Reliable Event Detection and Congestion Avoidance in Wireless Sensor Networks

Md. Mamun-Or-Rashid, Muhammad Mahbub Alam, Md. Abdur Razzaque, and Choong Seon Hong*

Networking Lab, Department of Computer Engineering, Kyung Hee University
Giheung, Yongin, Gyeonggi, 449-701 South Korea
{mamun,mahbub,razzaque}@networking.khu.ac.kr,
cshong@khu.ac.kr

Abstract. Due to dense deployment and innumerable amount of traffic flow in wireless sensor networks (WSNs), congestion becomes more common phenomenon from simple periodic traffic to unpredictable bursts of messages triggered by external events. Even for simple network topology and periodic traffic, congestion is a likely event due to time varying wireless channel condition and contention caused due to interference by concurrent transmissions. Congestion causes huge packet loss and thus hinders reliable event perception. In this paper, we present a congestion avoidance protocol that includes *source count* based hierarchical medium access control (HMAC) and weighted round robin forwarding (WRRF). Simulation results show that our proposed schemes avoid packet drop due to buffer overflow and achieves more than 90% delivery ratio even under bursty traffic condition, which is good enough for reliable 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 messages triggered by sensed events. Therefore, congestion happens due to contention caused by concurrent transmissions, buffer overflows and dynamically 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 degrade its performance drastically [4]. Congestion causes many folds of drawbacks: (i) increases energy dissipation rates of sensor nodes, (ii) causes a lot of packet loss, which in turn diminish the network throughput and (iii) hinders fair event detections and reliable data transmissions. Therefore, congestion control or congestion avoidance has become very crucial to achieve reliable event detection for the practical realization of WSN based envisioned applications.

* This research was supported by the MIC , Korea, under the ITRC support program supervised by the IITA, Grant no-(IITA-2006-(C1090-0602-0002)).

R. Perrott et al. (Eds.): HPCC 2007, LNCS 4782, pp. 521 – 532, 2007.

We find that one of the key reasons of congestion in WSN is allowing sensing nodes to transfer as many packets as they can. This is due to the use of opportunistic media access control. The high amount of data transferred by sensing nodes can overwhelm the capacity of downstream nodes, particularly the nodes near to sink. Hence, we propose *source count* (defined in section 3) based hierarchical medium access control (HMAC) that gives proportional access, i.e. a node carrying higher amount of traffic gets more access to the medium than others. Therefore, downstream nodes obtain higher access to the medium than the upstream nodes. This access pattern is controlled with local values and is made load adaptive to cope up with various application scenarios.

Congestion due to buffer overflow is not insignificant [9][10]. To avoid congestion, before transmitting a packet each upstream node must be aware whether there is sufficient free buffer space at the downstream node. To implement this notion, we restrict an upstream node from delivering packets when its downstream node has not sufficient amount of free buffer space. This is achieved by our proposed *source count* based weighted round robin forwarding (WRRF).

Even though the sensor network can tolerate a certain percentage of packet loss, data transmission in such network is termed as unreliable if the packet delivery ratio decreases to very low value so that the event can not be detected reliably. We thus seek an efficient way to avoid congestion within the sensor network to ensure good delivery ratio for reliable event detection. Integrated effort of our proposed *source count* based HMAC and WRRF reduces packet drop due to collision and avoids packet drop due to buffer overflow and finally achieves more than 90% delivery ratio which is good enough for reliable event perception.

The rest of the paper is organized as follows: section 2 briefly discusses the related works, section 3 articulates the network model and assumption, section 4 describes our proposed protocol, section 5 discusses the performance evaluation and finally we conclude in section 6.

