DYNAMIC DIFFUSION FOR CONGESTION AVOIDANCE IN WIRELESS SENSOR NETWORKS

by

Sri Divya Deenadayalan

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Computer Science

Boise State University

August 2012

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Sri Divya Deenadayalan

Thesis Title: Dynamic Diffusion for Congestion Avoidance in Wireless Sensor Networks

Date of Final Oral Examination: 15 August 2012

The following individuals read and discussed the thesis submitted by student Sri Divya Deenadayalan, and they evaluated her presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Murali Medidi, Ph.D.	Chair, Supervisory Committee
Sirisha Medidi, Ph.D.	Co-Chair, Supervisory Committee
Alark Joshi, Ph.D.	Member, Supervisory Committee

The final reading approval of the thesis was granted by Murali 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 family

ACKNOWLEDGMENTS

It is a pleasure to thank those individuals whose support and guidance has helped me to accomplish this thesis.

First and foremost, I wish to express my sincere gratitude to Dr. Sirisha Medidi, for her support and motivation throughout my thesis. My Master's degree would not have been possible without her effort, guidance, and encouragement.

My sincere thanks to Dr. Murali Medidi for his invaluable assistance and advice from the very early stage of my research. With his inspiration and great efforts to explain things clearly and simply, he changed the way I think today.

I gratefully thank Dr. Alark Joshi for his valuable inputs, and for serving as committee member to evaluate my thesis.

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

Lastly, and most importantly, special thanks to my husband, my parents, and my brother for their love and support.

ABSTRACT

Wireless Sensor Networks (WSNs) are employed for either continuous monitoring or event detection in the target area of interest. In event-driven applications, it is critical to report the detected events in the area, and with sudden bursts of traffic possible due to spatially-correlated events or multiple events, the data loss due to congestion will result in information loss or delayed arrival of the sensed information. Congestion control techniques detect congestion and attempt to recover from packet losses due to congestion, but they cannot eliminate or prevent the occurrence of congestion. Congestion avoidance techniques employ proactive measures to alleviate future congestion using parameters like queue length, hop count, channel conditions, and priority index. However, maintaining and processing such information becomes a significant overhead for the sensor nodes and degrades the performance of the network. We propose a congestion avoidance MAC protocol that uses the queue buffer length of the sensor nodes to estimate the congestion and diffuse traffic to provide a congestion-free routing path towards the base station. This protocol provides event reporting, packet delivery ratio, by dynamically diffusing the traffic in the network using multiple forwarders in addition to backup forwarding. We used the standard Network Simulator (NS2) to evaluate the performance of our protocol. Results show that our protocol significantly improves event reporting in terms of packet delivery ratio, throughput, and delay by avoiding congestion while diffusing the traffic effectively.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Wireless Sensor Networks

Recent technological advances have made use of small, inexpensive, low-power, distributed device that are capable of sensing, processing, and disseminating environmental data through wireless communication. Such devices are called wireless sensor nodes. These sensor nodes are equipped with a low-power radio transmitter, different sensors, a small battery unit, limited memory, and a microcontroller. A Wireless Sensor Network (WSN) [1] is a group of these self-organizing sensor nodes that cooperatively monitor the area of interest. The power of wireless sensor networks lies in the ability to install these sensor nodes which can coordinate among themselves to monitor the given physical environment. Unlike other networks, sensor networks depend on a dense deployment of the sensor nodes and coordination among them for successful data transmission.

Sensors usually communicate the sensed information to each other using a multi-hop approach and the flowing data end at a special node called the Base Station (commonly known as the sink). In some cases, the sink will query the sensor nodes for the required information or dispatch any information to all the sensor nodes in the networks. These sinks have better capability over simple sensor nodes, since they must do more complex data processing of the sensed information. Although WSN research

was initially developed for military applications, the usage scenarios for the wireless sensor networks range from intelligent battlefield, to target tracking, to monitoring of changes in environmental conditions, to ubiquitous computing environments, to health or equipment monitoring. They are also used for controlling the actuators that extend control from cyberspace into the physical world.

When the exact location of the incident is unknown, distributed sensing allows for placement of the sensors closer to the phenomenon than a single sensor would permit. Also, in most of the cases, multiple sensor nodes are required to overcome the environmental obstacles like obstructions, line of sight constraints, etc. to obtain detailed measurements of the particular environment in an unobtrusive manner. Usually the environment to be monitored does not have an existing infrastructure in terms of energy or communication. It is very important for sensor nodes to survive on small, finite sources of energy and communicate through a wireless communication channel. Due to these limiting factors, wireless sensor networks are required to have a greater number of nodes deployed and these nodes do not have an individual identity. Another important requirement of wireless sensor network is to avoid the collisions of the data packets which happens when packets from two or more closer nodes attempt to transmit at the same time. For example, in an environmental monitoring system used to detect harmful gas in a chemical plant, hundreds of sensor nodes can be scattered over an area that supports low data rate periodic sensing. In case of unpredictable bursts of traffic by the correlated events or multiple events sensed, the high data rate can easily cause congestion problems especially at intermediary nodes located closer to the sink. Congestion happens at a node by dropping the data packets that cannot be accomodated in the nodes' queue with limited capacity. Another important cause of network congestion even under periodic low data rate

traffic is the variation in the radio channels and concurrent data transmissions over different radio links that interact with each other.

1.1.1 Congestion in WSN

Congestion is detrimental to wireless sensor networks because it lowers the throughput of the network by dropping more packets containing critical sensed information and reduces the lifetime of the network due to decreased energy efficiency at each sensor node, especially for spatially-correlated events. With the buffers of the sensor nodes close to full, there will always be traffic at the node for the data packets, which results in increased contention, increased retransmissions, decreased packet delivery ratios, and increased energy consumption. As a result, data loss due to congestion may ultimately threaten the benefits of the WSN: like throughput, packet delivery ratio, latency, and energy efficiency. In event-driven applications, when there is a sudden increase in the traffic, congestion would degrade the performance of the network by the loss of the event packets or the delayed arrival of the packets to the sink. Congestion control is not only important to improve the overall throughput but also to lengthen the network lifetime and improve the end-to-end throughput, called accuracy level, by avoiding the packet loss due to congestion. Congestion, being one of the biggest problems for a sensor network, has to be avoided to improve the Quality of Service (QoS) in terms of throughput, packet delivery ratio, latency, and energy efficiency.

Congestion control in WSN has been widely about detecting the congestion in the network and controlling the congestion by adjusting the rate of the input traffic, or prioritization of the data packets, or load balancing among the sensor nodes. The traffic in the network is adjusted either hop-by-hop, at each sensor node, or end-to-end rate adjustment at the source nodes where the traffic is generated. While

congestion control concentrates on enabling the network to recover from packet loss due to the occurrence of congestion, congestion avoidance detects incipient congestion or estimates for the congestion in the network and tries to prevent its occurrence. For example, in an event-based approach, suitable congestion avoidance mechanism could help to detect the approaching congestion and react to the situation before the actual collapse takes place. Congestion avoidance is the core concept for this thesis model to proactively identify and alleviate congestion in the network and adjust the network to handle the future traffic.

1.2 Organization of Thesis

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

CHAPTER 2

RELATED WORK

In recent years, a number of research works have been studied for congestion problems in wireless sensor networks and have proposed different approaches to handle it. Congestion in a wireless sensor network is either controlled or avoided for improving the data transmission in both continuous monitoring and event-reporting applications. Rate control of the generated packet and traffic dispersion are the common approaches to reduce congestion in the network. Though these approaches try to improve data transmission, they suffer from other problems like delayed data arrival, reduced sensor nodes energy, and overhead of processing information for the sensor nodes. Some of the protocols that have been proposed to reduce congestion in WSNs can be broadly classified as

1. Congestion Detection and Avoidance
2. Rate Control
3. Routing
4. Medium Access Control

2.1 Congestion Detection and Avoidance

WSNs consist of a large number of sensor nodes densely deployed in the areas of interest. On detecting the event, sensor nodes generate packets and forward them to

the base station with the help of the neighboring sensor nodes. At the base station, these packets are processed and necessary action is taken. This clearly shows the importance of a timely delivery of data packets without loosing the information. But due to the limited capacity of sensor nodes and the error prone nature of WSNs, the data transmission is delayed or lost. Congestion is one of the major factors for these data losses in the network that must be handled. Several protocols have been proposed for detecting the congestion in the network and taking necessary action to avoid it.

Congestion is classified into two main types [30], node-level congestion and link-level congestion. Node-level congestion occurs at a node when its queue buffer overflows with packets, and link-level congestion occurs when multiple nodes try to access a common transmission medium. While node-level congestion causes packet losses and leads to retransmissions, link-level congestion increases packet service time and decreases the link utilization.

