

# Congestion avoidance and fairness in wireless sensor networks

Mohammad Z. Ahmad and Damla Turgut  
School of Electrical Engineering and Computer Science  
University of Central Florida  
Orlando, Florida 32816-2362  
Email: {zubair,turgut}@eecs.ucf.edu

**Abstract**—Designing a sensor network congestion avoidance algorithm is a challenging task due to the application specific nature of these networks. The frequency of event sensing is a deciding factor in the occurrence of congestion. Numerous sensors, simultaneously transmitting data, increase the probability of packet drops due to congestion close to the base station(s). In this paper, we propose a novel distributed congestion avoidance algorithm which uses the ratio of the number of downstream and upstream nodes along with available queue sizes of the downstream nodes to detect incipient congestion. Monitoring queue sizes of candidate downstream nodes helps ensure effective load balancing and fairness in our avoidance algorithm. Through simulation studies we observe a greater packet delivery ratio and higher network lifetime in comparison with other prevalent mechanisms.

## I. INTRODUCTION

Wireless sensor networks (WSNs) are an emerging class of networks with a wide variety of potential applications in the fields of health, military and environmental monitoring. While *congestion control* concentrates on enabling the network to recover from packet loss, *congestion avoidance* detects incipient congestion and prevents its occurrence. This paper studies effects of congestion in WSNs and proposes a congestion avoidance algorithm.

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-rate periodic sensing. The proposed tiered networks [1] can possibly be used in applications which require high transmission rates such as imaging [2]. 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 rates before congestion collapse occurs.

Another important effect of network congestion is the increase of packet collision at the MAC layer due to sensors overhearing each other's radio transmissions in densely populated areas. The goal of the chosen MAC protocol 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, hence

resulting in increased contention, greater retransmissions and decreased packet delivery ratios.

Routing in sensor networks 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 the limited energy. Proposed routing layer schemes can be divided into many categories such as attribute-based, hierarchical, flat, geographical and multipath routing methodologies [3]. Multipath routing techniques ([4] and [5] 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 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, it 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.

In this paper, we propose a congestion avoidance algorithm to prevent the network from entering the 'congestion collapse' state. Our contributions 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 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.
- We propose a congestion avoidance scheme with multipath routing for WSNs using the CR value and discuss 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 other existing protocols.

## II. RELATED WORK

Sensor networks are designed to be deployed in different infrastructures for various applications based on requirements. Tilak et. al. [6] 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 [7] proposes a simple scheme to throttle 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 which serves 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 as each node advertises  $1/k$  of its buffer size as available. We use a similar assumption of the presence of a multipath routing scheme in our congestion avoidance algorithm.

Ee and Bajcsy [8] introduce the concept of fairness in sensor networks and state 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. Rangwala et. al. [9] presents an interference aware fair rate control technique which monitors queue lengths to detect incipient congestion. We use a similar queue monitoring technique in our proposed scheme by making a packet forwarding decision based on the number of candidate downstream nodes along with their current queue sizes. Fusion [10], CODA [11] and [12] are other congestion mitigation schemes proposed to recover from congestion in the network.

## III. PROPOSED CONGESTION AVOIDANCE ALGORITHM

We consider a sensor network comprising of a number of sensors and multiple *sinks* (also called base stations). The assumptions include a set of stationary sensors distributed in a rectangular area and a collision free MAC. Packet loss due to collisions are independant design aspects and thus can be abstracted when studying a congestion avoidance and/or control protocol.

### A. 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 following a set of steps at each node to gather the required information for the congestion avoidance solution.

- 1) Neighbor list creation: Each node is required to maintain a list of *upstream* and *downstream* nodes which would be used to calculate the CR value at each node.

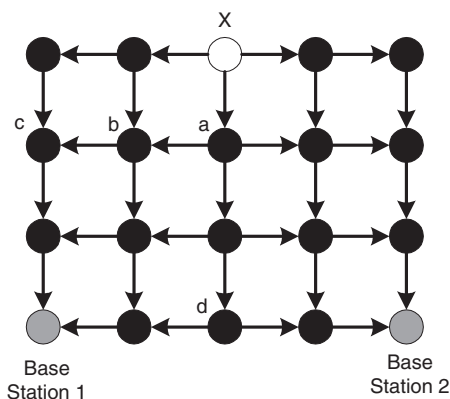


Fig. 1. A grid of sensor nodes with one source (X) and two base stations. The CR values at each node can be calculated by the in-degree and out-degree for that node as observed from the above figure.