2 Related Work

A good number of transport layer protocols have been proposed for wireless sensor network [1]-[10]. These works aimed to provide reliability guarantee either by congestion detection and control or by congestion avoidance [1][9][14]. Few of these techniques are described below:

ESRT [4] allocates transmission rate to sensors such that an application-defined number of sensor readings are received at a base station, while ensuring the network is not congested. 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. Also the assumption of congestion notification by the sink node is very optimistic. 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 receiver 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 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. Also PSFQ has several disadvantages (i) it cannot detect single packet loss since they use only NACK, (ii) it uses statically and slowly pump that result in large delay and finally (iii) it requires more buffer as hop-by-hop mechanism is used. As defined in Many-to-One Routing [2], event fairness is achieved when equal number of packets are received from each node. In this proposal, individual nodes divide its effective available bandwidth equally amongst all upstream nodes. This, in turn ensures fairness. Several disadvantages are including: (i) It provides no reliability guarantee. (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's 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. TARA [14] discusses the network hotspot problem and presents a topology aware resource adaptation strategy to alleviate congestion in sensor network.

In our approach two key reasons of packet loss have been taken into account: loss due to collision and loss due to buffer overflow. Our proposed HMAC scheme takes care of hierarchical medium access and thereby reduces packet drops due to collision. WRRF controls the number of packets to be received from upstream nodes in each round (single-hop control). Round operation is controlled by estimating buffer status at each individual downstream node using exponential moving average (EWMA). A downstream node allows packet from its upstream nodes only if there is available buffer and thereby avoid drops due to buffer overflow.

3 Network Model and Assumptions

We consider a network of N sensing nodes, deployed with uniform random distribution over an area A. Node density is defined as $\rho = N / A$. Therefore, the approximate number of nodes within the sensing 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; we do not consider mobile sensors that form a dynamic ad-hoc network.

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 4.1). 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.

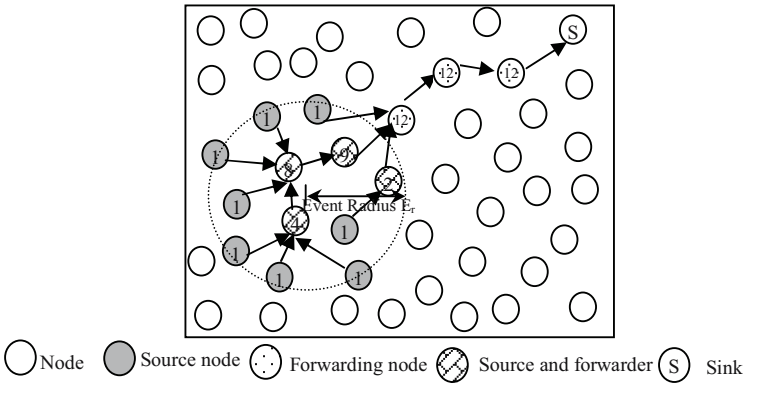


Fig. 1. Event to sink routing path and source count value of nodes

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 schemes of our proposed protocol.

4 Proposed Protocol

The key idea of our proposed congestion avoidance protocol is as follows. We do not allow any node the 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.

4.1 Hierarchical Medium Access Control (HMAC)

Due to many-to-one routing generalization [2] in sensor network (shown in Fig. 1), downstream nodes have to carry more traffic than upstream nodes. Therefore, as simple CSMA/CA gives equal opportunity to all contending nodes, it might cause huge loss of packets due to collision and increase media contention.

Sensing 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 hierarchical medium access control (HMAC) that gives proportional access based on *source count* value, i.e. a node carrying higher amount of traffic gets more accesses than others. Each node then calculates its contention window using equation (3).

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

We consider N_s, a global system parameter, in calculating W as because it has noteworthy impact in handling bursty traffic condition as well as aggregated load on downstream nodes. As we use implicit ACK, the transmitting nodes do not use binary exponential backoff procedure; instead they choose a uniformly random backoff value using equation (4).

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

This proportional media access significantly reduces the media contention and congestion due to collision. The value of W calculated from equation (3) is the approximate value of N_s, which may not be equal to the approximate number of nodes in a practical sensor network all the times. This may 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. So, 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.

4.2 Weighted Round Robin Forwarding (WRRF)

Even though the HMAC gives more accesses to nodes with higher *source count* values than others, it does not guarantee that upstream nodes will transfer their weighted-share amount of packets. Probability exists that a node may get multiple chances 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) event detection may be biased, (ii) nodes' battery will be drained quickly resulting early node failure and (iii) increased channel contention.

