CRAP: Cluster based Congestion Control with Rate Adjustment based on Priority in Wireless Sensor Networks

S. Jeya Shobana[1,*] and B. Paramasivan[2]

[1,*]Assistant Professor, Department of Computer Science and Engineering,
Rajas International Institute of Technology for Women, Nagercoil, India
[2]Professor and Head, Department of Computer Science and Engineering,
National Engineering College, Kovilpatti, India
jshobana.prakash@gmail.com[1], bparamasivan@yahoo.co.in

Abstract

Congestion in Wireless Sensor Networks (WSN) get worse when there are multiple and random flows of data in which some have superior significance over the others requiring fidelity in terms of packet delivery, QoS, energy efficiency and throughput. In node-level, congestion leads to impairment of packets that obviously reduces the QoS. In this paper, we present a Cluster based congestion control with Rate Adjustment based on Priority (CRAP) protocol, which self-organizes the sensor nodes into clusters. These clusters monitor congestion in a proactive manner within its confined range which exchanges information among them and adjusts the traffic rate when one cluster has high priority flow over the other. This rate adjustment is based on the exchange of traffic rate estimate among the clusters that reduces packet re-transmissions and energy loss. Our simulation expedites system wide rate control resulting in good throughput, very low packet loss probability and delay that deals with multiple, random flows of data.

Keywords: Cluster, Congestion, QoS, Wireless Sensor Networks

1. Introduction

Wireless Sensor Networks (WSNs) has small sensing devices which are spatially distributed to monitor the environmental conditions like temperature, pressure, sound etc. Sensor nodes range from a few hundreds to thousands based on deployment. It has a radio transceiver with an antenna, battery and a processor [1]. The size of each node varies from the size of a grain to the size of a shoe box which differs in terms of cost too. Routing of sensors is very important and it can be in the form of a simple star network to a multi hop mesh network. Improper routing leads to congestion which results in loss of packets and poor packet delivery probability [3]. One of the important constraints is to limit the use of battery as the power of the battery is fixed. Once the energy is exhausted, it becomes a critical task to replace it in a widely deployed network. Thus it has to be turned off when they are not used. WSN has a wide range of potential applications like healthcare, military surveillance, security, industry etc.

Sensors are deployed to collect and report periodic data in a timely manner which should be able to balance situations even when traffic load is high. But congestion grows gradually when individual nodes forward data towards the sink. Thus we move on to the clustering technique which is based on the grouping of nodes that has a cluster head and the vicinity of the sensor nodes connected to the cluster head. The main objective of clustering is to limit energy consumption within a cluster thereby decreasing the number of messages passed from the nodes to the sink as every communication goes only through the cluster head to the sink. Thus it reduces the number of beacon and control messages from the nodes to the sink.

In this paper, we present a protocol that deals with multiple and random flows of data which is of high and low level importance. We address the problem in real world applications like military surveillance where some flow of packets maybe important than the others [4] which has to be forwarded in a timely manner with low battery usage. Here the sensor nodes are self-organized into clusters and each cluster proactively monitors congestion within its confined range. When one cluster has priority over the other, traffic rate is adjusted using a rate adjusting method thereby reducing the chance of hotspot formation or congestion near the sink.

The rest of this paper is organized as follows: Section 2 presents the related works in congestion control and clustering in wireless sensor networks. In Section 3 we describe the design objectives of our protocol and the proposed protocol named CRAP (Cluster based congestion control based on Priority) is discussed in Section 4. In Section 5 we evaluate the performance of the proposed protocol. Finally the paper is concluded in Section 6.

2. Related Work

Many works have been conducted on congestion control in WSNs. Traditional congestion control approaches use end to end or hop by hop mechanism considering only a single class of packets. *Fen et al*. [5] proposed a MMLC routing protocol based on LEACH and uses the combination of dynamic cluster ideas and MDG nodes that reduces the communication volume around the cluster head nodes. Thus it reduces the transmission delay and extends the network life cycle. HNAMe [6] is an energy efficient hidden node avoidance mechanism for WSN which divides clusters of nodes into separate groups of non-hidden nodes through a cluster grouping strategy and assures no intervention between overlapping clusters. *Chatterjee et al* [7] proposed a cluster-based single hop tree topology that transmits the received data to the corresponding parent nodes hop-by-hop until they reach the sink and avoids congestion for both continuous and event based monitoring.

Congestion Control and Fairness (CCF) [8] is a distributed algorithm which ensures fair delivery of packets within a sensor network and eliminates congestion. In this algorithm, the average rate of each node is calculated and that rate is divided among the child nodes to adjust the rate when queues are about to overflow. Thus congestion information is implicitly reported and the rate adjustment is exactly based on the available service rate. Priority based Congestion Control Protocol (PCCP) was proposed in [9], which is an upstream congestion control protocol for WSNs. It measures congestion degree and utilizes a cross-layer optimization to manage congestion, achieving flexible weighted fairness for both single path and multipath routing. Event-to-sink reliable transport in wireless sensor networks (ESRT) was proposed in [10] that use end-to-end techniques in which the sending rate is regulated by the sink. The updated messages sent by the sink may be regulated at the source if there is higher traffic.

