2007

Congestion Avoidance And Fairness In Wireless Sensor Networks

Mohammad Ahmad
University of Central Florida

Congestion Avoidance and Fairness in Wireless Sensor Networks

by

MOHAMMAD ZUBAIR AHMAD
B.S. Visvesvaraya Technological University, 2005

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2007

Major Professor: Damla Turgut

ABSTRACT

Sensor network congestion avoidance and control primarily aims to reduce packet drops while maintaining fair bandwidth allocation to existing network flows. The design of a congestion control algorithm suited for all types of applications in sensor networks is a challenging task due to the application-specific nature of these networks. With numerous sensors transmitting data simultaneously to one or more base stations (also called sinks), sensor nodes located near the base station will most likely experience congestion and packet loss. In this thesis, we propose a novel distributed congestion avoidance algorithm which calculates the ratio of the number of downstream and upstream nodes. This ratio value (named Characteristic ratio) is used to take a routing decision and incorporate load balancing while also serving as a pointer to the congestion state of the network. Available queue sizes of the downstream nodes are used to detect incipient congestion. Queue characteristics of candidate downstream nodes are used collectively to implement both congestion avoidance and fairness by adjusting the node's forwarding rate and next hop destination. Such an approach helps to minimize packet drops, improve energy efficiency and load balancing. In cases of severe congestion, the source is signaled to reduce its sending rate and enable the network recovery process. This is essentially a transport layer algorithm and would work

best with a multi-path routing protocol and almost any MAC layer standard. We present the design and implementation of the proposed protocol and compare it with the existing avoidance protocols like Global rate control and Lightweight buffering. Our simulation results show a higher packet delivery ratio with greater node buffer utilization for our protocol in comparison with the conventional mechanisms.

To my parents!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Wireless sensor networks (WSNs) are an emerging class of networks with a wide variety of potential applications in the fields of health, military, environmental monitoring and so on. Lately there has been an increased focus towards developing transport protocols for WSNs in the research community. While *congestion control* concentrates on enabling the network to recover from packet loss, *congestion avoidance* detects incipient congestion and prevents its occurrence. This thesis studies effects of congestion in WSNs and proposes a novel distributed congestion avoidance protocol.

1.1 Motivation

Understanding the motivation behind congestion avoidance comes from the role of the structure and applications of WSNs in general. For example, in an environmental monitoring system, hundreds of sensors can be scattered over a flat area (thereby forming a *flat* WSN) which support low data-rate periodic sensing. The proposed tiered networks [17] can possibly be used in applications which require high transmission rates such as imaging [13]. Applications requiring high data-rate can easily cause congestion problem especially at intermediary nodes located closer to the sink. Suitable congestion avoidance schemes could help detect approaching congestion and reduce sending the data rates before congestion collapse occurs.

Another important effect of network congestion is the increase of packet collisions at the MAC layer due to sensors overhearing each other's radio transmissions in a densely populated area. The goal of the chosen MAC protocol [23] is to ensure correct delivery of packets to one-hop neighbors by using various techniques such as exponential backoff, coordinated sleep-wake schedules and virtual carrier sensing. With close to full buffers at each sensor, there will always be traffic at the queue, hence resulting in increased contention, greater retransmissions, and decreased packet delivery ratios. As a result, data loss due to congestion may ultimately threaten the benefits of the WSN.

1.2 Rate based vs window based approaches

Congestion control in WSNs has been largely implemented by using rate based approaches with the sources controlling the rate of the packets forwarded towards the sink. Periodic packet generation by the sources allow rate based congestion control as it enables an end-to-end adaptation which has been successful in developing the internet architecture [8]. However, nodes are not as relatively far apart from each other in WSNs (as in wired networks) with radio interference playing a major role in decreasing the successful packet delivery probabilities. An increase in the traffic generation leads to degraded channel quality with corresponding increases in the loss rates experienced by the nodes. Congestion avoidance on the other hand concentrates on regular monitoring of the buffer levels of intermediate nodes in the flow from source to sink. Conventional approaches to avoidance for both WSNs [3] and wired networks [9] have focused on examining queue sizes at intermediate hops in the route and detect incipient congestion, before transmitting notifications to the sources to reduce window sizes or data rates. As queue lengths are almost always defined by *number* of packets in the queue, we believe that window based approaches are more suitable for implementing congestion avoidance. Once the algorithm finds the buffer size exceeds a specific threshold, it can use a variety of techniques to notify the source (for instance, turning a congestion bit 'on' in the packet header) which in turn would immediately reduce the number of packets forwarded periodically. This means that the source would only need to increase the time between successive packet creation and not use extra resources in calculating packet generation times for different data rates.

1.3 Congestion avoidance based on queue sizes

1.3.1 Basic methodology

Buffer-based congestion avoidance is based on the premise that a sensor node should only forward a packet when the immediate destination node has a queue occupancy less than full or sometimes less than the congestion threshold. Sensor a should send a packet to sensor b only if the latter has enough of available memory in its buffer. As a result, each node should have prior information of the status of neighbor node queues, based on which it either sends or refrains from sending the packet. Packet drops due to congestion are thus minimized with the trade off being less than optimal network resource utilization. In addition, extra state has to be maintained at each node to store the instantaneous neighbor queue sizes which often turns out to be an ephemeral value (depending on packet arrival and service rate at each neighboring node).

Tracking neighbor buffer state at each node could be implemented in a couple of ways as follows:

- Piggybacking: Whenever sensor node a receives a packet, its buffer length changes (increases by one). The new buffer length is added onto a specified field in the packet header of the outgoing packet and sent to the immediate destination. When all nodes within radio range of node a hear this packet, they check their neighbor tables and update the corresponding queue length values. This method of piggybacking relevant data with regular messages helps in reducing extra packet overhead and hence saves battery power. Piggybacking would be ideal with packets at the MAC layer since control packets such as RTS/CTS are sent before the data packet and hence it would be the most updated information forwarded.

- Beacon broadcast: The problem with piggybacking is that for specific areas in the network where traffic is sparse, the node queue lengths could become 'stale' easily due to a very few packets forwarded at any given time. With reduction in traffic, most sensor MAC protocols propose periodic sleep/wake schedules [23] to conserve energy. Queue states could be broadcast as periodic beacons in coordination with other MAC or other routing layer beacons.

Both of the above techniques discussed have a few interesting points. Piggybacking queue information in any packet above the MAC layer, namely a network layer control packet or a transport layer payload, would require a minimum level of cross-layer interaction. The packet is created by the application and then a query is initiated to the network layer about the queue lengths of the one-hop neighbors.

Beacon broadcasts for informing queue lengths could actually be done away with the presence of timers at the sensor nodes. If the rate of packet forwarding decreases beyond a specific threshold (which may surely happen with sparse traffic), then it can be safely assumed that the neighbor queues are more or less empty. The sensor nodes have coarse timers implemented at the MAC layer, and a simple counter could also be implemented to count the number of packets sent/received during a specific time interval. If the count does not exceed a lower threshold, it would indicate low traffic density, less-full node queues leading to a very low frequency of queue broadcast beacons.

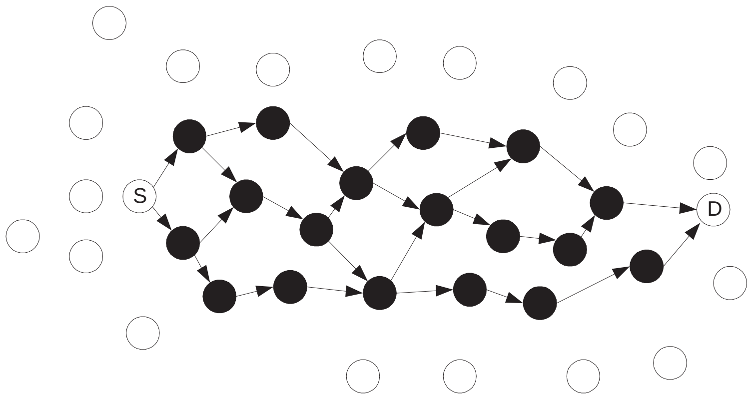


Figure 1.1: Meshed multipath routing

1.3.2 Multipath routing and fairness

Routing in sensor networks [1] has focused on energy efficiency as a key design metric. The transport layer should work in sync with the routing protocol to enable maximum conservation of limited energy. Proposed routing layer schemes can be divided into many categories such as attribute-based, hierarchical, flat, geographical, multipath routing methodologies, and so on. Multipath routing techniques ([5], [10] and [6] are prominent examples) focus on computing multiple paths from source to destination to effectively route around failed nodes or invalid links. If a single link fails, there is always a secondary route available to re-route the packets. This characteristic of multiple routes (see Figure 1.1) could also be used to implement congestion avoidance by re-routing packets when neighbor nodes' queues are gradually filling up. A packet drop on a broken link has the same effect as a packet drop due to buffer overflow. In such a situation, if multiple routes are available to the sender, the sender could make a routing decision on the fly and route its packet through a different node.