TABLE I  
TYPICAL RATIO VALUES FOR NODES IN CONSIDERED SCENARIO

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
Base stations (1,2)	0	2	Sink

- 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.
- 3) Congestion avoidance: It is achieved by using CR values at each node along with the candidate node queue lengths to forward packets to appropriate candidate nodes.

### B. Algorithm description

1) *Characteristic ratios at each node*: From the neighbor discovery protocol, each sensor node knows the total number of upstream and downstream neighbors. 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.

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, represented in the Fig. 1, is the same one considered by Chen and Yang [7]. '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’ and ‘c’ 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 one upstream node. 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. 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 in our example scenario. Table I lists some of the  $R_i$  values for various nodes in the assumed network.

The CR at a particular node presents useful information of the network state at that position. 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 achieving load balancing and fairness in WSNs.

2) *Congestion avoidance with multiple paths*: The individual CR values at each node can be used to make forwarding decisions. When a node has a packet to forward, it checks its  $R_i$  value before taking an action. If  $R_i > 1$ , it means that the node has multiple downstream nodes and may implement any fair queueing technique to forward packets. For prioritized packets, priority queueing may be implemented while Weighted Fair Queueing (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  $R_i < 1$ , it means there are more upstream nodes than the downstream nodes, thereby requiring a rate reduction to prevent incipient congestion. As the node’s queue fills up, it needs to inform its neighboring upstream nodes to send lesser number of packets. The chosen rate reduction scheme may be any of the proposed mechanisms such as the congestion bit [7], Fusion [10] or by using backpressure messages in [11]. If  $R_i = 1$ , then the node checks queue sizes of the candidate downstream nodes and route packets through them fairly while avoiding congestion. Pseudocode for the congestion avoidance algorithm is presented in Algorithm 1.

3) *Importance of characteristic ratio*: The CR value presents an approach towards 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 the aspects of fairness and load balancing. Conventional congestion avoidance and/or control mechanisms in WSNs aim to enable the network to recover from congestion collapse or throttle sending rates to minimize the chances of congestion. However, these decisions are dependent largely on the bottleneck node in the network. Downstream nodes to the bottleneck could be numerous and packets could have been routed through these with a small decrease of sending windows in the upstream nodes and greater buffer usage, without the actual need for decreasing sending rates at the source. In general, the scheme would work well for a situation with a generous mixture of nodes with all three ranges of CR values (which would be the case in random deployment scenarios). This is because fair queueing and queue monitoring would be required for effective

---

#### Algorithm 1 Congestion avoidance with single sink

---

```

1: Setup initial neighbor list  $L_i$ 
2: Current buffer  $B_i$  is empty
3: while packets are being transmitted in the network do
4:   if packet received and requires forwarding then
5:     Buffer packet
6:     Check packet header for relevant queue information
7:     Update  $L_i$  with received info
8:     Check  $L_i$  length ( $len_i$ ) and compare with  $l_{th}$ 
9:     if  $len_i \geq l_{th}$  then
10:      Run neighbor table management algorithm
11:    end if
12:    if Current buffer size equals queue threshold  $Q_{th}$  then
13:      Calculate characteristic ratio  $R_i$ 
14:      if  $R_i > 1$  then
15:        Implement fair queueing and forward packets
16:      else if  $R_i < 1$  then
17:        Use chosen rate reduction technique to reduce sending rate
18:      else if  $R_i = 1$  then
19:        Check queue size in candidate downstream nodes from  $L_i$ 
20:        Route traffic through these nodes as per fair queueing
21:      end if
22:    end if
23:  end if
24: end while

```

---

load balancing and congestion avoidance. Thus, a mix of all types of nodes (with all CR’s) would be an ideal scenario for the proposed scheme.

#### C. Neighbor table management

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. They may inadvertently miss a more suitable node for packet forwarding. Network density  $\eta$  can be defined as the average number of nodes per node in the network. The neighbor table management algorithm in our scheme uses  $\eta$  value to keep track of the number of neighbors in a static sensor network.

We have implemented the piggybacking procedure to update queue sizes in the neighbor table. Once a packet reaches a node ( $N_k$ ), queue details are obtained from the header and the corresponding records searched in the existing table. If a record exists, then it is updated otherwise an extra record is added and the neighbor record with the lowest queue size is deleted.