There are several congestion detection and avoidance [30] protocols that have been developed. One of the popular congestion avoidance schemes is CODA [33], which is an upstream congestion mitigation strategy where the congestion detection mechanism is based on queue length at the intermediate nodes and channel load. If buffer occupancy or wireless channel load exceeds a threshold, it implies that congestion has occurred. CODA involves three different strategies: congestion detection, open-loop hop-by-hop backpressure and closed-loop end-to-end multisource regulation. In this approach, when congestion is detected, a backpressure message is sent to the neighboring nodes to indicate that no more packets should be sent until an indication to resend is sent. The nodes send the message to the next nodes to stop sending the data packets. The open-loop backpressure message is designed to

handle short-term transient hotspots. In this situation, CODA enforces a change in transmission policy whereby all packets must be ACK'ed before another packet can be sent. This congestion detection mechanism tries to converge with backpressure messaging, which increases the delayed arrival of the packets at the base station.

Another popular protocol, ESRT [26], provides event-level reliability from sensors to sink by controlling the congestion in the network. The current network state is what determines the current congestion condition in the network and end-to-end source rate adjustment is done to achieve the perceived reliability at the base station. It has been tailored for use in sensor networks with adaptibility to dynamic topology, collective identification, energy conservation, and biased implementation at the base station. Reliability is measured by the number of data packets received at the base station. The end-to-end source rate adjustment in ESRT follows two basic rules: if the current reliability perceived at the base station exceeds the desired value, ESRT will multiplicatively reduce the source rate. Otherwise, the source rate is additively increased if the required reliability is not met, unless there is congestion in the network. To detect the current state of the network, the base station must be able to detect congestion in the network. The sensor nodes detect congestion using the queue buffer size and set the congestion notification bit. Once the base station receives a packet with its bit set, it knows that congestion took place and will update the reporting frequency accordingly. ESRT does not support end-to-end reliable data delivery and it is impractical to vary transmission rates of the nodes depending on the applications.

Compared to these end-to-end congestion control protocols there are also several hop-by-hop congestion control protocols [17, 27, 39, 12, 23] where congestion is controlled at each hop level of the network instead of the base station, and these protocols

show improvement in performance and faster congestion control. Congestion is also detected based on packet loss rate [9], packet service ratio [20], and the measure of congestion degree which is obtained from packet inter-arrival time over packet service time [35, 36]. The combination of multiple parameters like buffer size at the node, hop count, current channel busy ratio, and the MAC overhead have been used in [28] to control congestion by setting the congestion bit when the node rank calculated from these parameters exceeds the threshold. If the congestion bit is set, the downstream node calculates the Rate Adjustment Feedback based on the rank and propagates this value upstream towards the source nodes. The source nodes will adjust their transmission rates dynamically based on this feedback. It is an overhead for the sensor nodes to compute all these values periodically to identify the congestion and the final transmission rate is controlled at the source instead of individual sensor nodes.

Hop-by-hop congestion control [20], based on packet service ratio, is used to measure the congestion level at each node, which is the ratio of average packet service rate and packet scheduling rate. The output rate of a node is adjusted by adjusting the scheduling rate. In [2], the authors have used various parameters like received signal-to-noise ratio, relay traffic, buffer length, and energy level of the nodes to determine if it can participate in the communication or not. The objective of the cross-layer protocol is highly reliable communication with minimal energy consumption, adaptive communication decisions, and local congestion avoidance. Using this initiative, the cross-layer module (XLM) performs congestion control, hop-by-hop reliability, and distributed duty cycle operation. CONSEQ [3] is also a cross-layered congestion control mechanism where congestion is estimated based on queue length and the channel conditions at an one-hop neighborhood. Based on the

estimate, each node dynamically adapts its packet transmission rate and balances the load among its one-hop neighbors to avoid creating congestion and bottleneck nodes. Here each node employs CONSEQ to estimate the degree of congestion in its on-hop forwarder set and accordingly control its rate of scheduling packet transmission to the forwarder set and adapt load balancing decisions. Each node computes a virtual queue length for each node in its forwarder set and the load is balanced among the forwarder set based on the virtual queue value, thereby decreasing the packet scheduling rate.

PCCP [35] is another cross-layer optimization where congestion is detected based on the congestion degree and utilizes a node priority index. PCCP consists of three components: intelligent congestion detection, implicit congestion notification, and priority-based rate adjustment, which are all part of the basic congestion control algorithms. If the sink wants to receive more detailed information from a certain set of sensors, the corresponding sensors will receive higher priority. RCS [38] employs prioritized queuing to provide service differentiated, soft, real-time guarantees where there are multiple queues maintained at the nodes for service differentiated applications.

2.2 Rate Control

There are many research works [13, 24, 8, 16, 14, 25] where congestion is detected and the data flow rate at the source or at each intermediate node is adjusted to control congestion in the network. A token bucket scheme is used in [13] to regulate each sensor's send rate. A sensor accumulates one token every time it hears its parent forward packets. The sensor is allowed to send only when its token count is above zero, and each send costs one token. WRCP [31] is an explicit and precise distributed

rate-based congestion control protocol that associates capacities of the links instead of the nodes. The available capacity at each node is based on the receivers capacity, rate of the forwarding packets from neighbors, number of flows, and the set of neighbors. LACAS [19] employs a simple autonomous learning machine, called automata, which is placed at each node to control the packet rate flow at the intermediate nodes based on the probability of the number of packets that are likely to get dropped. The authors approach is to make the processing rate at the nodes equal to the transmitting rate, so that the occurrence of congestion in the nodes is seamlessly avoided. The rate control technique in [13] involves three mechanisms: hop-by-hop flow control, rate limiting source traffic, and a prioritized Medium Access Control (MAC) protocol. In the first mechanism, congestion is detected based on the queue size and is signaled to the other nodes. In the second approach, each sensor node uses a token bucket scheme to regulate the sending rate after monitoring the current transit traffic. Finally, the third technique gives a backlogged node priority to the sensor node over the non-backlogged sensor node to provide access to the shared medium. RCRT [16] is an end-to-end rate control protocol where loss recovery is used to detect congestion and end-to-end reliability is achieved by rate adaption and rate allocation mechanisms. In WCP [25], congestion is detected using the exponential weighted moving average of the queue size and the congestion information is shared among the neighbors. The WCPCap in [25] uses a distributed rate controller for estimating the available capacity within the congested region and distributing it fairly among the relevant flows. In CONSEQ [3], each node adjusts the rate of packet injection to the underlying MAC layer to avoid congestion in the forwarder set using the fuzzy rate controller. All of these approaches detect or estimate congestion, notify the neighboring nodes, and adjust the rate of traffic flowing through the nodes.

2.3 Routing

Instead of the geographic routing scheme where the packets can be routed only to the neighboring node with the shortest distance to the destination, the routing schemes developed in recent times can find the optimized path to the destination based on various factors. Congestion is highly vulnerable in geographic routing schemes where there is only one defined forwarder for a sender node. In [3, 18, 40, 38], the authors have used dynamic forwarding to avoidance congestion by balancing the load in the network. In ECR-MAC protocol [40], the Dynamic Forwarding Scheme (DFS) algorithm selects the forwarder dynamically based on the distributed duty cycles of the sensor nodes. DFS assigns multiple potential forwarders for a sender and each forwarder employs independent wake-up schedules without synchronization to reduce the protocol overhead. Instead of waiting for a particular forwarder, each sender hurls packets as quickly as possible to any one of the nodes termed as potential forwarder that can help transmit packets. This modified MAC protocol handles spatially-correlated contention efficiently and scales well with network density. HMAC [18] is also a modified MAC protocol that uses source count value to decide the node that gets more access when compared to others. The weighted round robin forwarding implements hop-by-hop fair packet scheduling to guarantee that upstream nodes will transfer their weighted share amount of packets. In all these approaches, the forwarders are dynamically chosen from the available one-hop neighbors of a sensor node, which helps to reduce packet drops due to collisions and achieves more packet delivery ratio. CaEe [10] is a routing protocol where the in-network storage model uses the sleeping nodes as data buffers to avoid data loss from congestion. In this protocol, once the buffer of a sensor node reaches a threshold limit, then the cluster

head node selects another sleeping node as data buffer to which the data will be redirected.

PSFQ [34] is a transport protocol that addresses reliable communication from sink-to-sensor nodes. It consists of three operations: pump, fetch, and report where packets are injected slowly into the network and perform a hop-by-hop recovery in case of packet loss. Each intermediate node should maintain a data cache to be used for the in-sequence data delivery and local loss recovery. RMST [32] guarantees successful transmission of packets in the upstream direction using the concept of directed diffusion. The two modes: cached and non-cached are used for caching the packets to recover from the losses. There is always an overhead to cache the packets and provide recovery in case of packet losses. When compared to these protocols, the authors in [29, 17, 37] have developed reliable hop-by-hop transport protocols.