LEACH (Low-Energy Adaptive Clustering Hierarchy) [11] is a self-organizing clustering protocol that uses randomization for allotting energy fairly among the sensor nodes. In this scheme, the nodes organize themselves into a cluster that can behave as a cluster head also. Since the cluster head position is swiveled to all the nodes, energy consumption is equal. This approach is very useful when some nodes are far away from the base station and when the sensor node deployment area is large. *Soroet et. al*. [12] proposed an unequal clustering model for organizing the network, which leads to uniform energy dissipation among all the cluster heads and increases the lifetime of the network. *Chan et al*. [13] analyses the problem of prolonging the lifetime of the network by regulating the best cluster size in a network. Thus it maximizes the life time and minimizes energy consumption. RCRT (Rate Controlled Reliable Transport in Wireless Sensor Networks) [14] is a centralized transport protocol that has rate allocation and rate

adaptation around the sink. Though its performance is good, it cannot differentiate flows in critical regions.

Interference-aware Fair Rate Control in Wireless Sensor Networks (IFRC) [15] uses multiple buffer thresholds for each node. When the buffer size of a node is about to exceed a predefined threshold level, it requests its neighbour to decrease the sending rate thereby ensuring fairness. *Wan et al.* [16] uses a reliable transport protocol namely Pump Slowly Fetch Quickly (PSFQ) which supports a scalable transport mechanism for meeting the needs of different data applications and provides reliability.

3. Design Objectives

In this approach, we have an overview of the objectives of the protocol design and as an example, we assume the scenario of the military surveillance system.

3.1. Multiple Class of Data Flows

Scenarios like the one we have discussed earlier is called event based environments in which one flow of data may be important than the other. It is only partially based on buffer occupancy or congestion degree because some packets may have an emergency to be forwarded immediately to the sink. Thus multiple classes of data flows that have different range of priorities arise and managing them becomes much critical in event based systems. The reason is that there will be no time for calculating the shortest paths in a per node basis that will obviously lead to congestion. The situation gets much worse when a particular area has more number of data packets to be sent which may either lead to very high packet loss probability or very low data delivery ratio. An important thing to be noted is that congested areas cannot be bypassed and concentrated on other areas in forwarding packets because, sensor nodes will be scattered throughout the network. This situation has to be avoided particularly in event based systems in which some have less importance and some others may have more importance and that is why we are motivated to the concept of clusters. In such environments our ultimate aim is to give preference to the flows having much traffic rate when compared to the flows of low traffic rate.

3.2. Requisite of Rate Control

A Cluster is a group of nodes and each cluster has a Cluster Head (CH) which has an entire control over all the nodes within it and monitors them periodically. Any data that has to be forwarded from each node can pass only through the CH. The traffic rate of one cluster will be quite different from the other and this requires a rate control between them. When a cluster has high priority data flow, it has to notify the other clusters so that they may coerce reducing their sending rate so as to avoid congestion. Thus it requires a lightweight rate control mechanism so that data is smoothly passed through the congested areas based on the level of importance of data flows. If such a mechanism is used, then packet dropping probability will be greatly reduced even in congested situations.

3.3. Dimension of Traffic in Clusters

We need a profound congestion detection mechanism to identify the level of traffic in each cluster. Usually sensors send their traffic rate in control bits while forwarding beacon messages. Though this information is simple, it has to be sent to all the nodes in the network, which proliferate unnecessary traffic and results in energy loss which is a serious constraint to be noted. But according to clustering, this information is sent only to the CH which calculates the intensity of traffic and forwards packets considering all the information of the nodes collectively. Thus traffic is captured in a more precise manner which helps in analyzing the priority level and saves the battery power which is limited for all the nodes.

4. CRAP Protocol Design

The proposed CRAP protocol uses clustering method as a basis for congestion control and it uses a rate adjustment mechanism when multiple classes of traffic arise in the network. As mentioned earlier, a cluster is a group of nodes and each cluster has a CH which monitors and maintains the information of all the nodes within it. This helps to alleviate unnecessary messages that could possibly be forwarded in networks that work in node basis. The information that is collected by the CH has the responsibility to reduce congestion and so it is considered to be very effective.

Nodes can be classified into three in a clustered approach. They are 1) CH that schedules transmissions and allocates resources within a network. 2) Gateways that connects adjacent clusters which is a member of one cluster with a link to the member of another cluster when it is within the sector radius of two CHs. 3) Member Nodes which are neither CHs nor gateways and are considered ordinary. Figure 1 shows the clustered architecture in which the CHs, gateways and member nodes are denoted differently.
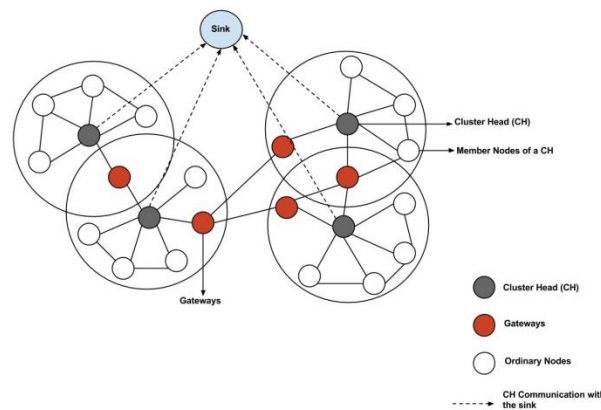

