PAPER *Special Section on Ubiquitous Sensor Network*

Congestion Avoidance and Fair Event Detection in Wireless Sensor Network

Md. Mamun-Or-Rashid [†a)], *Non Member,* Muhammad Mahbub Alam[†b)], *Non Member*, Md. Abdur Razzaque[†c)], *Non Member* and Choong Seon Hong[†d)], *Member*

Summary
Congestion in WSN increases the energy dissipation rates of sensor nodes as well as the loss of packets and thereby hinders fair and reliable event detection. We find that one of the key reasons of congestion in WSN is allowing sensing nodes to transfer as many packets as possible. This is due to the use of CSMA/CA that gives opportunistic medium access control. In this paper, we propose an energy efficient congestion avoidance protocol that includes *source count* based hierarchical and load adaptive medium access control and weighted round robin packet forwarding. We also propose in-node fair packet scheduling to achieve fair event detection. The results of simulation show our scheme exhibits more than 90% delivery ratio even under bursty traffic condition which is good enough for reliable event perception.

Key words:
Sensor Network, Congestion Avoidance, Hierarchical Medium Access, Fair Event Detection.

1. Introduction

Wireless Sensor Networks (WSNs) are densely deployed for a wide range of applications in the military, health, environment, agriculture and smart office domain. These networks deliver numerous types of traffic, from simple periodic reports to unpredictable bursts of packets triggered by sensed events. Therefore, congestion happens due to contention caused by concurrent transmissions, buffer overflows and time varying wireless channel condition [1][2][3]. As WSN is a multi-hop network, congestion taking place at a single node may diffuse to the whole network and degrades its performance drastically [4]. Congestion causes many folds of drawbacks: (i) it increases energy dissipation rates of sensor nodes, (ii) it causes a lot of packet loss, which in turn diminish the network throughput, and (iii) it hinders fair event detections and reliable data transmissions. Therefore, congestion avoidance and fair packet delivery become a crucial research issue for the practical realization of WSN based envisioned applications.

1.1 Problem Description

Congestion control algorithms detect a congestive state by monitoring either queue occupancy or channel utilization factor. Once congestion is detected, the congested node in [1], [4] and [5] propagate its notification towards source nodes using hop-by-hop backpressure. This approach might not work in a highly congested network where nodes become unable to transmit or receive any packet. Another problem with this approach is that severe congestion may happen before controlling it. Because, even in lightly congested network, the propagation delay of congestion notification increases as network diameter grows in size. Therefore, congestion avoidance is more preferable in WSN to congestion control.

We find that one of the key reasons of congestion due to collision in WSN is allowing sensing nodes to transfer as many packets as they can. This is due to the use of opportunistic medium access control (CSMA/CA). The high amount of data transferred by sensing nodes can overwhelm the capacity of downstream nodes, particularly the nodes near the sink. Hence, we propose *source count* based hierarchical medium access control (HMAC) that gives proportionate access to the medium. We define *source count* of a node as the total number of source nodes for which it is forwarding data. Therefore, downstream nodes get higher access to the medium than their upstreams. This access pattern is controlled by local values and is made load adaptive to cope up with various application scenarios.

Detection of an event may be biased if highly inequity number of packets is received at the sink from the event sensing nodes. Unfair packet delivery may also lead to false event detection. We define that an event is fairly detected if the sink node receives almost equal number of packets from all source nodes. For this purpose, we need to ensure that each node can deliver its weighted-share amount of packets. Another problem with the data collection networks is the congestion due to buffer overflow, which is not also insignificant [9][10]. To avoid buffer drops, before transmitting a packet each upstream node must be aware of whether there is sufficient free buffer space at the downstream node or not. An upstream node must be restricted from

Manuscript received April 02, 2007.
Manuscript revised June 30, 2007
[†]The authors are with the Department of Computer Engineering, Kyung Hee University. 1 Seocheon, Giheung, Yongin, Gyeonggi, 449-701 Korea
* This research was supported by the MIC and ITRC Project. Dr. C.S. Hong is the corresponding author
a) Email: mamun@networking.khu.ac.kr
b) Email: mahbub@networking.khu.ac.kr
c) Email: razzaque@networking.khu.ac.kr
d) Email: cshong@khu.ac.kr

transmitting packets when its downstream node does not have sufficient amount of buffer space. Even though HMAC gives higher medium access to a downstream node than that of its upstreams, it does not guarantee that (i) upstream nodes will transfer their weighted-share amount of packets and (ii) there will be no buffer drop at the downstream node. We propose *source count* based weighted round robin forwarding (WRRF) mechanism to address above two issues.

Although the sensor network can tolerate a certain percentage of packet loss, higher packet delivery ratio is desirable for detecting an event reliably. The combined effort of HMAC and WRRF highly increases the data transmission reliability of the network as well as the level of fairness significantly. We also propose a packet scheduling mechanism based on *source count* value namely in-node fair packet scheduling (FPS) to fine tune the level of fairness of the proposed protocol.

1.2 Contributions

The key contributions of this paper are: (i) energy efficient congestion avoidance protocol that includes *source count* based hierarchical medium access control (HMAC) and weighted round robin forwarding (WRRF), (ii) fair event detection and reliable data transmission, and (iii) probabilistic analysis of HMAC.

The rest of the paper is organized as follows. Section 2 describes network and system model, section 3 presents the proposed protocol and section 4 contains the probabilistic analysis. Performance evaluation is carried out in section 5, section 6 discusses related works and the paper concludes in section 7.

2. Network and System Model

We consider a network, where N sensing nodes are deployed with uniform random distribution over an area A. Node density is defined as $\rho = \frac{N}{A}$. Therefore, the approximate number of nodes within the event radius of a particular event is calculated as

$$N_s = \pi \rho R_s^2 \tag{1}$$

Where, R_s is the sensing range of each node. We consider a single sink in the network, placed at anywhere within the terrain.

