International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 6631-6649

http://www.ripublication.com

Classifier Level Congestion Control based on Priority in Wireless Sensor Networks

S. Jeya Shobana[+]

Research Scholar, Manonmanium Sundaranar University

Tirunelveli 627 012, India.

jshobana.prakash@gmail.com

B. Paramasivan

Professor and Head, National Engineering College

Kovilpatti 628 503, India.

bparamasivan@yahoo.co.in

Corresponding Author: S. Jeya Shobana[+]

Abstract

The improvements of Wireless Sensor Networks (WSNs) have opened up a wide gateway in the research of congestion control that concentrates more in low packet loss and increased throughput. A well planned infra-structure, secure data collection and real time responses are critical issues to be considered, as congestion results in discontinuous data flow resulting in reduced performance, packet loss & capitalizes on energy consumption. It is therefore essential to develop a congestion control framework based on priority such that it results in an increased throughput, avoids packet loss and possibly giving uniform data collection. To accomplish this goal, a Classifier Level Congestion Control Protocol (CLCCP) is proposed so that the classifier categorizes the packets based on priority and schedules them using a weighted fair queuing (WFQ) algorithm. Thus the packets with maximum buffer occupancy is first sent to the sink thereby delaying the medium and low priority packets to wait, resulting in low packet dropping probability. This algorithm is designed by concentrating on the priority, distance of the nodes and the queue length. This helps the nodes by tending not to be a hotspot near the sink which is a serious issue in WSNs. The results of simulation shows that it has improved implications over the current trends and has low overhead with fairness for the wide area WSN.

Keywords: Congestion, Priority, Wireless sensor network, Classifier, Weighted Fair Queuing.

1. INTRODUCTION

Congestion is a critical problem in Wireless Sensor Networks (WSN) which has an impact of deprived performance like very low throughput, loss of packets & increased energy consumption which is threatening in fields like military, surveillance etc. In traditional networks, data do not travel towards a common point and seem to be very irregular. But when it is compared to WSN, the nodes move toward a common sink and that's why WSN is different from the other networks. We've to consider a lot of parameters, especially with the behavior of nodes as it may be either random or dense. Most of the works that were carried out earlier were mainly engrossed only on the traffic control because it would fortunately lessen the congestion towards the sink and around it. End to end and hop by hop mechanisms helped in reducing congestion which was not suitable for two reasons and they are: First is minimizing or restricting source traffic which was not a good choice in terms of emergency (example include monitoring).Second, traffic bursts during congestion and relying only of the feedback of it may not give the necessary time for calculating the impact of traffic and resolve it or try a new elucidation as the impact would be abnormal.

Though there are many congestion control mechanisms like end to end and hop by hop traffic control, routing etc., the selection of a good scheme that matches the characteristic of the problem always determines a good solution. In this paper, we propose a Classifier Level Congestion Control based on Priority (CLCCP) to alleviate congestion by classifying the nodes based on the buffer occupancy into High Priority node (HPN), Medium Priority Node (MPN) and Low Priority node (LPN).

The main idea is to create two independent crossbreed concepts namely length of the queue (LOQ) and distance from the node to its destination (DND). LOQ maintains a constant threshold value which should not grow larger than the desired one as it assures congestion.DND calculates the distance from the node to its destination which results in finding the shortest path for the packets to be sent. HPN, MPN, LPN are categorized based on the analysis of the combined values of DND and LOQ respectively which are maintained as weightage in the packet header. The high priority packets are clearly navigated through the sink as the buffer which is almost full or full should not create congestion. The medium priority packets are strained and made wait until the HPN are fully forwarded.Finally, the low priority packets are sent to the sink after the HPN and MPN are fully sent. When there are more than one node in a priority category, it is sorted based on FIFO order and the decision is made using the Weighted Fair Queue (WFQ) scheduling algorithm.

WFQ schedules all the packets in a fair manner resulting in a fair data share that is easily manageable. Thus, even the congested packets have no or very less packet dropping probability in critical situations and in changing traffic.

The rest of this paper is organized as follows: in Section 2, we bring out the related works regarding congestion control in WSNs and why we are motivated to design this Classifier Level Congestion Control based on Priority. Section 3 describes the ideas of our solution and the three modules. Section 4 illustrates the evaluation of performance in a network that is arbitrarily deployed over the network and compares with the other mechanisms. We conclude this paper with Section 5.

2. RELATED WORK

A very important and critical issue in WSN is congestion control. Basically there are two kinds of congestion. The first one is link level congestion that comes up due to sharing of countless nodes in media access control (MAC) which results in collision and captures the channels at the same time. Next is the node level congestion that is usually due to overflow of buffer in the node resulting in loss ofpacket. The work done by Ee and Bajcsy et al.[9] predicts a tree like structure from all source nodes to the sink. Each sensor packets are transmitted and received from the upstream neighbors and computes a fair rate such that data is not sent beyond that particular rate.

The congestion control mechanism given by CODA [10] outcomes with an idea that congestion is detected by having an eye on the queue length of the packets at the intermediate nodes. A backpressure message upstream is broadcasted as soon as congestion is detected by a node which lightens congestion. It also involves the additive increase multiplicative decrease (AIMD) scheme is a rate based mechanism in which the sending rate is increased by a constant value by the intermediate nodes. In Adaptive rate control (ARC), the source and the intermediate nodes are changed based on the constant bit rate (CBR) used. FUSION [13] initiates congestion control by a prioritized MAC, limiting source rate and hop-by-hop flow control mechanism where the neighboring nodes stop emitting data packets when congestion is detected. Thus better fairness and higher throughput are achieved when compared to the other schemes. Event to Sink Reliable Transport (ESRT) [14] is an unswerving congestion control mechanism in which the source sending rate is periodically configured by the sink. The transmission rate allocation is computed centrally, *i.e.* the number of received sensor readings are periodically counted by the base station and reschedules the sensors by broadcasting a new transmission rate. Interference-Aware Fair Rate Control (IFRC) [15] adjusts the departing rate on each link based on AIMD scheme which employs the static queue threshold for congestion control. It uses a tree that is routed at each and every sink to route the data and the rates of flows on the interfering trees are choked. RCRT [16] is a reliable transport protocol that uses end to end explicit recovery and grants flexibility in rate allocation. Here, congestion control is concentrated towards the sink. In [17] a congestion avoidance scheme is proposed by Chen and Yang based on light weight buffer management which follows the hop by hop flow control. SPEED [18] manages congestion by rerouting the incoming traffic around the hotspot.