Figure 1. Clustered Architecture

4.1. Cluster Generation and CH Selection

Each sensor node deployed in the network can join with some other nodes and be self-structured into a cluster which is how clusters are generated. These generated clusters send the data, traffic rate and other information to the CH in a one hop basis. Thus a general framework has to be followed for selecting the CH as all nodes cannot become the CH. During the CH selection phase, each sensor node broadcasts a CH proclamation message pm_n along with the lingering energy le_n that it wishes to be the CH within a predefined time t_m, a timer within which the neighbouring nodes should respond.

$$CH \; selection \; message = pmn + le_n \tag{1}$$

This message is sent to all its neighbouring nodes within the transmission range t_r. The nodes that are within t_r and that responds within the expiry of t_m, will be joined to a group called as Cluster and the node that has sent the CH proclamation message will be selected as the CH. Here, lingering energy means the energy that is left over for that particular node to survive in the network. It is formulated as a metric called CHM (Cluster Head Metric) which is as follows:

$$CHM_a = \frac{le_{na}}{\sum Nei[en_{a \to b} * \beta_{a \to b}]} - \frac{le_{na}}{\sum Nei[\delta(a \to b)^2 * Pow * \beta_{a \to b}]} \tag{2}$$

The following table lists the values used in Equation (2).

Table 1. Values of Equation 2

Values	Elucidation
CHM_a	Cluster Head Metric of Node a
le_{na}	Lingering energy of node a
Nei	Neighbour
$en_{a \rightarrow b}$	Energy that is required for sending a single bit of data from node a to node b
$\beta_{a \rightarrow b}$	Total number of bits sent from node a to node b
$\delta(a \rightarrow b)$	Distance from node a to node b
Pow	Power transfer for a single bit of data

If none of the nodes respond to the message and t_{ch} expires, it selects itself to be the CH based on a probability ρ_n where ρ is the probability and n is the number of times it tries to become the CH. ρ_n is deducted by 1 on each attempt to become the CH based on the probability $(1-\rho_n)$ and it becomes ρ_{n+1} on its next try based on the following formula:

$$P_{n+1} = P_n + (1-P_n)^{\theta n} \qquad (3)$$

where θ is a constant that increases as many times as the probability P_n rises with respect to n. Thus it is vibrant that any one of the following two will occur. *i.e.*, a sensor node either becomes a CH or joins a member of the CH.

4.2. Computational Example

Let us consider Figure 2 representing CH selection from which the CHM is computed and see how the CH is selected based on it. Each node is set to 1 for β and *Pow* and the other values are taken as per the figure for computation.
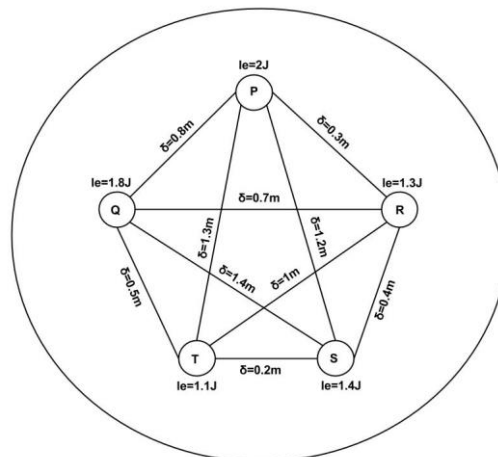

Figure 2. CH Selection

$$Node_P = \left[\frac{2J}{0.8^2 + 1.3^2 + 1.2^2 + 0.3^2} \right] = \frac{2}{3.86} = 0.518$$

$$Node_Q = \left[\frac{1.8J}{0.8^2 + 0.5^2 + 0.7^2 + 1.4^2} \right] = \frac{1.8}{3.34} = 0.530$$

$$Node_R = \left[\frac{1.3J}{0.3^2 + 0.7^2 + 1.0^2 + 0.4^2} \right] = \frac{1.3}{1.74} = 0.747$$

$$Node_T = \left[\frac{1.1J}{0.5^2 + 1.3^2 + 1.0^2 + 0.2^2} \right] = \frac{1.1}{2.98} = 0.369$$

Thus from the nodes P to T, it is found that node R has the highest CHM and so, it is selected as the CH which assures an unvarying energy depletion among all the neighbouring sensor nodes.

