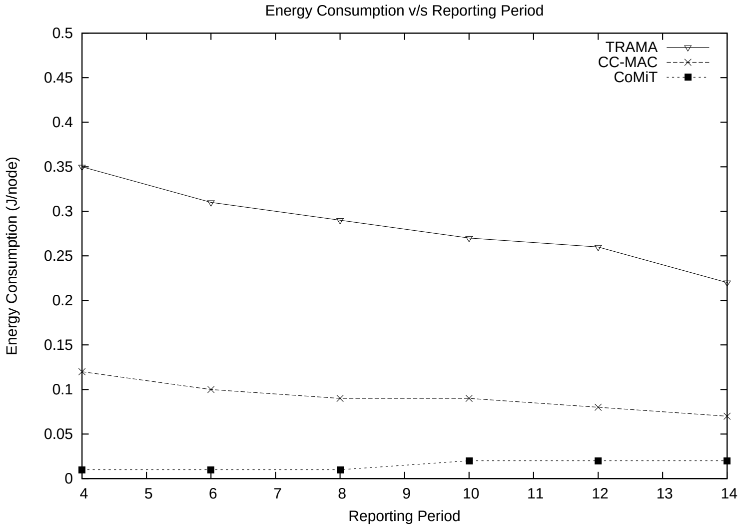


Figure 4.3: Energy Consumption v/s Reporting Period

delay that occurs in the *First Contention Phase.* TRAMA has a delay of 10s, which is due to the complex schedule-based medium accessed scheme.

CoMiT provides significant energy savings without compromising latency when compared to CC-MAC and TRAMA. Figure 4.3 shows the comparison of the three protocols in terms of energy consumption. With the help of grouping and scheduling, spatially-correlated contention is decreased, and as a result CoMiT achieves 84% less energy consumption compared to CC-MAC and 95% compared to TRAMA. CC-MAC has more energy consumption because it has to select representative nodes more frequently and as a result there is an increase in the number of collisions.

In our protocol, grouping the nodes helped in reducing spatially-correlated contention. Our protocol is consistent in providing a high packet-delivery ratio irrespective of the variable reporting time. By having a forwarder-selection mechanism that reduces packet drops, our protocol design makes it tolerant to highly congested network scenarios, thereby increasing the packet-delivery ratio.

4.3 CoMiT Protocol Evaluation

To evaluate the performance of our protocol with grouping, for base-line comparison, we compare our protocol in the case of not having grouping. We assess our protocol for packet-delivery ratio, average delay, and energy consumption, and also evaluate scalability in terms of traffic load.

Partitioning the network into many groups may be energy efficient but then delay can be a trade off due to the scheduling of all the nodes. In the same way, partitioning a network into less groups may give less delay, but the packet-delivery ratio might decrease. After thorough experimentation, we decided that partitioning the network into three groups may improve overall protocol performance in terms of packet-delivery ratio, energy efficiency, and delay.

4.3.1 Comparison without Grouping

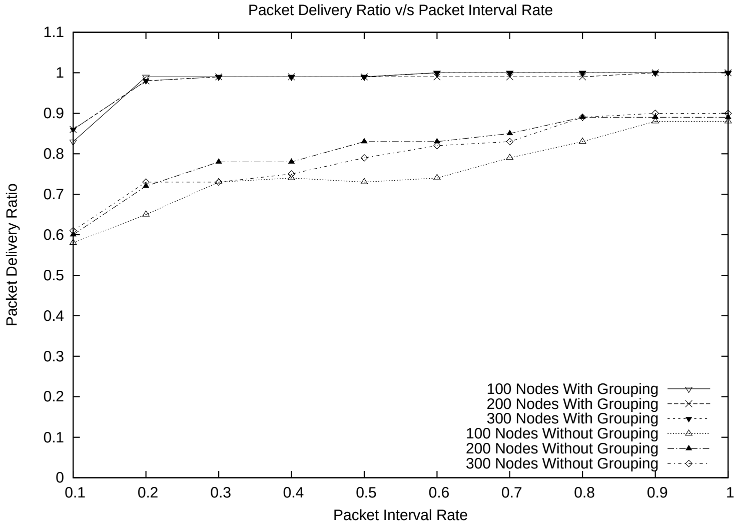


Figure 4.4: Packet-Delivery Ratio v/s Packet Interval Rate

Figure 4.4 shows that during high traffic load our protocol with grouping outperforms our protocol without grouping. This is because of the reduction in collisions due to the scheduling of groups. Our forwarder-selection mechanism assists in reducing congestion by dispersing traffic through alternate paths, hence improving packet delivery. Note that even with the increase in network density, there is not much change in the packet-delivery ratio when we employ grouping. However, without grouping, the network becomes highly unstable and the packet-delivery ratio falls during high traffic loads. This shows that our protocol is scalable in terms of network density and provides a high packet-delivery ratio.

