Middle-East Journal of Scientific Research 24 (9): 2668-2674, 2016
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2016.24.09.23877

Improving Performance of Wireless Sensor Networks Using Alternative Route Selection Congestion Control Algorithm

[1]*Srigitha S. Nath and* [2]*K. Helenprabha*

[1]Saveetha Engineering College, Chennai, India
[2]R.M.D. Engineering College, Chennai, India

Abstract: Congestion manipulation is discovered to be one of the most momentous trials in Wireless Sensor Networks (WSNs) that is counted for resource restraint specification and the number of sensor nodes. Congestion results in longer latency, packet drops and power consumption in Sensor networks. Many ideas have been introduced in the past involving routing protocols, to detect and control congestion mechanism. In the earlier schemes, the congestion avoidance was done by the sink node that reasons topology reset and abundant traffic drop. In continuation, the latest mentioned congestion control protocol will detect and control congestion avoidance at the node level itself. Various sensed packets needs disparate priorities such as fairness, reliability, transmission rate, etc, . The paper proffers ARS, a congestion control algorithm using layer based dynamic alternative node selection for diverse traffic in the congested network. Link capacity and Packet Service Ratio (PSR) are computed for congestion notification. We used priority queue alongside prioritized traffic to confirm fairness.

Key words: Congestion Control · Link capacity · Packet service ratio · Priority queue

INTRODUCTION

Wireless sensor Networks (WSN) [1-5] is a pack of assorted wireless devices installed with different kinds of sensor nodes to amass data from the environment. The amassed data is forwarded from node to node, employing a multi-hop routing protocol towards the sink. Data aggregation and analysis are done at the sink. Energy consumption, data storage capacity, processing skills is some of the challenges in a sensor node [6-9].

Real time values need constant sensing of parameter for important application such as nuclear stations monitor. Periodic sensing method is adopted whenever continuous sensing is not possible. Sensor nodes broadcast packet only when an event occurs is known as event driven data transmission for examples such as Forest fire, Tsunami alert. Higher priority is always offered to event driven application compared to continuous flow, then the event driven sensing devices directs to uncertain network load and cause congestion and disrupts the network data transmission.

Congestion happens in two main situations. When the packet scheduling rate becomes larger than the packet service rate, congestion occurs. The probability of congestion arising in the nodes (node level) nearer to the Base Station is high. On the other hand, congestion occurs when the random channel varies and causes bit error rate in the links (link level).

To compute congestion, we used packet service ratio and channel link capacity by adopting a hybrid technique. The concept of "priority based alternative path selection" is employed in this paper to control congestion. The system architecture and alternative path selection analysis are elaborated in sections 2 and 3 respectively. The performance analysis and the observation results are analyzed in section 4.

Related Work: CODA [10] is a congestion control technique that uses two parameters to detect congestion, 1) remaining Buffer occupancy 2) channel load. Congestion is detected whenever the threshold value of above two parameters exceeds. Now, BS (Base Station) transmits ACK packet that force the nodes to reduce the data rate. CODA does not ensure reliability and whenever network is heavily congested, the response time will increase.

Corresponding Author: Srigitha S. Nath, Saveetha Engineering College, Chennai, India.

ESR [11] is to guarantee a preferred reliability using adjusting the sensors reporting frequency. ESRT detects congestion by buffer occupancy. ESRT is well suitable for energy conservation and reliability but does not give attention to interference.

In PCCP [12], congestion is detected by, using the ratio between PSR and inter arrival time of packets. It supports both node level and link level congestion and when congestion is detected, nodes decreases their sending rate based on priority index. But it provides priority only for nodes in the network not for different traffic flows.

PHTCCP [13] is a hop-by-hop rate adjustment where congestion is detected by packet service ratio and it uses weighted fair queuing for scheduling. It provides priority to traffic to achieve required reliability, but it does not support link level congestion.

ECODA [14] uses two parameters for congestion detection.1) Weighted buffer difference 2) Dual buffer thresholds. ECODA dynamically estimates channel loading with an implicit manner and optimizes channel utilization. It uses an AIMD scheme for congestion control in the network.

System Model: The network, analyzed in this paper, is a heavily deployed WSN with nodes random as well as uniformly spaced. These nodes set up alongside similar number of assorted sensor boards. Each node can send disparate kinds of data simultaneously to BS. The highest priority is allocated for the event driven traffic by the Sink.