This method of functioning simultaneously results in a self-regulating system of fairness and load balancing in the network. Packets are forwarded to nodes where the chances of packet drop due to buffer overflow is minimum and coupled with fair queuing techniques leads to a higher degree of packet fairness in the sensor network.

1.3.3 MAC effects

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is the standard MAC protocol used to reduce the probability of collisions at the radio level due to the well-known *hidden* [19] and *exposed* [2] terminal problems. The RTS-CTS-DATA-ACK exchange presents a great opportunity for the transfer of neighbor queue information from one node to the other just before a packet transmission. However, since routing decisions are being made at another layer, the neighbor queue update will not affect the current data packet. The advantage lies in the fact that the neighbor list will be updated often.

Hidden and exposed nodes in the network will not be able to update their information correctly at all times since they may not overhear all the packets being exchanged. In such a situation, a periodic beacon broadcast would do the job efficiently or else by neighbor discovery messages. Probable reasons which lead to stale neighbor information due to the MAC layer at a node are:

- Sleep state: Inability to receive a packet from a neighbor if it is in the sleep/energy saving state. On waking up, the neighbor queue information has already become stale.

- MAC interference: Regular radio interference could disable a node from overhearing a nearby transmission.

- Hidden node influence: Collisions between packets due to hidden nodes could also play a huge part in keeping neighbor lists updated.

All the above problems can be addressed by using the method of piggy-backing queue information with concurrent use of RTS-CTS exchange to keep the neighbor lists updated. In some cases as in CSMA, there is no RTS-CTS exchange and hence the queue information has to be forwarded along with the DATA packets. The chances of a packet drop due to congestion in this case is noticeably higher and if the packet is lost, the sender would not receive the corresponding ACK packet and would in-turn go into the exponential backoff mode. By the time it is ready to resend the packet, the receiver queue would have much less number of packets and be able to receive the DATA packet.

1.4 Drawbacks in sensor network simulation

Network Simulator (ns-2) is perhaps the most well-known and used academic simulator in the wireless networking domain. Commercial simulators include OPNET, QualNet, GLO-MOSIM and so on. The TinyOS Simulator (TOSSIM) was developed in UC Berkeley as an extension to the TinyOS operating system designed for the Berkeley motes. While there have been a few more simulators developed with the purpose of simulating sensor networks, all of these were designed with specific applications and protocols in mind. Existing simulator support for WSNs is not highly developed as of now. In this section, we consider some of the standard simulators and discuss their shortcomings as follows:

- Ns-2: This is the most widely used and accepted network simulator currently in the networking research community. Ns-2 is a discrete event simulator written in C++ and TCL which provides substantial support for TCP, routing and multicast protocols for wired and wireless (ad hoc and satellite) networks. Sensor networks however require additional support - namely phenomenons generating information to be sensed by the sensor node and forwarded along the network to the base stations or sinks. These phenomenons have different types as applications where sensor networks are deployed differ significantly from one scenario to another. For example, a continuous monitoring system has constant rate traffic for long durations in operation with sudden specific spikes in packet generation rate when a particular event occurs. Sensor network extensions for ns-2 have been proposed as the NRL extensions, which implements a specific PHENOM class generating these phenomenon packets. However, ns-2's complex class hierarchy makes it difficult to implement cross-layer interactions. While lower to upper layer information can be passed through packet headers, it is extremely difficult for an upper layer agent to access parameters from a lower layer. Energy traces are also not easily supported and coupled with the problems associated with traffic generation, makes it a hard task to successfully simulate most aspects of the sensor network.

- TinyOS: This operating system is used extensively in the field of sensor networks. It was developed at UC Berkeley to work with the Berkeley mote. TinyOS uses 'NesC', a C-style development language which in collaboration with the Java powered TOSSIM simulator is used to simulate a sensor network. Since this simulator has been developed with the requirements of a sensor network in mind, much of the shortcomings of a regular network simulator such ns-2 has been removed. However, the problem of acquiring lower level information at a higher layer beforehand is still another problem and requires extensive coding. The TOSSIM also requires a large amount of memory and processor speed to execute and an average simulation could take up a long time on a standard desktop PC.

- Commercial simulators: Simulators like OPNET, QualNet and JavaSim all have some characteristic deficiencies in simulating sensor networks as most of these have been designed with regular wired and wireless networks in mind. The problems encountered while trying to develop WSN simulations in these simulators are almost the same as those that have been detailed in ns-2. Moreover, availability of these simulators to the research community at large is also limited. With fewer users there is less support available and inadequate documentation. There are also fewer development releases and bug fixes making these simulators difficult to use for newer users.

Due to problems with the most common network simulators, the researchers working on congestion avoidance and control devise their own simulation engine to perform simulation studies. Therefore in our work, we choose to do the same.

1.5 Contribution

In this thesis, we propose a congestion avoidance algorithm to prevent the network from entering the 'congestion collapse' state. Our contributions in this paper are as follows:

- We propose the term *Characteristic Ratio (CR)* as the ratio of downstream and upstream nodes for a particular node and define its usage and characteristics.
- We propose a congestion avoidance scheme which works well with multipath routing for WSNs using the CR value and argue the importance of this value in determining definite characteristics of a sensor node.
- A detailed simulation study of the proposed CR scheme is carried out in comparison with standard protocols like global rate control and lightweight buffering alongwith a scheme with no congestion avoidance implemented. The simulation parameters include packet delivery ratio, average and instantaneous queue size, average number of transmission and network lifetime. Simulation results show an increase in packet delivery ratio with correspondingly better network resource utilization. The mean network

lifetime with our congestion avoidance protocol is higher showing it to be an energy efficient algorithm.

1.6 Organization

The organization of this thesis is as follows. We follow this chapter with a review of related literature of congestion avoidance/control in wireless sensor networks. Chapter 3 presents the detailed description of the proposed algorithm.

Chapter 4 gives a brief discussion of the protocols studied and compared with our proposed protocol. Chapter 5 provides the simulation framework which we have used to evaluate the workings of the protocol with performance comparison of the three protocols.

Conclusions and future work are discussed in Chapter 6.

CHAPTER 2
RELATED WORK

In this chapter, we discuss the related literature for both congestion avoidance and the fairness concept in WSNs. Fairness has generally been discussed as part of transport layer protocols in sensor networks and so we discuss some standard transport protocols proposed.

2.1 Congestion avoidance

Sensor networks are designed to be deployed in different infrastructures for various applications. [18] is one of the earliest works which study the effect of congestion in a sensor network and determines probable techniques for congestion avoidance. The authors perform simple experiments and increase the deployed sensor density thereby increasing overall network load. They show that a proper deployment infrastructure is a natural form of congestion avoidance and should always be kept in mind during network setup. They also specify that the congestion avoidance scheme should ultimately *converge* on a reporting rate which meets the actual network requirements.

Chen and Yang [3] proposes a simple scheme to throttle the source sending rates by refraining from forwarding a packet if the receiver queue is full. This produces a cascade effect as multiple forwarding nodes stop forwarding their local copies serving as an implicit feedback mechanism to the source node. Using multipath routing, a sensor will have a list of neighbors to which it can forward packets while node advertises $1/k$ of its buffer size as available. Using analysis and simulations, the authors suggest that with the sensor nodes advertising a buffer length of 1/6 of the residual buffer capacity, congestion due to hidden terminals can be avoided completely. The reason for using one-sixth of the length is that the surrounding area of a sensor node can be divided into six convex regions (assuming every sensor has a circular transmission range of equal radius) with sensors in each region able to overhear transmissions in the region of the considered node. For sensor nodes not having uniform circular transmission ranges, the authors propose an adaptive approach to buffer advertisement by dynamically increasing or decreasing the value of 'k' based on the actual amount of buffer overflow occurrence in the network.

Fusion [11] presents a scheme for congestion mitigation using a combination of various schemes to recover from congestion in the WSN. It measures queue lengths, uses hop-by-hop backpressure and a priority based MAC protocol to reduce congestion by emptying near-full queues. Instead of using explicit congestion notification packets, it piggybacks congestion information over the regular packets transmitted. Upstream neighbors of the node setting the congestion bit overhear this event and refrain from transmitting till the congestion bit is cleared. The prioritized MAC protocol ensures that congested nodes get higher priority access to the channel and clear up their queues as fast as possible.

CODA [21] is a similar algorithm which listens to the radio channel to determine the amount of channel utilization. It uses both an open-loop hop-by-hop backpressure mechanism which detects congestion and sends backpressure messages to neighbors to reduce sending rates. CODA also uses an end-to-end acknowledgment scheme to enable the network to recover from enduring congestion by using an ACK stream from the sink. This enables the sink to regulate the source rate when required.