To address this issue, we propose *source count* based weighted round robin forwarding (WRRF) that implements hop-by-hop fair packet scheduling. 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 R packets to be transmitted by all of its upstream nodes in a round and SC_d is its *source count* value, the weighted-share number of packets of any of its upstream node i, S_w, is calculated as follows:

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

Round is controlled by downstream node and a single bit field, *round control*, is appended with each packet forwarded from it. Rounds cycle with 0 and 1 values, a new round is started with 0 in *round control* bit 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

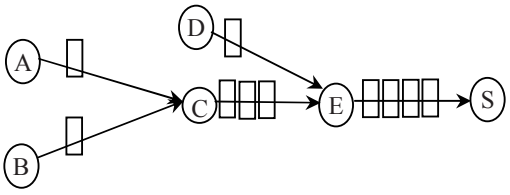


Fig. 2. Weighted round robin packet forwarding

it completes its share in that round. Thereafter, the downstream node switches *round control* bit to 1 and transmits further packets.

For instance, we consider R=6 in Fig. 2, the downstream node C allows 2 packet from A and 2 packet 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 are explicitly discussed in section 4.3.

Thus controlling packet transmission in round robin fashion provides fair packet delivery in each routing path. It decreases channel contention and also takes care of nodes energy by allowing equal packets from all nodes.

4.3 Congestion Avoidance

The proposed mechanism avoids packet drops due to congestion by not allowing upstream nodes to transmit if there is not available buffer. This is achieved by controlling the transmission round explained in section 4.2. A downstream node changes the round based on its buffer status. In a round, the downstream node allows R packets to be transmitted by all the upstream nodes. Given the condition that the downstream node has proportionally higher probability to access the medium, even then the downstream node may not be able to forward all R packets. So, in the next round, the downstream node will have some packets from the previous round, which might cause congestion within few successive rounds. Therefore, two important things to be considered are:

- The value of R and its relations with buffer size
- After getting R packets from the upstream node whether the downstream node immediately change the round or not. More specifically, should the downstream node forward all R packets before changing the round?

Definitely, this will guarantee no packet drop due to congestion. However, such strict round robin forwarding will block the upstream nodes to transmit, even though there may be empty buffer in the downstream node. Therefore, it is important to know R_e the number of empty buffers in the downstream node, where $R_e \leq R$ is required to change the current round. If the number of packets forwarded by the downstream node is F, when it receives the last packet of that round from one of the downstream nodes, then the downstream node should have at least $R_e = R - F$ empty buffers.

Similarly, in the next round, the downstream node requires R_e empty buffers to change the round and thus can avoid congestion. But R_e is not a constant and depends on the network load. We therefore find the value of R_e using the exponential weighted moving average (WEMA) and given by:

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

Finally, as long as there are not at least $R_{eestimited}$ number of empty buffers, the downstream node will not change the transmission round. This ensures near to zero packet drops and at the same time ensures efficient buffer utilization.

5 Performance Evaluation

To evaluate the performance of our proposed schemes we have performed extensive simulations using *ns-2*[15]. Proposed protocol implementation includes a tree based hierarchical static routing protocol, HMAC and WRRF. 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 and WRRF calculates the weighted-share number of packets to transmit.

Following matrices 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 packets successfully received at sink.
- Packet Drop: The ratio of dropped packets due to collision and buffer overflow to the number of sent packets
- Efficiency: Number of hops traveled by each successful reception of a packet at the sink divided by the total number of transmission required for the packet in entire path.
- Energy Dissipation: Amount of energy dissipated per node per unit time, measured in Joule

We have compared our protocol with the following four mechanisms:

- No Congestion Control (NoCC): Under this scheme packets are transmitted using a hierarchical routing 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 load adaptive hierarchical medium access (HMAC) and weighted round robin forwarding (WRRF).