Wang et al. [19] proposed a Priority based Congestion Control protocol (PCCP) which uses the rate adjusting algorithm that provides equal fairness to each sensor node and adjusts the priority of each traffic source for congestion control. Reliable Multi Segment Protocol (RMST) [20] is a hop by hop transport protocol in which packet loss is recovered in a hop by hop manner and guarantees reliability. The rate of data transmission is set by the system administrator in this approach thereby mitigating congestion. Congestion Aware Routing (CAR) discovers the congested area of the network which exists between the data sink and the data source and concentrates more on the high priority traffic and forwards it. An Interference-Minimized Multipath Routing (I2MR) [21] increases throughput by identifying the paths that are disjoint for load balancing requiring minimal localization support. Biased Geographical Routing (BGR) was developed by Popa et al. where traffic is splitted on congestion detection. It needs local node information by GPS or some other source which results in worst congestion at times and is not much useful. It uses dynamic routing and sounds good as it minimizes the cost of static multipath routing.

Based on this understanding, in this paper, we have designed a Classifier Level Congestion Control protocol that steers the nodes towards the sink and alleviates congestion thereby meeting the reliability requirements.

3. THE PROPOSED CONGESTION CONTROL ALGORITHM

Some research have been carried out earlier based on the potential based routing especially which was carried out by Basu et al. [22] which was not popular because it led to severe overhead. The PCCP [23] also has some drawbacks which had a solution that is not suitable for considering the provision of priority in random service time. One another problem is that often the physical link varies and considering only the fixed service time based on the output rate of each node cannot be a feasible solution. It also doesn't care about the priority index which either increases or decreases based on the level of congestion. Also practically it is tedious to develop a field considering all the parameters that gives out a good solution. Thus we have developed theCLCCP scheme based on all these issues which fine-tunes the traffic ratio whenever congestion is sensed by any node.

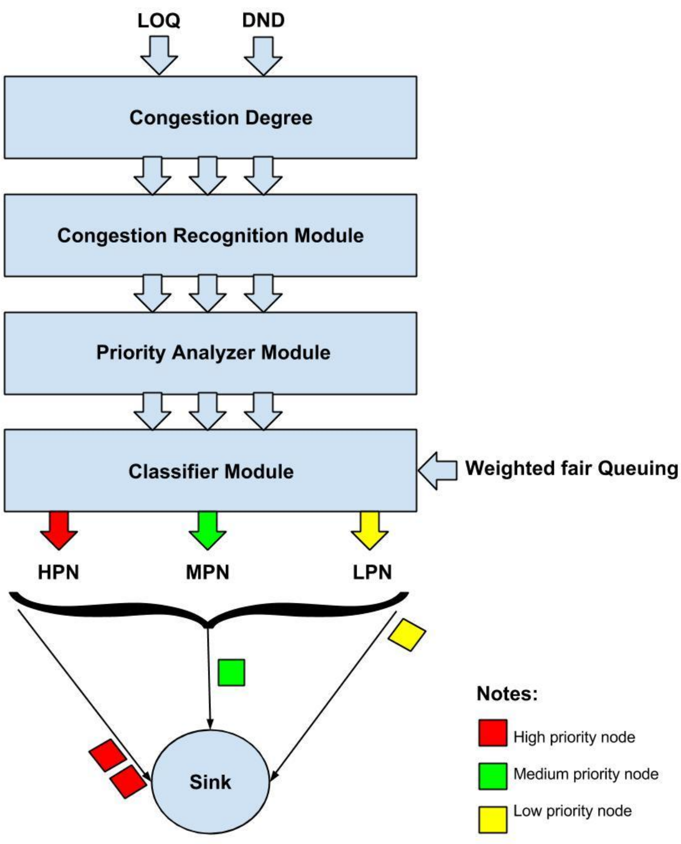

Figure 1. Structure of the proposed protocol

The model we have proposed is called Classifier Level Congestion Control Protocol (CLCCP). Figure 1 represents the structure of the proposed protocol which has three modules. They are Congestion Recognition Module (*CRM*), Priority

Analyzer Module (*PAM*) and Classifier Module (*CM*). We describe each unit in detail in the following subsections.

3.1 Congestion Recognition Module

The Congestion Recognition Module is used to recognize the impact of congestion among the nodes. Each node has its packets collected in its queue. The length of the queue (*LOQ*) and the distance from the node to its destination (*DND*) *i.e.* the sink is calculated separately. The collective result of both helps us to find out the level of congestion to obtain the congestion degree so as to recognize congestion. Since all the nodes have queues, a queue scheduler will be helpful for scheduling the queues without overloading the packets which leads to congestion or loss of packets which should be avoided. Thus it recognizes congestion in the earlier stage itself before transmitting data to the next-hop nodes. Now we have to broadcast this information to the neighboring nodes in a periodic manner. Now let us discuss about the fields LOQ and DND.

3.1.1 Length of the Queue (LOQ)

Each sensor node maintains a limited buffer which should not overflow. If not, loss of packets would result. A large queue results in a hotspot which is a critical issue to be considered. Now the normalized *LOQ*N_l at a node *x* is defined by

$$N_l(x) = \frac{n_p(Q)}{BS(x)} \qquad (1)$$

where $n_p(Q)$ denotes the total number of packets in the queue and $BS(x)$ denotes the buffer size of node *x*. *LOQ* is very important in deciding the congestion degree which can have three ranges of values: 1 and above, a value between 0.5 and 1, and a value between 0 and 0.5. If the value exceeds 1, it is termed to be critical and if it is between 0.5 and 1 it is termed to be moderate and the other case is termed to be low level.

For example, $N_l(x) = \begin{cases} \frac{07}{20} = 0.35 \to Low \\ \frac{10}{20} = 0.50 \to Medium \\ \frac{25}{20} = 1.25 \to High \end{cases}$

3.1.2 Distance from Node to Destination (DND)

Usually sensor nodes have its packet flow towards the sink, which is defined by the Distance from the Node to its Destination (*DND*). Thus if the sensor node has to transmit a packet to its next hop node, it should have the knowledge of the distances of the adjacent / neighboring nodes. It is because, only based on this the next hop node is chosen. It is formulated as follows:

$$N_d(x) = \frac{d_y - d_x}{CL_{xy}} \qquad (2)$$

whereCL_{xy} denoted the cost of link from the node *x* to the destination *y*. *DND* can have the values between 0 and 1 as if the node hops two times away, then it cannot be considered as the next hop node. Based on the values of *LOQ* and *DND*, we can have the congestion degree C_d. This value is compared by the priority analyzer module so as to categorize the nodes such as *HPN*, *MPN* and *LPN*.

$$C_d(x) = \delta N_1(x) + (1-\delta) N_d(x) \quad (3)$$

where $C_d(x)$ is the congestion degree by combining *LOQ* and *DND*. It should be notes that $0 \leq \delta \leq 1$. The variable δ is adjustable based on the degree of influence.