Pseudo code of cluster generation and CH selection

1: Status = Cluster Generation
2: Transmission of CHSM to all neighbouring nodes
3: if (CHSM > t_r)
    Response from node = Inactive && Do not join the cluster
  Else
    If (CHSM < t_r)
    Response from node = Active && Joins the cluster as a member
4: Repeat Step 3 until there are no neighbouring nodes
5: Status = CH Selection
6: Calculate the distance δ for all neighbouring nodes
7: Update the information in NDT
8: Compute CHM
9: if (CHM > CHM of all neighbouring nodes)
    {Node with the highest CHM is selected as CH}
10: if Node $\geq t_r$ of more than one CH
    {Node is selected as a member of the adjacent CH with a link to the member of another}

4.3. Routing Methodology

For appropriate formation of clusters, it needs the accurate neighbourhood data in a timely manner. So a good proactive part of a routing protocol should be used for helping the clusters in reducing the control messages and latency while determining the routes quickly without delay. In our approach, we are using the Zone Routing Protocol (ZRP) that helps to find loop free routes to the destination and assists in the formation of clusters. It exploits the feature of proactive and reactive way of routing. Proactive part of the protocol is constrained to a small locality of a node called routing sector that collects information about all the nodes within it. This information is sustained in a routing table so that it can find a route to any node from the table whenever it needs. If the reactive part of the protocol is used, it can route across the network in which we do not focus on.

A routing sector is a collection of nodes whose minimum distance *i.e.,* hops should not be greater than the sector radius. The maximum sector radius can be three for cluster heads. In link clustered architecture, all the members of the cluster are within one hop of the CH and two hops of another where the maximum can be three hops. Routing strategy is of two parts. They are I) Intra Sector Routing – the packet is sent within the routing sector of the source node to reach the other nodes. Each node broadcasts a hello message

known as sector notification message which expires in one hop after reaching its neighbouring nodes. II) Inter Sector Routing – it determines routes to the nodes by sending a route demand message to the other nodes and acquires a route response message back to the source node.

4.4. Traffic Rate Estimation

We have many nodes inside a cluster which is dependent on the coverage of t_r. So once these CHs are generated with member nodes inside, we need to calculate the traffic rate within a cluster, called local congestion. This is because, if we try to alleviate congestion within a CH, then it will not be propagated to the other CHs also. The traffic rate that is estimated for each node will be fed as input to the CH as it has the only control to the sink. This is done by sending control messages to the sink and our motive is to reduce the number of control messages that is usually transferred to the sink. Also, the CH will estimate the overall traffic. The rate estimation has to be done both inside and outside the clusters with different classes of traffic and this is a tough task to be accomplished in a real sensor network that is deployed over a wide area. We need a network where each node is modeled as a queue and the queues constitute together into a cluster. In our paper, we consider the BCMP network of queues.

Assume that a WSN that has m sensor nodes with each node modeled to be a queue and two types of traffic classes high and low, each having its own importance. Here, importance is not based on buffer occupancy of queue where the urgency of data to be transferred has the impact of importance. The routing of packets among the queues is assumed to be state dependent and we signify the network state to have a product form of all the queues. This model provides low overhead and makes computational aspects to be less complicated which is very much needed for a clustered environment. The traffic rate estimation has the following scenario. It is formulated as,

$$\alpha_m = \frac{\mu_m}{\lambda_m} \tag{4}$$

where α_m is the traffic estimate of the queue of sensor node m, μ_m is the arrival rate of packets based on the flow of data and λ_m is the mean service rate for the sensor node m which may depend on its state. Let the number of packets in a queue be $P(q)$ and the probability distribution of a single queue is given as,

$$P(q) = (1 - P_m)P_m^q \tag{5}$$

In Equation (5), P_m is the probability of a sensor node m and P_m^q is the probability of a sensor node m with respect to q number of packets in it. When there are X distinct queues, the n-ary product form of the BCMP network is as follows:

$$P(q_1, q_2, q_3 \dots q_n) = \prod_{m=1}^{x} (1 - P_m)P_m^q \tag{6}$$

We represent the traffic rate estimation of a cluster as ω with the consideration that at least a single sensor node will have buffer occupancy greater than 0.

$$\omega = 1 - \prod_{m=1}^{x} (1 - P_m) \tag{7}$$

According to the above mentioned assumptions and formulas, the estimated traffic rate of each sensor node m is given by α_m within a predefined time quantum and is broadcasted to the CH. The collective traffic rate of all the sensor nodes are obtained by the values of α_m where $1 \leq m \leq X$. This is called the ω estimate which determines the congestion degree and is exchanged between the CHs on its route to the sink. When the value of ω becomes high $i.e.$ a sensor's queue is about to be full and encumbered, that particular CH

is discerned to be congested. We need to set a threshold value ω_t so that we can have a clear idea about when a cluster becomes congested.

We estimate ω_t from all the nodes in which some nodes may have high priority flow and some may have low priority flow. A predefined threshold value is set for ω_t which should not exceed during high priority data flows. As per our simulation results, even when the cluster sizes vary greatly, ω_t is much approximate and has very low deviance for less number of sensor nodes in a cluster. Our motive is that the value of ω_t should be less than or equal to 0.6 and is likely to be the tolerance level for congestion in clusters. When ω_t exceeds 0.6, it is concluded deliberately that the path that follows that CH is congested.

4.5. Traffic Rate Adjustment

There are two ways of communication between clusters. They are Intra cluster and Inter cluster communication. The former one has communication between sensor nodes within a cluster and the latter one has communication between CHs. Initially, no clusters will have idea about other clusters regarding congestion. If this information is needed to be exchanged among CH's, it will result in overflowing or flooding of information which is a bottleneck to be avoided. ZRP manages routing of packets and after a certain time, the CHs send updates between them which are the initial setup in the clustered architecture.