All sensors are static and the network is homogeneous *i.e.*, all nodes have the same processing power and equal sensing and transmission range. Data generation rate of each sensing node is also assumed to be equal. Since receiving explicit ACK from the downstream node incurs huge overhead on energy constraint sensor nodes [6][7] [8], we have used snoop-based implicit acknowledgement. Modified CSMA/CA is used as MAC protocol. We do not need binary exponential *backoff*, as we exclude ACK packets in response of reception of data packets. Therefore, the backoff value is uniformly random within the range $0 \sim (W-1)$, where W is the size of contention window. The value of W is dynamically updated as traffic load varies (subsection 3.2). We consider that all data packets have the same size and the amount of buffer at each node is represented by the number of packets it can store. We also consider that congestion does not occur if there is no data transmission in the network.

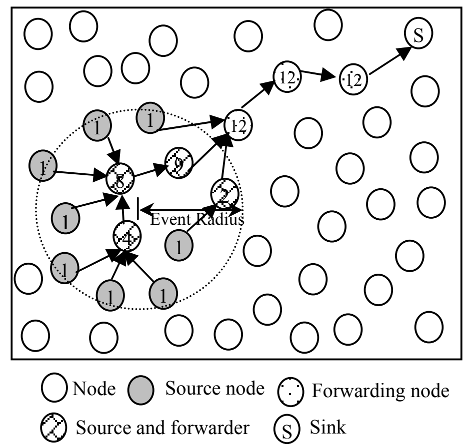

Fig. 1 Event to sink routing path and *source count* values of nodes

We have considered a tree based hierarchical static routing protocol. Hence, the route from each source to the sink is predetermined and unchanged during the data delivery of a certain event. The network is event driven; nodes within the event radius generate traffic and the sensed data eventually reach to the sink, forwarded by intermediary downstream nodes. Downstream node may also generate its own data by sensing the vicinity of its sensing range. As defined in subsection 1.1, *source count* value of any node *i,* denoted as SC_i, is the total number of source nodes for which it is forwarding data. If S_u represents the set of single hop upstream nodes of *i,* its *source count* value is calculated as follows

$$SC_i = \sum_{\forall k \in S_u} SC_k + I \tag{2}$$

Where, I is the source indicator function; $I=1$, if the node *i* is generating data, 0 otherwise. We do not use any type of control packets in designing different schemes of the proposed protocol. Since a downstream node requires knowing its *source count* value whenever it has some data packets to send, it is sufficient to propagate *SC* value along with the data packet. While transmitting data packets, each upstream node inserts its *source count* value in the packet header and the downstream node can easily calculate its *SC* value using Eq. 2. An upstream node learns the *source count* value of its downstream by snooping packets transmitted by the latter. Note that, a transient state exists between the event occurrence and the *SC* values of all downstream nodes are being stabilized.

SC value of a downstream node is stabilized whenever it receives at least one packet from all of its upstream nodes and therefore the network enters into steady state when the sink node receives at least one packet from each source node. Since the duration of transient state is very short (less than a second in our simulation), the effectiveness of the proposed protocol is not hampered. It is notable that, *source count* values of each node along the routing path are updated without transferring any additional control packets. Fig. 1 shows the updated *source count* values for each node in the routing path. This *source count* parameter works as a driving entity for all three schemes of our proposed protocol.

3. Proposed Protocol

3.1 Basic Idea

The key idea of the proposed protocol is as follows. We do not allow any node to get opportunistic access to the medium; rather we grant proportionate access that does not overwhelm the capacity of downstream nodes. From each downstream node, we allow upstream nodes to transfer their weighted-share number of packets in a round robin fashion. An upstream node forwards packets if its downstream node has sufficient buffer space. Downstream nodes maintain fairness in forwarding packets (received) from each of its single-hop upstream nodes. We use implicit block-ACK scheme to handle lost packets.

3.2 Hierarchical Medium Access Control (HMAC)

Due to many-to-one routing generalization [2] in sensor network (Fig. 1), downstream nodes have to carry more traffic than upstream nodes. Since CSMA/CA gives equal opportunity to all contending nodes, it causes huge loss of packets due to collision and increases medium contention. Upstream nodes must not transfer so high amount of data that can overwhelm the capacity of downstream nodes, particularly the nodes near to sink. Hence, we propose *source count* based hierarchical medium access control that gives proportional medium access to the nodes, *i.e.*, a node forwarding higher amount of traffic gets more access than others. Each node then calculates its contention window value using Eq. 3.

$$W(i) = CW_{\min} \times \frac{N_s}{SC_i} \tag{3}$$

We consider N_s, the number of nodes within the event radius, in calculating W because it has noteworthy impact in handling bursty traffic as well as aggregated load on downstream nodes. As discussed in section 2, the transmitting nodes choose a uniform random *backoff* value using Eq. 4.

$$backoff = rand(0 \sim (W-1)) \tag{4}$$

This proportional medium access significantly reduces the medium contention and congestion due to collision. Since N_s varies with the node density of the network, the value of W, calculated from Eq. 3, may be too high or too low. This may either lead to low medium utilization or overshoot the network capacity. To ensure the optimal contention window value, we incorporate the packet loss rate of each individual node for calculating W and the load adaptive equation is expressed as follows

$$W(i) = CW_{\min} \times \frac{N_s}{SC_i} \times \frac{1}{\alpha} \tag{5}$$

Where, α is a scaling factor that ranges from 0.5 to 1.5 based on channel contention. When a node has a packet to transmit, it gets the *backoff* value using Eq. 4 and transmits the packet when *backoff* value reduces to zero. Therefore, if the number of contending neighbors of a transmitting node is very low, lower value of α simply increases the medium access delay and reduces the network throughput. On the other hand, if the number of contending neighbors of a transmitting node is very high, a higher value of α increases the collision probability and thereby increases packet loss. The value of α is initialized to 1, which nullify its effect. Later on, to ensure efficient medium utilization, the value of α should be set carefully. A sharp increase or decrease of the value of α may also hinder the throughput of the network. Therefore, we have divided the range of α into 10 discrete values. Each node can easily identify the number of contending neighbors during the time between *backoff* assignment and packet transmission. In a round, if a node experiences collision with the current number of contending neighbors, it decreases the current value of α by 0.1 for the next round. Accordingly, if the node does not experience any collision, the α value is increased by 0.1.