Equation (3) can be rewritten as,

$$\beta = \frac{C_d(x) - C_d(y)}{CL_{xy}} \quad (4)$$

Here CL_{xy} = distance $x \rightarrow y$ which defines the link cost from node x to destination y. y Є neighbor (x)
β is the combination of both *LOQ* and *DND*.

Mathematically it is understood as,

$$C_d(x) = \begin{cases} y \; \varepsilon \; neighbor(x), \; if \; 0 < distance \; (x \rightarrow y) \leq 1 \\ y \; \phi \; neighbor(x), \; if \; distance(x \rightarrow y) > 1 \\ y \; \phi \; neighbor(x), \; if \; distance(x \rightarrow y) > dp(x) \end{cases} \quad (5)$$

where $d_p(x)$ is the permissible distance and $d_p(x)$ should be between 1 and 5. The algorithm of CRM is as follows:

Algorithm for Congestion Recognition Module

A1: Calculate the normalized LOQ $N_l(x)$
A2: Obtain the congestion degree $C_d(x)$
A3: If $N_l(x) > 1$
  $C_d(x)$ = High
  Else
  If $0.5 \leq N_l(x) \leq 1$
    $C_d(x)$ = Medium
  Else
  If $N_l(x) < 0.5$
    $C_d(x)$ = Low
  End if
  End if
  End if
  A4: Calculate $N_d(x)$ and β

3.2 Priority Analyzer Module

This module analyzes the priority of the sensor nodes based on the level of congestion degree. Since each node is installed based on the severity of the environment and its application, a priority analyzer analyzes by comparing the source traffic of each node and the congestion degree that is exchanged which leads us to the second module Priority Analyzer Module.

To compute the source traffic rate, we first identify the service time $S_t(x)$ of the current packet in node x. The average service time can be calculated as:

$$\frac{1}{S_t(x)} = (1-p)\frac{1}{S_t(x)} + p * S_t(x) \qquad (6)$$

where p is a constant coefficient. $1/S_t(x)$ is the time taken from the time the packet is sent through the network layer until the packet is successfully transmitted to the MAC layer after acknowledgement. Now, the source traffic rate is calculated as follows:

$$R(x) = S_t(x)^{-1} \qquad (7)$$

where $R(x)$ is the source traffic rate of the node x.

Now we have the new traffic rate of each sensor and the congestion degree which has three categories namely High Priority Node (*HPN*), Medium Priority Node (*MPN*) and Low Priority Node (*LPN*). Here, priority is implemented based on the congestion degree. Thus each sensor node is categorized into any of the three groups. If this categorization is not made, a few nodes to a cluster of nodes will try to become a hotspot trying to be congested near the sink. This condition becomes a drawback for the sink and it has no other way than dropping the packets. But since packets are classified based on the classifier, packet dropping is very little when compared to the other algorithms.

To achieve this we are introducing the Classifier Level Congestion Control algorithm based on priority (CLCCP) which is a neighborhood based algorithm and energy efficient with very less complexity. Neighbors are discovered based on making a list of one hop neighbors. This list doesn't change until the next dynamic network change. Figure 2 represents the selection of one hop neighbor.

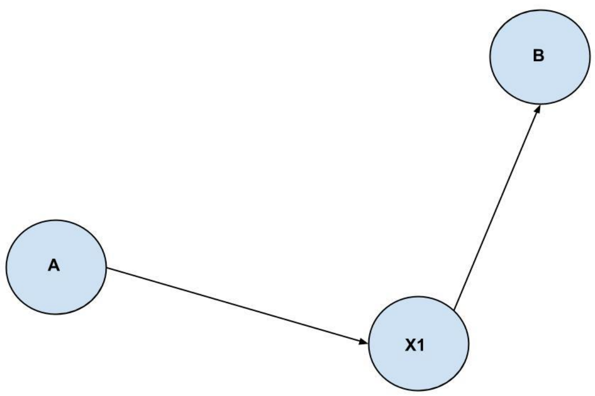

Figure 2. *x1*becomes one-hop neighbor between source *A* and Destination *B*

Though two hop nodes are also considered as next hop nodes in general, we consider a next-hop as neighboring node. Figure 2 shows how *x1* becomes a one hop neighbor between source *A* and destination *B*. When there are two nodes *x1* and *x2* becomes a one hop neighbor and *x2* becomes the two hop neighbor. A procedure is adopted to find out the next hop node based on two radio aware link metrics namely, anticipated distance (*AD*) and anticipated transmission time (*ATT*). We consider these two parameters because if the packet transmission error increases, *ATT* increases thereby slowing down the rate of transmission which leads to congestion. Also *ATT* indicates the one hop transmission time which is expressed as,

$$ATT=\frac{n_p(Q)}{TR}ATN \tag{8}$$

where*ATN* is the anticipated no of transmissions in a single hop. It is derived using

$$ATN = \frac{1}{(1-CL_{xy})} \tag{9}$$

The next link metric ATD is defined as,

$$ATD = \frac{D}{ATT^{\lambda}} \tag{10}$$

where D is the distance and μ is the weight which is used to balance the bond between *D* and *ATT*. As λ increases, *ATT* increases.

We know that the node ID and position information is exchanged periodically by a beacon message. Along with this, the packet error rate is also exchanged which is maintained in a neighbor data table (NDT) by each node. So the neighboring information is known to all nodes using the table. This makes easier to find out the closest node which is one hop away. The table is represented in table 1.