In some of the recent works [4, 11, 22], routing protocols have been developed that can avoid packets flowing through the congested area in the network. In order to avoid packet drops at the hotspots, route discovery is performed dynamically by selecting the path that is loss-free and is also the shortest path to the sink. Expected Transmission Count (ETX) [7] is the metric to find the high-throughput path to transmit the packets where the chance of link loss is less. In [22], the congested zone, conzone, is identified from the area that generates high priority packets and the nodes in this area will mark themselves as on conzone nodes. CAM [4] introduces the Relative Success Rate (RSR) of each neighboring node that is periodically broadcasted to avoid the congested nodes. RSR is the ratio of the number of packets transmitted from the MAC layer to the number of packets forwarded from the network layer over a small period of time. This RSR at the application layer is also used for determining the data transmission rate adjustment. All these different routing protocols need

periodic information from their neighboring nodes to choose the best route to the sink.

2.4 Medium Access Control

The Medium Access Control (MAC) sublayer of data link layer controls which sensor node will participate in the communication at any point of time. It provides flow control and error control to provide reliability in the wireless sensor network. The fundamental mechanism to access the medium, which is the Distributed Coordination Function [5], is a carrier sense multiple access with collision avoidance (CSMA/CA) scheme of IEEE 802.11 MAC. The four-way handshake technique of 802.11 MAC, known as RTS/CTS/DATA/ACK, has been widely used as a standard mechanism for data transmission to avoid contention among the nodes in the shared medium. The RTS/CTS technique increases the performance of the network by reducing the collision in the network when multiple sensor nodes compete to use the channel at the same time.

Routing protocols are usually based on a criteria (e.g., number of hops) to choose the best optimal path to the sink. Receiver contention based dynamic forwarding in used in [38] for convergecast packet routing. The dynamic forwarding process is combined into the RTS/CTS exchanging period of real-time MAC design. Here in real-time MAC design, if a sender wins during a polling contention period and gains the channel access after the exponential back-off period, it will initiate a RTS transmission containing its own group ID. All the nodes within the transmission range will overhear this RTS message and enter the receiver contention period. In the receiver contention period, only the sensor nodes with the same or lower group

ID, become the qualified next-hop forwarders.

Anycasting is method of routing where a source node with data packets can send the packets to any node that belongs to a given set of destinations. MAC-layer anycasting is a scheme that utilizes the knowledge of the current channel condition to select the next downstream neighbor in the set for each data transmission. As mentioned in [6], the routes chosen by the network layer are optimal on a longer time scale, and ignore the possibility of transient variations in link conditions. Here the anycast group is formed at the network layer and is handed to the MAC layer along with the packets. The MAC layer decides the optimal path from the anycast group neighbors based on the channel condition. IEEE 802.11 DCF is also used to exploit the path diversity [15] to select the best next hop to forward the packets. Here the authors have extended the 802.11 MAC to support for anycast routing. The CTS reply from the group of anycast nodes are timed based on the shorter path to the destination and least numbers of packets waiting at the interface queue of the next-hop node. This modified 802.11 MAC has been proved to provide better packet delivery ratio relative to the standard 802.11 MAC in a variety of ad-hoc network models but will increase the delay in the network.

CHAPTER 3

DYNAMIC DIFFUSION MEDIUM ACCESS CONTROL

3.1 Motivation and Design Consideration

In an event-driven WSN application, data packets are generated by several source nodes when events occur in the areas of interest. In case of spatially-correlated events or multiple events, the network will generate a high volume of data packets containing the event information and increase the traffic in network. The data generated from detecting these events are of utmost importance, and loss of such data can violate the purpose of deploying a sensor network for event reporting. Congestion of data packets at the sensor nodes is likely to occur during this crisis period, which is detrimental to sensor networks because it lowers the packet delivery level by dropping a lot of data packets that contain critical sensed information. This will lead to events not being reported at the base station. There are many sources for congestion: sensor nodes' buffer overflow, concurrent data transmission, packet collision and many to one traffic nature. Congestion is a critical problem in WSNs because it causes packet loss, which in turn reduces throughput and energy efficiency, and increases packet delay. Most importantly the detected event might not get reported at the base station. Therefore congestion in WSNs need to be avoided to improve event reporting and Quality of Service (QoS) in terms of packet delivery ratio, throughput, latency, and energy efficiency.

Congestion control in WSN has been widely about detecting the congestion in the network and controlling the congestion by adjusting the rate of the input traffic, or prioritizing of the data packets, or load balancing among the sensor nodes. Traffic in the network is adjusted by either hop-by-hop at each sensor node or by end-to-end rate adjustment at the source node where traffic is generated. Previously proposed traffic control strategies in [13, 26] suggest some mechanisms that might be unsuitable for transient congestion and the traffic in network to get quickly adjusted during a sudden burst of traffic. Once the congestion is detected, usually by monitoring the sensor node's queue buffer, control messages are transferred among the nodes based on which transmission rate or load is balanced to ease the current congestion at the nodes. The network takes more time to converge to a congestion free state and eventually the detected event is also reported late. While congestion control concentrates on enabling the network to recover from packet loss, congestion avoidance detects incipient congestion and prevents its occurrence. For example, in an event-based approach, suitable congestion avoidance mechanism could help to detect the approaching congestion and react to the situation before actual collapse takes place. Applications requiring high data-rate can easily cause congestion problems especially at intermediary nodes located closer to the sink. Suitable congestion avoidance schemes could help detect approaching congestion and reduce sending data rates before congestion collapse occurs. Congestion avoidance is the core concept of Dynamic Diffusion Medium Access Control (DDMAC) protocol to proactively identify the approaching congestion in the network to alleviate it and adjust the network to handle future traffic.

Congestion is detected in the network when the sensor nodes' queue overflows and packets start to drop. The node is said to be congested and cannot handle any

further packets until its buffer starts to clear. Many congestion avoidance mechanisms [2, 3, 32, 35] use queue buffer to identify congestion in the network. In [2] protocol, the initiative determination is computed for each RTS packet using four different parameters: received signal-to-noise ratio (SNR) of an RTS packet, relay packet rate, buffer size, and remaining energy of the node. The combination of all these parameters will determine if a node can participate in the data transmission. Computing these values for each RTS packet is definitely an overhead for the sensor nodes and will degrade the entire network lifetime. CONSEQ [3] balances the traffic load based on the channel conditions and queue buffer availability among the one-hop forwarder set. In addition to the node's queue, each node also maintains a neighbor queue for through-traffic packets. The queue buffer availability is based on the computed virtual queue length of each node. When a node i has a data packet to transmit, it computes the virutal queue length for all its forwarder set nodes, based on the number of packets in node i*'s* queue, the number of packets in the neighbor's queue, and the number of packets dropped by node i due to an excessive number of retransmissions after the most successful transmission. Though this protocol does not use any additional control messages, it has to maintain multiple queues and compute the virtual queue length for all forwarders for each data packet. The main objective of this thesis is to avoid control messages and unwanted computation that will degrade the network performance. When a source node wants to send data packet to any node that belongs to a given set of destinations, it is called anycasting

In this thesis, the *queue buffer length* is used as an important parameter to identify the congested nodes in the network. This can help to dynamically choose an alternate, better path for data transmission, and also by using only the queue buffer length as a measure, the overhead of computations at a node can be eliminated. Unlike

other protocols where the sender node chooses its receiver node based on the current congestion level, the DDMAC protocol provides the option for each node to decide if it can participate in the data transmission. With this approach of anycasting, the response time for a node to reply could be completely avoided.

The data transmission decisions of anycasting could be handled at the MAC layer. This will make the DDMAC protocol very light weight and more efficient. Further, instead of having one forwarder for data transmission, there are multiple potential forwarders for each node. This could provide the opportunity for the data packets to get transmitted in a congestion-free path much faster. Since the forwarders get the opportunity to make the participation decision, the proposed protocol can help to alleviate congestion in the network efficiently by managing the data transmission at each hop level dynamically. In order to improve event reporting, a backup mechanism could support data transmission when all the potential forwarders are not available. Overall, the proposed protocol attempts to improve event reporting by proactively identifying congestion at the nodes and provides congestion-free paths for the data packets. In the following sections, the design challenges for this thesis are identified and how the protocol addresses these issues are described.

- **Link Failures:**

There are several reasons for packet losses in WSNs. Due to errors in links between two nodes, packets may not be delivered. These errors can occur due to signal attenuation. 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. Packet losses could also occur when two nodes try to transmit data simultaneouly. When two nodes try to send data packet

at the same time, they may collide and packets from either of the nodes might get dropped. In order to provide reliable event reporting, the designed protocol should have an ability to recover packets in case of link failures.

- **Node Failures:**

Due to a drop in energy level or by any other unforseen events, nodes in a sensor network are subject to 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 ensure packet-level reliability or event reporting, the protocols should be designed in such a way that packets being dropped should be identified and retransmitted. This will help mitigate packet losses, and thereby increases event reporting.

- **Congestion:**

When the rate of generation of events is more than the rate at which nodes forward the data packets, congestion occurs in the network. The network will have more traffic flowing and the sensor nodes will start buffering the packets if they are not able to transmit them immediately after receiving. However, since the buffer size of a sensor node is limited, any packet that arrives at the time when the buffer is already full will be dropped. 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 the infrastructure to handle the network in congested scenarios.