2.2 Fairness and transport protocols

[22] presents the Adaptive Rate Control (ARC) technique where each node estimates the number of downstream nodes and splits the bandwidth proportionally for packets originating at the node or being routed through it. Preference is given to those packets originating elsewhere but being forwarded through the current node. By specifying the bandwidth allocations to each node in equal amounts, the resource allocation in these networks are approximately fair.

[7] introduces the concept of fairness in sensor networks and states that fairness is achieved when an equal number of packets are received at each node in the network. The authors present a simple solution for congestion control by calculating the mean packet generation rate to enable fair rate assignment. They measure the average rate at which packets can be forwarded from a node and divide that rate by the number of downstream nodes, which is the per-node packet generation rate. This rate is compared with the parent sending rate

sent and the minimum of these two rates is used for packet forwarding. They also use a Probabilistic Selection or an Epoch-based Proportional Selection algorithm which enables equal number of packets to reach the base station.

An event-to-sink reliable transport protocol (ESRT) is proposed in [16] which uses congestion feedback and broadcasts notifications to the entire network to reduce sending frequencies. This scheme depends on the congestion duration and correspondingly the feedback latency. If the congestion occurring in the network is ephemeral with a high feedback latency, then it serves no purpose since by the time the notification arrives there is actually no more congestion present. This protocol is not suited for very large network with huge diameters as in these cases the feedback latency would also be very high.

[20] presents Pump-Slow Fetch-Quickly (PSFQ), a reliable transport layer protocol for sensor networks. However, PSFQ assumes that major losses in a sensor network are due to radio transmission errors rather than congestion. The authors propose to *pump* packets into the network slowly but *fetch* it back quickly when it is dropped. By fetching it means that a copy of the packet should be retransmitted quickly once it has been dropped, to enable reliable service to the application. It uses minimum signaling and is designed to be responsive to higher error rates in the radio channel.

[14] presents an interference aware fair rate control (IFRC) technique which monitors queue lengths to detect incipient congestion. The authors use relevant concepts from Internet congestion control literature like TCP congestion control [12] and queue management

algorithms like Random early detection [9]. Intuitively queue occupancy can be taken as a sufficiently accurate indication of network congestion. Another interesting characteristic the authors point out is that potential interferers of a sensor node need not only be child nodes in the node's routing tree, but also those in its parent's neighbor subtree. Their basic mechanism to attack the congestion problem is to assign to the 'most congested' node a particular rate to which all neighbors should adhere to when sending packets. An expression of the exponentially weighted moving average of instantaneous queue length is derived as a measure of node congestion and a technique of multiple thresholds are also considered to determine the current sending rate and control congestion.

[4] presents an aggregate fairness algorithm (AFA) for sensor networks which focuses on allocating bandwidth fairly to the various flows through a sensor node. A significant aspect of this work is that the authors study the network level fairness of *end-to-end* flows instead of the more popular *one-hop* MAC layer fairness. The authors point out that resolving congestion in a network does not ensure fairness. Each node in the network makes an estimate of the number of flows passing through it and based on the AFA algorithm, bandwidth is allocated proportionally. To characterize high priority data flows, 'weights' are assigned to flows from which the aggregate flow weight is calculated. These aggregate flow weights are now used to limit sending rates along a particular flow and ensure fairness is also maintained.

CHAPTER 3
PROPOSED CONGESTION AVOIDANCE PROTOCOL

The proposed congestion avoidance scheme needs to be simple enough to run unhindered on individual sensor nodes without overly taxing the system resources. It needs to be distributed as it will be running independently on each sensor. There should be an appropriate status dissemination mechanism to enable all nodes to keep track of the number of upstream/downstream nodes along-with their respective queue sizes. A distributed algorithm makes the entire system more robust to failures with a greater degree of energy efficiency due to lesser control overhead [7].

3.1 Characteristic ratios at each node

From the neighbor discovery protocol, each sensor node has a count of the total number of upstream and downstream neighbors present. We define *upstream nodes* as nodes on a flow which are closer to the source while *downstream* nodes are those which are closer to the sink. An intermediate node will hence have a specific number of candidate downstream and upstream nodes based on its location. Source nodes and sinks are special nodes with no upstream and downstream nodes respectively.

Table 3.1: Typical ratio values for nodes in the scenario considered

Node ID	Number of downstream nodes	Number of upstream nodes	Characteristic Ratio (CR)
a	3	1	3
b	2	2	1
c	1	2	0.5
d	2	1	2
X	3	0	Source
BS1	0	2	Sink

We define *Characteristic Ratio (CR)* R_i as the ratio of the number of downstream and upstream nodes for the considered node in the network and Q_a as the queue size of that particular node. Our example scenario is represented in the Figure 3.1 which is the same as the one considered in [3]. 'X' denotes the sensing node and the arrows denote the various paths possible to any of the two sinks. Even in the presence of a single base station, there could always be multiple paths from the source. We consider nodes 'a','b', 'c' and 'd' as the intermediate hops from source to sink. According to the definition of upstream and downstream nodes, we observe that node 'a' has three downstream and is a one-hop neighbor of 'x'. As a result, ratio R_a will be 3/1 which equals to 3. Similarly, R_b and R_c becomes 1 and 1/2 respectively. Again, let us remember that nodes with zero upstream nodes are sources and those with zero downstream nodes are base stations. In some special cases, sensor nodes within the network may have zero upstream/downstream nodes, which essentially means that they are disconnected from the network. We assume the absence of such sensor nodes

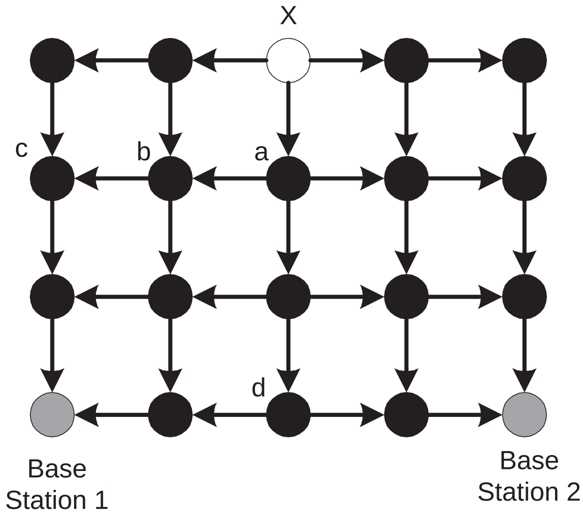


Figure 3.1: Sensor network model

in our example scenario. Table 3.1 lists the R_i values for various nodes in the assumed network.

The CR value of a particular node presents useful information of the network state at that location. Nodes can determine the probable measure of incoming traffic and decide on the output link to route the traffic. This technique yields itself well to achieve load balancing and fairness in WSNs and can relate to better congestion avoidance within the network.

3.2 Congestion avoidance model

Queue monitoring provides a mechanism to check congestion levels of nodes in the network, thereby enabling the implementation of congestion avoidance. Our avoidance model requires a set of steps to follow at each node to gather the required information for the congestion avoidance solution. The main steps in the proposed algorithm are:

1. Neighbor list creation: Each node is required to maintain a list of *upstream* and *downstream* nodes. This count of the specific number of nodes would be used to calculate the CR value at each node. This neighbor list and CR values would be ideally stored at the network layer of a node and be accessible to the transport layer protocol which would set its sending window in coordination with this number. Current queue sizes of each node in the list should also be maintained in the neighbor list. Essentially the neighbor list should comprise of three fields: (node_id,upstream/downstream,last_updated_queue_size).

2. Queue length advertisements: Sensor nodes need to regularly advertise their current queue sizes to enable neighbor nodes to update the queue sizes in their respective neighbor tables. As discussed in the previous chapter, this can be achieved by piggybacking queue state on packets or by a periodic beacon broadcast. Both of these mechanisms come with their share of pros and cons. We have used the former technique in our simulations by piggybacking queue state in routing messages as they require minimal change to the underlying protocols while providing energy efficiency and increased reliability with higher update frequency.

3. Congestion avoidance: The basic mechanism of congestion avoidance is to use the CR values at each node along with the candidate node's queue length and forward packets to appropriate candidate nodes. Instead of blindly forwarding the packets to

downstream nodes on the route, each node should make an informed decision in this regard.

The following subsections discuss the proposed algorithm in detail.

3.3 Network model

We assume a set of sensors in a rectangular area with the following properties:

1. The sensor nodes in the network are stationary.
2. Multiple base stations may be present in the network. Data sensed by the nodes can be forwarded to any of the base stations.
3. All nodes are structurally similar with equal transmission and communication capabilities.
4. Nodes do not have external location aids such as GPS, hence they are not location-aware.
5. A collision-free environment is assumed with the presence of a perfect MAC. The goal is to study the workings of the proposed algorithm with relation to congestion drops. Packet loss due to collisions are independent design aspects and thus can be abstracted when studying a congestion avoidance and control protocol.