5.1 Simulation Setup Parameters

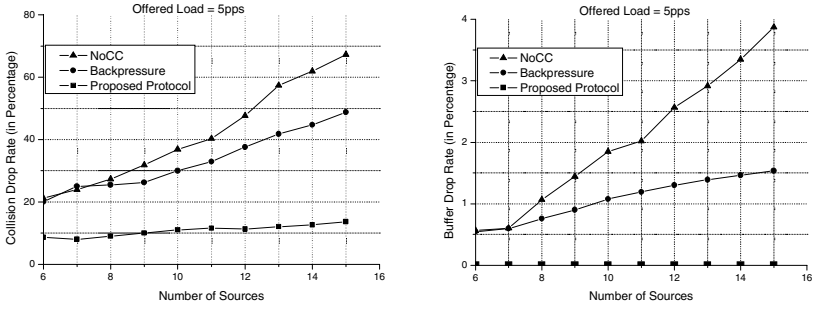
Table 1. Simulation parameters

Parameter	Value
Total Area	100m X 100m
Number of nodes	100
Initial Energy	5 Joule/Node
Transmission power	5.85e-5
Receive signal threshold	3.152e-20
Data rate	300 kbps
Transmission Range	30m
Packet size	64 bytes
Initial α value	1.0
Range of α	0.5~1.5
Buffer size	20
Data Sources	1~15
Offered load	4~6 pkts. per sec. (pps)
Sink location	[3.6148, 99.2246]
Simulation Time	50 Sec

5.2 Simulation Results

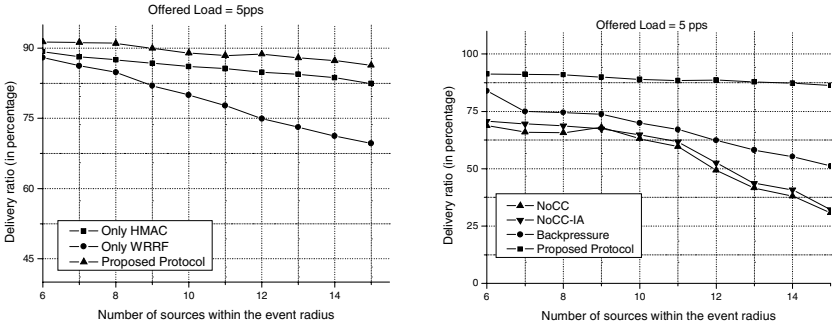
Fig. 3.a shows collision drop rate for various protocols. In case of NoCC, drop rate increases sharply with the increased data sources. Since nodes within the event radius generate bursts of data packets and opportunistic medium access (CSMA/CA) provides equal share to all nodes, huge collision occurs 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.

Buffer drop rates of different protocols are plotted in Fig. 3.b 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



a. Drops due to collision **b.** Drops due to buffer overflow

Fig. 3. Packet drop rate due to collision and buffer overflow with increasing number of sources



a. Comparison among individual schemes **b.** Comparison with other protocols

Fig. 4. Delivery ratio of individual schemes as well as their integrated effort and other protocols with a load of 5 pps

algorithms experience lower packet drops than NoCC, since they use rate control mechanism to control congestion. However, our algorithm completely avoids packet loss due to buffer overflow.

Fig. 4.a 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. 3.a and Fig. 3.b, 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

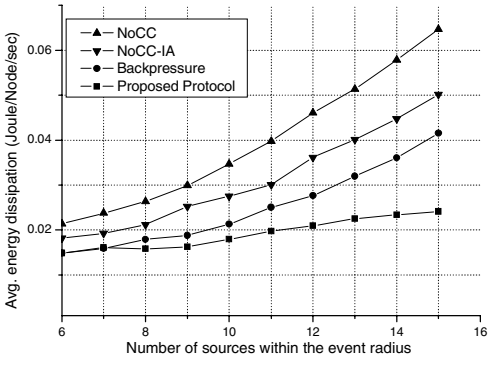


Fig. 5. Average energy dissipation with a load of 5 pps

HMAC and WRRF in our proposed protocol provides better delivery ratio than other protocols (NoCC, NoCC-IA and Backpressure) as depicted in Fig. 4.b.