In Intra cluster communication, each sensor node estimates the arrival rate (μ_m) by computing the total number of incoming packets based on the flow of importance and the service rate (λ_m) by analyzing the time taken for servicing the incoming packets depending on its state. This is not done quite often, where it is done sporadically for every time interim t_i. We move on to clustered architecture because of its advantages like less number of control messages transmitted and low probability of retransmission of dropped packets. In order to save the energy of sensor nodes the traffic rate ω is forwarded to the CH only when it exceeds the threshold value $\omega_t \geq 0.6$ as conferred earlier. Some nodes may not have its buffer to exceed the threshold value but may have high level of importance. Such sensors also set their threshold value to exceed the pre-set level to ensure that the CH is in an emergency to forward its information to the sink. While estimating ω_t, the CH finds the congested node congested and tries to forward it to the sink. This is how Intra cluster communication takes place.

Inter cluster communication is the communication of information between CHs. The traffic rate estimate is calculated periodically by each sensor node within a cluster and the congestion degree is thereby analyzed by the source cluster *i.e.* the CH that broadcasts the other CHs during congestion. In such a scenario, the route following the congested CH will be also congested. When a CH finds ω_t to exceed beyond the requisite level, it broadcasts the beacon information among the other CHs by setting the congestion degree to be high in its packets' control message. The traffic rate will be maximum for congested CHs *i.e.* $\omega_t = \gamma_{max}$ and this information is overheard by the other CHs with $\omega_t = \gamma_{min}$ (*i.e.* clusters other than the nodes that have high traffic) along its route and adjusts their traffic rate when it is lesser than the observed congested rate. Also when some CHs have low important data flows, congested CHs are forwarded to the sink first where the other clusters delay itself waiting for another chance to reach the sink. This incurs some energy expenditure which is not too high when compared to the other protocols consuming more energy for retransmitting the lost packets or retrieving the dropped packets during congestion.

5. Performance Evaluation

The proposed congestion control algorithm is evaluated in this section. For simulation, we have used the Network Simulator NS-2 version 2.29. The simulation parameters are described in Table I.

Table 2. Simulation Parameters

Parameter	Value
Number of sink nodes	1
Number of sensor nodes	50 to 99
Simulation Area	100 * 100 m^2
MAC Protocol	802.11
Routing Protocol	ZRP
Sector Radius	3 hops
Average packets per node	60
Packet Size	512 bytes
Threshold value for CH	0.8
Threshold value for sensor node	0.6
Transmission Range	30m
Beacon interval	1 second
Simulation Time	100 seconds

5.1. Simulation Setup

We have used the network topology deployed in a simulation area of 100 * 100 m^2. There is one sink node and the sensor nodes vary from 50 to 99. A simple MAC protocol is used and the ZRP protocol manages the routing of sensor nodes with a sector radius of 3 hops. The transmission range is restricted to be within 30 m with the packet size of 512 bytes and the average packets per node to be 60.The threshold value for CH is 0.8 and that of each sensor node is 0.6. The radio coverage i.e. the transmission range is 30 m with a beacon interval of 1 second. The total setup is simulated for a 100 seconds and is represented in Figure 3.
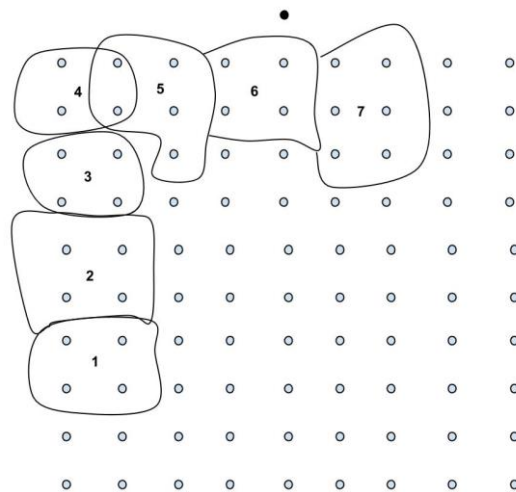

Figure 3. Setup of Clusters in a Network

Initially we are setting up the network with a single sink and 99 sensor nodes that has seven clusters with two different flows namely Low Priority Flow (LPF) and High Priority Flow (HPF). We have compared our technique with CRAP and without CRAP experiments. This is done for every 15 seconds and the event burst is analyzed for every 3 seconds by considering 7 clusters for our experiments. Consequently, we have 7 runs of