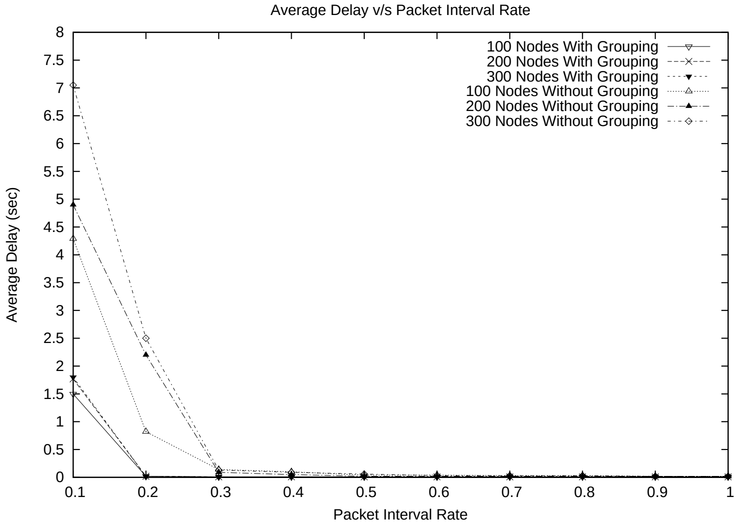


Figure 4.5: Average Delay v/s Packet Interval Rate

In Figure 4.5, the average delay of our protocol with grouping is lower than the delay incurred by our protocol without grouping. As our forwarder-selection mechanism assists in forwarding the packets through the shortest path based on hop count, the delay is lowered. However, we observe that with an increase in network

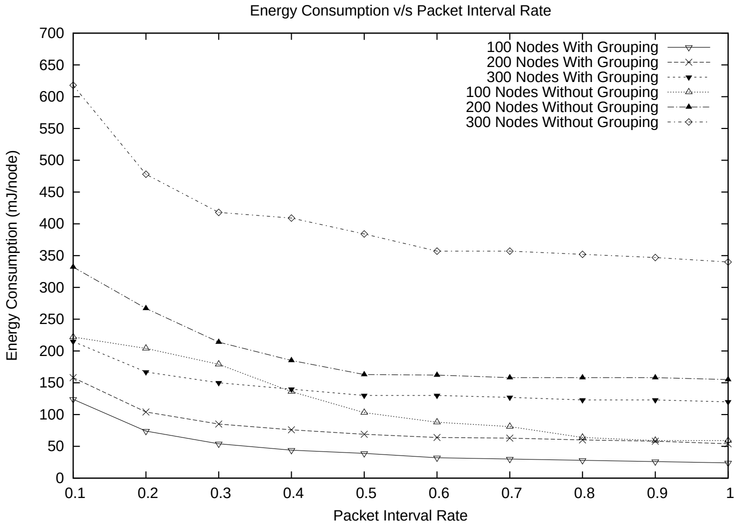


Figure 4.6: Energy Consumption v/s Packet Interval Rate

density as the number of hops increase, there is a slight increase in delay. This delay is significantly less when compared to our protocol without grouping.

Figure 4.6 shows that our protocol with grouping has more energy savings when compared to our protocol without grouping. With an increase in the network density, the energy consumption slightly increases as the number of nodes that act as forwarders increase. Our protocol with 300 nodes achieves 67% less energy consumption when compared to our protocol without grouping. These graphs show that the grouping-based approach with scheduling is energy efficient because of the decrease in spatially-correlated contention and collisions.

CHAPTER 5

CONCLUSION

Wireless sensor networks are used for monitoring purposes in various fields. As many sensors detect the same event, or when multiple events happen at the same time, sensor nodes forward the data to other nodes, which causes high traffic load and degrades the network performance by increasing collisions, congestion, delay, and energy consumption. Several techniques such as data aggregation, queue management, prioritizing packets, and etc. are used in some applications to reduce the congestion in the network. However, these techniques fail to address the problems caused due to spatially-correlated contention, such as loss of event reports that are being delivered to the base station and also excessive energy consumption. Hence, there arises a need for a protocol that mitigates congestion and also handles spatially-correlated contention. Also, as sensor nodes are energy constrained, energy efficiency is one of the primary concerns in designing protocols for these networks.

To reduce the collisions caused due to spatially-correlated contention, we developed a protocol in which we divide network into groups and employ scheduling among these groups. Each group works independently and each node in the group chooses its forwarder through a forwarder selection mechanism technique, thereby reducing collisions. For upstream traffic, several protocols have been proposed to alleviate the problem of spatially-correlated contention; [13] and [14] are some of the protocols that address this problem. However, in these protocols, scalability in terms of network density and traffic load was not addressed due to the complexity of their protocols. Unlike the proposed protocols, by adapting the grouping mechanism and a forwarder-selection mechanism that restricts the number of nodes acting as a forwarder to at most two, our protocol mitigates the congestion with improved packet delivery while maintaining energy efficiency and scalability.