3.3 Weighted Round Robin Forwarding (WRRF)

Even though the hierarchical medium access control gives more access to the nodes with higher *source count* values than others, it does not guarantee that (i) upstream nodes will transfer their weighted-share amount of packets and (ii) there will be no buffer drop at the downstream nodes. Buffer drop occurs if upstream nodes transfer packets even though there is no free buffer space at downstream. Probability also exists that a node may get multiple chances of medium access in succession and injects packets more than its share, depriving other nodes and worsening the fairness. Highly imbalance number of packets received at sink from different nodes engenders several potential problems: (i) unfair event detection, (ii) quick drainage of nodes' battery, and (iii) increased channel contention. To address these issues, we propose *source count* based weighted round robin forwarding that implements hop-by-hop fair packet delivery. It also ensures zero buffer drops, discussed in subsection 3.5.

In each round, a downstream node allows all of its upstream nodes to transmit their weighted-share amount of packets. If the downstream node allows its upstreams to transmit at most R packets in total in a round, any upstream node u can calculate its weighted-share number of packets as follows

$$S_w(u) = R \times \frac{SC_u}{SC_d} \tag{6}$$

Where SC_u and SC_d are *source count* values of upstream and downstream node, respectively. Each downstream node controls round individually, no synchronization is required among the nodes. For round controlling a single bit field, *round control*, is appended with each packet forwarded from it. Rounds cycle with 0 or 1, a new round is started with 0 and it remains unchanged until downstream node receives weighted-share number of packets from all of its upstream nodes. Upstream nodes get round value by snooping packets transmitted by its downstream node. An upstream node restricts itself from transmitting any further packet if it completes its share in that round. Thereafter, the downstream node switches *round control* bit to 1 and allows upstream nodes to transmit further packets. Thus, WRRF provides fair packet delivery in each routing path. It decreases channel contention and also takes care of balanced energy consumption by allowing equal packets from all nodes.

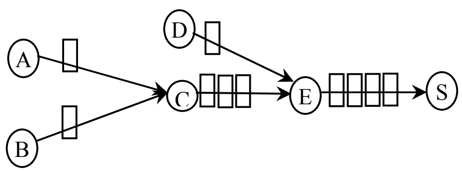

Fig. 2 Weighted round robin packet forwarding

For instance in Fig. 2, we consider R =6, the downstream node C allows 2 packets from A and 2 packets from B in each round. Similarly, E allows 4 packets from C and 1 packet from D in a round. The value of R and the round control mechanism are much related with the amount of buffer space in a node and its delivery rate, which is explicitly discussed in subsection 3.5.

3.4 In-node Fair Packet Scheduling (FPS)

Reliable data transmission does not guarantee that the event occurred in the network would be detected fairly. An event is said to be fair if the sink receives almost equal number of packets from all sensing nodes within the event radius. Thus, fair event detection requires the integrated effort of the following three:

- Proportionately fair medium access by the nodes
- Weighted-share packet delivery from upstream nodes
- In-node fair packet scheduling at each downstream node

The first two requirements are already achieved in subsections 3.2 and 3.3 respectively. The third requirement is addressed below.

A downstream node should forward packets received from each of its upstream nodes prioritized with their *source count* values. To accomplish this, the downstream node maintains separate virtual queues for each of its upstream nodes and a queue for the empty buffer management (depicted in Fig. 3). When a downstream node receives a packet, head of the bucket of the empty virtual queue is assigned and the bucket is placed at the tail of the respective virtual queue of the upstream node. After the successful transmission (acknowledgement and retransmission mechanisms are described in subsection 3.6) of a packet, bucket is made free and placed at the tail of empty virtual queue.

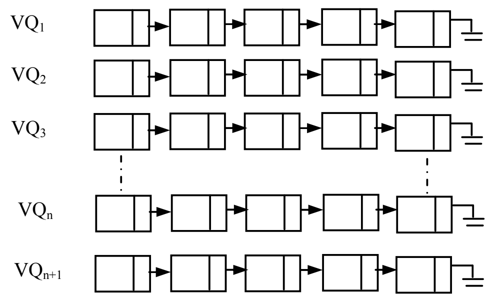

Fig. 3 $VQ_1 \sim VQ_n$ corresponds to virtual queues of upstream nodes. Queue VQ_{n+1} holds the list of free buffers

On reception of the first packet from an upstream node, downstream node assigns it a virtual queue and the queue weight is calculated according to Eq. 7.

$$w(VQ_{new}) = \max_{i \in 1 \sim n}(w(VQ_i)) + 1 \tag{7}$$

Where, VQ_1 to VQ_n are the existing virtual queues at the downstream node. When a packet is scheduled to be transmitted, the node selects packet from the virtual queue which has lowest weight. On transmission of each packet from the queue VQ_i, its weight is updated as Eq. 8.

$$w(VQ_i) = w(VQ_i) + \frac{P_{size}}{SC_i} \tag{8}$$

Where, P_{size} is the size of a packet in bytes and it varies from application to application. This guarantees proportionate packet delivery from each downstream node and thus ensures fair packet queueing in the nodes.

3.5 Congestion Avoidance

Congestion avoidance is a preemptive measure that manages the buffer of a node carefully so that it can avoid packet drops. This subsection describes how WRRF implements congestion avoidance at each downstream node.

Let's consider, in the current round, a downstream node forwards F packets to its next hop and receives R packets from its upstream nodes. If $R \le F$, the question of buffer drop never exists. But if $R > F$, then ($R - F$) packets are carried to the next round and if this is happened in several consecutive rounds, the buffer of the downstream node eventually becomes full and the packets are dropped. This may happen due to the randomness of medium access mechanism. Therefore, the round control operation in WRRF should address the following two conditions very carefully.

- Should the downstream node forward all R packets before changing the current round?
- Should the downstream node change the current round even though it has less than R empty buffers?

If the first condition is applied, it is guaranteed that there will be no buffer drop. We term this as hard congestion avoidance (HCA) since it is very conservative. However, as HMAC provides higher medium access probability to the downstream node than its upstreams over long period of time, downstream node may unnecessarily restricts its upstreams from transmitting packets even though it has sufficient buffers to receive more packets, which in turn decreases the network throughput.