Average energy dissipation of individual nodes for various protocols is depicted in Fig. 5. 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.

6 Conclusions

Reliable event perception is essential for collaborative actions in many envisioned applications of sensor networks. Congestion in WSNs causes huge packet loss and thereby hinders reliable event detection. We found that two major reasons of congestion are (i) congestion due to collision and (ii) congestion due to buffer overflow. To reduce packet loss and achieve a fair delivery ratio we propose a *source count* based hierarchical medium access control (HMAC) and weighted round robin forwarding (WRRF). HMAC ensures a hierarchical access of the medium according to the *source count* value. While WRRF assigns a weighted-share of packet delivery to the downstream node. These two schemes together greatly reduce media contention and thereby congestion due to collision. Congestion due to buffer overflow is completely avoided. We have utilized the source count value as a driving parameter for the schemes of our proposed protocol. Our proposed protocol greatly reduces packet loss due congestion, exhibits a higher delivery ratio, which is very necessary for reliable event detection.

References

1. Wan, C.Y., Eisenman, S.B., Campbell, A.T.: CODA: Congestion Detection and Avoidance in Sensor Networks. In: the proceedings of ACM SenSys, Los Angeles, pp. 266–279. ACM Press, New York (2003)
2. Ee, C.T., Bajcsy, R.: Congestion Control and Fairness for Many-to-One Routing in Sensor Networks. In: the proceedings of ACM SenSys, Baltimore, pp. 148–161. ACM Press, New York (2004)
3. Wang, C., Sohraby, K., Li, B., Daneshmand, M., Hu, Y.: A survey of transport protocols for wireless sensor networks. IEEE Network Magazine 20(3), 34–40 (2006)
4. Sankarasubramaniam, Y., Ozgur, A., Akyildiz, I.: ESRT Event-to-Sink Reliable Transport in wireless sensor networks. In: the proceedings of ACM Mobihoc, pp. 177–189. ACM Press, New York (2003)
5. Hull, B., Jamieson, K., Balakrishnan, H.: Mitigating Congestion in Wireless Sensor Networks. In: the proceedings of ACM SenSys, pp. 134–147. ACM Press, New York (2004)
6. Woo, A., Culler, D.: A Transmission Control Scheme for Media Access in Sensor Networks. In: the proceedings of ACM MobiCom, pp. 221–235. ACM Press, New York (2001)
7. Xie, P., Cui, J.H.: SDRT: A Reliable Data Transport Protocol for Underwater Sensor Networks. UCONN CSE Technical Report: UbiNet-TR06-03 (2006)
8. Park, S., Vedantham, R., Sivakumar, R., Akyildiz, I.: A Scalable Approach for Reliable Downstream Data Delivery in Wireless Sensor Networks. In: the proceedings of ACM MobiHoc, pp. 78–89. ACM Press, New York (2004)
9. Chen, S., Yang, N.: Congestion Avoidance Based on Lightweight Buffer Management in Sensor Networks. IEEE Transaction on Parallel and Distributed Systems 17(9), 934–946 (2006)
10. Zhang, H., Arora, A., Choi, Y., Gouda, M.G.: Reliable Bursty Convergecast in wireless Sensor Networks. In: the proceedings of ACM MobiHoc, pp. 266–276. ACM Press, New York (2005)
11. Wan, C.Y., Campbell, A.T., Krishnamurthy, L.: Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks. IEEE Journal of Selected Areas in Communication 23(4), 862–872 (2005)
12. Stann, R., Heidemann, J.: RMST Reliable data transport in sensor networks. In: the proceedings of IEEE SPNA, Anchorage, Alaska, pp. 102–112. IEEE Computer Society Press, Los Alamitos (2003)
13. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: the proceedings of ACM MobiCom 2000, Boston, MA, pp. 56–67. ACM Press, New York (2000)
14. Kang, J., Zhang, Y., Nath, B.: TARA Topology Aware Resource Adaptation to Alleviate Congestion in Sensor Networks. IEEE Transaction on Parallel and Distributed Systems 18(7), 919–931 (2007)
15. The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/index.htm