Our simulation results show that this technique has a very good improvement in energy efficiency and an excellent packet-delivery ratio with very little delay even in congested scenarios. However, our scheduling structure requires tight synchronization among different groups. In general, tight synchronization would affect the performance of the network.

In the future, we would like to extend the grouping-based mechanism and implement internal scheduling to reduce spatially-correlated contention within the same group. This way network performance will be improved in terms of data delivery and energy efficiency.

BIBLIOGRAPHY

[1] C.Y. Wan, S.B. Eisenman and A.T. Campbell. "CODA: congestion detection and avoidance in sensor networks". *SenSys '03:Proceedings of the 1st international conference on Embedded networked sensor systems (2003)*, pp.266-279.

[2] Y. Sankarasubramaniam, .B. Akan and I.F. Akyildiz. "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks". *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '03, 2003.

[3] M. Zawodniok and S. Jagannathan. "Predictive Congestion Control Protocol for Wireless Sensor Networks". *IEEE Transactions on Wireless Communications, November*, 2007.

[4] S. Rangwala, R. Gummadi, R. Govindan and K. Psounis. "Interference-Aware Fair Rate Control in Wireless Sensor Networks". *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, 2006.

[5] C. Wang, K. Sohraby, V. Lawrence, B. Li and Y. Hu. "Priority-based Congestion Control in Wireless Sensor Networks". *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing -Vol 1 (SUTC'06) - Volume 01*, SUTC '06, 2006.

[6] C.T. Ee and R. Bajcsy. "Congestion control and fairness for many-to-one routing in sensor networks". *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, 2004.

[7] K. Yedavalli. "Using wireless advantage for congestion control in wireless sensor networks". *USC Technical Report*, October, 2005.

[8] P.A.K. Acharya, A. Sharma, E.M. Belding, K.C. Almeroth and K. Papagiannaki. "Congestion-Aware Rate Adaptation in Wireless Networks: A Measurement-Driven Approach". *Proceedings of SECON'2008*, pp.1-9.

[9] B. Hull, K. Jamieson and H. Balakrishnan. "Mitigating congestion in wireless sensor networks". *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004. SenSys '04.

[10] L. Popa, C. Raiciu, I. Stoica and D.S. Rosenblum. “Reducing Congestion Effects in Wireless Networks by Multipath Routing”. *Proceedings of the 2006 14th IEEE International Conference on Network Protocols*, 2006. ICNP ’06.

[11] S. Chen and N. Yang. “Congestion Avoidance Based on Lightweight Buffer Management in Sensor Networks”. *IEEE Transactions on Parallel and Distributed Systems*, 2006.

[12] K.K. Sharma, Dr. H. Singh and Dr. R.B. Patel. “A Hop by Hop Congestion Control Protocol to Mitigate Traffic Contention in Wireless Sensor Networks”. *International Journal of Computer Theory and Engineering*, vol. 2, no. 6, pp. 986-991, 2010.

[13] M.C. Vuran and I.F. Akyildiz. “Spatial correlation-based collaborative medium access control in wireless sensor networks”. *IEEE/ACM Transactions on Networking*, 2006.

[14] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy-efficient, collision-free medium access control for wireless sensor networks”, *in Proc. ACM SenSys 2003*, Los Angeles, CA, Nov. 2003, pp. 181192.

[15] Joa-Hyoung Lee and In-Bum Jung. “Adaptive-Compression Based Congestion Control Technique for Wireless Sensor Networks”. *Sensors* 2010, 10(4), 2919-2945; doi:10.3390/s100402919.

[16] R. Chakravarthi, C. Gomathy, S.K. Sebastian, K. Pushparaj and V.B. Mon. “A Survey on Congestion Control in Wireless Sensor Networks”. *International Journal of Computer Science & Communication*, Vol. 1, No. 1, January-June 2010, pp. 161-164.

[17] V. Deshpande, P. Sarode and S. Sarode. “Root Cause Analysis of Congestion in Wireless Sensor Network”. *International Journal of Computer Applications*, vol. 1, issue 18, pp. 31-34, 2010.

[18] Q. Pang, V.W.S. Wong and V.C.M. Leung. “Reliable data transport and congestion control in wireless sensor networks”. *International Journal of Sensor Networks 2008* - Vol. 3, No.1 pp. 16 - 24, 2008.

[19] F.B. Hussain, Y. Cebi and G.A. Shah. “A Multievent Congestion Control Protocol for Wireless Sensor Network”. *EURASIP Journal on Wireless Communications and Networking*, Volume 2008, 2008.