#### D. Buffer advertisements and rate reduction

Current node queue sizes are advertised to neighbor nodes to enable regular updates of the neighbor tables. The piggybacking scheme for neighbor updates are advantageous as the control packet overhead is minimized leading to reduced processing and frequent updates.

We used a rate reduction protocol similar to Chen and Yang [7]. In this approach, a node only forwards a packet if

---

**Algorithm 2** Neighbor table management algorithm

---

```
1: Receive packet from neighbor node  $N_k$ 
2: Check packet header for queue details
3: for all Records  $R_i$  in header do
4:   Search for  $R_i$  in existing table  $T_i$ 
5:   if  $R_i$  found in  $T_i$  then
6:     Retrieve  $R_i$ 
7:     Replace existing record with current record
8:   else
9:     Check size of table  $T_i$  ( $S_i$ )
10:    if  $S_i = \eta$  then
11:      Choose and delete record with least queue size for any
        neighbor node  $N_j$ 
12:      Insert  $R_i$  into position emptied
13:    else
14:      Insert  $R_i$  into available position
15:    end if
16:  end if
17: end for
```

---

it is aware that the next hop will be able to accommodate the packet in its queue. If not, the node holds on to the packet thereby filling up its own queue. This produces a ripple effect upstream from the point of congestion and acts locally in times of transient congestion. However, in times of extreme congestion the source rate will ultimately be reduced as multiple downstream nodes start filling up their queues and refrain from forwarding packets.

#### E. Node forwarding rates

In our implementation, we consider the forwarding rates for upstream and downstream nodes to be the same. Generally, the arrival rate of packets at a node is larger than the service rate since typically multiple nodes forward packets to bottleneck nodes. Thus even though forwarding rates could be same, the actual intervals between packet arrivals is lesser than packet forwarding. Overall, the forwarding rates at congested nodes are automatically decreased at times of congestion and increased otherwise.

### IV. SIMULATION STUDY

#### A. Simulation environment and metrics

Extensive simulations are carried out to analyze the effectiveness of the proposed congestion avoidance mechanism. 100 sensor nodes are randomly placed in a 400 x 400 area with transmission ranges of 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 buffer sizes at each node is set to 15 packets. We assume the presence of two sinks, one at the right-bottom edge of the simulation grid while the other is placed at the left bottom edge of the grid (similar to Figure 1). We also assume the presence of a multipath routing protocol which realizes multiple shortest paths from the source to sink. Table II summarizes the simulation parameters used in our experimental setup.

TABLE II  
SIMULATION PARAMETERS

Number of nodes	100
Total area	400x400
Active sources	1-25
Traffic type	Variable rate
Data packet size	30 bytes
Node density ( $\eta$ )	10
Buffer size	15 packets

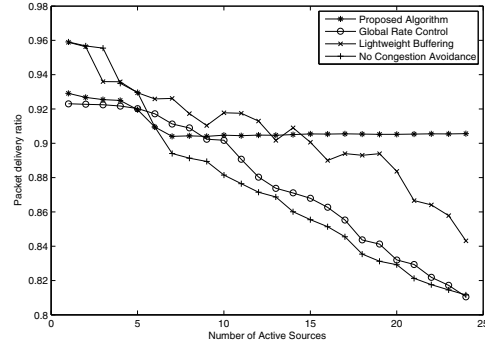


Fig. 2. Number of active sources vs. packet delivery ratio. Note that for most schemes there is a decrease in the delivery ratio with increasing sources whereas the proposed algorithm remains more or less constant. This is due to increased buffer memory usage.

From our simulations, we study the effect of the proposed algorithm using characteristic ratio with the following simulation metrics: packet delivery ratio, average and instantaneous queue sizes, average number of transmissions and mean network lifetime. We compare our proposed protocol to the lightweight buffer management [7] protocol, global rate control protocol [9] and a mechanism without any congestion avoidance and control.

#### B. Simulation results

1) *Packet delivery ratio*: Figure 2 denotes the average packet delivery ratio with varying 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 on the other hand show a decrease in the packet delivery ratio with an increase of active nodes in the network. This graph presents an interesting characteristic of the proposed algorithm. It is expected that an increase in the active sources would result in a greater number of packet drops. However, the CR scheme 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.