On the other hand, the second condition implies soft congestion avoidance (SCA), which tries to keep the packet dropping closer to zero but utilizes the network resources efficiently. In this case, current round is changed as soon as possible. To implement this notion, a downstream node estimates the number of empty buffers ($R_{e\_estimated}$) requires to change the current round that might avoid buffer drops in the next round, where $0 < R_{e\_estimated} \le R$. If $R_{e\_current}$ represents the number of empty buffers when the downstream node receives all R packets from its upstreams, then the downstream node changes the current round if the condition $R_{e\_current} \ge R_{e\_estimated}$ holds, otherwise it waits until $R_{e\_current}$ raises to $R_{e\_estimated}$.

Now, it is critical to estimate $R_{e\_estimated}$ as packet delivery rate of a sensor node is unstable. However, for a realistic sensor network, current delivery rate depends on the historical delivery rate. Hence, we use exponential moving average formula (EWMA) to estimate $R_{e\_estimated}$ as follows:

$$R_{e\_estimated} = (1-\beta)\times R_{e\_estimated} + \beta \times R_{e\_current} \quad (9)$$

Where, β is a weight factor. A value of β closer to 1, calculates the estimated free buffers giving more emphasis on the current value, $R_{e\_current}$. Whereas, for very smaller value of $\beta(\beta << 1)$, the estimated value represents the historical behavior.

To find out a value of β that produces the best result, we have done an extensive evaluation in our simulation to measure packet drops of a downstream node that has 4 upstreams for various values of β . The results are shown in Table 1, a value of β = 0.1 produces the best results.

Table1: Effect of β value on the drop rate

Parameters	β	Total packet drops	Drop rate (%)
R =20 Rounds = 1500 Packets =30000	0.10	111	0.37
	0.15	281	0.93
	0.20	477	1.56
	0.25	603	2.01

3.6 Hop-by-Hop Implicit Block Acknowledgement and Loss Recovery

As described in subsection 3.4, each downstream node maintains separate virtual queues for its upstream nodes. Like synchronous explicit ACK, nodes do not wait for explicit acknowledgement. Rather they transmit as soon as they get the opportunity. The loss recovery mechanism is described as follows:

- When a packet is arrived at downstream node N_d from upstream node N_u , it is stored in the free buffer pointed by the head of the virtual queue VQ_{n+1}. Then that buffer is removed and added at the tail of VQ_u (the virtual queue maintained for node N_u at node N_d).
- When a packet of node N_u is forwarded from node N_d to the next hop, N_d inserts highest packet sequence number received in-sequence to indicate the block-ACK.
- N_u snoops packets forwarded by N_d and it learns successful reception of packets up to the sequence number found in the header. N_u then removes the list of buffers pointed by the successfully delivered packets and moves to the list of free buffers. On the other hand, if the snooped highest sequence number is not the same as the highest sequence number sent, N_u identifies that the snooped sequence number packet has been lost. If a loss is detected, the node retransmits the packet.
- After the retransmission of a packet, N_u removes the buffer holding the packet in the virtual queue and it is added at the tail of the free buffer list.
- Usually sink of a sensor network delivers data to the external network. As a consequence, nodes sending data directly to the sink will be unable to snoop packets. Sink implements aggregated ACK for the packets received from one hop away of it. This method is similar as described in [10].

Our simulation shows that with retransmission limit 1, proposed protocol achieves more than 90% delivery ratio and 98% efficiency.

4. Analytical Analysis

In this section, we analyze the hierarchical packet delivery rate and fairness of the proposed scheme.

4.1 Hierarchical Packet Delivery Rate

Before transmitting data, each node randomly selects a *backoff* value based on its contention window, W and waits until the *backoff* reduces to zero. Also, as we use implicit ACK, the transmitting node uses a uniform random *backoff* value from $0 \sim (W-1)$ instead of using binary exponential *backoff* value. Let *b(t)* be a stochastic process that represents the *backoff* counter of a particular node. If the medium is found free then each node decrements its *backoff* value by one in each slot time. Suppose p is the probability that the medium is busy in a slot time. Then the system can be modeled as a discrete time Markov chain and is given in Fig. 4. The non-zero one-step transition probabilities for this Markov chain are given in Eq. 10.

$$\begin{aligned} &P\{k\,|\,0\} = 1/W && k \in (0, W-1) \\ &P\{k\,|\,k\} = p && k \in (1, W-1) \\ &P\{k\,|\,k+1\} = 1-p && k \in (0, W-2) \end{aligned} \tag{10}$$

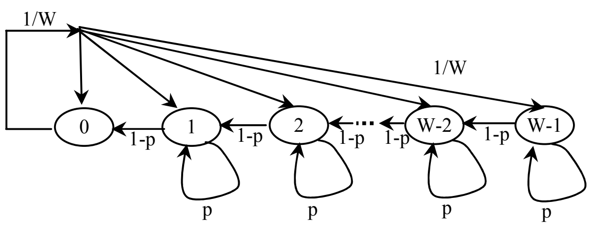

Fig. 4 Markov model of *backoff* counter

Let $b_k = \lim_{t\to\infty} P\{b(t) = k\}$ represents the stationary distribution of the chain, where $k \in (0, W-1)$. Then in steady state, the following relation is valid

$$b_k = \frac{W-k}{W}\frac{b_0}{1-p} \quad , 0 < k < W \tag{11}$$

Now, the value of b_0 can be determined by using the normalization condition

$$b_0 + \sum_{k=1}^{W-1} \frac{W-k}{W}\frac{b_0}{1-p} = 1 \tag{12}$$

Solving Eq. 12, we have the probability, b_0 that a node transmits in a slot time.

$$b_0 = \frac{2(1-p)}{W+1-2p} \tag{13}$$

Since a node only competes to transmit whenever the medium is free, to find the probability of transmission in a free slot time we can ignore *p.* Therefore, the steady state probability, τ, that a node transmits in a free slot time is

$$\tau = \frac{2}{W+1} \tag{14}$$

Since the steady state probability indicates the frequency that a node will be in a particular state in the long run, τ represents the fraction of packet transmissions of a node. Therefore, the ratio of the number of transmissions by a downstream node (N_d^{trans}) to that of an upstream node (N_u^{trans}) is given by

$$\frac{N_d^{trans}}{N_u^{trans}} = \frac{\tau_d}{\tau_u} \tag{15}$$

Putting the value of τ from Eq. 15 and combing with Eq. 3, we have

$$\frac{N_d^{trans}}{N_u^{trans}} = \frac{SC_d}{SC_u}\frac{CW_{\min}.N + SC_u}{CW_{\min}.N + SC_d} \tag{16}$$

Where, Eq. 16 can be approximated as,

$$\frac{N_d^{trans}}{N_u^{trans}} \approx \frac{SC_d}{SC_u} \tag{17}$$

Observation 1 - The ratio of number of transmissions by a downstream node to that of upstream node is equal to the ratio of their *source count* values.