Table I: An example of the neighbor data table

Node ID	D (*m*)	ATT (*ms*)	AD (*m/ms*)
1	80	8.4	9.52
2	117	10.5	11.14
3	142	13.6	10.95

Using NDT, selecting the next hop neighbor is made simple by choosing one which has the maximum AD. The next module is the Classifier module which serves as a gateway for all the nodes trying to reach the sink. Our main motive is to dynamically update the priority level, which should be carefully done as the node should be aware of when to update and check whether it is periodically updated. Here updating refers to updating LOQ and DND. Both are designed to be 8 bytes each. We don't send the CD because floating point number occupies an immense space because the buffer size is limited. Thus, sending two integer numbers saves space. Next, there should be a clear idea of when to update, because if it often updated, loss of battery results and if it is not properly updated, degradation results.

Algorithm for Priority Analyzer Module

B1: Find *R(x)* for each packet in node *x*

```
B2:   If (R(x) = Max &&Cd(x) = HPN)
  Route the nodes directly to the sink based on FIFO.
      Else
      If (R(x) = Med &&Cd(x) = MPN)
            Route the nodes after HPN are fully sent.
       Else
      If (R(x) = Min &&Cd(x) = LPN)
      Wait behind MPN.
      End if
      End if
  End if
B3:   For each child node
      Maintain a NDT
  End for
B4:   for j=1 to Tjdo
            Tentry++;
            Tupdate--;
  End for
B5:   while Tj>Tmax
      Tentry--;
      Tupdate++;
  End While
```

CLCCP updates only when there is a time burst of the maximum updating interval, variation of LOQ and variation of D. The pseudo code of CLCCP is as follows.

Pseudo code of CLCCP: When to update

```
1:   if (Time Burst of the Max Interval) then
2:   Transmit (Message Update)
3:   else
4:   Message Update Pending = TRUE;
5:   Wait for = n µsecs;
6:   end if
```

According to our algorithm, nodes get updated only based on the time interval. Until the maximum time interval expires, message updating will be pending and the pending time is for some n µsecs. This updated information has to be shared among the next hop nodes periodically.

3.3 Classifier Module

The third module is shown in the figure 3. The unordered nodes are sent via the classifier which has all the three categories of buffer occupancy namely HPN, MPN and LPN.The nodes are classified
based on a queue scheduling algorithm called the Weighted Fair Queuing (WFQ) which classifies and sorts the queues and sends to the sink in a fair order such that it results in low packet dropping probability. All the nodes can reach the sink only through the classifier and no nodes are allowed directly to the sink. The role of a classifier is to classify the nodes based on the congestion degree as shown in equation (5). As there are many queues, the WFQ algorithm allows different priorities to multiplexed data flows helping to smooth out the flow of data by sorting the packets.

It is very popular as it approximates scheduling of packets regardless of when it arrived. WFQ allows different sessions to have different service shares. Also it minimizes the average latency and prevents exaggerated discrepancies if there are m data flows with weights $\mu_1, \mu_2 \ldots \mu_m$, data flow i will achieve an average data rate of

$$\frac{R\mu_i}{(\mu_1+\mu_2+\cdots+\mu_m)} \quad (11)$$

thereby regulating the weights dynamically. Thus it is utilized for controlling the quality of service and guarantees end to end delay bound for achieving fair data rate. To achieve such fairness, weights are set to $\mu_i = C_i^{-1}$ where C_i is the cost per data bit of data flow i. WFQ is shown in the figure 4.

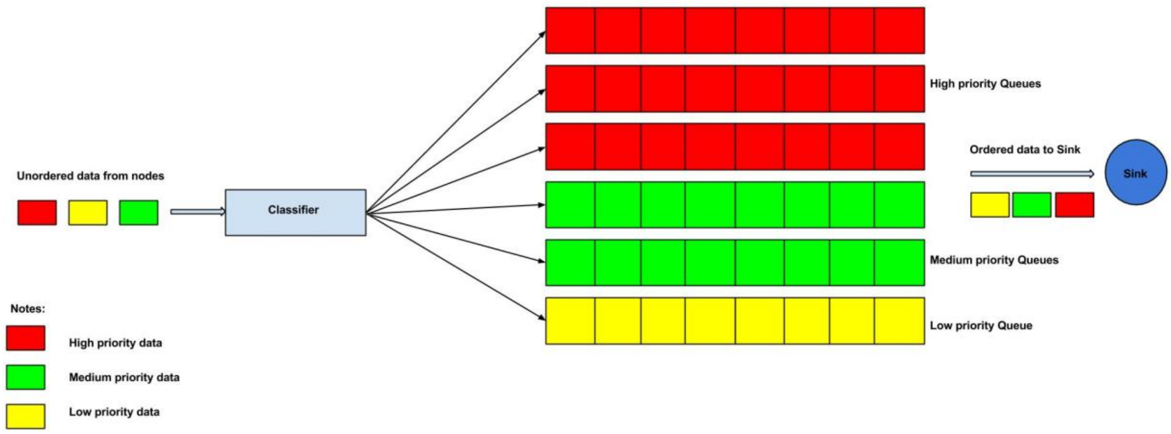

Figure 3. Classifier Module.

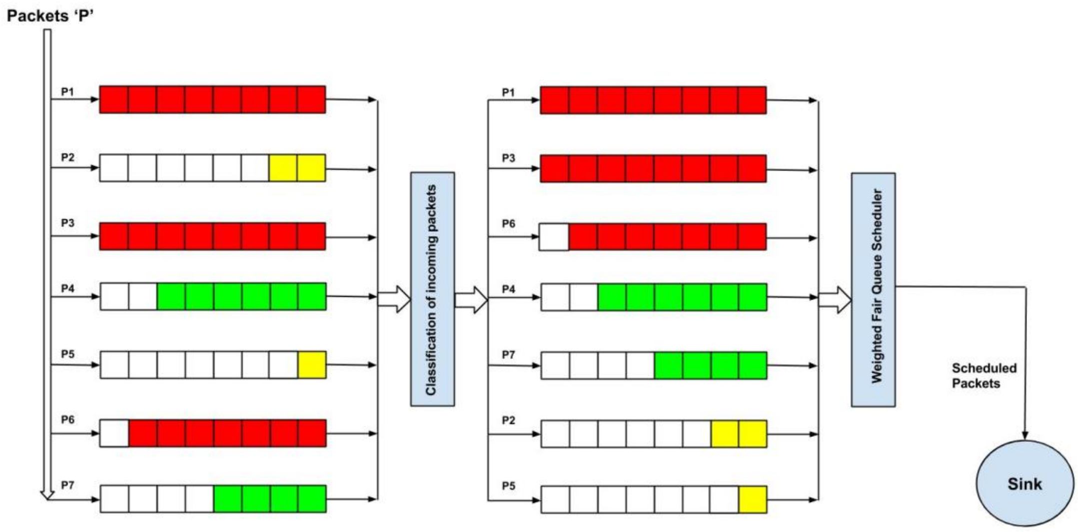

Figure 4. Weighted Fair Queuing

The incoming queues pass through the classifier which is of three categories based on buffer occupancy namely HPN, MPN and LPN. Among them the HPN are sent to the sink based on the congestion degree. If there are more than one HPN, it is forwarded based on the FIFO algorithm. Thus all the other queues are halted to send the traffic from the highest priority based on the time of arrival. Meanwhile the MPN are queued for some time until the HPN are successfully passed through the sink based on emergency. Once all the HPN are transmitted, the MPN gets priority and are sent accordingly. Some packets are delayed or dropped which is comparatively less, where fairness is ensured as it automatically smooth out the data flow. The LPN with the low congestion C_d are sent thereafter ensuring very little packet loss or delay.