results and understand that clusters 4 and 5 collide in which congestion is realized in the earlier stages causing packet drops. From Figure 4, we understand that there is a substantial change in the ω estimate when we move from LPF to HPF which shows clearly that ω rises quickly during HPF (above 0.6) when compared with LPF (below 0.2) where congestion is either low or there is no congestion at all. Thus energy can be greatly saved as the clusters need not send any update messages to CHs during LPF and updates only when a particular threshold value exceeds.
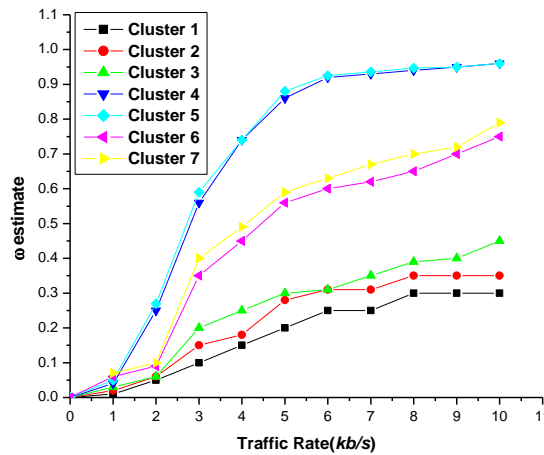
Figure4. ω Estimate of All Clusters Versus Traffic Rate

5.2. Packet Transfer Delay

The Packet Transfer Delay (PTD) is usually affected by the number of routes towards the sink and the level of congestion. Factors of PTD include processing of sensor nodes, time delay for transmission, broadcast delay and packet waiting time in the queue.

$$Time\ delay\ for\ transmission = \frac{P_l}{B_l}$$

where P_l is the length of packet measured in bits and B_l is the link associated bandwidth measured in bits per second.

$$Broadcast\ delay = \frac{L_p}{B_s}$$

Where L_p is the length associated with the physical link and B_s is the broadcast speed of packets measured in *m/sec*. Though processing of sensor nodes is not a big issue to be considered, it delays packets in checking errors and regulating the output link. One of the other factor is the time delay of packets waiting in the queue and the scheduling of queues. According to our protocol, these factors do not override because the sink estimates the time delay of a packet to reach it from the source node. This delay information is piggybacked to the CH as $\omega_t = \gamma_{max}$ from which the packet was sent. If this exceeds the level of packet dropping, it broadcasts this information to all the sensor nodes in its cluster. Thus the other sensor nodes adjust their rates to $\omega_t = \gamma_{min}$ and a new transmission rate is estimated and sent periodically.

We have plotted the average packet transfer delay versus traffic rate in both lightly loaded and densely loaded environments with CRAP and without CRAP setup which is represented in Figure 6. In CRAP with 50 nodes setup, the delay is substantially low when compared to CRAP with 90 nodes setup. On the other hand, a network without

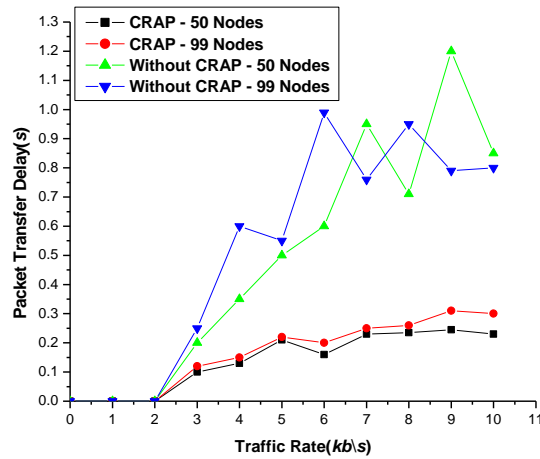CRAP results in unacceptable queuing delays which cannot even sustain when traffic exceeds 3 kbps.

Figure 5. Packet Transfer Delay Versus Traffic Rate

5.3. Packet Delivery Ratio

Packet Delivery Ratio (PDR) is defined as the ratio of the number of packets delivered to the destination over a certain period of time. Figure 6 shows the observed throughput with CRAP for LPF as well as HPF and throughput obtained without CRAP. We understand that PDR is above 0.75 for all the clusters having HPF and LPF in CRAP based network and the results are similar to each other. Still, HPF has high PDR than LPF and is good to have very low packet dropping probability. An important point to note is that the values remain constant up to 3 kbps for all the three comparisons. But after 3 kbps, PDR without CRAP shows very high packet drops for every successful packet delivery which leads to congestion.
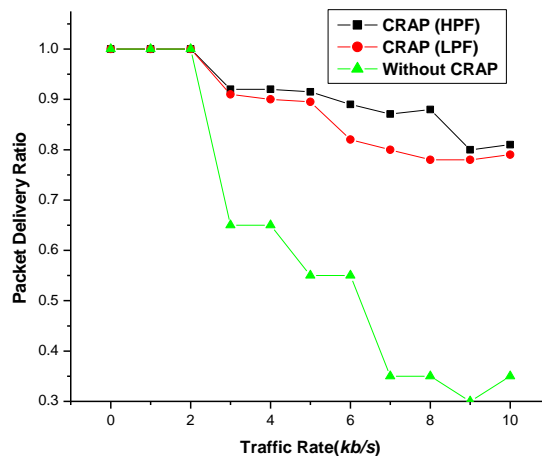