Observation 2 - Since the *source count* value of a downstream node is at least equal to the sum of the *source count* values of all upstream nodes, the number of transmissions by the downstream node is equal to the sum of individual transmissions of all upstream nodes.

Observation 3 - Eq. 17 and weighted round robin forwarding also ensure that if a downstream node receives R packets from all upstream nodes, then packet forwarded by the upstream node u is equal to $R \times \frac{SC_u}{SC_d}$. That is, data forwarded by the upstream nodes are proportional to their *source count* values.

4.2 Fair Event Detection

The integrated employment of hierarchical medium access, weighted round robin and in-node fair packet scheduling ensures that nodes only forward packets based on their respective *source count* values. Therefore, it can easily be shown that the number of packets received by the sink from any source node i is

$$\eta_i = \left\lfloor \frac{P_{recv}}{SC_i} \right\rfloor \tag{18}$$

Where η_i is the number of packets received from source *i*, P_{recv} is the total number packets received by the sink.

5. Performance Evaluation

To evaluate the performance of our proposed protocol we have performed extensive simulations using *ns-2*[16].

5.1 Protocol Implementation

Proposed protocol implementation includes a tree based hierarchical static routing protocol, HMAC, WRRF and FPS. The tree based hierarchical static routing module creates parent (downstream) and child (upstream) hierarchy among the nodes in the network. The routing tree is constructed using Warshall's algorithm so that the sensed data could reach the sink with shortest number of hops. It may be mentioned here that, the choice of downstream nodes does not depend on any traditional parameters of sensor network routing *e.g.*, energy or delay.

An event is generated at a random location and we have assigned source IDs randomly to the nodes within the event radius. We have modified the CSMA/CA MAC implementation of *ns-2* as follows. We have added two additional fields in the MAC frame header: *source count* and *round control*. At each downstream node, virtual queues are created using link list data structure for storing packets from individual upstream nodes. Dissemination and update operation of *source count* value is described in section 2. When a node has data to transmit, HMAC takes care of assigning its *backoff* value, WRRF calculates the weighted-share number of packets to transmit and finally FPS determines which packet to schedule.

5.2 Simulation Metrics and Parameters

The following metrics are used to realize the performance of proposed schemes:

- Delivery Ratio: It indicates the ratio of number of packets sent by the sources to the number of unique packets successfully received at sink.
- Collision Drop Rate: The ratio of dropped packets due to collision to the number of sent packets.
- Buffer Drop Rate: The ratio of dropped packets due to buffer overflow to the number of sent packets.
- Efficiency: It is measured as the ratio of number of hops traveled by each successful reception of a packet at the sink to the total number of transmissions required for the packet in the entire path.
- Fairness: This is the ratio of number of unique packets received by the sink from each source node to the number of packets sent by them.
- Energy Dissipation: Amount of energy dissipated per node per unit time, measured in Joule.

The performance comparisons of the following four mechanisms are carried out.

- No Congestion Control (NoCC): Under this scheme packets are transmitted without controlling transmission rates at the sources and forwarders.
- No Congestion Control with Implicit ACK (NoCC-IA): This is the same scheme as NoCC without RTS-CTS-DATA-ACK handshake. It uses snoop based implicit ACK.
- Backpressure: It is a hop-by-hop rate control based congestion control mechanism explained in CODA [1]. If a sensor gets congested, the mechanism advertises congestion using explicit congestion notification bit to reduce the transmission rate of its upstream sensors by a factor of 0.5. If an upstream neighbor is a data source, the neighbor reduces the rate at which it generates new data by the same percentage.
- Proposed Protocol: It includes HMAC, WRRF and FPS. Where, WRRF implements soft congestion avoidance (SCA).

Table 2. Simulation parameters

Parameter	Value
Total Area	100m X 100m
Number of nodes	100
Initial Energy	5 Joule/Node
Transmission power	5.85e-5 watt
Receive signal threshold	3.152e-20 watt
Data rate	300 kbps
Transmission Range	30m
Packet size	64 bytes
Initial α value	1.0
Range of α	0.5~1.5
β value	0.1
Buffer size	20
Event radius	14m ~22m
Number of Sources within the event radius	6~15
Offered load	4~6 pkts. per sec. (pps)
Sink location	[3.6148, 99.2246]
Simulation Time	50 Sec

5.3 Simulation Results

Fig. 5 shows collision drop rate for various protocols. In case of NoCC, drop rate increases sharply with the increased data sources. Since nodes within the event radius generate bursts of data packets and opportunistic medium access (CSMA/CA) provides equal share to all nodes, huge collision occurs and it becomes severe with the increase of number of sources. Algorithms using backpressure also cannot reduce the collision drop rate by a significant amount, since they also use opportunistic medium access for all nodes. On the other hand, the proposed protocol exhibits very less collision drop due to the use of hierarchical and controlled medium access which is implemented by the integrated employment of HMAC and WRRF. We have found that without HMAC, the proposed protocol experiences comparatively large number of collisions but still somewhat less than backpressure algorithms. This is obvious as we do not need to propagate congestion notification message in backward direction (like backpressure algorithms), rather each upstream node's transmission is controlled by its downstream (single-hop control).

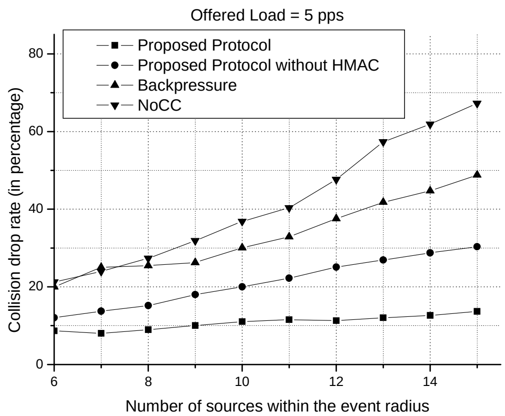

Fig. 5 Collision drop rate with increasing number of source nodes.