- **Packet Loss Identification:**

Detecting packet loss can be done at various levels. Nodes sending data packets can detect packet loss by using ACK/NACK messages sent by receivers. Receivers

can detect packet loss based on timers or by means of packet sequence number (*i.e.,* whenever a node receives an out of sequence numbered packet, it assumes the expected sequence packet is lost). Packet loss is also detected at the MAC layer, when a packet is dropped even after several retransmissions are made. The protocol that provides event-level or packet-level reliability must identify the packet losses as it enables the protocol to recover from the lost packet.

• **Scalability:**

As sensor networks contain a very large number of sensor nodes, networks should be scalable enough to provide event reporting. The hardware scalability involves sensing, communication bandwidth of the radio, and power usage. Whereas software scalability involves reliability of data transfer, and management of large volume of data. The protocol need to be distributed in nature in order to reduce the overhead caused in the case of very high traffic.

Considering all the above challenges, in the proposed protocol to alleviate congestion in WSNs for improved event reporting and better network performance, we use the following standard measures to evaluate performance. We also compare our protocol's performance with CONSEQ [3] which monitors queue buffer to avoid congestion in the network.

• **Packet Delivery Ratio:**

In order to measure the event reporting in terms of the packet delivered, the ratio of the number of packets successfully delivered to the base station to the total number of packets originating from all sources is measured. This packet delivery ratio gives the measure of successful event reporting by alleviating congestion in the network.

- **Throughput:**

The network performance in terms of throughput is critical. Throughput as a measure of network capability in delivering the data packets will be calculated over a period of time. A high throughput implies better performance of the protocol.

- **Delay:**

Another standard metric for evaluating network performance is delay. Depending on the nature of the applications using sensor networks, delay in the network plays a crucial role. In event-reporting applications, an event detected needs to be reported to the base station in real-time. Average delay is measured to identify the latency in forwarding the packets to the base station.

3.2 Assumptions

The following assumptions are made in the proposed protocol while considering the network for alleviating congestion to improve event reporting:

- The network is densely deployed to report any event to the base station.

- All nodes know their one-hop forwarder information by local broadcast mechanism.

- For simplifying the explanation, the network deployment does not have any physical holes and the outer boundary is identified.

3.3 Forwarder Configuration for Traffic Diffusion

3.3.1 Potential Forwarders Setup

Once the sensor nodes are deployed in the required area of interest, the potential forwarders for each node in the network is set. Traditionally, there is one forwarder or receiver for a sensor node in the same transmission radius and it is usually one hop level closer to the base station. The node with a data packet should always depend on the availability of this single forwarder. In situations like queue buffer full or link broken, the data transmission to this forwarder will not be possible and the packets will be held at the sender node. This is one of the main issues in event-reporting applications that interrupts the critical event report transmission to the base station and lowers the network reliability.

The DDMAC protocol addresses the issue with single forwarder by adopting *multiple potential forwarders* for each node. By providing multiple potential forwarders, the data packets can be transmitted through any possible forwarder without being dropped at the sender. The sender node does not have to depend on the availability of a single forwarder and thereby the events can be reported much faster. Overall, the multiple forwarder setup has many advantages compared to the single forwarder setup, like increased network reliability, reduced congestion, and increased Quality of Service (QoS).

The potential forwarders configuration procedure involves the following steps:

1. Identifying one-hop neighbors.

2. Choosing the forwarder set from the neighbors.

The base station is located at the top-left corner of the deployment area and all the sensor nodes know their exact location in the area. The sensor nodes also have the knowledge of its hop distance from base station. With this information, the one-hop neighbors are identified. Each sensor node initially identifies all its one-hop neighbors based on transmission radius (i.e, the ones that are within the transmission radius). In order for these neighbors to be active forwarders, they should be closer to the base station and also one hop level above the node. Now, from the list of one-hop neighbors, the neighbors that are closer to the base station are identified using the hop distance to base station. All other nodes are avoided from being chosen as potential forwarders.

In message exchanging, a node, in order to find its forwarders, will broadcast control messages and wait for reply messages. This involves more message transfers among the nodes, which increases the setup latency and reduces the node's energy. In the DDMAC protocol, the setup latency is minimized by avoiding such message transfers. From the reduced list of one-hop neighbors, four neighbors are chosen based on their hop distance to the base station. The sensor nodes closer to the base station or within the transmission radius of the base station, which are basically in one-hop distance, have only base station as its forwarder/receiver. Some of the sensor nodes, due to their location, might not be able to find four forwarders and might even have a single forwarder. With atmost four potential forwarders, the network performance is improved by providing multiple data paths for transmission. The data packets, instead of being dropped at a node due to the unavailability of a forwarder, can be transmitted to the base station through any potential forwarder.

To better illustrate the forwarder setup, consider the Figure 3.1. The hop levels are identified for the sensor nodes with level *i-2* being the closest to base station and

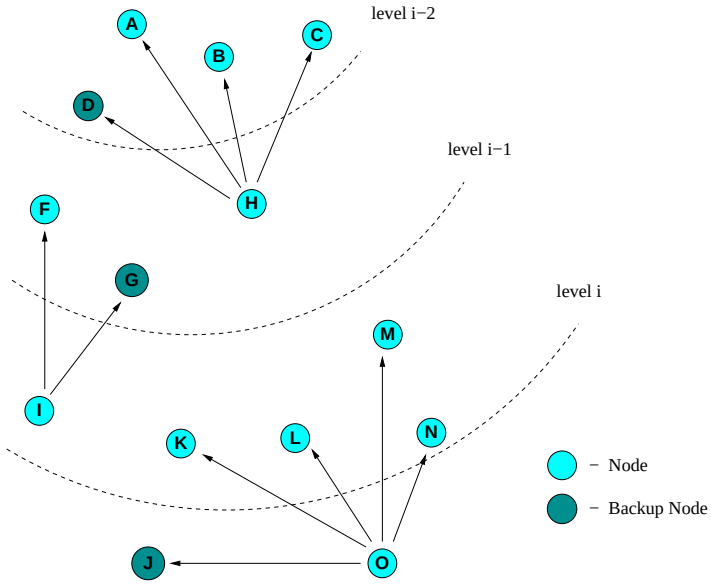


Figure 3.1: Potential Forwarders and Backup

level i being the farthest from base station. There are nodes labelled from A to O that can sense the events and forward the data packets. For the sensor node H, there are three potential forwarders: A, B, and C. For node I, there is only a single potential forwarder F. And the node O contains the full set, K, L, M, and N, as potential forwarders. Based on the location of the sensor nodes and its transmission radius, the number of potential forwarders will vary. Since the area for event monitoring can be anywhere in the network, the sources are randomly chosen and hence a source node is also a potential forwarder. Similar to these sensor nodes, all other nodes in the network will configure their potential forwarders as part of the initial setup. Each of these forwarders could provide different paths to the base station. The choice of selecting a forwarder dynamically is explained in later sections.

The protocol could reduce contention by avoiding all the potential forwarders involved in the data transmission. With fewer nodes contending for the channel, the contention in the network could reduce as compared to all the potential forwarders

involved in the data transmission. The potential forwarders setup promotes dynamic data path selection based on the availability of the forwarders. This mechanism does not need any data path construction, which incurs more setup latency. Allowing each node to have mutiple forwarders not only reduces congestion and contention, but also achieves shorter delay since it can significantly reduce the latency at each hop level.

3.3.2 Backup Forwarder Setup

In addition to potential forwarders, each node also has a backup forwarder. A backup node is used only when none of the potential forwarders are available for data transmission. This node is chosen similar to the potential forwarders, in addition to being the farthest among the one-hop neighbors. Each node knows its distance to its one-hop neighbors based on its location in the deployment. With the help of this information, the farthest one-hop neighbor is chosen as a backup node. This process could ensure reliability of the event reporting. In situations when all the potential forwarders are congested, usually because of a queue buffer being full, the forwarders in the higher hop level will also be unavailable. Trying to wait for the successive forwarders to become available might take more time and the event reporting will be delayed. Instead, if a node uses the backup forwarder, which is located far away, and if any potential forwarders for the backup forwarder is available then the data packets will be moved further. In Figure 3.1, the backup node is set for each of the sensor nodes. Node D is the backup forwarder for H, G is the backup forwarder for I, and J is the backup forwarder for O. When a sensor node is able to find only four potential forwarders and no possible backup forwarder in the hop level above, the backup forwarder can also be chosen in the same hop level. This can be seen in Figure 3.1, the backup forwarder J is in the same level as node O. Though this backup

forwarder setup in the same hop level is very rare, the nodes still try to have a backup. This ensures data transmission without packets being held at the intermediate nodes due to the unavailability of potential forwarders and thereby relieving congestion in the network.