2) *Average queue occupancy*: We monitor both the instantaneous and the average queue sizes of the sensor nodes in the network. Figure 3 denotes the average queue sizes of all the nodes with varying number of active sources. It is evident

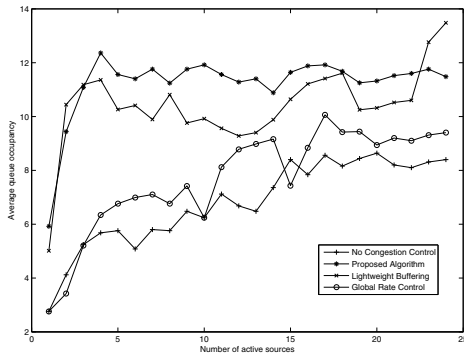


Fig. 3. Number of active sources vs. average queue sizes. The proposed algorithm shows a consistently high queue occupancy denoting the full use of available buffer space.

that the proposed scheme makes greater use of the available queues in the network than any of the standard protocols with a consistently high queue occupancy. This result also explains Figure 2 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 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 would be 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 Figure 3, 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.

3) *Instantaneous neighbor queue occupancies*: Figure 4 denotes 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 since neighbor queues are central to the workings of most congestion avoidance protocols. In Figure 4 we see that for most time instances, the neighbor queue occupancies are much higher in our algorithm. 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. 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

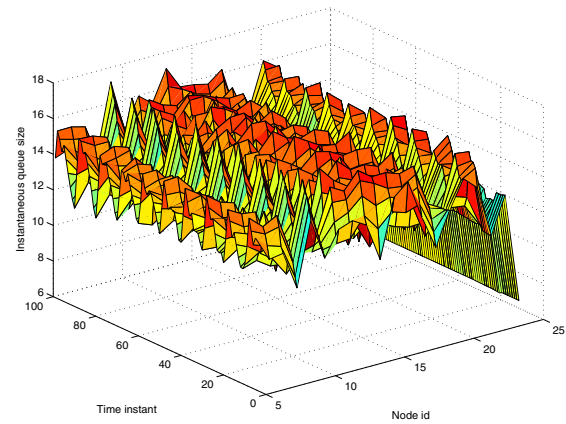


Fig. 4. Plot comparing the instantaneous queue sizes of neighbors of a particular node during the entire simulation run for the proposed CR scheme. For simplicity, we only vary source node ids for one quarter of the total number of nodes in the network.

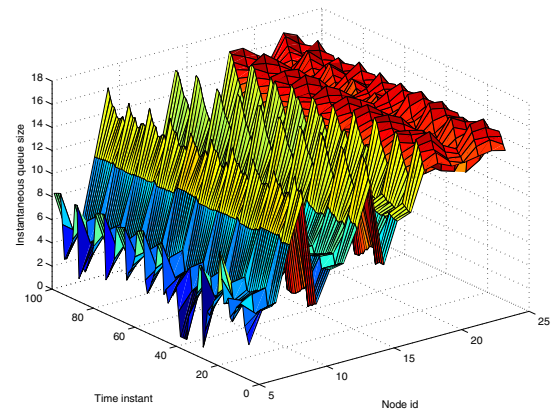


Fig. 5. Plot comparing the instantaneous queue sizes of neighbors of a particular node during the entire simulation run for the scheme with no congestion avoidance.

drops and highly unfair resource allocation. In our proposed algorithm (Figure 4), 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.

At first glance, high neighbor queue occupancies throughout the course of network operation seems to defeat the purpose of an avoidance algorithm as higher queue sizes inevitably lead to congestion. However, there is an upside to this characteristic if we monitor the net packet loss ratio obtained. As it is evident from Figure 2 the proposed scheme accounts for an improved delivery ratio compared to other existing mechanisms even though Figure 4 shows a higher queue occupancy. Using available memory at each node throughout the network in an effective manner is definitely more advisable than leaving queues unfilled in some nodes while others overflow and ultimately drop packets.

4) *Energy consumption and number of transmissions*: We denote average transmissions as the mean number of packets forwarded throughout at every instant of time. Figure 6

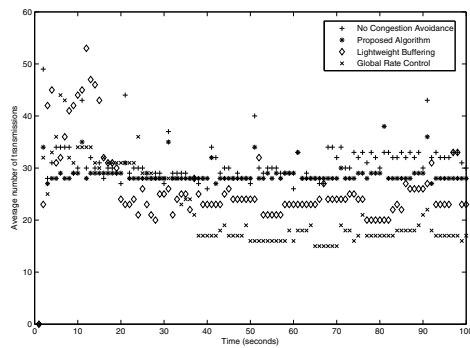


Fig. 6. Plot comparing the average number of transmissions in the network with increasing time. The proposed scheme shows more or less constant transmissions, indicating a controlled scenario unlike a widely oscillating scheme present without congestion avoidance.