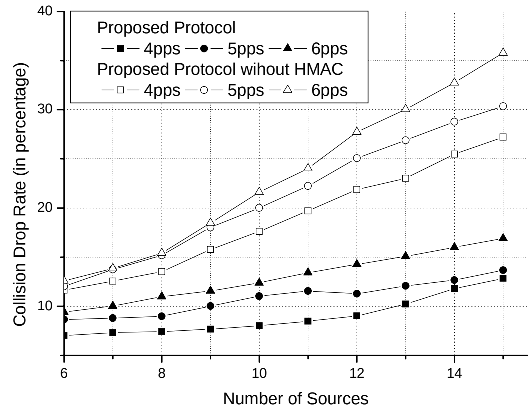

Fig. 6 Collision drop rate under various load, ranges from 4~6 pps

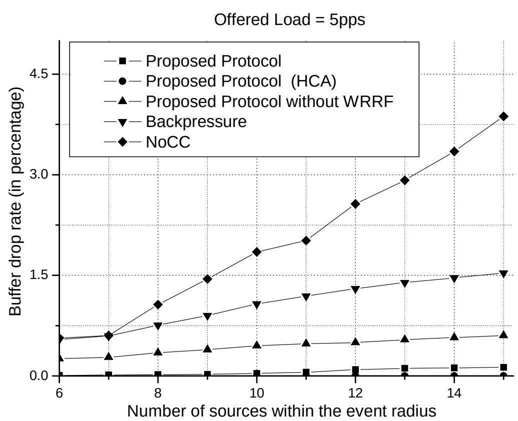

Fig. 7 Buffer drop rate with increasing number of sources

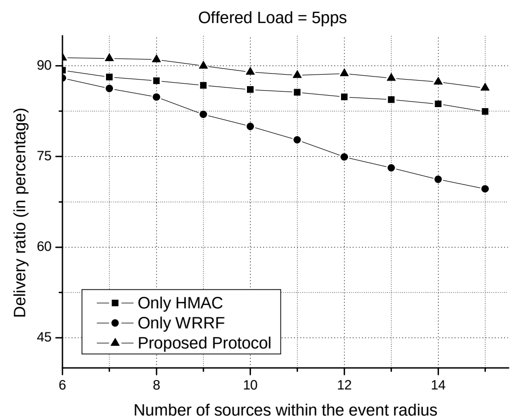

Fig. 8 Delivery ratio of individual schemes and the proposed protocol

Fig. 6 depicts that the collision drop rate experienced by proposed protocol increases with the increased offered load and number of sources within the event radius. Also, without HMAC (*i.e.*, protocol with only WRRF) the proposed protocol has a higher collision drop rate, because in this case all nodes will equally contend for the medium irrespective of their *source count* values.

Buffer drop rates of different protocols are plotted in Fig. 7. In fact, buffer drop is relatively much less than collision drops [5]. Uncontrolled rate of transmission mechanism in NoCC is the main reason of higher packet drops. Backpressure algorithms experience lower packet drops than NoCC, since they use rate control mechanism to control congestion. However, our proposed protocol experiences zero buffer drop with HCA and very less drops (near to zero) with SCA and the results are congruent with the theoretical expectation. Even if we do not use WRRF, packet drop rate is very low. This is obvious since HMAC ensures that the delivery rate of a downstream node is almost equal to the sum total rates of its upstream nodes, except in the condition that the forward link is very bad.

Fig. 8 depicts the effectiveness of proposed HMAC and WRRF schemes individually and their combined effort in terms of delivery ratio. Only WRRF can achieve the least delivery ratio since in this case all nodes equally contend for the medium irrespective of their source count values and, thereby, increase the collision drop rate. HMAC exhibits lower delivery ratio than the combined effort of HMAC and WRRF as it does not care about the buffer drops. Our proposed protocol can achieve around 90% delivery ratio.

According to Fig. 5 and Fig. 7, since both NoCC and backpressure algorithms experience comparatively large amount of collision and buffer drops, their ultimate delivery ratio is very poor. While the integrated employment of HMAC and WRRF in our proposed protocol provides better delivery ratio as depicted in Fig. 9.

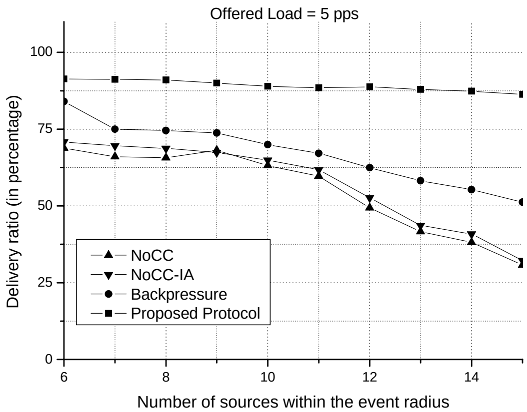

Fig. 9 Delivery ratio with a rate of 5pps

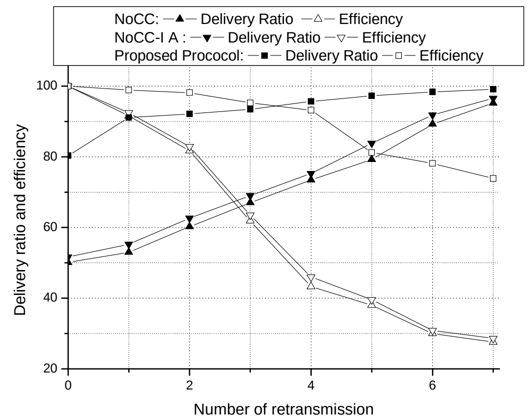

Fig. 10 Delivery ratio and efficiency with a load of 5pps

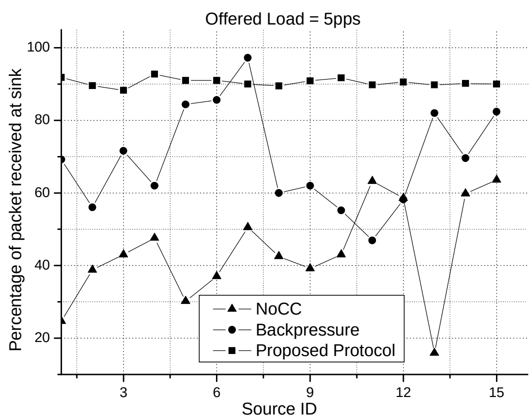

Fig. 11 Percentage of unique packets received at sink from different sources.