3.4 Algorithm description

3.4.1 Congestion avoidance with multiple paths

Let us consider the case of multipath routing with a single sink in the network and nodes n_i with $1 \leq i \leq N$ where N is the total number of nodes, L_i is the neighbor list maintained at each node which holds the set of upstream and downstream nodes followed by the queue sizes of these nodes. These queue sizes are updated either when a packet is received or a transmission is overheard. n_i could overhear a packet transmission from n_{i+1} to n_{i+2} and update its own list by decreasing n_{i+1}'s queue size and simultaneously increasing n_{i+2}'s. This list should be saved in ascending order of queue sizes as it would enable the node to immediately decide as to which node it should forward the packet to once it is received. For nodes with relatively less number of neighbors, maintaining such a table will not require extensive resources. However with a large number of neighbors, maintaining such a state table could become an additional drain on the system resources. Sorting and updating such large tables would be unadvisable towards deploying an energy efficient protocol. We thus propose the use of an upper threshold to limit the total number of table entries. Some candidate nodes may have to be removed from L_i to accommodate the others, but we believe that such an action would not affect the overall working of the protocol. There would be many alternatives already present for a node to forward its packet in the neighbor table from which the node would choose the best option.

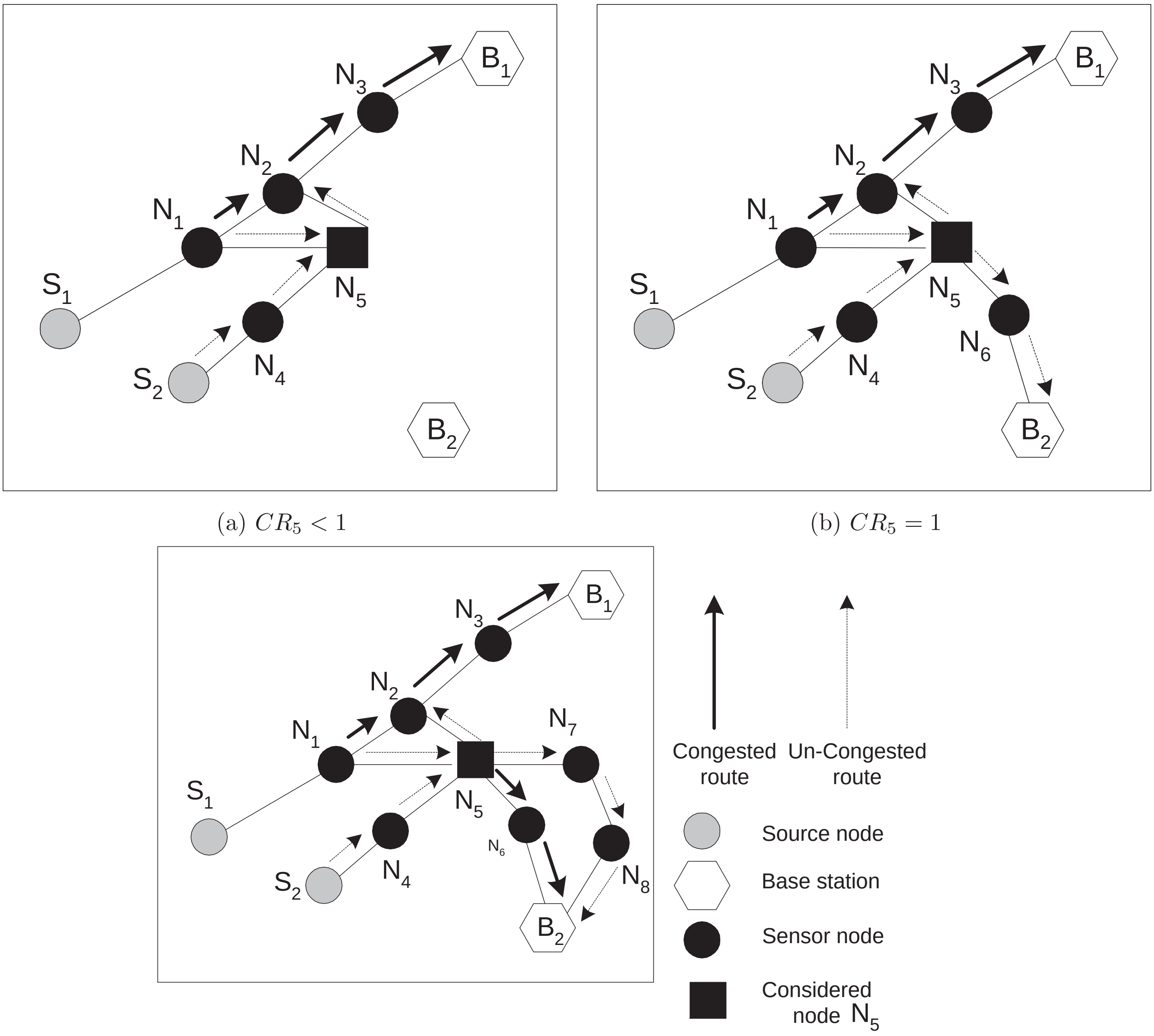
Setup initial neighbor list L_i;
Current buffer, B_i, is empty;
while packets are being transmitted in the network **do**
  if packet received and requires forwarding **then**
    Buffer packet;
    Check packet header for relevant queue information;
    Update L_i with received info;
    Check L_i length (len_i) and compare with l_{th} (Length threshold);
    if $len_i \geq l_{th}$ **then**
      Run neighbor table management algorithm
    end if
    if Current buffer size equals queue threshold, Q_th, **then**
      Calculate characteristic ratio CR_i;
      if $CR_i > 1$ **then**
        Implement fair queueing and forward packets
      else if $CR_i < 1$ **then**
        Use chosen rate reduction technique to reduce sending rate;
      else if $CR_i = 1$ **then**
        Check queue size in candidate downstream nodes from L_i;
        Route traffic through these nodes fairly;
      end if
    end if
  else if Current buffer size $> Q_{th}$ **then**
    Reduce sending rate;
  else if Current buffer size $< Q_{th}$ **then**
    Continue forwarding regularly
  end if
end while

Figure 3.2: Congestion avoidance with single sink

The individual CR values at each node can be used to make forwarding decisions. When a node has a packet to send out, it checks its CR_i value before taking a corresponding action. If CR_i is greater than one, it means that the node has a greater number of downstream nodes in comparison with upstream nodes. If so, it can implement any Fair Queuing (FQ) technique and forward the packet to the appropriate downstream node. For prioritized packets, priority queuing may be implemented while Weighted Fair Queuing (WFQ) can also be applied. The simplest technique would be to forward the packet to the candidate downstream node with the least queue occupancy. If CR_i is less than one, it means there are more upstream nodes than the downstream nodes. Thus, a rate reduction is required to prevent incipient congestion. As the node's queue fills up, it needs to inform neighboring upstream nodes to start sending fewer packets. The chosen rate reduction scheme may be any of the proposed mechanisms such as the congestion bit [3], Fusion [11] or backpressure messages used in [21]. If CR_i value is one, then the node could check queue sizes of the candidate downstream nodes and route packets fairly through them while also avoiding congestion. Algorithm 1 presents the pseudo code for the proposed congestion avoidance mechanism.

3.4.2 Congestion avoidance with multiple sinks

In the presence of multiple sinks in the network, an extra provision must be made with respect to the route the packets are forwarded. A route to a particular sink may be less congested in comparison to another route to a different sink. In such a scenario, an ideal congestion avoidance algorithm would need to route packets to the latter sink to ease the congestion on the remaining routes. Consider the scenarios shown in Figure 3.3. We consider an intermediate node (N_5) with two source nodes (S_1 and S_2) and two base stations (B_1 and



(a) $CR_5 < 1$

(b) $CR_5 = 1$

(c) $CR_5 > 1$

Figure 3.3: (a) Congestion avoidance scenario with $CR_5 < 1$ (b) Congestion avoidance scenario with $CR_5 = 1$. (c) Congestion avoidance scenario with $CR_5 > 1$.

Table 3.2: Tabulated characteristics for N_5 in Figure 3.3

Figure	Characteristic Ratio (CR)	Comments
(a)	0.5	Rate reduction necessary
(b)	1	Route traffic based on available queue sizes
(c)	1.5	Packets routed fairly through candidate downstream nodes

B_2) in the network. Various routes from source to destination are denoted by the arrows with the darker arrows representing congested routes and the lighter ones representing a lightly loaded route. Traffic should ideally be sent through the un-congested route in such a scenario to avoid congestion drops. Table 3.2 lists the basic characteristics of the various scenarios considered.