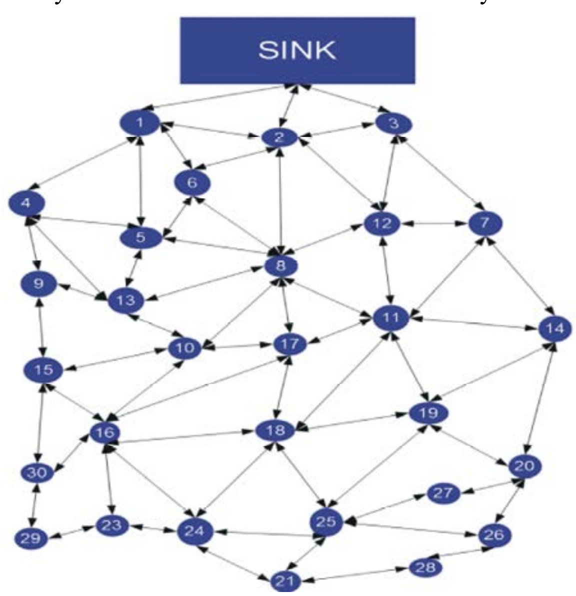

Fig. 1: Network Topology before Topology control Algorithm

Assigning Levels to SN: Consider the Fig. 1, Sink broadcast PING message with its level value. The level value is '0' for BS. The node which receives Ping message from BS is now come to know that it can directly receives transmits/receives data to/from Sink, it adds one to the level value and rebroadcast PING message with its ID and Level value=1. If a node receives PING message from more than one node, it waits until it receives PING message from all the neighbors. Then searches the neighbor table with the lowest level and add one to this level. Now, the node rebroadcast with its own Id and updated level value in the network. This process continues until all the nodes set their level value. The layer number is dynamically updated during the whole lifetime of the node due to lossy links [15]. Fig. 2 shows the level value of each node in the network.

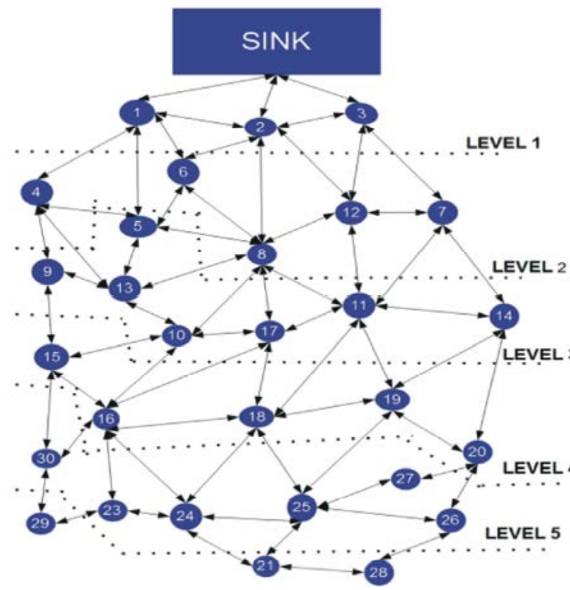

Fig. 2: Network Topology after Topology control

Table 1: Neighbor Table of node 26

Node_ID	Buffer Occupancy	Flag	Link Capacity	Residual Energy	Layer Level
20	0.62	1	1.1	0.95 J	1
25	0.74	1	0.8	0.87 J	1
28	0.80	1	0.75	0.78 J	2

Neighbor Table: In the Neighbor table, as well the ID of the adjacent nodes and their layer value, data are being retained concerning their buffer occupancy, their residue energy, as well as the Flag and link capacity. If the neighbor node is available, the Flag Id is set to zero, else it remains 1. A node sets its flag 0, when its power is fully drained and again resets to 1 when sufficient energy is acquired.

We considered that the nodes posses 'n' number of uniform sized priorities queue for various traffics. The network has mainly two kinds of packets such as,

1) Source Traffic 2) Transit Traffic.

As the transit traffic has already traversed many hops, priority queue assigns higher priority to it, when compared to source traffic. This is because loss in transit traffic causes resource deficiency. As mentioned before event driven packets have second higher priority than continuous flow.

Gentle Plan

Congestion Detection: The node experiences congestion, when its incoming rate is greater than the outgoing rate [16].

$$PSR = Tr_{MAX} \bigg/ \sum_{i=0}^{n} (R_r) \quad (1)$$

EWMA concept is used to evaluate Tr_{MAX}. When PSR = 1, no occurrence of congestion. On the other hand, when PSR < 1, the node's buffer begins heaping with incoming packets, finally dispatches congestion notice.

Algorithm 1: Congestion Detection

```
if (R(i) = 1)
 no_congestion();
else
congestion_occurs();
end
```

For example, here from the fig 2, it is found that node 11 receives packets from nodes 18 and 19. This node 11 can forward packets to 7, 8 and 12. When the PSR<1, for the node 11 and if no data loss preventive methods are initiated, then packets drop. It is interesting to learn that this congestion process can happen even when there is only one flow similar to the one mentioned above. That is even when the network is not loaded; just two flows are more than enough to create congestion in a single node. In such a situation, application of traffic control mechanism will compel the nodes to lesser their transmission rate when compared to the remaining unused resources in the network.