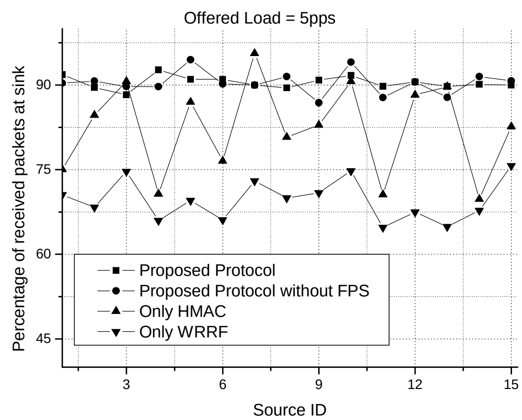

Fig. 12 Percentage of unique packets received at sink from different sources for individual schemes and proposed protocol.

Fig. 10 shows the delivery ratio and efficiency for different retransmission limits. From the graphs in the figure, it can easily be perceived that the higher number of retransmissions increases the delivery ratio but lowers the efficiency. However, lower efficiency is undesirable for energy constraint sensor network. Our proposed protocol can achieve more than 90% delivery ratio and 98% efficiency with retransmission limit 1, which is good enough for reliable event perception.

Fig. 11 shows comparative results of various protocols for fair event detection. Since WRRF ensures that a downstream node receives weighted-share number of packets from its upstream nodes and FPS fairly schedules packets from the virtual queues of the downstream node, the proposed protocol guarantees that almost equal number of packets from each source is received by the sink. On the other hand, NoCC and backpressure algorithms suffer from poor fairness due to the use of opportunistic medium access control and FCFS in-node packet forwarding policy.

It can easily be observed from the graphs of Fig. 12 that WRRF is the most dominant for achieving fairness, while the combined effort of HMAC and WRRF gains a significant amount of fairness. Moreover, the incorporation of FPS fine tunes the fairness level of the proposed protocol. The results of Fig. 13 depicts that irrespective of the offered load, the proposed protocol guarantees fair event detection. Fig. 14 shows the same result in terms of the number of packets received.

Average energy dissipation of individual nodes for various protocols is depicted in Fig. 15. Proposed protocol achieves better energy efficiency than NoCC, NoCC-IA and backpressure algorithms, on an average, approximately by a factor 1.998, 1.586 and 1.808 respectively. The rationale behind this result can be explained as follows: *firstly*, loss of energy due to packet drops (collision and buffer drops) is greatly reduced in the proposed protocol as compared to the existing ones and *secondly*, number of retransmissions at each downstream node is also reduced which in turn saves energy. Finally, the use of snoop based acknowledgement further reduces energy consumption.

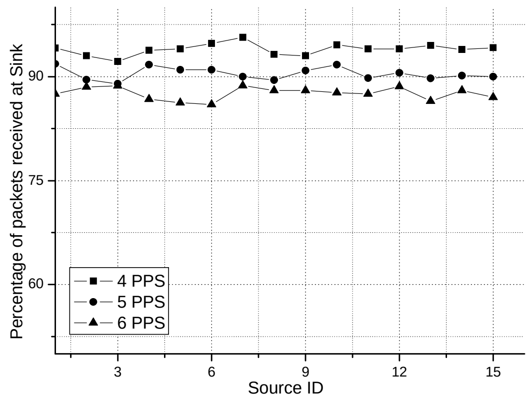

Fig. 13 Percentage of unique packets received at sink from different sources under a varying traffic load between 4~6 pps.

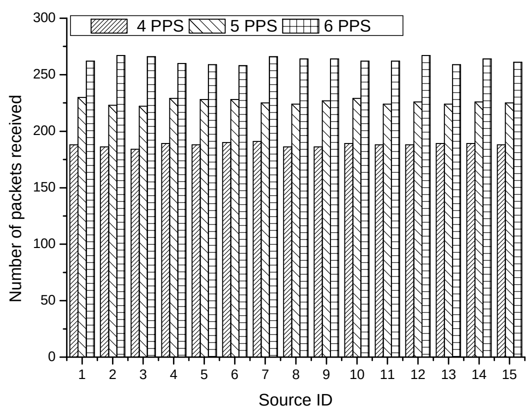

Fig. 14 Number of unique packets received at sink from different sources under a traffic load between 4~6 pps

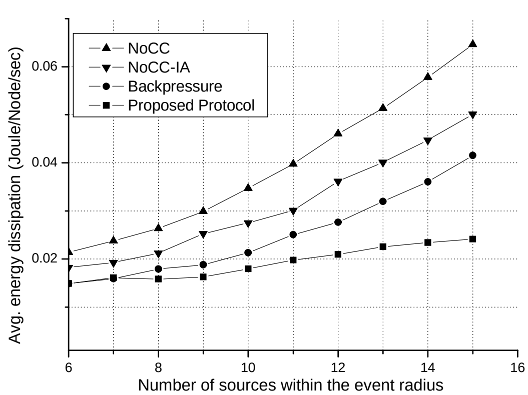

Fig. 15 Average energy dissipation

6. Related Work

In ESRT [4], on reception of packets with congestion notification bit high, sink node regulates the reporting rate by broadcasting a high energy control signal so that it could reach to all sources. This high powered congestion control signal may disrupt some other transmissions.

CODA [1] uses a combination of the present and past channel loading conditions and the current buffer occupancy, to infer accurate detection of congestion at each downstream node with low cost. As long as a node detects congestion, it sends backpressure messages to upstream nodes for controlling reporting rate hop-by-hop. It is also capable of asserting congestion control over multiple sources from a single sink in the event of persistent congestion. Even though it overcomes some of the limitations of ESRT [4], it doesn't consider the event fairness and packet reliability at all. PSFQ [11] is a scalable and reliable transport protocol that deals with strict data delivery guarantees rather than desired event reliability as it is done in ESRT. However, this approach involves highly specialized parameter tuning and accurate timing configuration that makes it unsuitable for many applications. As defined in Many-to-One Routing [2], event fairness is achieved when equal number of packets is received from each node. In this proposal, individual node divides its effective available bandwidth equally amongst all upstream nodes, which in turn ensures fairness. Several disadvantages are including: (i) it provides no reliability guarantee and (ii) the effective throughput may decrease due to implementation of ACK in transport layer.