- Figure (a) represents the situation when CR value of node N_5 is less than one. It is evident that there is only one candidate downstream node, N_2, for N_5 here, which in itself is part of a congested route and has a high queue occupancy. This characteristic of N_2 forces N_5 to immediately invoke its rate reduction scheme and refrain from forwarding packets to N_2 till its queue frees up. Even though there is another base station (B_2) where the packets could potentially have been forwarded, there are no active neighbors of N_5 to forward packets to this sink.

- Figure (b) represents a scenario where the CR value of N_5 is one. N_5 has a choice between N_2 and N_6 to forward packets it has received from the pair of sources. Since the route through N_2 is congested, queue sizes at that node would be above the threshold indicating congestion. N_5 checks its CR value, and the queue sizes of the candidate downstream nodes (N_2 and N_6). Accordingly it starts routing traffic through N_6 to B_2. As fair queueing is used to route traffic through these candidate downstream nodes, packet level fairness in the network is also maintained.

- Figure (c) represents the scenario when the CR value of N_5 exceeds one since it has 3 downstream and 2 upstream nodes. With a higher number of downstream nodes, the sender would be able to fairly route traffic through all these nodes preferably in round robin fashion. N_5 could send packets to N_6, N_7 and N_2 in this case. A question arises, should N_5 be forwarding packets to a congested route such as N_2 fairly? There can actually be two solutions to this dilemma. N_5 either checks queue sizes of all downstream nodes and takes a forwarding decision based on these values, or else it can implement fair queuing and forward the packets. The disadvantage with the former approach is that in the case of the high values of CR (meaning that numerous available downstream nodes), the current node would have to check numerous entries from its neighbor table before making a decision to forward. Based on table sizes this could take extra time and would also dependend on the freshness of the information. Also, a great deal of energy may be expended if this process has to be repeated frequently. We believe the latter approach is better even though there is a risk of congestion loss due to packets forwarded through congested links, energy consumption and delay is minimized. Also, by the time packets are sent to various routes (using fair queuing) and its the turn of the earlier congested route, more packets in its queue would have been serviced leading to lesser loaded conditions. In this scenario as in Figure (c), while N_2 is congested, N_5 could first send packets to N_6 and N_7. In this time, N_2 could have some packets serviced from its queue and be able to accommodate some additional packets, even though the congestion scenario would not have been averted. Packets to N_2 will not come from N_1 since the CR value of N_1 is less than 1. In this case, rate reduction should be implemented. With few packets from that source, N_2 will not be in *congestion collapse* and would most likely be able to accommodate packets at sparse intervals from N_5.

3.4.3 Importance of characteristic ratio

The CR value helps making forwarding decisions at the nodes, specially at times of incipient congestion. This value allows the node to take an informed choice of forwarding a packet to a downstream node keeping in mind both the aspects of fairness and load balancing. A conventional congestion avoidance and control mechanism in WSNs aims to recover the network from congestion collapse or throttle sending rates to minimize the chances of congestion. However, these decisions depend largely on the bottleneck node in the network. Downstream nodes to the bottleneck node could be numerous and packets could have been routed through these nodes with a small decrease of sending windows in the upstream nodes.

There is also greater buffer usage, resulting in no actual need for decreasing sending rates.

3.4.4 Neighbor table management

As mentioned in section 3.4.3, managing the size of the neighbor lists of the node is an important aspect in the workings of our algorithm. Large lists would delay the searching or sorting of neighbor nodes, while too small lists would not represent sufficient choices for the node to forward its packets and may inadvertently more suitables node for packet forwarding. Optimum neighbor list size depends on the deployment criteria of the sensor nodes in the WSN. For example, if the nodes are deployed in a grid fashion, the minimum number of neighbors will be four (one in each direction), while the maximum number will actually depend on the transmission range of each node. Nodes scattered randomly would mean any number of neighbors, the total number of nodes in the network and the network density being the upper bound. Network density η can be defined as the average number of nodes per node footprint in the network [15]. The Neighbor table management algorithm (NTMA) in our scheme uses η value to keep track of the number of neighbors in a static sensor network. For a mobile scenario, the best technique would be to use the periodic beaconing scheme. However, we concentrate only on fixed WSNs in this work.The neighbor tableneeds to be updated either by piggybacking or periodic beacons.

As we have discussed earlier, we have implemented the piggybacking procedure to update queue sizes in the neighbor table. Once a packet reaches a node (N_k), from the packet header, it determines the header information such as the source of the packet and the piggybacked information stored by the sending node. Each extracted record is first searched in the existing table. If found, it is replaced by the new record. If it is not found, then it is added into the table and hence the current table size is compared with the network density η value at that node. If the table size is already equal to η, then space must be made by deleting a record before the new record is inserted. We propose to delete the neighbor record with the least queue size. This is because monitoring of the congested nodes is more important for congestion avoidance instead of those which do not have near-full queues. Thus, the record with the lowest queue size is deleted and the new record is inserted at that position. Algorithm 2 presents the workings of our neighbor table management scheme.

3.4.5 Buffer advertisements

Current node queue sizes are advertised to neighbor nodes to enable regular updates of the neighbor tables. As discussed in the Section 3.4.4, piggybacking or periodic beacon broadcasts can be implemented for this purpose. We have chosen the piggybacking scheme for neighbor updates (Algorithm 2) due to some inherent advantages as listed below:

```
Receive packet from neighbor node N_k
Check packet header for queue details
for all Records R_i in header do
    Search for R_i in existing table T_i
    if R_i found in T_i then
        Retrieve R_i
        Replace existing record with current record
    else
        Check size of table T_i (S_i)
        if S_i = η then
            Choose and delete record with least queue size for any neighbor node N_j
            Insert R_i into position emptied
        else
            Insert R_i into position which just became available
        end if
    end if
end for
```

Figure 3.4: Neighbor Table Management Algorithm (NTMA)

- Minimized control packet overhead: Explicit control packets are not required to be exchanged periodically between neighbor nodes, leading to reduced traffic load on the network. Additional memory requirements per packet will not be extensively high.

- Reduced processing overhead: Less number of packets leads to energy efficiency while the additional processing of packet headers does not cause an additional drain on the battery power of the node.

- Ease of updates: Lists are updated whenever packets are forwarded which reduces the possibility of stale information retained at a node for a long period of time. For areas with low traffic, tables will not be updated regularly, but this is not detrimental to the network since these nodes will most likely have empty queues. If there is a sudden burst of packets, queue sizes will get updated with the initial packet transmissions and the congestion avoidance algorithm will get initiated.

3.4.6 Rate reduction

With signs of incipient congestion, a rate reduction is certainly required. Our avoidance algorithm tries to route packets across different nodes before the onset of congestion; however in the event of the CR value of less than one (which denotes fewer choices for downstream nodes), the best approach is to initiate rate reduction. There have been various rate reduction schemes proposed in sensor network literature, some of which are drawn from traditional wired and wireless network protocols.

The buffer based scheme proposed in [3] has been used in our avoidance algorithm. This is a simple but effective scheme where a node does not forward a packet if the neighbor does not have enough buffer space. The packet remains at the node itself which starts filling up its own buffer. This creates a ripple effect as the node's upstream neighbors stop sending packets once they notice this considered node's buffer is filling up and ultimately this information reaches the source. The advantage of such a packet based scheme is that it removes the need for complicated signaling process as in the rate based schemes and decrease overall network performance.

CHAPTER 4
PROTOCOLS COMPARED

We compare our proposed protocol to the Lightweight buffer management protocol, Global rate control protocol and a mechanism without any congestion control and avoidance. The brief description for each protocol is presented in this chapter.

4.1 Lightweight buffer management

Sensor nodes in this scheme only forward data packets if the next hop neighbor queues can accommodate incoming packets. This is a packet based congestion avoidance scheme where there is no emphasis on any rate reduction algorithm at the onset on congestion. For rate based schemes, it is always difficult for upstream nodes to determine the exact amount of rate reduction required at downstream nodes as a response to congestion. Moreover, the bandwidth assigned to a sensor node keeps varying with time due to radio interference, multipath fading and changes in the number of active neighbors. Thus by allowing a packet level scheme, the need for complicated signaling procedures is discarded and upstream nodes can easily adapt to rates which would represent close to optimal network resource utilization.

The lightweight buffering scheme avoids packet drop due to buffer overflow completely. Once the node's buffer fills up to the advertised queue size, it simply refrains from forwarding the packet. There are a couple of advantages to such an approach as described below:

1. Local congestion: At a bottleneck node where there may be sudden traffic overflows, there could be instances of ephemeral local congestion. Such a congestion occurrence does not merit a complete reduction in packet sending rate from the source node and the network could easily stabilize in a few seconds of time with minimal packet loss. At instances of local congestion, this protocol responds well by taking a conservative approach and using the available memory at each node. With packets not immediately forwarded by a single node, its one-hop downstream node has greater time to service its buffer and is thus given full opportunity to clear up the congestion. Thus, local congestion can easily be handled by this mechanism.