4. PERFORMANCE EVALUATION

In this section, we evaluate the proposed congestion control algorithm. For this, we use the network simulator NS2 version 2.29. The simulation parameters are described in Table II.

Table IISimulation parameters

Parameter	Value
Number of sink nodes	1
Number of source nodes	60
MAC Protocol	802.11
Routing Protocol	DSR
Simulation Area	1000 * 1000 m^2
Average packets per node	30
Packet Size	512 bytes
Radio Range	100m
Life time of NDT	5 seconds
Beacon interval	1 second
Simulation Time	300 seconds

4.1 Performance Metrics

We have some quantitative metrics for performance evaluation and they are as follows.

1. **Throughput** is defined as

$$\text{Throughput} = \frac{Packets\ obtained\ by\ the\ sink}{Packets\ sent\ by\ the\ source\ node}$$

Throughput is usually measured to find out the bandwidth capacity to find out whether the system is stable or manageable or more than the expected state. It is affected by a huge number of factors and they include high signal to noise ratio, low bandwidth, poor channel utilization, RC losses, termination of wires etc. It is measured over a short period of time in bits/second or bps.

2. **Energy Efficiency** – It is defined as the average energy consumed by each packet which should be comparatively less. The sensor nodes should be energy efficient as the life time depends heavily on the limited energy resource. So the radio power supply should be put off when it is not in use. As WSNs are deployed over remote and hostile environments, the energy resource that is scarce should be conserved less and only then it can be energy efficient.

3. **Packet loss probability** –Packets may be dropped in switches and routers when the packet queues are almost full due to congestion. When the rate of a packet that arrives exceeds the rate of the packet service time, then packet loss probability will be high and if it is high, data has to be retransmitted again. Packet loss occurs due to signal degradation, network dissuasion etc. which leads to highly noticeable performance issues and overhead. As packet loss increases as traffic increases, the performance of each sensor node is measured not only in terms of delay but in terms of packet loss probability also and that's why it is termed to be an important performance metric.

4. **Fairness** – It is the fair share of bandwidth that is shared among all the nodes. Fairness defines the fair channel allocation given to all the nodes leading to the average rate of a successful data transfer through a communication path. The algorithm should result in maximum fairness when compared to the other algorithms. The difference between throughput and fairness is that the average throughput is less than or equal to the channel capacity in addition to the implementation overhead.

4.2 Simulation Setup

For simulation, we have a single sink with 60 source nodes in an area of 1000 * 1000 m^2. For finding out the next hop node we are maintaining a neighborhood data table whose life time is 5 *seconds*. We have used the Mac protocol 802.11 and the radio range is 100 *m*. the packet size is 512 *bytes* with a beacon interval of 1*second* and the simulation was conducted for 300 *seconds*. As the energy supply is limited, sensor nodes should avoid sending data continuously as it lessens the life time. The other parameters are assumed to have their default values and the simulation details are summarized in Table II as simulation parameters.

4.3 Comparative Analysis

We compare the performance of our proposed protocol CLCCP with the other existing solutions PACC, No Congestion Control and Backpressure (50 %).

A. Throughput Comparison

The throughput of No Congestion Control, Backpressure (50%), PACC and CLCCP are compared with respect to traffic and we have simulated for 300 *seconds*. Let us start with No congestion control which suffers more than the others as no rule is adopted to achieve the expected throughput. It has an uncontrolled flow of packets and because of this, the number of packets a node receives is lesser than it transmits thereby resulting in decreased throughput. It peaks at 200 *Kbps* and the throughput level increases initially, which is easily mitigated to congestion,but falls down after 800 *Kbps* reaching only 170 *Kbpsi.e.* far below the acceptable level. Next, the

Backpressure (50%) has a good throughput levelup to 750 *Kbps* and falls down as the time goes on which means that, it is not able to sustain well under pressure reaching a minimum level of throughput. It is true that when the traffic load grows high, the probability of congestion is higher which results in an unpredictable level of throughput. But our simulation has the highest throughput rate and the comparative results are shown in figure 5.

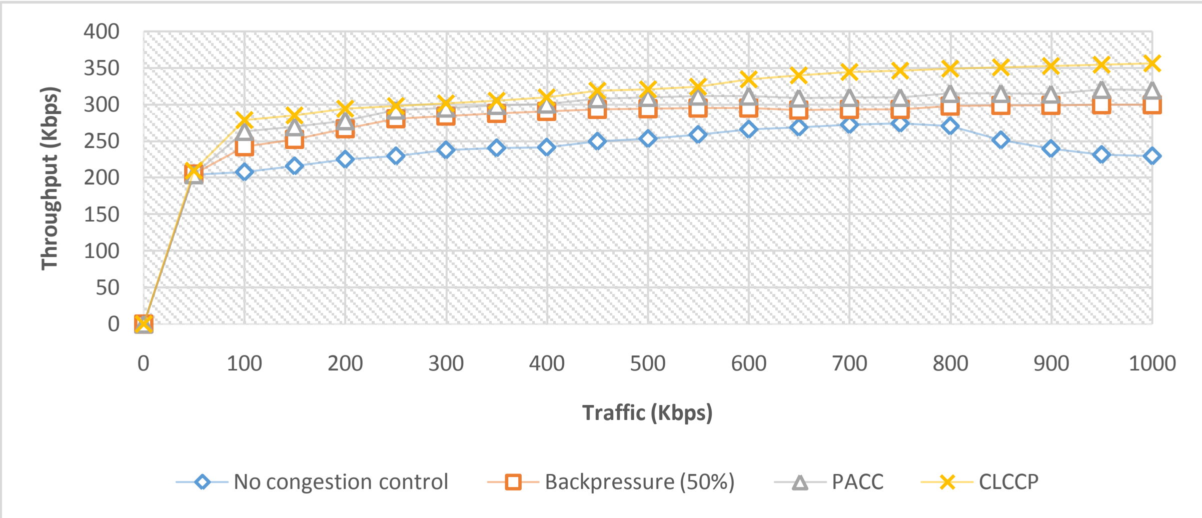

Figure 5. Throughput with respect to traffic.