3.4 Medium Access Control Enhancement

The Open Systems Interconnection (OSI) model places the responsibility for channel access in the medium access control (MAC) sublayer of the data link layer. The MAC layer controls medium access among the nodes, but it also offers support for roaming, authentication, and power conservation. The IEEE standard 802.11 MAC specifies the most famous family of WLANs, which offers services in wireless networks. The three basic access mechanisms have been defined for 802.11 MAC: the mandatory basic method based on carrier sense multiple access with collision avoidance (CSMA/CA), an optional method of avoiding the hidden terminal problem, and finally a contention-free polling method for time-bounded service. The first two methods are also summarized as Distributed Coordination Function (DCF). The 802.11 MAC protocol ensures access mechanism based on CSMA/CA, which is a random access scheme with carrier sense and collision avoidance through random backoff. The hidden terminal problem is another issue that is handled at the MAC layer. It occurs if one sensor node can receive two others' packets, but those two nodes cannot receive each others. The two sensor nodes may sense the channel is idle, send a packet, and cause collision at the receiver in the middle. To deal with this problem, the IEEE 802.11 standard defines an additional mechanism using two control packets, RTS and CTS.

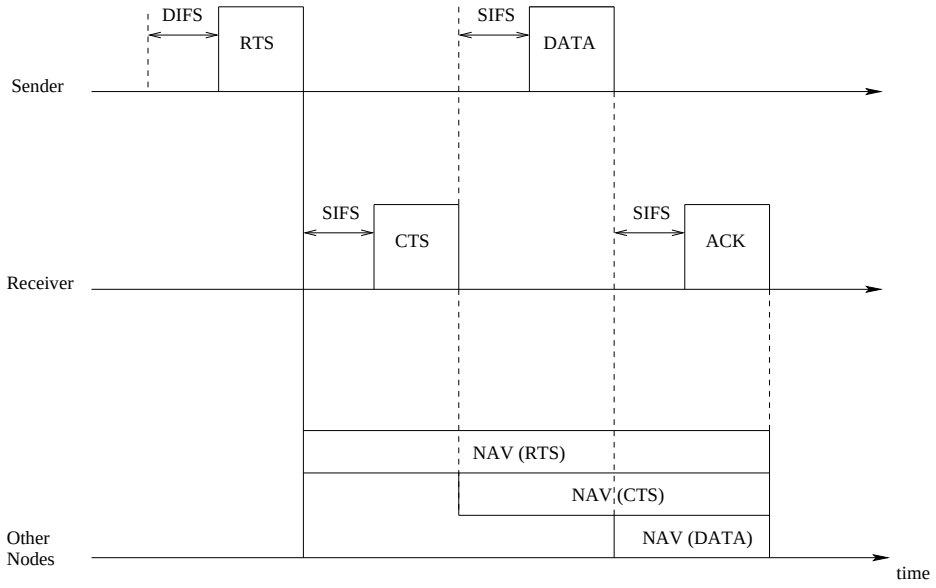


Figure 3.2: IEEE 802.11 Contention-Free Access

Figure 3.2 illustrates the use of RTS and CTS. After waiting for random backoff time (DIFS), the sender node issues a resquest-to-send (RTS) control packet. If the receiver of the data transmission receives this RTS packet, it replies with a clear to send a (CTS) control packet after waiting for SIFS amount of time. DIFS is DCF (Distribution Coordination Function) inter-framing spacing, which is the longest waiting time for medium access, and SIFS is Short inter-framing spacing, which is the shortest waiting time for medium access. Now all the nodes within the transmission radius of the sender and receiver are informed that they have to wait more time before accessing the medium and so these nodes adjust their Negative Allocation Vector (NAV) accordingly. Basically, this mechanism reserves the medium for one sender exclusively. The sender now can send the data frame after SIFS time and the receiver waits for SIFS time and then acknowledges whether the data transmission was correct. The transmission is now completed, the NAV in each node marks the

medium as free and the standard cycle starts again.

This thesis augments the basic mechanism of 802.11 MAC to provide anycasting. Anycasting is one of the means to divulge the information in the network. There are multiple ways to transfer the information in the network to reach its final destination, either as unicast, anycast, multicast, or broadcast. Unicasting is a method of routing, which involves the transmission of data packets between two nodes whereas the other methods involve multiple receivers for the information. When a source node wants to send a data packet to any node that belongs to a given set of destinations, it is called anycasting. Since event-based applications do not care about which intermediate nodes receive the information, anycasting provides an effective way of routing the information in the network so that it reaches the base station early. It reduces the one-hop delay by choosing any possible receiver for the data packet. The following sections explain the enhancements to 802.11 MAC to support anycasting and alleviate congestion and contention in the network.

3.4.1 RTS Broadcast

Traditionally, the RTS control packet is sent to a single receiver. There are situations when this receiver will not repond with a CTS packet: a receiver's queue buffer is full, no channel access, a receiver's battery is drained out, the link between the two nodes is broken, the RTS packet itself is not received, etc. In such cases, the critical data packet of the sensed information will be held at the sender node and many such packets will start to get congested at the node, which eventually leads to packet drops. Instead of this unicasting of RTS packet, DDMAC protocol broadcasts RTS packet to gain the channel access. When a sensor node has a data packet, it will broadcast the RTS control packet, which will be received by all the one-hop neighbors within

the transmission radius. The data packet will now move forward and not depend on a single receiver, and thereby reduce congestion at a node. From the way in which the potential forwarders and the backup forwarder is set for a node, this RTS packet will be received by all of them. The protocol ensures only the forwarder is set to react to this RTS packet.

3.4.2 CTS Response

According to anycasting in 802.11 MAC, once a receiver node replies with a CTS control packet, the other receivers in the anycast group will adjust their NAV for the entire data transmission period. When the sender receives a CTS, it transmits the DATA frame to the sender of this CTS after SIFS interval. This ensures that other potential RTS receivers in the anycast group will not send a CTS until another SIFS interval and will suppress any further CTS transmission. All such receivers then set their NAV until the end of ACK period. This avoids contention among the nodes. The DDMAC protocol ensures that the first node that replies with a CTS will be the best forwarder among the potential forwarders. The choice of the best forwarder is made dynamically based on the current queue buffer availability of the node. Upon receiving a RTS broadcast, the potential forwarders check their own queue buffer availability. Based on the current status of queue availability, each potential forwarder proportionally times their CTS replies, which is within the CTS time limit. In detail, the CTS receiving time from the potential forwarders are timed so that it does not exceed the CTS duration limit and is completed before the next SIFS interval. In Figure 3.3, it can be seen that the best forwarder will be the one with maximum queue availability among the potential forwarders. In the case of sensor node H, the queue availability is more in node B than in A and C. And in the

case of the node O, all the potential forwarders are filled up and the backup forwarder J accepts data transmission. But in case of node I, it is to be noticed that, though the backup forwarder G has an empty queue, the potential forwarder F has accepted the transmission. This happens because node F has some queue availability and is ready to accept the data transmission.

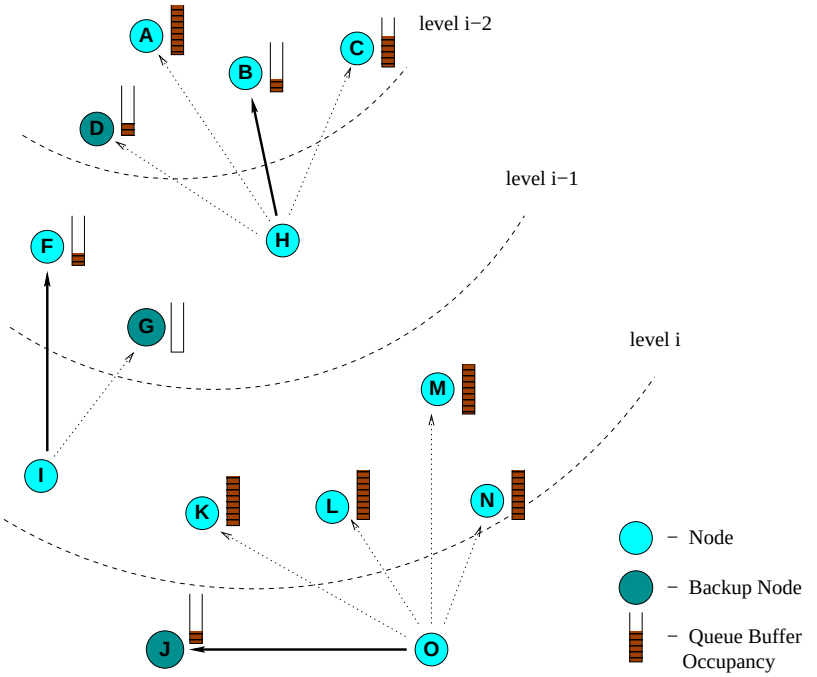


Figure 3.3: Forwarder Selection Based on Queue Availability

The decision for a node to be a fowarder is done at the MAC layer so that the time taken for any additional control messages is completely avoided. Moreover, since the CTS replies from potential forwarders are timed so that the RTS, CTS, DATA, and ACK time frames are not disturbed, there is no delay incured in the forwarder decision process. Unlike other protocols, DDMAC needs no information about the queue status of other nodes to make the decision. Figure 3.4 shows an example of time frames of the control packets and how the nodes backoff. When node H has a data

packet to send to the base station, it broadcasts RTS. The potential forwarders *A, B,* and *C* receives this RTS and waits for SIFS interval before replying with CTS. Now each of these forwarders checks its own queue buffer availability. As shown in Figure 3.3, node *B* has more queue availability than others, so its CTS reply will be received first by node *H*. As soon as the CTS reply is received by node *H*, it starts to transmit the data frame after SIFS interval. According to anycasting using 802.11 MAC, this transmission of data frame can be overheared by all the potential forwarders. Though the other nodes *A* and *C* also try to send CTS, they will backoff since the node *H* had already started sending the data frames. It can be seen that nodes *A* and *C* adjust their NAV and backs off from current data transmission. Therefore, this mechanism ensures that only the node with more queue availability gains the channel access and acts as a forwarder. This will reduce contention among the potential forwarders and reduce the congestion at the nodes, which improves event reporting.