2. Global congestion: This occurs when the source sending rate is too high for the network to handle and most node buffers in the route start filling up. Even in such a scenario, the downstream nodes are the ones which start filling up early. This has a cascade effect on the upstream nodes as these nodes refrain from sending the packets till the downstream node buffers are below the congestion threshold. Ultimately, if these queues are not processed fast enough, the source finds its immediate downstream neighbors with full queues and stops forwarding packets to them automatically,

decreasing the sending rate. Such an approach thus works well both with local and global congestion in a sensor network.

Hidden nodes in the network could also inadvertently be the cause of congestion in a sensor network and hence this protocol makes each sensor node advertise a queue size of one-sixth its total buffer size. This is because considering circular transmission range of each node, there can be a maximum of six overlapping neighbors of a sensor node which could be hidden from another. By advertising a buffer size of one-sixth of the total size, even if all six sensor nodes send packets at the same time, none of them will be dropped. Rate reduction is carried out automatically with the node queues filling up and hence refraining from sending packets either till congestion clears, or until the source reduces it packet generating rate.

4.2 Global rate control

As the name suggests, this protocol has a centralized body (basically the sink) controlling the rate of all the sources in the sensor network. The sources initially start sending their data packets to these sinks. With the onset of congestion at a particular node (queue size extending above a predefined threshold), a congestion bit is turned 'on' in the packet header before forwarding it to the next hop. The sink receiving those packets with the congestion-notification bit set realizes the onset of congestion and broadcasts a lower reporting rate throughout the network. The source nodes receive the new rate broadcast by the source and reduce their sending rates to this lower rate. As a result, all the sources have their rates controlled by the sink thereby removing congestion from all parts of the network ultimately resulting in an uniform sink reporting rate.

The disadvantage with such an approach is that the sending rate of the sources becomes dependent on the sending rate of the route experiencing highest congestion in the network. If the congestion occurs in a small part of the network with very little consequence on the entire scenario, the base station will take a conservative approach and force all the sources in the network to reduce their sending rate to the lowest level possible. This ultimately results in inefficient resource utilization as the network is probably capable of transmitting a higher number of packets; however, it but is not allowed to do so by the sink. Moreover, the periodic beacon broadcasts by the sink would also interfere in all the communications in the network thereby resulting in greater packet loss. This event occurs assuming the sensors are capable of transmitting and hearing only one channel instead of two or more. By forcing the noncongested sources to a conservative reporting rate there could also be a loss in the application fidelity in the network coupled with suboptimal resource utilization. One of the major design goals for this protocol is to achieve reliability by sending the event notifications only when needed. Some of the basic characteristics can be listed as follows:

1. The main goal is to adjust the reporting rates of sources and obtain proper reliability of the packets being forwarded.
2. The focus is on the actual event, not on individual data pieces.
3. It is application driven since the application defines the event reporting rate.

4. This algorithm runs mainly on the sink and hence does not require high processing overhead or extra memory in the sensor nodes. This enables higher overall network lifetime.

4.3 No congestion avoidance or control

A regular scenario where packets are dropped on congestion with no change in the sending rates. This serves primarily as the reference point for comparisons between different protocols with the various metrics used.

CHAPTER 5
SIMULATION STUDY

5.1 Simulation environment

Extensive C++ simulations are carried out to analyze the effectiveness of the proposed congestion avoidance mechanism compared to the existing protocols discussed in Chapter 4. One hundred sensor nodes are randomly placed in a 400 x 400 area with transmission ranges of the sensor nodes chosen randomly in the range [20,40]. The number of active sources are varied from 1 to 25 with each source generating on-off traffic at a maximum sustainable rate of 10 packets per second. Each data packet is 30 bytes long and each node can buffer upto 15 packets. We assume the presence of two sinks, one based at the right-bottom edge of the simulation grid while the other is placed midway at the bottom edge of the grid (similar to Figure 3.1). We also assume the presence of a multipath routing protocol providing multiple shortest paths from the source to the sink. Each simulation point is obtained as an average of 10 simulation runs with different random number seeds running for 100 seconds, unless otherwise specified. Table 5.1 summarizes the simulation parameters used in our experimental setup.

Table 5.1: Simulation parameters

Number of nodes	100
Total area	400x400
Transmission range	20-40 (meters/sec)
Active sources	1-25
Traffic type	On-off
Packet generation rate	10 packets/sec
Data packet size	30 bytes
Buffer size	15 packets
Number of sinks	2

5.2 Simulation parameters

From our simulations, we analyze the effect of the proposed algorithm with packet delivery ratio, average and instantaneous queue sizes, average number of transmissions, and mean network lifetime metrics.

- Packet delivery ratio: It is defined as the ratio of the number of unique packets successfully received at the sink to the number of packets forwarded by the sources. Due to the presence of multipath routing protocols, copies of a packet may be forwarded along different routes to the sink and these duplicate packets are not considered in our calculations.

- Average queue sizes: We monitor the queue size of each node in the network over the entire duration of the simulation. This metric represents the degree of queue utilization by the congestion avoidance protocol. The buffers should be used as much as possible with a minimum packet drop rate. An optimum queue utilization with the least amount of packet drops would translate into higher protocol performance.

- Instantaneous neighbor queue sizes: Neighbor queue sizes are an important aspect in the running of any congestion avoidance scheme since packets are forwarded based on the advertised queue lengths. This metric helps us to understand the change in neighbor queue lengths over time and represents the degree to which neighbor queues are being handled/used by the avoidance mechanisms. Frequent increase/decrease in the values of this metric denote ineffectiveness as neighbor queues are not populated uniformly which would lead to higher congestion drops.

- Average number of transmissions: This metric denotes the total number of transmissions made by the network, namely the total number of packets forwarded and received over the entire course of the simulation. A high number of transmissions with a lower packet delivery ratio would mean greater number of packets dropped during network communication. Schemes with frequent rate reductions would also have a lower number of transmissions since less number of packets will be generated over time.

- Mean network lifetime: This metric denotes the total time of active communication in the network. The average number of transmissions and the mean network lifetime have a direct relationship. Greater the number of transmissions would lead to a greater energy expenditure which in turn would decrease the network lifetime. Any sensor

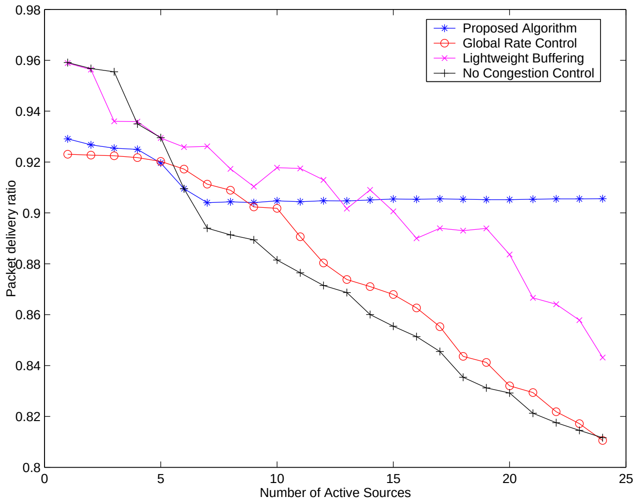


Figure 5.1: Average delivery ratios with increasing number of active sources in the network

network protocol should be designed with the energy conservation in mind and try to maximize the network lifetime.

5.3 Simulation results

5.3.1 Packet delivery ratio

Figure 5.1 denotes the average packet delivery ratio with varying the number of active sources in the network. After an initial period of restlessness, we see that with an increased number of active sources, the packet delivery ratio for the proposed algorithm hardly varies. The other algorithms also show a decrease in the packet delivery ratio with the number of active nodes in the network increase. If no congestion control or avoidance is used, the delivery ratio suffers due to the increase on the total number of packets generated. This graph presents an interesting characteristic of the proposed protocol. It is expected that with an increase in the active sources in the network would result in a greater number of packet drops. However, the CR scheme tries makes full use of the available queues in the network by monitoring queue sizes of neighboring downstream nodes before taking a forwarding decision. As a result, packets are transmitted only if they can be accommodated by candidate downstream nodes and not dropped due to congestion. The congestion avoidance scheme cannot however guarantee the complete exclusion of packet drops due to congestion. Aside from packet loss due to MAC layer collisions, congestion loss will occur if the network is suddenly flooded with a deluge of packets from the sources. Another reason for probable packet drop is *stale* network information. In an example scenario with four nodes, node's 1, 2 and 3 may both monitor the queue size of the same downstream node (say node 4, which probably has a current queue size just equal to the congestion threshold). Packets may then be sent at times not farther away from each other, which may not interfere at the radio level. In such a scenario, the first two packets arriving at node 4 will be enqueued leading to dropping of the final packet as the node queue is already full. Only during the next iteration will the nodes update their local information regarding the queue of the downstream node, which is currently full, and thus should not have any packets routed to it. The same issue with stale network information is also a factor in the number of packets dropped with the other protocols. Logically, packet drops should be non-existent in an avoidance protocol which only forwards packets if the next hop neighbor queue has available space. However if a