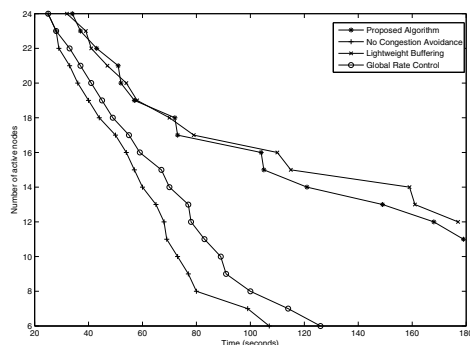


Fig. 7. Plot comparing network lifetimes for a restricted 25 node scenario. Energy consumption in networks without congestion avoidance is much higher and hence the network lifetime is much lower than in the proposed CR scheme.

compares the transmissions in the entire network at every time instant between the CR scheme and the three other protocols described earlier. It is evident that the number of transmissions in the proposed scheme is largely lesser and also remains 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. The number of transmissions are also indicative of the net delay in successful packet reception at the sink. A lesser number of transmissions mean greater delays, but within the larger avoidance goal this is a necessary tradeoff. 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 can be seen in Figure 6. A packet drop actually results in a double loss. Not only is energy wasted in the unsuccessful transmission, but also more battery power is needed for the retransmissions, which in itself could be required several times. The CR based algorithm hence leads to better energy efficiency and greater network lifetime due to less number of packet transmissions.

Network lifetime is compared in Figure 7. As can be seen

from the plot the 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 avoidance 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 are expected to be further magnified in larger networks running for greater intervals of time.

## V. CONCLUSION

In this paper, we proposed a congestion avoidance algorithm by introducing characteristic ratio (ratio of downstream nodes to upstream nodes) concept in WSNs. The CR values at each node is used to make number of decisions aimed at reducing packet drops due to congestion and enable a higher degree of fairness. Higher overall delivery ratios with lesser transmissions leading to an increase in the lifetime of the network characterizes the proposed avoidance algorithm. Simulation results verify the effectiveness of the proposed scheme against global rate control and lightweight buffering algorithms.

## REFERENCES

- [1] R. Szewczyk, A. M. Mainwaring, J. Polastre, J. Anderson, and D. E. Culler, "An analysis of a large scale habitat monitoring application." in *SenSys*, November 2004, pp. 214–226.
- [2] M. H. Rahimi, R. Baer, O. I. Iroezji, J. C. Garcia, J. Warrior, D. Estrin, and M. B. Srivastava, "Cyclops: in situ image sensing and interpretation in wireless sensor networks." in *SenSys*, November 2005, pp. 192–204.
- [3] A. Boukerche, M. Ahmad, B. Turgut, and D. Turgut, "A taxonomy of routing protocols in sensor networks," in *Handbook on Algorithms and Protocols for Wireless Sensor Networks*, A. Boukerche, Ed. Wiley, 2008, ch. 6, pp. 129–160.
- [4] S. De, C. Qiao, and H. Wu, "Meshed multipath routing with selective forwarding: an efficient strategy in wireless sensor networks," *Computer Networks*, vol. 43, no. 4, pp. 481–497, October 2003.
- [5] 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*, vol. 5, no. 4, pp. 10–24, October 2001.
- [6] S. Tilak, N. B. Abu-Ghazaleh, and W. R. Heinzelman, "Infrastructure tradeoffs for sensor networks." in *WSNA*, September 2002, pp. 49–58.
- [7] S. Chen and N. Yang, "Congestion avoidance based on lightweight buffer management in sensor networks." *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 9, pp. 934–946, 2006.
- [8] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks." in *SenSys*, November 2004, pp. 148–161.
- [9] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware fair rate control in wireless sensor networks." in *SIGCOMM*, September 2006, pp. 63–74.
- [10] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks." in *SenSys*, November 2004, pp. 134–147.
- [11] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "Coda: congestion detection and avoidance in sensor networks." in *SenSys*, November 2003, pp. 266–279.
- [12] Q. C. Lujiao Li, Yun Li and N. Nie, "PTCP: Phase-divided tcp congestion control scheme in wireless sensor networks," *Mobile Ad-Hoc and Sensor Networks, Lecture Notes in Computer Science*, vol. 4864, pp. 462–485, 2007.