3.4.3 Backup Response

A backup forwarder is used to support data transmission when all the potential forwarders fail to respond. In Figure 3.3, the backup node *J* responds to node *O* since all other potential forwarders *K, L, M,* and *N* have their queue buffers full. In such a scenario, a backup node eases the traffic and relieves the congestion at a node. When a node sends a RTS packet and does not get a CTS reply after waiting for SIFS interval of time, it retries with another RTS packet. In the standard IEEE 802.11 MAC, the maximum number of RTS retries is seven. In a situation in which all of the potential forwarders have their queue buffers full, it might take more RTS attempts to receive a CTS reply, or no CTS reply is received. The packets will get held at the sender's queue and start to create congestion. In order to ease such a situation and

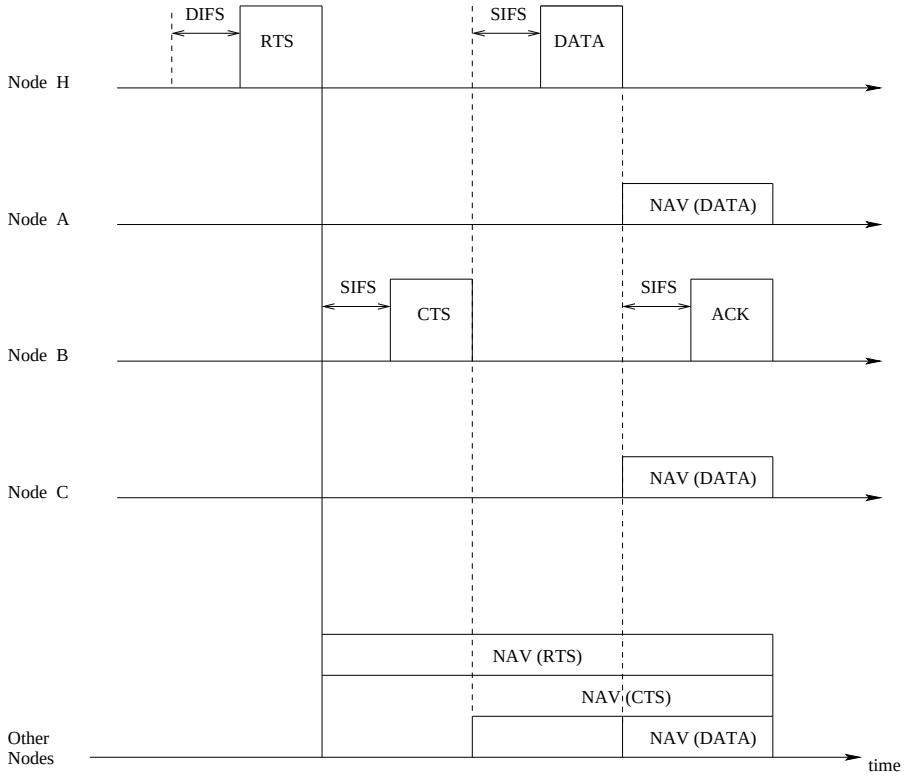


Figure 3.4: Dynamic Diffusion MAC Anycasting

also reduce the delay incured in waiting for a CTS reply from the potential forwarder, the backup forwarder is used in the DDMAC protocol.

Figure 3.5 shows a congested situation when a group of sensor nodes at multiple hop levels have their queue buffer full. These nodes cannot receive data packets unless they clear the current packets in the queue. This will lead to packets getting dropped at the sender. The DDMAC protocol will avoid such a situation by diffusing the data packets using a backup forwarder. The condition for selecting the backup forwarder to be far from the other potential forwarders helps to avoid the congested zone and tries to find a congestion-free forwarder to send the packets. Though the potential forwarder mechanism using queue availability can alleviate congestion, the backup

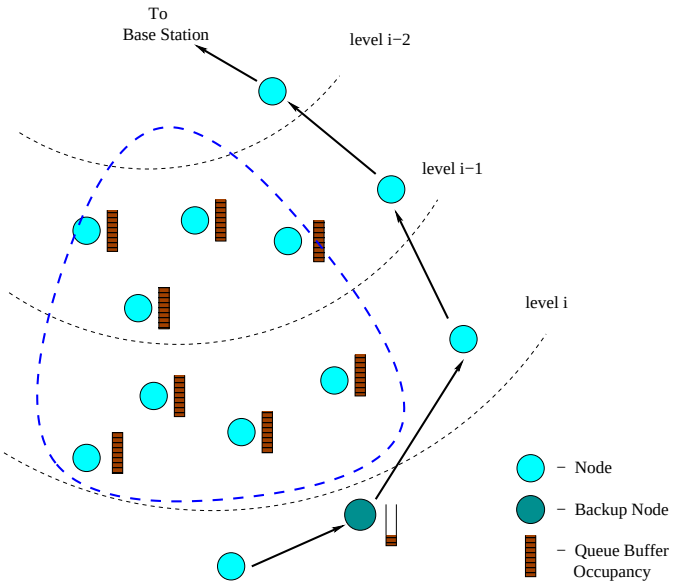


Figure 3.5: Traffic Diffusion

technique provides more support to avoid congestion in some back-logged situations.

CHAPTER 4

PERFORMANCE EVALUATION

DDMAC protocol is implemented in the standard Network Simulator (NS2) [21] to evaluate its performance. Extensive experiments were conducted to test the protocol's performance under various traffic loads. The protocol is compared with CONSEQ [3], which exploits multiple forwarders to reduce congestion in the network. CONSEQ also uses queue buffers to monitor the congestion status at the node and balance the load to control congestion at each hop level. The contribution of backup forwarding in traffic diffusion is also tested under two different congested scenarios. This is performed to evaluate the performance of backup forwarding when potential forwarders are unavailable for data transmission. We used the standard network performance metrics such as packet delivery ratio, throughput, and average end-to-end delay for evaluation of our protocol. We also measured the number of packets delivered and their delay to compare with CONSEQ results.

4.1 Simulation Setup

A summary of the simulation parameters is given in Table 4.1. All the nodes in the area are distributed uniformly and randomly. For the data packets, each source generates Constant Bit Rate (CBR) traffic and the number of sources are varied to evaluate the performance of the protocol at different loads. For all the experiments,

Table 4.1: Simulation Parameters

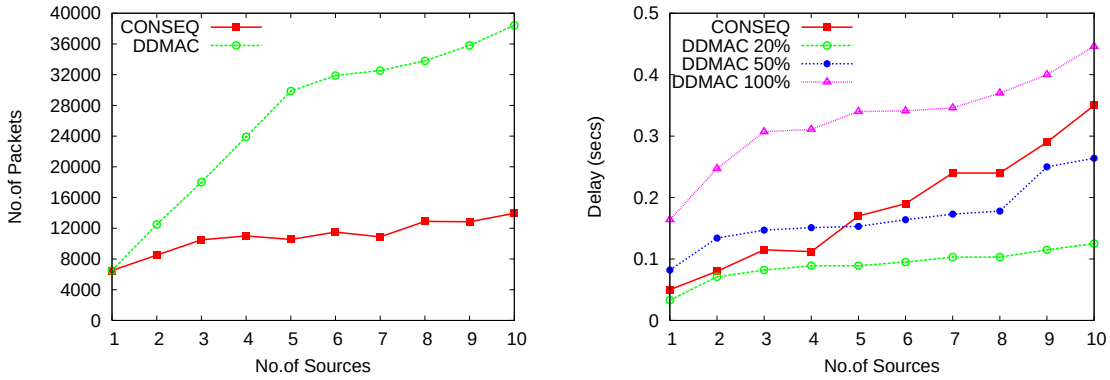
Parameter	Value
Area	$1000m$ x $1000m$
Deployment Strategy	$Uniform\ Random$
Transmission Radius	250 m
Total Number of Nodes	100
Number of Sources	$1 - 16$
Data Packet Size	64 $bytes$ & 512 $bytes$
Number of Packets Sent	$1pkt/sec - 10pkts/sec$
Queue Size	25 & 50

each data point taken is an average of 20 independent runs under various topologies and randomly chosen sources.