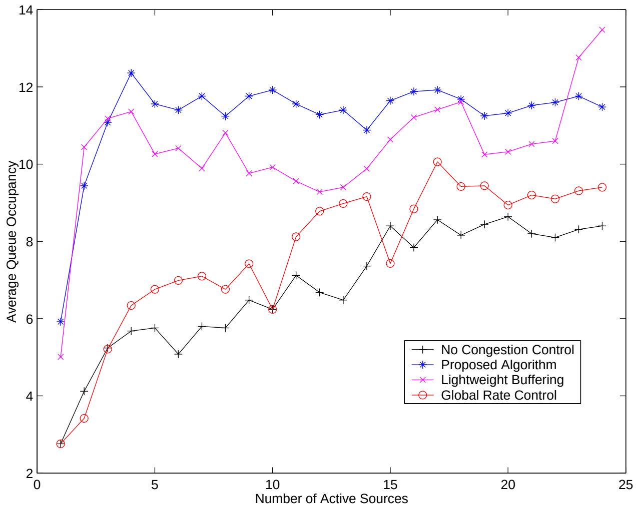


Figure 5.2: Average queue sizes with increasing number of active sources in the network

situation such as above arises, then there is an increased likelihood of packet drops, specially in areas with a high number of active sources.

5.3.2 Average queue sizes

Due to node queue sizes being of paramount importance in the workings of our protocol, we monitor both the instantaneous and the average queue sizes of the sensor nodes in the network. Figure 5.2 denotes the average queue sizes of all the nodes in the networkwith varying number of active sources. It is evident from Figure 5.2 that the proposed scheme makes greater use of the available queues in the network than any of the standard protocols. This result also explains Figure 5.1 in which the average delivery ratio remains more or less constant with the increase in number of active sources. While other schemes show a gradual

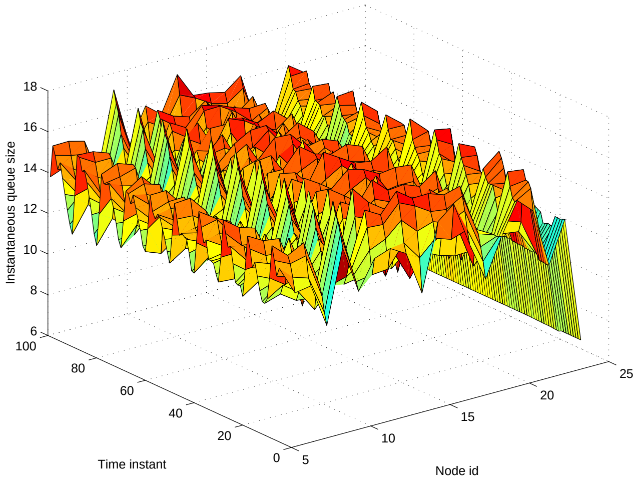


Figure 5.3: Instantaneous queue sizes of neighbors of a particular node during the entire simulation run for the proposed CR scheme. For simplicity, we only vary source node id's for one quarter of the total number of nodes in the network

increase in the average queue occupancy, the proposed mechanism consistently uses the extra available resources to minimize the packet drops due to congestion. A consistently high queue occupancy rate is actually desirable for the network (provided there are fewer packet drops), as it shows greater resource utilization and a higher degree of fairness. It avoids a common drawback of overloading specific bottleneck queues while other neighboring nodes which could participate in the packet forwarding process are left with largely unused queues. From the Figure 5.2, it can be observed that the lightweight buffering scheme also uses large parts of its available buffer space since it simply waits on a packet before transmission. However, our scheme has a better overall memory usage since the packets are forwarded based on not only the neighbor queue characteristics, but also the number of downstream and upstream nodes. Load balancing in such a scenario is more effective.

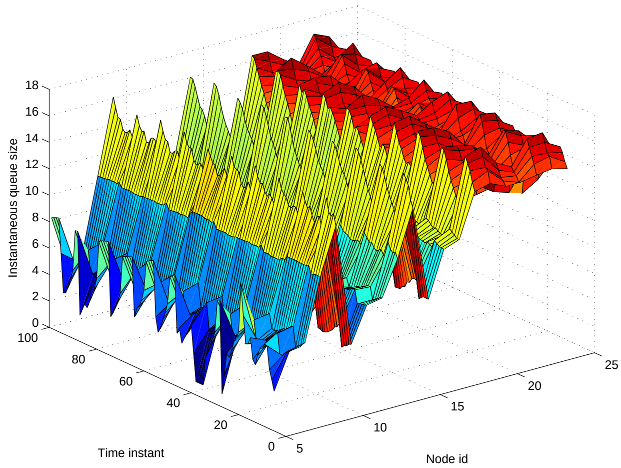


Figure 5.4: Instantaneous queue sizes of neighbors of a particular node during the entire simulation run for the regular scheme with no congestion control. For simplicity, we only vary source node id's for one quarter of the total number of nodes in the network

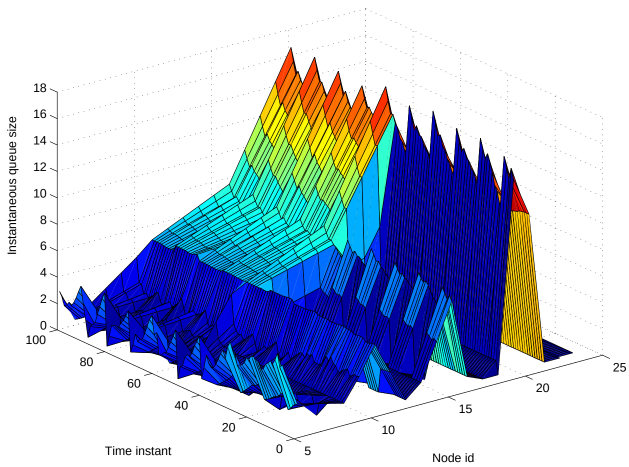


Figure 5.5: Instantaneous queue sizes of neighbors of a particular node during the entire simulation run for Global rate control. For simplicity, we only vary source node id's for one quarter of the total number of nodes in the network

5.3.3 Instantaneous neighbor queue sizes

Figures 5.3 - 5.5denote the instantaneous queue occupancies of those nodes which are immediate neighbors of the active sources. We observe the variation in neighboring queue sizes with time. Neighbor queues are central to the workings of most congestion avoidance protocols. In Figure 5.3 we see that for most time instances, the neighbor queue occupancies are much higher in our protocol. As explained in sub-section 5.3.2, this is due to the fact that most source nodes refrain from forwarding packets if the neighbor queues are close to being full. Let us note the greater number of abrupt rises and declines in Figure 5.4. This shows that a scheme with no congestion avoidance and control will naturally populate neighbor queues much faster leaving other candidate queues empty. Also once congestion occurs and no packets are forwarded to a particular route, the packets in the queue are serviced and the imminent queue size decreases. These drastic oscillations in queue sizes lead to greater packet drops and highly unfair resource allocation. Figure 5.5 also shows that the neighbor queue occupancy is not structured with extensive peaks and crests throughout. This shows an inefficient utilization of the queues which is more prone to congestion losses due to buffer overflow. In our proposed CR scheme, the queue sizes converge to a more constant size and hence represent a greater network resource utilization which leads to a higher degree of fairness in the network.

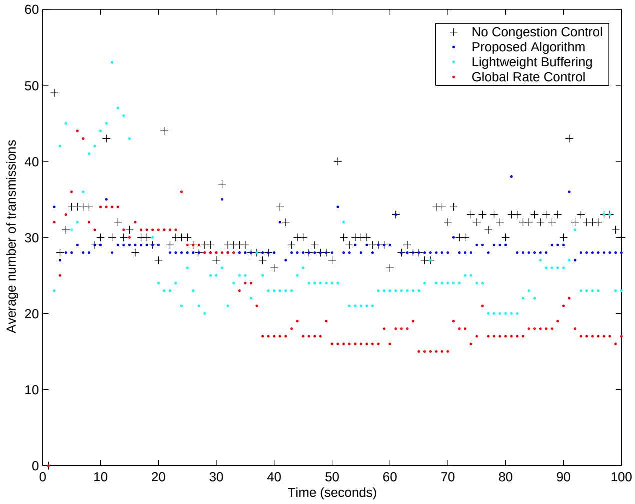


Figure 5.6: Comparison of average number of transmissions in the network with increasing time

5.3.4 Energy consumption and number of transmissions

Energy efficiency is a major design goal when developing sensor network protocols. While energy constraints are issues mainly handled at the MAC or the network layers, the number of forwarded packets contribute to the overall energy consumption. Due to energy efficiency being such an important metric for any sensor network protocol, we analyze the effectiveness of the CR scheme in terms of energy usage.