The ARS deals this situation with a suitable plan called "gentle plan" as described here. From the previous discussion, whenever a node receives packets from more than one node, then each node enters into alert condition. On getting alert, the node tries to avoid data reception from more than one flow. During avoidance, the node notifies its previous node through a "back pressure" message. To obtain this, node sets the "next sequence number" field in the ACK packet header to "False" for the source node ID that was disturbed during transmission. The node which is selected for reception is programmed to check for a new route. Each node selects the path with higher transmission rate. The node 11 in fig 2 is assumed to receive packets with higher rate from 18, then node 19 will receive a False value in its ACK packet header to the 'next sequence number" field. As a result, the node B begins looking for another feasible path in its neighbor table.

In this Gentle plan algorithm, the light is focused on the fact that this algorithm "advices" the nodes to find a new path using level value. If the disturbed nodes are not able to find a new path, then they continue forwarding data packets through the same upstream node. Relative paths are selected as the routes which can streamline the packets transmission as well as will not disturb the performance of the network. In this fig 2, an excellent route for node 19 can begin from node 14, as 14 is in the same level as node 11.

Algorithm 2: Congestion Notification

```
if (Pack_Service_ratio < 1)
{
Node_n=
congestion_notification_message("backward_node")
Priority1: If (backward_node ? transit_traffic())
{
Send ("Congestion_notification" );
}
Priority2: else
{
If (backward_node->low_transmission_rate())
{
Send ("Congestion_notification" );
}}
end
else
No Congestion();
}
end
```

Rigid Plan: During the process of gentle plan, a node is just "advised" to vary its forwarding route as we discussed earlier. Therefore the sender can now take decision about forwarding data packets to the same forwarding node. The flag decision algorithm is initiated, if this condition is not satisfied. The flag decision algorithm is an easy method and purpose is to alter the value of the field "flag" in the neighbor table from true to false or vice versa, during favorable situation.

A node's presence or absence for packet reception is clearly displayed by the flag field. Every time this status of the flag field is changed by the node, that node announces its neighbor immediately through an improved ACK packet it.

The status of the node shows "not available" due to below mentioned facts.

Buffer Occupancy: Is reaching its upper limit. If the receiving rate of a node is greater than its transmitting rate, then it will result in over flow and packets drop. When the buffer capacity of a node reaches a particular threshold value, the node changes its flag status to false and intimates this message through ACK packets to all its neighbor nodes. Ultimately the neighbor nodes halt forwarding data packets to that node and commence the process of searching a feasible route. After recovering from the over flow situation, the congested node resets its flag status to true again and sends message to neighbors.

Residue Energy: When the energy of a node starts draining, it also converts its flag to false. The application or the hardware mechanism may decide the lower limit of energy. After gaining enough energy through any energy saving mechanism, the flag can be changed to true status.

Lower Level Node Unavailability: Whenever there is a shortage of nodes the nearer to BS, that node tends to change its flag to false. During this critical situation, that node cannot forward any of the received data packets and will result in congestion. Here that node will turn its flag to false and remains till any of the neighbor nodes which are at lower level than itself, becomes available again.

Substitute Path Creation: The number of hops to the sink node may vary when the node's state in the same layer as the sink varies, because here the algorithm is dynamic. After ignoring the congested node, the selection of the new node to transmit packet relies on its availability and also the number of hops to the sink node. Applying this technique, every node can find the next node with less evaluation in very smooth manner.

It separates the number of available nodes in increasing order of hops from the BS and transmits the packet to the top node in the list. If the node is unavailable, then the forwarding node next selects the second node in the list. The table opts to separate with respect to residue energy if there is more than a single node at the same level. Lastly, nodes are separated based on the buffer occupancy when there is more than one node above the threshold. In the worst case, if this threshold values are similar for more than one node, then the algorithm selects the node with the lower node ID to transmit the packet. This algorithm gives importance to the consistent evaluation of metrics like the mean time for the forwarding of data packets from a source to the sink, as well as to the network's even distribution of energy usage, thereby not permitting one formation of energy and routing "holes".

A weighted function can be designed and need depending on the application, taking into account the idea of the new feasible path. As mentioned earlier, priority factors support the idea of the new path, availability (flag), number of hops to BS, residue energy and buffer occupancy. For example, if there is a request for less delay, the number of hops to BS should have the highest weight. Next if the life of the network is more important, then residue energy must be given more priority.