B. *Energy Expenditure*

The total energy spent by the sensor nodes are calculated based on the cumulative number of transmissions made in the network divided by the number of transmissions successfully delivered to the sink. A single transmission moves a packet one hop adjacent to the sink. The energy consumption rate is analyzed with respect to traffic and time. Figure 6 shows the energy expenditure / packet with respect to several source rates. As the bandwidth utilization is very good in our scheme, it is more energy efficient. Figure 7 shows the energy expenditure / packet with respect to time and we have simulated for 500 seconds. On comparing with No congestion control, it outcomes in a poor way with a lot of energy spent meaning that very little and truncated packets had reached the sink. PACC scores better than No congestion control as well as PACC. Yet, CLCCP has very less energy consumption making sure that the number of transmissions made to the sink is also higher.

Figure 6. Energy Expenditure / packet with respect to traffic.

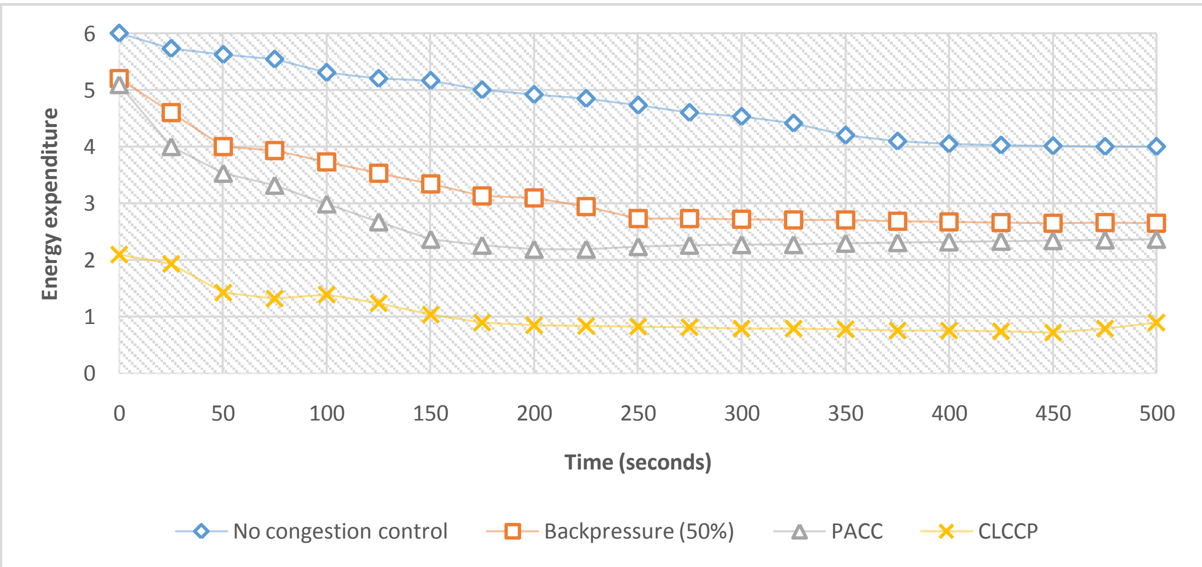

Figure 7. Energy Expenditure/ packet with respect to time.

C. Packet Loss Probability

When packets are dropped, the possible reasons are collisions, buffer overflows, network traffic or congestion. Figure8 shows the packet loss probability with respect to time under the traffic of 1000 *Kbps* per low. With No Congestion Control the packet dropping rate is very high and grows exponentially. Backpressure (50 %) brings up backpressure algorithms with 50 % reduction percentage in a sensor's data rate with respect to the backpressure messages. It is noted that it results in a significant number of packets dropped. On the contrary, our congestion control scheme tries to assign a fair rate of bandwidth for each flow and lessens the packet drops. Figure 9 shows the packet loss probability over traffic *i.e.* source rate. It compares the number of packets dropped over the initial rate at which the source nodes generate data. It is noted that when the source rate is higher. The packet dropping probability is higher. But our congestion control scheme is less sensitive to the initial source rate and has less probability in losing packets.