[20] J. Wang, S. Medidi and M. Medidi. “Energy-efficient k-coverage for wireless sensor networks with variable sensing radii”. *Proceedings of the 28th IEEE conference on Global telecommunications*, GLOBECOM’09, 2009.

[21] S.C. Ergen and P. Varaiya. “TDMA scheduling algorithms for wireless sensor networks”. *Journal: Wireless Networks - WINET* vol. 16, no. 4, pp. 985-997, 2010.

[22] Y. Zhou and M. Medidi. “Energy-efficient Contention-Resilient Medium Access for Wireless Sensor Networks”. *IEEE International Conference on Communications - ICC’07*, pp. 3178-3183, 2007.

[23] M.M. Monowar, M.O. Rahman, and C.S. Hong. “Multipath Congestion Control for Heterogeneous Traffic in Wireless Sensor Network”. *Advanced Communication Technology*, 2008. ICACT 2008.

[24] W. Fang, J. Chen, L. Shu, T. Chu, and D. Qian.“Congestion avoidance, detection and alleviation in wireless sensor networks”. *Journal of Zhejiang University-SCIENCE C (Computers & Electronics)*, vol. 11, no. 1, pp. 63-73, 2010.

[25] `http://en.wikipedia.org/wiki/Wireless_sensor_network`

[26] `http://en.wikipedia.org/wiki/Sensor_node`

[27] X. Qiu, D. Ghosal, J. Yick and D. Li. “Priority-Based Coverage-Aware Congestion Control for Multihop Wireless Sensor Networks”. *The 28th International Conference on Distributed Computing Systems Workshops*, ICDCSW ’08, 2008.

[28] The Network Simulator - ns-2. `http://www.isi.edu/nsnam/ns/`

[29] M.Z. Ahmad and D. Turgut. “Congestion Avoidance and Fairness in Wireless Sensor Networks”. *IEEE Global Telecommunications Conference 2008*, IEEE GLOBECOM 2008.

[30] H.M. Ammari and S.K. Das. “Scheduling protocols for homogeneous and heterogeneous k-covered wireless sensor networks”. *Pervasive and Mobile Computing*, Volume 7, Issue 1, Pages 79-97, 2011.

[31] Y. Zhou and M. Medidi. “Sleep-based Topology Control for Wakeup Scheduling in Wireless Sensor Networks”. *4th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks - SECON*, pp. 304-313, 2007.

[32] A. Keshavarzian, H. Lee and L. Venkatraman. “Wakeup scheduling in wireless sensor networks”. *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc ’06, 2006.

[33] M.O. Rahman, M.M. Monowar and C.S. Hong. “A QoS Adaptive Congestion Control in Wireless Sensor Network”. *10th International Conference on Advanced Communication Technology, 2008.* ICACT 2008.

[34] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu.“Upstream congestion control in wireless sensor networks through cross-layer optimization”. *IEEE Journal on Selected Areas in Communications*, 2007.

[35] Y. Ding, C. Wang and L. Xiao. “A connectivity based partition approach for node scheduling in sensor networks”. *Proceedings of the 3rd IEEE international conference on Distributed computing in sensor systems*, DCOSS’07, 2007.

[36] S. Al-Omari and W. Shi. “Availability Modeling and Analysis of Autonomous In-Door WSNs”. *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, 2007.

[37] `http://en.wikipedia.org/wiki/Network_congestion`

[38] S. Lindsey and C.S.Raghavendra. “PEGASIS: Power-Efficient Gathering in Sensor Information Systems”. *IEEE Aerospace Conference Proceedings*, 3:11251130, 2002.

[39] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks”. *ACM MobiCom*, pages 8596, 2001.

[40] Y Xu, J. Heidemann, and D. Estrin. “Geography-informed energy conservation for adhoc routing”. *ACM MobiCom*, pages 7084, 2001.

[41] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. “Optimizing sensor networks in the energy-latency-density design space”. *IEEE Transactions on Mobile Computing*, pages 7080, 2002.

[42] H. Chen, L. Cui, and V.O.K. Li. “A joint design of oppurtunistic forwarding and energy-efficient mac protocol in wireless sensor networks”. *IEEE GLOBECOM*, 2009.

[43] M. Dhanaraj, B. S. Manoj, and C. S. R. Murthy. “A new energy efficient protocol for minimizing multi-hop latency in wireless sensor networks”. *IEEE PerCom*, pages 117126, 2005.

[44] F. Aurenhammer. “Voronoi diagramsA Survey of a Fundamental Geometric Data Structure”. *ACM Comput. Surv.*, 23(3):345405, 1991.

[45] J. G. Leonidas, D. E. Knuth, and M. Sharir. “Randomized Incremental Construction of Delaunay and Voronoi Diagrams”, volume 7. Springer New York, 1992.