4.1.1 Comparison with CONSEQ

The performance of CONSEQ [3] is evaluated by varying the number of sources/cameras. The simulation setup of CONSEQ shows high power settings and high bandwidth. The data packet generation from each source is 100 Kbps and the packet size is 64 bytes [3]. We use the same data packet generation, packet size, queue size (25), and simulation duration of 60 seconds with NS2's default power setting to compare DDMAC's performance with CONSEQ. The total number of packets delivered and end-to-end delay metrics are compared.

Figure 4.1(a) shows DDMAC has a significant improvement in the number of packets delivered when compared to CONSEQ. The traffic diffusion approach to proactively avoiding congestion at the nodes makes our protocol deliver more packets even with a high traffic loads. In CONSEQ, though each source transmits 100 Kbps towards the base station, the rate controller at each node adjusts the packet sending rate at each hop level and reduces the actual packets generated. DDMAC does not



(a) No.of Packets Delivered (b) Delay

Figure 4.1: Comparison with CONSEQ

employ any rate controlling mechanism to avoid congestion, instead it adjusts the traffic with the given load, since we consider all the packets important. All the sources generate data packets at a rate of 100 Kbps in DDMAC and the number of sources are varied from 1 to 10 as in CONSEQ [3]. As observed from Figure 4.1(a), the DDMAC outperforms CONSEQ even under high traffic load. Since the actual number of packets generated after rate control is not clear in CONSEQ, it is hard to compare their packet delivery ratio [3] with DDMAC. Table 4.2 shows the number of packets sent and received by the CONSEQ and DDMAC protocol. The approximate number of packets sent from different numbers of sources is calculated for CONSEQ from the protocol's packet delivery ratio and the number of packets delivered graphs [3]. From the Table 4.2, it is clear that DDMAC delivers more packets than CONSEQ.

Figure 4.1(b) shows the delay comparison of DDMAC and CONSEQ. The average end-to-end delay for the first 20%, 50%, and 100% of the packets are measured for DDMAC protocol because the number of packets delievered in both of the protocols are different. The average end-to-end delay of the first 20% of packets delivered by DDMAC is less than CONSEQ's delay for less sources. For a greater number of

Table 4.2: Number of Packets Sent and Received

Number of Sources	CONSEQ No.of Packets Sent	CONSEQ No.of Packets Received	DDMAC No.of Packets Sent	DDMAC No.of Packets Received
1	6500	6500	6500	6500
2	8800	8800	13000	12500
3	10932	10900	19500	17997
4	11357	11300	26000	23954
5	11167	11000	32500	29935
6	12154	11850	39000	35883
7	11650	11300	45500	36785
8	13471	13000	52000	37178
9	13542	13000	58500	39369
10	14631	13900	65000	41808

sources, the 20% delay of DDMAC cannot to be compared with CONSEQ's delay since the number of packets delivered is different. The average delay of the first 50% of packets is compared with CONSEQ's delay for more sources. The delay for 50% of packets is less than CONSEQ's delay, though 50% of the total number of packets delivered is more than CONSEQ's total number of packets. In DDMAC, a node does not need the queue status of its neighbors to involve in data transmission, whereas in CONSEQ, load is balanced after computing the virutal queue length of all the fowarders. This mechanism of DDMAC reduces the transmission delay at each hop level even at high traffic. The delay for all the packets (100%) is higher than CONSEQ delay because the total number of packets delivered by DDMAC is significantly more than CONSEQ from Table 4.2. On an average, the difference between DDMAC and CONSEQ in end-to-end delay of all the packets delivered (100%) for different number of sources is 0.15 seconds. While incuring only 0.15 seconds more, DDMAC delivers more packets than CONSEQ. Overall, the DDMAC protocol delivers significantly

more packets with considerable delay than CONSEQ protocol.

4.1.2 Backup Forwarding Evaluation

To evaluate the performance of backup forwarding in traffic diffusion, two different scenarios are created to form a congested state in the network. The base station is located at the center of the deployment area for this experiment. This setup will make the traffic converge from all directions and will create a congested state. We use a packet size of 512 bytes and queue size of 50. For some event-reporting applications, it is desirable to allow the first few reports to reach the base station as soon as possible, which enables the base station to handle the events quickly. We show the end-to-end delay of the first 10% of the reports. The number of packets received at the base station and the end-to-end delay of the first 10% of the packets delivered are measured.

Distributed Traffic:

With the base station at the center, 15 sources are randomly chosen. The network is loaded such that traffic converges towards the base station from different directions. Traffic from these sources are sent at different rates. Table 4.3 shows the simulation results with backup forwarder and without backup forwarder. When backup forwarding is employed, more packets are received in a shorter duration than without using backup. Though the packets get diffused through backup forwarders, the delay is less, because the time taken for the packets to reach the base station through backup forwarder is less than the wait time for the potential forwarders when their queues are full.

Table 4.3: Distributed Traffic

	Without Backup		With Backup	
No.of Packets Sent	No.of Packets Received	Delay (secs)	No.of Packets Received	Delay (secs)
1500	1015	0.16	1302	0.14
2250	1555	0.18	1872	0.15
4500	2791	0.20	3171	0.12

Table 4.4: Distributed Traffic with Hotspots

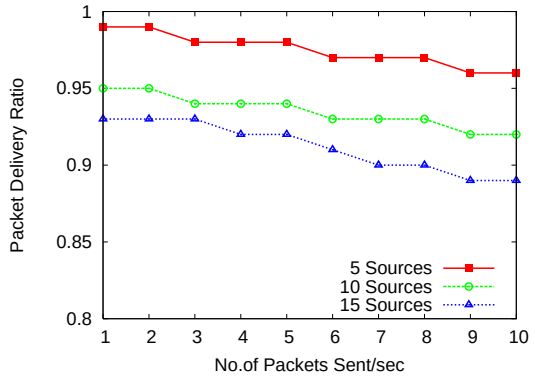
	Without Backup		With Backup	
No.of Packets Sent	No.of Packets Received	Delay (secs)	No.of Packets Received	Delay (secs)
1500	433	0.29	597	0.18
2250	1159	0.28	1208	0.27
4500	2397	0.25	2604	0.23

Distributed Traffic with Hotspots:

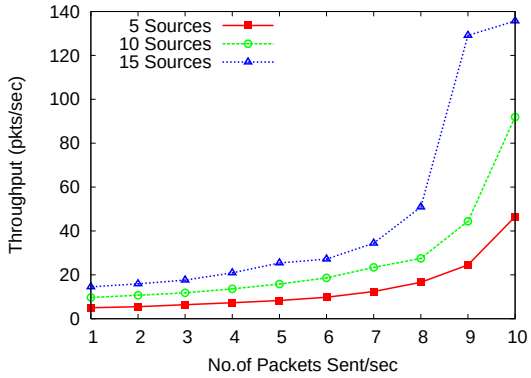
In order to further evaluate the backup forwarding technique's performance, the network is loaded to create congested hotspots. Three groups of 5 source nodes are chosen at random locations in the deployment area. The sources are close to each other to form hotspots in the network, which makes the potential forwarders unavailable for transmission. Table 4.4 is the result from the simulation experiments conducted using the backup forwarders and not using the backup forwarders. Traffic from three hotspots forms congestion closer to the base station and eventually load all the potential forwarders in the data path. Results show that traffic diffusion using backup fowarders even in a severe congested state delivers more packets in a shorter duration than not using backup forwarders. As expected, backup forwarding contributes to relieving congestion in the network and improves the event reporting.

4.1.3 DDMAC Protocol Evaluation

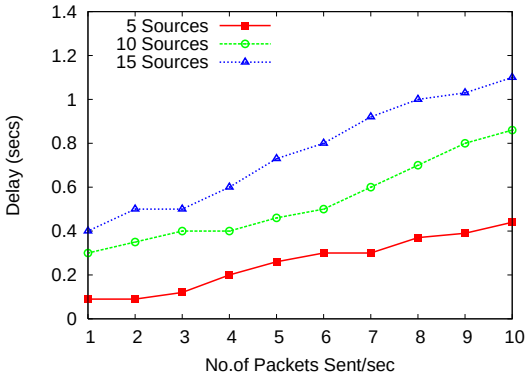
To evaluate our protocol, simulations are conducted with a base station located at the top-left corner of the deployment area to create different hop distances to the base station. Packet delivery ratio, throughput, and end-to-end delay are the important metrics to evaluate a protocol's performance in attempting to avoid congestion in the network. DDMAC is evaluated with a packet size of 512 bytes to show how the protocol handles a huge packet and by varying the number of sources detecting an event and sending the packets. The total number of packets handled by the network varies with the number of sources.