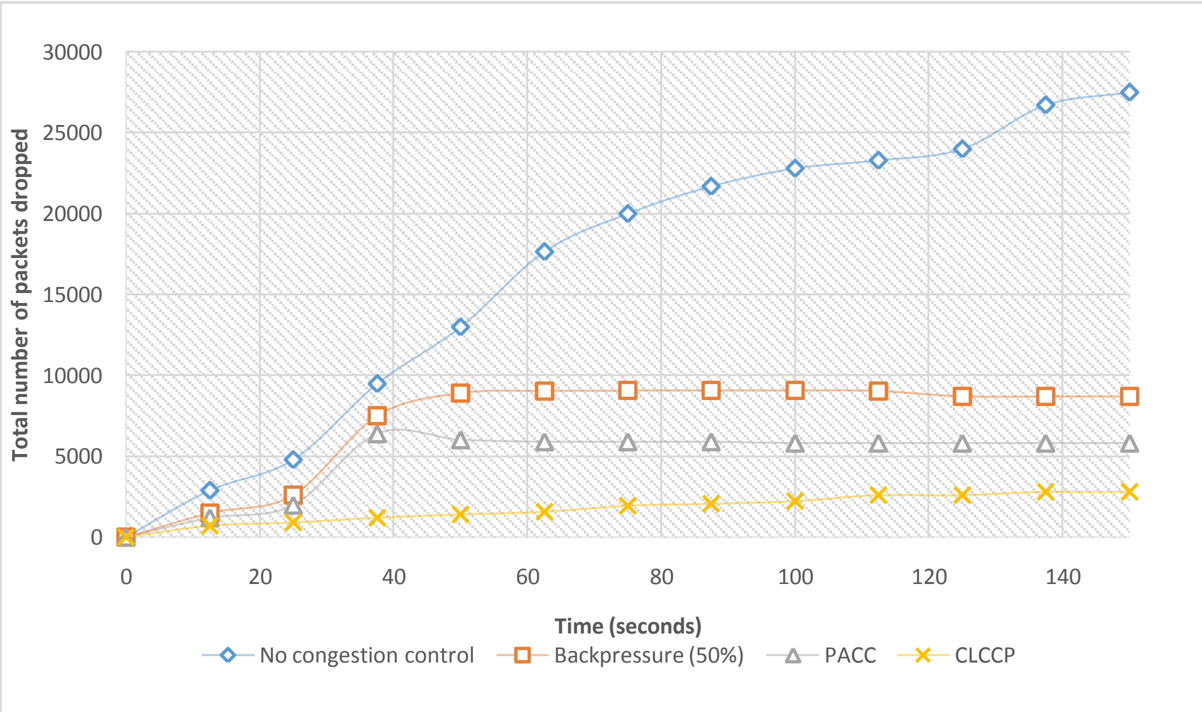

Figure 8. Packet loss probability over time.

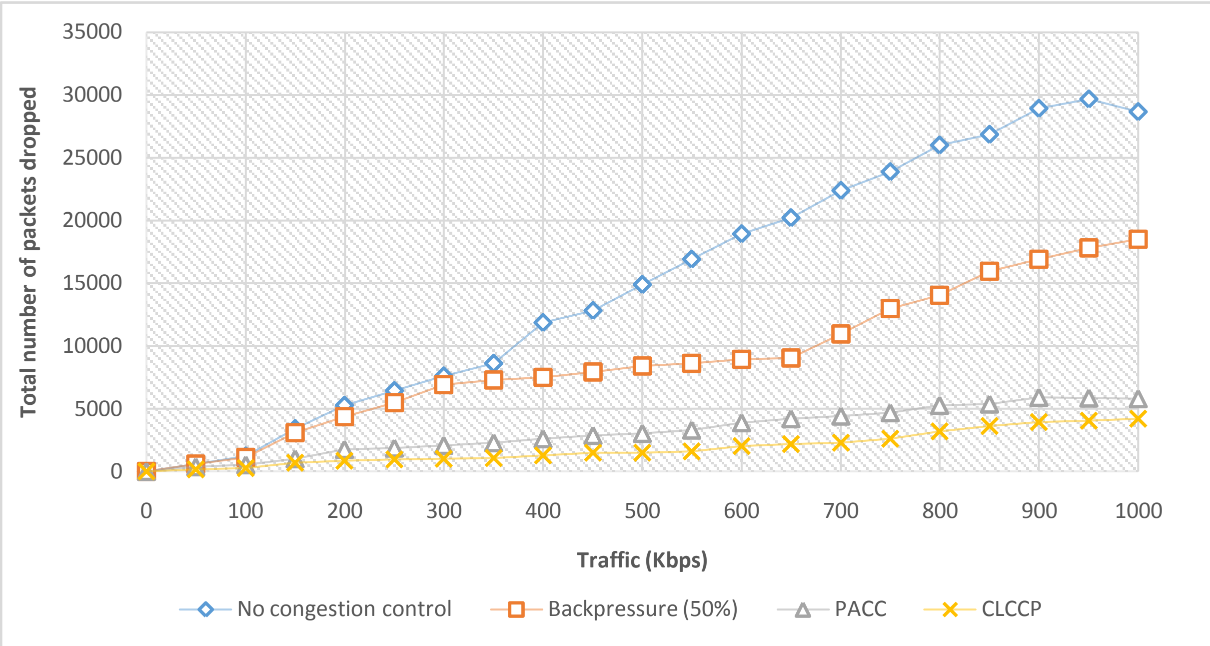

Figure 9. Packet loss probability overtraffic.

D. Fairness Comparison

Fairness ensures a fair bandwidth share among all the sensor nodes for providing smooth transmissions with no congestion or negligible congestion with each flow at 1000 *Kbps*. Figure 10 shows the comparison of fairness among the other algorithms. Contrarily, no congestion control guarantees failure in achieving fairness. PACC whose ultimate attention is on priority acquires a fair share of bandwidth, but fails to completely achieve an acceptable fairness. Moreover, only if the bandwidth is fair and the channels are not busy, sensor nodes can successfully transmit packets, as priority changes from time to time. Comparing all the data flows different nodes suffer from different interference. Flow 1 is different from flow 11 and No congestion control services well only for shorter flows. Flow 10 achieves maximum throughput compared to all the flows ensuring that longer flows do not starve behind shorter flows. Thus, CLCCP achieves better fairness assigning fair data share by calculating the available bandwidth more precisely and achieves maximum throughput.

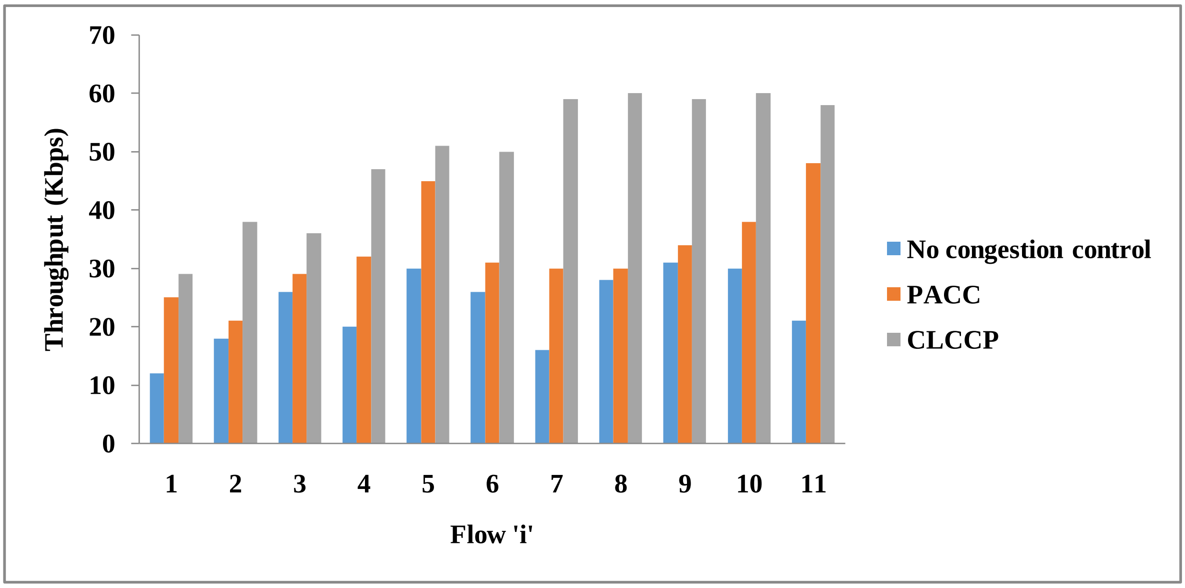

Figure 10. Fair data share with respect to different data flows.