We denote average transmissions as the mean number of packets being forwarded throughout the network at every instant of time. Figure 5.6 compares the transmissions in the entire network at every time instant between the CR scheme and the three other protocols mentioned. It is evident that the number of transmissions in the proposed scheme is largely lesser and is also more or less constant over time. This characteristic is again due to the controlled packet forwarding based on the Characteristic ratio of the candidate downstream and upstream nodes. It reduces unnecessary packet transmissions which inevitably lead to unwanted packet drops. The Lightweight buffer management and the Global rate control protocols take conservative approaches in defining their sending rates. While the former refrains from forwarding on indication of congestion, the latter ensures that all sources transmit at the minimum possible rate. These lead to lesser transmissions as is evident from Figure 5.6. A packet drop actually results in a double loss. Energy is not only wasted in the unsuccessful transmission, but more battery power is needed for the retransmissions, which in itself could be required several times. The CR scheme hence leads to better energy efficiency and greater network lifetime due to less number of packet transmissions.

Comparing average number of transmissions with other prevalent mechanisms is not actually required since this metric depends on the rate reduction technique used by the congestion avoidance or control protocol. Global rate control, which controls the sending rate of the sources will have a much less average number of transmissions since it is not aggressive in forwarding as many packets as possible. Similarly, lightweight buffering would also have reduced number of transmissions due to holding back the packets instead of forwarding them when required. As a result, in Figure 5.7 we compare the performance of the proposed scheme with a mechanism which does not implement any congestion avoidance and control algorithm.

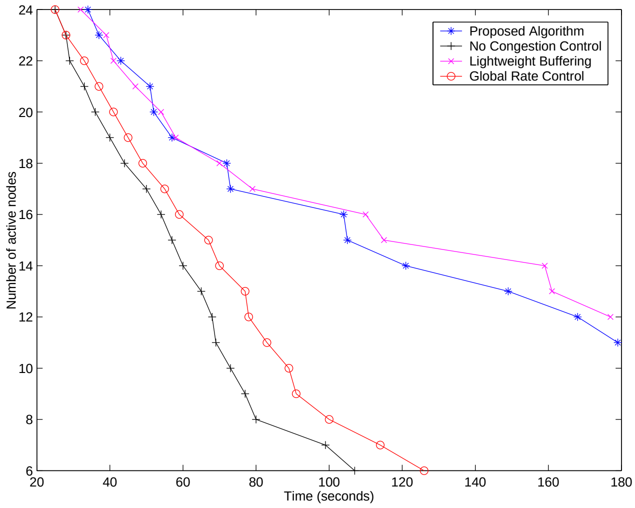


Figure 5.7: Comparison of network lifetimes for a restricted 25 node scenario with the proposed scheme being outperformed by the Lightweight buffering protocol

Network lifetime is compared in Figure 5.7. It is evident from the plot that network lifetime for the proposed CR scheme is much higher than any of the other mechanisms. We model the decrease in the number of active nodes with time. It can be clearly observed that the rate of nodes leaving the network due to power drain is significantly lower in the proposed scheme, resulting in a higher network lifetime. We can see that the network with no congestion control implemented simply runs out of power within 110 seconds of the simulation while a network running with the CR based congestion avoidance still has around 12 active nodes after 180 seconds of simulation time. These gains will be further magnified in larger networks running for greater intervals of time.

CHAPTER 6
CONCLUSION AND FUTURE WORK

In this thesis, we proposed a congestion avoidance algorithm by introducing the concept of characteristic ratio (CR) which is the ratio of number of downstream to upstream nodes in wireless sensor networks. The CR values at each node can be used to make a number of decisions aimed at reducing packet drops due to congestion, keeping fairness issues under consideration. In prior work, fairness has mostly been considered in one-hop scenario, i.e. at the MAC layer. We consider end-to-end fairness at the network layer with the help of multi-path routing in the network. The presence of multiple routes from source to sink is used for load balancing by routing packets through each route with any prevalent fair queuing technique thereby enabling both congestion avoidance and packet level fairness. We presented the design and implementation of the proposed protocol and compare it with the existing avoidance protocols like Global rate control and Lightweight buffering. Our simulation results show a higher packet delivery ratio with greater node buffer utilization for our protocol in comparison with the conventional mechanisms.Simulation results verify the effectiveness of the proposed scheme against other standard mechanisms.

Implementing a standard MAC layer like CSMA/CA or 802.11 to study the effects of congestion due to radio level interference would be a natural course of action for future work.

A rigorous mathematical analysis of queue statistics would also lead to a greater understanding of the state of neighbor queues from which an appropriate fair queuing mechanism could be deployed. Other measures such as calculating the Jain's Fairness Index would be suitable in determining the degree of fairness obtained by the proposed algorithm.

LIST OF REFERENCES

[1] Jamal N. Al-Karaki, Raza Ul-Mustafa, and Ahmed E. Kamal. Data aggregation in wireless sensor networks - exact and approximate algorithms. In *Proceedings of IEEE Workshop on High Performance Switching and Routing (HPSR)*, pages 241–245, April 2004.

[2] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LAN's. In *Proceedings of ACM SIGCOMM '94*, pages 221–225, August 1994.

[3] Shigang Chen and Na Yang. Congestion avoidance based on lightweight buffer management in sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 17(9):934–946, September 2006.

[4] Shigang Chen and Zhan Zhang. Localized algorithm for aggregate fairness in wireless sensor networks. In *MOBICOM*, pages 274–285, September 2006.

[5] Swades De, Chunming Qiao, and Hongyi Wu. Meshed multipath routing with selective forwarding: an efficient strategy in wireless sensor networks. *Computer Networks*, 43(4):481–497, October 2003.

[6] Budhaditya Deb, Sudeept Bhatnagar, and Badri Nath. ReInForM: Reliable information forwarding using multiple paths in sensor networks. In *LCN*, pages 406–415, October 2003.

[7] Cheng Tien Ee and Ruzena Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks. In *SenSys*, pages 148–161, November 2004.

[8] Sally Floyd and Kevin R. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Trans. Netw.*, 7(4):458–472, May 1999.

[9] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.*, 1(4):397–413, August 1993.

[10] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing Communication Review*, 5(4):10–24, October 2001.

[11] Bret Hull, Kyle Jamieson, and Hari Balakrishnan. Mitigating congestion in wireless sensor networks. In *SenSys*, pages 134–147, November 2004.

[12] Van Jacobson. Congestion avoidance and control. In *SIGCOMM*, pages 314–329, August 1988.

[13] Mohammad H. Rahimi, Rick Baer, Obimdinachi I. Iroezi, Juan C. García, Jay Warrior, Deborah Estrin, and Mani B. Srivastava. Cyclops: in situ image sensing and interpretation in wireless sensor networks. In *SenSys*, pages 192–204, November 2005.

[14] Sumit Rangwala, Ramakrishna Gummadi, Ramesh Govindan, and Konstantinos Psounis. Interference-aware fair rate control in wireless sensor networks. In *SIGCOMM*, pages 63–74, September 2006.

[15] Saikat Ray, Jeffrey B. Carruthers, and David Starobinski. Evaluation of the masked node problem in ad hoc wireless lans. *IEEE Trans. Mob. Comput.*, 4(5):430–442, September 2005.

[16] Yogesh Sankarasubramaniam, Özgür B. Akan, and Ian F. Akyildiz. Esrt: event-to-sink reliable transport in wireless sensor networks. In *MobiHoc*, pages 177–188, June 2003.

[17] Robert Szewczyk, Alan M. Mainwaring, Joseph Polastre, John Anderson, and David E. Culler. An analysis of a large scale habitat monitoring application. In *SenSys*, pages 214–226, November 2004.

[18] Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Rabiner Heinzelman. Infrastructure tradeoffs for sensor networks. In *WSNA*, pages 49–58, September 2002.

[19] F. Tobagi and L. Kleinrock. Packet switching in radio channels: Part ii–the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Transactions on Communications*, 23(12):1417–1433, 1975.

[20] Chieh-Yih Wan, Andrew T. Campbell, and Lakshman Krishnamurthy. Psfq: a reliable transport protocol for wireless sensor networks. In *WSNA*, pages 1–11, September 2002.

[21] Chieh-Yih Wan, Shane B. Eisenman, and Andrew T. Campbell. Coda: congestion detection and avoidance in sensor networks. In *SenSys*, pages 266–279, November 2003.

[22] Alec Woo and David E. Culler. A transmission control scheme for media access in sensor networks. In *MOBICOM*, pages 221–235, July 2001.

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