RMST [12] is a transport layer paradigm designed to complement directed diffusion [13] by adding a reliable data transport service on top of it. It is a NACK based protocol like PSFQ, which has primarily timer driven loss detection and repair mechanisms. It does not provide with any congestion control mechanism.

Rather than rate based congestion control technique, TARA [14] uses resource based congestion alleviation. Like other algorithms it addresses hotspot problem [15] in sensor network. Capacity analysis model of TARA estimates the end-to-end throughput of a network under different topology and finally it controls the sleep and wake status of nodes to alleviate congestion.

7. Conclusions

The significant contribution of this paper is the introduction of *source count* based HMAC and WRRF schemes. The integrated employment of HMAC and WRRF greatly reduces the collision drops and avoids buffer drops, which in turn increases the data transmission reliability of network. Also, the joint effort of WRRF and

FPS achieves fair event detection. The steady state probabilistic analysis states that the ratio of number of transmissions by a downstream node to that of an upstream node is equal to the ratio of their *source count* values. Finally, the proposed protocol achieves comparatively lower energy consumptions than other protocols.

References

[1] C. Y. Wan, S.B. Eisenman, and A.T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," Proc. ACM SenSys, Los Angeles, pp. 266-279, Nov. 2003.

[2] C.T. Ee and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," Proc. ACM SenSys, Baltimore, pp. 148-161, Nov. 2004

[3] C. Wang; K. Sohraby, B. Li, M. Daneshmand, Y. Hu," A survey of transport protocols for wireless sensor networks," Proc. IEEE Network Magazine, Vol. 20, no. 3, pp. 34-40, May 2006.

[4] Y. Sankarasubramaniam, A. Ozgur, I. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in wireless sensor networks," Proc. ACM Mobihoc pp. 177-189, 2003

[5] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks," Proc. of the ACM SenSys, pp. 134-147, Nov. 2004

[6] A. Woo, D. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," Proc. ACM MobiCom, pp. 221-235, July 2001.

[7] P. Xie, J. H. Cui, "SDRT: A Reliable Data Transport Protocol for Underwater Sensor Networks," UCONN CSE Technical Report: UbiNet-TR06-03. February 2006.

[8] S. Park, R. Vedantham, R. Sivakumar and I. Akyildiz, "A Scalable Approach for Reliable Downstream Data Delivery in Wireless Sensor Networks," Proc. ACM MobiHoc, pp. 78-89, 2004

[9] S. Chen, N. Yang, "Congestion Avoidance Based on Lightweight Buffer Management in Sensor Networks," Proc. IEEE Transaction on Parallel and Distributed Systems, Vol. 17, No. 9, pp. 934-946, 2006

[10] H. Zhang, A. Arora, Y. Choi, M. G. Gouda, "Reliable Bursty Convergecast in wireless Sensor Networks," Proc. ACM MobiHoc, pp. 266-276, 2005

[11] C.Y. Wan, A.T. Campbell, L. Krishnamurthy, "Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks," Proc. IEEE JSAC, pp.862-872, 2005

[12] R. Stann and J. Heidemann, "RMST: Reliable data transport in sensor networks," Proc. IEEE SPNA, Anchorage, Alaska, pp. 102-112, May 2003

[13] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," Proc. ACM MobiCom '00, Boston, MA, pp. 56-67, 2000

[14] J. Kang, Y. Zhang and B. Nath, "TARA: Topology Aware Resource Adaptation to Alleviate Congestion in Sensor Networks," Proc. IEEE Transaction on Parallel and Distributed Systems, Vol. 18, Issue 7, pp. 919-931, July, 2007

[15] M. Perillo, Z. Cheng, W. Heinzelman, "An Analysis of Strategies for Mitigating the Sensor Network Hot Spot Problem," Proc. MobiQuitous, pp. 474-478, July 2005

[16] The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/index.html

Md. Mamun-Or-Rashid received the B.S. and M.S. degree in Computer Science from University of Dhaka, Bangladesh, in the year 2000 and 2002 respectively. During 2003-2004, he worked as a faculty member in Department of Computer Science and Engineering, University of Dhaka, Bangladesh. His research interest is in the area of modeling, analysis and optimization of protocols and architectures for wireless sensor networks, distributed systems, etc. He is now PhD candidate in the Department of Computer Engineering, Kyung Hee University, Korea.

Muhammad Mahbub Alam received the B.S. degree in Applied Physics and Electronics and M.S. degree in Computer Science from the University of Dhaka, Bangladesh in 1998 and 2000 respectively. His research interests include wireless and mobile networking and performance modeling and analysis of networking systems. Now he is a PhD student in the Department of Computer Engineering, Kyung Hee University, Korea.

Md. Abdur Razzaque received the B.S. degree in Applied Physics and Electronics and M.S. degree in Computer Science from the University of Dhaka, Bangladesh in 1997 and 1999 respectively. He worked as a faculty in the Department of Computer Science and Engineering, University of Dhaka. His research interest is in the area of modeling, analysis and optimization of protocols and architectures for wireless sensor networks, distributed systems, etc. He is now PhD student in the Department of Computer Engineering, Kyung Hee University, Korea.

Choong Seon Hong received his B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, Korea, in 1983, 1985, respectively. In 1988 he joined KT, where he worked on Broadband Networks as a member of the technical staff. From September 1993, he joined Keio University, Japan. He received the PhD degree at Keio University in March 1997. He had worked for the Telecommunications Network Lab, KT as a senior member of technical staff and as a director of the networking research team until August 1999. Since September 1999, he has been working as a professor of the School of Electronics and Information, Kyung Hee University. He has served as a Program Committee Member and an Organizing Committee Member for International conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN and TINA. His research interests include ad hoc networks, network security and network management. He is a member of IEEE, IPSJ, KIPS, KICS and KISS.