5. CONCLUSION

In this paper, we have developed a congestion control protocol named CLCCP. It uses a classifier level congestion control based on weighted fair queuing and can manage the traffic rate as nodes are sent to the sink only after the scheduler schedules or decides which queue to send to the sink based on the congestion degree for each sensor node. This traffic control can alleviate congestion which is really hard to satisfy the fidelity by other algorithms. Some WSNs are application oriented and different applications have different desires. We believe that a common framework is necessary to account all other factors which can be extended as our future work. The performance of our proposed protocol was evaluated based on computer simulations. We have shown that CLCCP has better performance in terms of throughput, energy consumption, packet loss probability and fairness. The results ensure that CLCCP can achieve very less probability of packet loss.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol.40, no. 8, pp. 102-104, Aug. 2002.

[2] M. Zawodniok and S. Jagannathan, " Predictive Congestion Control Protocol for Wireless Sensor Networks," *IEEE Trans. Wireless Comm.*, vol. 6, no. 11, pp. 3955-3963,Nov. 2007.

[3] T. He, F. Ren, C. Lin, and S. Das, "Alleviating Congestion Using Traffic-Aware Dynamic Routing in Wireless Sensor Networks," *Proc. Of the 5th*

Annual IEEE Communication Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'08), pp.233-241, June 2008.

[4] S.Vural and E.Ekici, "Analysis of hop-distance relationship in spatially ransom sensor networks," in *Proc. ACM MobiHoc '05,* May 2005, pp. 320-331.

[5] C.G. Wang. B Li, K.Sohraby, et.al. Upstream congestion control in wireless sensor networks through cross-layer optimization, *IEEE Journal on Selected Areas in Communications* 25 (4) (2007), pp. 786-795.

[6] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *ACM Mobile Computing Commun. Review,* vol. 1, no. 2, pp. 10-24, 2002.

[7] I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks J.,* vol. 38, pp. 393-422, 2002.

[8] A.Basu, A. Lin, and S. Ramanathan, "Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm," *Proc: ACM SIGCOMM,* pp.37-48, 2003.

[9] C.T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," in *Proc. ACM Sensys,* Nov. 2004.

[10] C. Y. Wan, S. B. Eisenman, and A. Campbell, "CODA: Congestion detection and avoidance in sensor networks," in *Proc. 1st ACM Conf. Embedded Netw. Sensys,* Los Angeles, CA, Nov. 2003, pp. 266-279.

[11] A. Woo and D. C. Culler, "A transmission control scheme for media access in sensor network," in *Proc. ACM Mobicom,* July 2004.

[12] M.H.Yaghmaee and D.A. Adjeroh, "Priority–Based Rate Control for Service Differentiation and Congestion Control in Wireless Multimedia Sensor Networks," *Computer Networks,* vol.53,no. 11, pp. 1798-1811, 2009.

[13] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," *in Proc. 2nd ACM Conf. Embedded Netw.Sensys, Baltimore, MD,* Nov. 2004, pp. 134–147.

[14] Y. Sankarasubramaniam, O. Akan, and I. F. Akyildiz, "ESRT: Event-to-sink reliable transport in wireless sensor networks, *in Proc. 4th ACM Int. Symp. Mobile ad hoc Netw. Comput. MobiHoc, Annapolis, MD,* Jun. 2003, pp. 177–188.

[15] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware fair rate control in wireless sensor networks," *in Proc. SIGCOMM, Pisa, Italy,* Sep. 2006, pp. 63–74.

[16] J. Paek and R. Govindan, "RCRT: Rate-controlled reliable transport for wireless sensor networks," *in Proc. 5th ACM Conf. Embedded Netw. Sensys, Sydney, Australia,* Nov. 2007, pp. 305–319.

[17] S. Chen and N. Yang, "Congestion avoidance based on lightweight buffer management in sensor networks," *IEEE Trans. Parallel Distrib. Syst.,* vol. 17, no. 9, pp. 934–946, Sep. 2006.

[18] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "Speed: A stateless protocol for real-time communication in sensor Networks," *in Proc. IEEE Int. Conf. Distrib. Comput. Syst., Providence, RI,* May 2003, pp. 46–55.

[19] C. Wang, K. Sohraby, V. Lawwrence, L. Bo, and Y. Hu., “Priority-based congestion control in wireless sensor networks”.

[20] F. Stann and J. Herdemann, “RMST: Reliable data transport in sensor networks,*” in Pro. 1st IEEE Workshop SNPA, Anchorage, AK,* Nov. 2003, pp. 102–112.

[21] J.Teo, Y. Ha, and C. Tham, “Interference-Minimized Multipath Routing with Congestion Control in Wireless Sensor Network for High – Rate Streaming,”*IEEE Trans. Mobile Computing,* vol. 7, no.9, pp. 1124-1137, Sept 2008.

[22] A. Basu, A.Lin, and S. Ramanathan, “Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm,” *Proc. ACM SIGCOMM,* pp. 37-48, 2003.

[23] Wang, Chonggang, et al. "Priority-based congestion control in wireless sensor networks." *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on*. Vol. 1. IEEE, 2006.

ABOUT AUTHORS

S. Jeya Shobana has accomplished her B.E. and M.E. in Computer Science and Engineering. She is a research scholar in Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli, India. Her areas of interest are Wireless Sensor Networks, Routing, Congestion Control and Mobile ad-hoc Networks.

Dr. B. Paramasivan has accomplished his B.E. and M.E. in Computer Science and Engineering. He accomplished his Ph.D degree in Information and Communication Engineering from Anna University. He is working as a Professor and Head in department of Computer Science and Engineering, National Engineering College, Kovilpatti, India. His areas of interest are Wireless Sensor Networks, Routing, Congestion Control and Mobile Ad-hoc Networks.

“The authors declare that there is no conflict of interests regarding the publication of this article.”