Figure 6. Packet Delivery Ratio Versus Traffic Rate

5.4. Throughput

It is defined as the ratio of the number of packets sent to the number of packets received within an observable amount of time. In Figure 7, we observe the number of

packets sent versus received over the entire network with an increasing traffic rate for both 50 nodes and 99 nodes for both HPF and LPF.
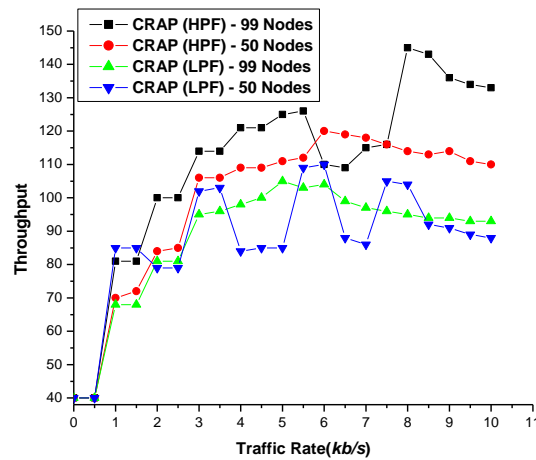

Figure 7. Throughput Versus Traffic Rate (HPF and LPF)

Throughput is comparatively high for CRAP with 99 nodes setup under HPF and CRAP with 50 nodes setup falls in an acceptable level of throughput gained. Now, when we observe our experiments for network setup without CRAP for both 50 nodes and 99 nodes, it is very less resulting either in packet delay or packet dropping where both should be avoided.

5.5. Energy Consumption

One of the main advantages in CRAP is saving energy and we have demonstrated it in Figure 8. For every successful packet delivery without CRAP, the number of packets dropped (P_d) rises in multiple hops with the packet size (P_s) which gives the number of wasted packet transmissions. It is formulated as

$$P_w = P_d . P_s$$

Also, when the packets are dropped, the number of retransmission of control packets (R_c) have to be considered for analysis and it is the product of the size of control packets C_s and R_c. Thus the total energy consumed (ξ) is formulated as:

$$\xi = P_d . P_s + C_s . R_c$$

The total energy saved (ξ_s) using CRAP is formulated as:

$$\xi_s = \frac{P_w - \xi}{\xi}$$

ξ_s is plotted in Figure 8 and observed that there is no energy consumption for very low traffic rate and becomes negative. When this rate is increased, energy consumption is stable and similar in the network setup with 50 and 99 nodes until it does not exceed a certain threshold. When it exceeds the limit, CRAP with 99 nodes have higher energy dissipation compared to that of 50 nodes which is quite common.
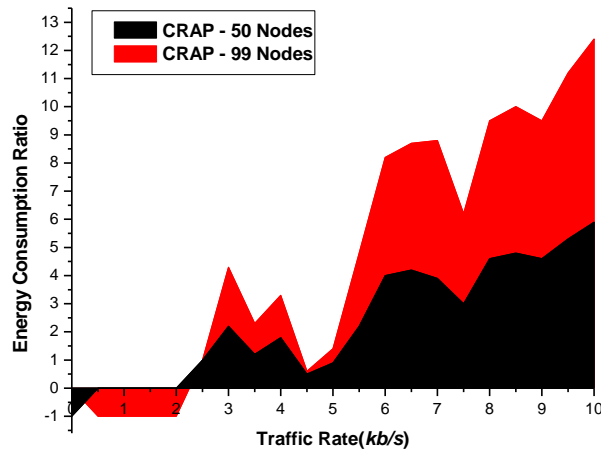
Figure 8. Energy Consumption Versus Traffic Rate

The number of control messages needed for retransmitting the packets is an issue to be considered as this wastes the energy spent by the CH as well as the sensor nodes. The ratio of control message overhead with respect to network density is shown in Figure 9.This does not have much impact in the clustered network because updates are made only when it exceeds ω_t which restricts the need for updating of both CH as well as sensor nodes.
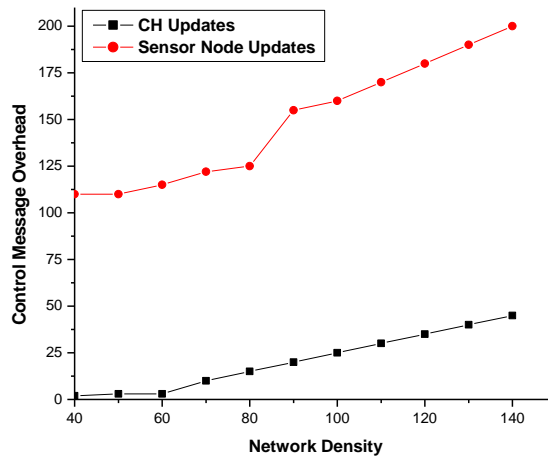

Figure 9. Control Message Overhead Versus Network Density

6. Conclusion

In this research work, we propose our protocol that supports different classes of data flows in a widely deployed sensor network. Our goal is to provide prominence to HPF thereby adjusting the traffic rate of other sensors having LPF even during congestion. This adjustment is done in a clustered environment and the battery power and energy is greatly saved as the sink is updated only when a CH finds a sensor node to be congested while exceeding the predefined threshold value. Thus sensor nodes do not have unnecessary energy dissipation which happens quite often in a non-clustered sensor network. Because of this, packet dropping is greatly avoided by controlling congestion and provides fidelity in packet delivery for both types of flows. This work can be

extended by analysing the medium priority flows *i.e.* when the threshold value is moderate when compared to HPF and LPF.