(a) Packet Delivery Ratio



(b) Throughput



(c) Delay

Figure 4.2: Performance of DDMAC Varying the Number of Sources

The packet delivery ratio is calculated as the ratio of the number of packets delivered to the number of packets generated at the sources. Figure 4.2(a) shows the performance of DDMAC in terms of packet delivery ratio for 5, 10, and 15 sources. Even for higher loads with 15 sources and 10 packets sent per second, DDMAC delivers close to 90% of the generated packets. The multiple forwarders along with backup ensures that more packets are delivered even under heavy load. This can also be seen from the throughput in Figure 4.2(b). The number of packets delivered per second increases exponentially as the packets sent increases. The results show that DDMAC can deliver more packets by effectively alleviating congestion in the network. Figure 4.2(c) is the average end-to-end delay of the first 10% of the packets delivered at the base station. The average delay increases as the number of packets delivered increases. The delay for 15 sources at 10 packets sent is 1.1 second, which is very low when delivering close to 90% of the packets generated. When more packets are sent, the potential forwarders will become unavailable and the backup forwarder disperses the traffic from the congested area. This makes the traffic diverged from the shortest path to the base station. The delay incured for the packets to reach the base station through a diverged path is less than the time taken for the packets to be held at a node. This ensures that the events are reported fast and are also not lost. The multiple forwarders with backup forwarding mechanism improves event reporting and reduces the transmission delay.

CHAPTER 5

CONCLUSIONS

Wireless Sensor Networks are built to transfer the sensed information to the base station without information loss. The sensor nodes are very constrained, restricts the purpose of the deployed network. In terms of event-reporting applications, the sensed information is very critical and has to be reported soon. But due to the processing capability of these nodes, the information arrival time gets delayed or sometimes even gets lost. Packet loss due to congestion at the nodes is one of the key factors affecting the performance of the network. Alleviating congestion at the node will ease the traffic and make the events reported much faster.

Congestion control and congestion avoidance techniques used in some applications attempt to detect and avoid congestion at the nodes. Congestion avoidance mechanism is more practical since it avoids the occurrence of congestion instead of detection and control. But the amount of information processing in congestion avoidance is an overhead for the network, which degrades the performance.

Sensor nodes queue buffers are commonly used to learn the congestion status at the nodes. CONSEQ [3] uses queue buffers of different forwarders to balance the traffic load among them. When there is a packet to send, the node needs the queue buffer information of all its forwarders to perform the necessary computations. In order to alleviate congestion without over burdening the nodes with several computations,

we developed a congestion avoidance mechanism, DDMAC, which diffuses the trafffic more efficiently in the network. Congestion in the network is proactively alleviated by employing multiple forwarders and backup forwarding mechanisms. Unlike other protocols, DDMAC handles congestion at the MAC layer.

Extensive simulations have been done to compare our proposed protocol with CONSEQ. Simulation results show that DDMAC can deliver more packets than CONSEQ in a shorter duration. Also the results for evaluating the protocol show that DDMAC can handle congestion very well and deliver more packets. This mechanism improves the packet delivery ratio and shortens the transmission latency.

In the future, we want to look at energy consumption of the nodes using our protocol and develop an energy-efficient DDMAC protocol. There are multiple data transport protocols such as data aggregation, duty cycle, subsetting of nodes, etc. to reduce the energy consumption in the network and provide reliable data delivery. We want to explore the effect of these techniques to provide an energy-efficient DDMAC protocol.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 2002.

[2] I. F. Akyildiz, M. C. Vuran, and O. B. Akan. A cross-layer protocol for wireless sensor networks. *IEEE 40th Annual Conference on Information Sciences and Systems*, 2006.

[3] C. Basaran, K. Kang, and M. H. Suzer. Hop-by-hop congestion control and load balancing in wireless sensor networks. *IEEE Conference on Local Computer Networks*, 2010.

[4] M. M. Bhuiyan, I. Gondal, and J. Kamruzzaman. CAM: Congestion avoidance and mitigation in wireless sensor networks. *IEEE Vehicular Technology Conference*, 2010.

[5] G. Bianchi. Performance analysis of the IEEE distributed coordination function. *Selected Areas in Communications*, 2000.

[6] R. R. Choudhury and N. H. Vaidya. MAC-Layer anycasting in ad hoc networks. *ACM SIGCOMM Computer Communications Review, Vol.34, No.1, 2004*, 2004.

[7] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. *MobiCom*, 2003.

[8] C. Tien Ee and R. Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks. *SenSys 2004 Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004.

[9] E. Felemban, C. Lee, E. Ekici, R. Boder, and S. Vural. Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks. *INFOCOM*, 2005.

[10] W. N. Gansterer, M. I. Khan, and G. Haring. Congestion avoidance and energy efficient routing protocol for wireless sensor networks with a mobile sink. *Journal of Networks*, 2007.

[11] T. He, F. Ren, C. Lin, and S. Das. Alleviating congestion using traffic-aware dynamic routing in wireless sensor networks. *IEEE SECON*, 2008.

[12] J. B. Helonde, V. Wadhai, V. Deshpande, and S. Sutar. EDCAM: Early detection congestion avoidance mechanism for wireless sensor network. *International Journal of Computer Applications*, 2010.

[13] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. *In Proceedings of ACM SenSys*, 2004.

[14] F. B. Hussain, Y. Cebi, and G. A. Shah. A multievent congestion control protocol for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2008.

[15] S. Jain and S. R. Das. Exploiting path diversity in the link layer in wireless ad hoc networks. *Journal Ad Hoc Networks, Vol.6, Issue 5, 2008*, 2008.

[16] J.Paek and R. Govindan. RCRT: Rate-controlled reliable transport for wireless sensor networks. *SenSys 2007 Proceedings of the 5th international conference on Embedded networked sensor systems*, 2007.

[17] H. Kim, R. Sankar, J. Lee, and I. Ra. Hop-by-hop based reliable congestion control protocol for wireless sensor networks. *International Symposium on Advanced Intelligent Systems (ISIS)*, 2007.

[18] Md. Mamun-Or-Rashid, M. M. Alam, Md. A. Razzaque, and C. Hong. Reliable event detection and congestion avoidance in wireless sensor networks. *Springer-Verlag Berlin Heidelberg*, 2007.

[19] S. Misra and M. S. Obaidat. LACAS: Learning automata-based congestion avoidance scheme for healthcare wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 2009.

[20] M. M. Monowar, Md. O. Rahman, and C. S. Hong. Multipath congestion control for heterogeneous traffic in wireless sensor network. *Advanced Communication Technology, ICACT*, 2008.

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

[22] N. Premalatha and A. M. Natarajan. Congestion control in wireless sensor networks. *IJCSNS International Journal of Computer Science and Network Security*, 2010.

[23] Md. O. Rahman, M. M. Monowar, and C. Hong. A QoS adaptive congestion control in wireless sensor network. *Advanced Communication Technology, ICACT*, 2008.

[24] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. *SIGCOMM*, 2006.

[25] S. Rangwala, A. Jindal, K. Jang, K. Psounis, and R. Govindan. Understanding congestion control in multi-hop wireless mesh networks. *MobiCom 2008 Proceedings of the 14th ACM international conference on Mobile computing and networking*, 2008.

[26] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz. ESRT: Event-to-sink reliable transport in wireless sensor networks. *MobiHoc*, 2003.

[27] B. Scheuermann, C. Lochert, and M. Mauve. Implicit hop-by-hop congestion control in wireless multihop networks. *Ad Hoc Networks*, 2007.

[28] K. K. Sharma, H. Singh, and 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, 2010*, 2010.

[29] K. K. Sharma, H. Singh, and R. B. Patel. A reliable and energy efficient transport protocol for wireless sensor network. *Global Journal of Computer Science and Technology*, 2010.

[30] P. Sharma, D. Tyagi, and P. Bhadana. A study on prolong the lifetime of wireless sensor network by congestion avoidance techniques. *International Journal of Engineering and Technology*, 2010.

[31] A. Sridharan and B. Krishnamachari. Explicit and precise rate control for wireless sensor networks. *In Proceedings of ACM SenSys*, 2009.

[32] F. Stann and J. Heidemann. RMST: Reliable data transport in sensor networks. *IEEE International Workshop on Sensor Net Protocols and Applications (SNPA)*, 2003.

[33] C. Wan, S. B. Eisenman, and A. T. Campbell. CODA: Congestion detection and avoidance in sensor networks. *In Proceedings of ACM SenSys*, 2003.

[34] C.Y. Wan, A.T. Campbell, and L. Krishnamurthy. Pump-slowly, fetch-quickly (PSFQ): A reliable transport protocol for sensor networks. *IEEE Journal of Selected Areas in Communication 23(4), 862872 (2005)*, 2005.

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

[36] C. Wang, K. Sohraby, V. Lawrence, B. Li, and Y. Hu. Priority-based congestion control in wireless sensor networks. *IEEE International Conference on Sensor Networks*, 2006.

[37] J. Wang and S. Medidi. Topology control for reliable sensor-to-sink data transport in sensor networks. *ICC*, 2008.

[38] Y. Xue, B. Ramamurthy, and M. C. Vuran. A service-differentiated real-time communication scheme for wireless sensor networks. *Local Computer Networks*, 2008.

[39] Y. Yi and S. Shakkottai. Hop-by-hop congestion control over a wireless multi-hop network. *IEEE/ACM Transactions on Networking*, 2007.

[40] Y. Zhou and M. Medidi. Energy-efficient contention-resilient medium access for wireless sensor networks. *IEEE ICC*, 2007.