References

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

[2] E. Zarei, Mashad and S. M. Mazinani, "An Obstacle Aware Clustering Algorithm for Wireless Sensor Network", Proceedings of IEEE Symposium on IS3C, (**2012**), pp. 483-486.

[3] Z. Hanzalek and P. Jurcik, "Energy Efficient Scheduling for Cluster-Tree Wireless Sensor Networks with Time Bounded Data Flows: Application to IEEE 802.15.4/ Zigbee", IEEE Transactions on Industrial Informatics, vol. 6, no. 3, (**2010**), pp. 438-450.

[4] B. A. Sabarish and K. Sashi Rekha, "Clustering based energy efficient congestion aware protocol for Wireless Sensor Networks", Proceedings of IEEE Conference on ICETECT, (**2011**), pp. 1129- 1135.

[5] F. Nie and Z.-F. Fu, "MMLC mobile clustering routing scheme based on LEACH in Wireless Sensor Network", Proceedings of IEEE Conference on WCICA, (**2010**), pp. 6793-6798.

[6] A. Koubaa, R. Severino, M. Alves and E. Tovar, "Improving Quality-of-Service in Wireless Sensor Networks by Mitigating "Hidden-Node Collisions", IEEE Transactions on Industrial Informatics, vol. 5, no. 3, (**2009**), pp. 299-313.

[7] A. Chatterjee, B.C. Roy, A.K. Mukhopadhyay and D. Mukherjee, "A transport protocol for congestion avoidance in Wireless Sensor Networks using cluster-based single-hop-tree topology", Proceedings of IEEE Third International Conference on EAIT, (**2012**), pp. 389-393.

[8] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks", Proceedings of ACM Sensor systems, (**2004**).

[9] C. G. Wang, B. Li and K. Sohraby, "Upstream congestion control in wireless sensor networks through cross-layer optimization", IEEE Journal on Selected Areas in Communications, vol. 25, no. 4, (**2007**), pp. 786-795.

[10] Y. Sankarasubramaniam, O. Akan and I. F. Akyildiz, "ESRT: Event-to-sink reliable transport in wireless sensor networks", Proceedings of 4th ACM International Symposium on Mobile ad hoc Networks, Annapolis, MD, (**2003**), pp. 177–188.

[11] M. O. Farroq, A. B. Dogar and G. A. Shah, "MR- LEACH: Multi-hop Routing with Low Energy Adaptive Clustering Hierarchy", Proceedings of IEEE Conference on SENSORCOMM, (**2010**), pp. 262-268.

[12] Ye, M. Li, C. F. Chen and J. Wu, "EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks", International Journal of Ad Hoc Sensor Network, vol. 3, (**2007**), pp. 99-119.

[13] H. Chan and A. Perrig, "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation", In Proceedings of the First European Workshop on Sensor Networks, (**2004**).

[14] P. Jeongyeup and G. Ramesh, "RCRT: Rate –Controlled Reliable Transport for Wireless Sensor Networks", Proceedings of ACM Sensor Systems, (**2007**).

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

[16] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy, "PSFQ: A reliable transport protocol for wireless sensor networks", In Proceedings of ACM MMOBICOM, (**2006**).

[17] M. M. Bhuiyan, I. Gondal and J. Kamruzzaman, "Dual-Channel based energy efficient event clustering and data gathering in WSNs", Proceedings of IEEE Asia Pacific Conference on APCC, (2011), pp. 241-246.

[18] J. F. Hayes and T. Babu, "Modeling and Analysis of Telecommunication Networks", 2nd edition Wiley – Interscience, New Jersey, (**2004**).

[19] Y.-U. Yun, J.-K. Choi, N. Hao and S.-J. Yoo, "Location–based spiral clustering for transmission scheduling in wireless sensor networks", Proceedings of IEEE 12th International Conference on ICACT, (**2010**).

[20] K. Mori, K. Naito and H. Kobayashi, "Distributed Backoff Mechanism for Traffic Adaptive Active Period Control in Cluster-Based IEEE 802.15.4 WSNs", Proceedings of 73rd Vehicular Technology Conference, (**2011**), pp. 1-5.

Authors

S. Jeya Shobana, she has accomplished B.E. and M.E. in Computer Science and Engineering and at present she is working as an assistant professor in department of Computer Science and Engineering, Rajas International Institute of Technology for Women, Nagercoil, India. Her areas of interest are Wireless Sensor Networks, Congestion control, Protocol design, Cryptography & Networks and Database Management Systems.

B. Paramasivan, he has accomplished his Ph.D. in Information and Communication Engineering from Anna University, Chennai, India. He is working as a Professor and Head in department of Computer Science and Engineering, National Engineering College, Kovilpatti. His areas of interest are Wireless Sensor Networks and Wireless Ad-hoc Networks. He is a life member in IE, CSI, ISTE, IEEE and ACCS.
.