In our work, we treat all factors equal, giving more importance to hop count. An example of a weighted function is given here. Again viewing Fig. 2 and we take into account that node 2 is not available further. The balance buffer occupancy is performed for the idea of the new path when compared to the delay as the main reason for an application. Therefore it uses a weighted function which can handle this problem as projected in equation [2].

$$\text{Priority factor} = [(B_O * 6) + (RE * 2) - (H_C * 2)] * F \qquad (2)$$

where, B_O and RE are Buffer occupancy and Residual energy respectively. H_C is Hop count and F is Flag. If the flag is set as 'false', then the node is unavailable. Since flag is "zero", then Priority factor is equal to "zero". From Table 2, node 11 will be selected as an alternative one rather than node 5, 6 and 12.

Table 2: Example of neighbor table for the Priority Function

Node_ID	Buffer occupancy	Residual .power	No of hops	Flag	Priority Factor
2	0.78	0.45 J	2	0	0
5	0.59	0.5J	2	1	0.2
6	0.	0.6J	2	1	0
11	0.92	0.5J	3	1	1
12	0.6	0.3J	2	1	0.2

Channel Estimation: Node level congestion alone can be taken care of by adopting packet service ratio mechanism. For instance, looking at Fig. 3, node 2 undergo packet collision and path loss resulting in packets drop.

This matter is not known to node 3 as its transmitting rate (25pkts/sec) is equal to its receiving rate (25pkts/sec). As a result, for congestion detection accuracy, a hybrid mechanism is needed using PSR and link load.

In order to eliminate congestion, radio link estimation is required and also to evaluate the various capacities, every node (x) estimates radio link relating to its parent node (y), by transmitting burst of packets during a precise amount of time (T). After collecting the previous packet's ACK or after a time out of previous packet, the next packet is transmitted. The link capacity is determined by

$$\text{Channel capacity } (C_{x,y}) = \text{Total number of ACK packets}/T \quad (3)$$

Each node transmits the estimated link capacity along with its parent ID to the sink after estimating link capacity. Fairness is guaranteed by BS by evaluating the exact number of source nodes in the network that is represented by 'σ'. Thus the parent node of each source node may be allowed to transmit at least 'σ' packets during a single frame.

By using packet service ratio, we can control only node level congestion. For example, In Fig. 3, Consider that node 2 experience channel loss or packet collision, then there is packets drops. But node 3 doesn't aware of this, because its transmitting rate (25 pkts/s) is equal to the receiving rate (25 pkts/s -node 1's transmitting rate). Therefore, for accurate congestion detection, a hybrid approach is required using packet service ratio and channel load.

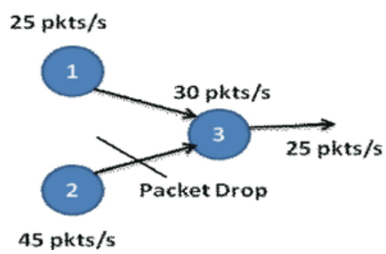

Fig. 3: Packet Collision

Performance Evaluation: In this section, we evaluate ARS and its performance is compared against existing algorithms. ARS uses priority queue to provide fairness and reliabilty to data packets. Two schemes has been proposed here, 1) Gentle Plan 2) Rigid Plan. We simulated our result using Mannasim patch with ns-2.29. The BS is placed at the upper edge of the simulation region. All the nodes which sense the event are considered as source nodes, then rest of the nodes used as forwarder node to relay the detected data. If we take away Gentle scheme from ARS, the Rigid scheme demonstrates worst results compared to other algorithms. It shows the importance of balancing traffic in the early stage itself.

Table 3: NS-2 parameters

Area of Sensor Field	500 × 500 m^2
Node Deployment	Randomly(Uniform Distribution)
Preliminary number of deployed sensors	100
Sensing Range	30 m
Communication Range	50 m
Packet Size	1024 bits
Buffer Size	512 k
Insert Layer	CSMA/CA
Simulation Time	60 s

Delivery Ratio: The ratio between the numbers of successfully delivered packets to number of packets generated. If no congestion control algorithm is used, then the performance of networks get degraded, since the forwarded data will not reach the BS. The networks performance increases a lot whenever the congestion control algorithms are applied. The result shows ARS is efficient and balance the traffic load. In comparison with existing congestion control protocols ARS can able to produce better result in delivery ratio. If we neglect Gentle plan in ARS, Rigid Plan provide low preformance when compared to PHTCCP and PCCP. It shows the importance of balancing traffic in the network at an earlier before any possibility of congestion occurs. ARS can able to deliver 25-30% additional packets compared to PHTCCP and PCCP.

Throughput: ARS can able to provide highest throughput in comparison with other algorithms. It uses a hybrid method by jointly considering packet service ratio and channel condition.Therfore we can provide better throughput.

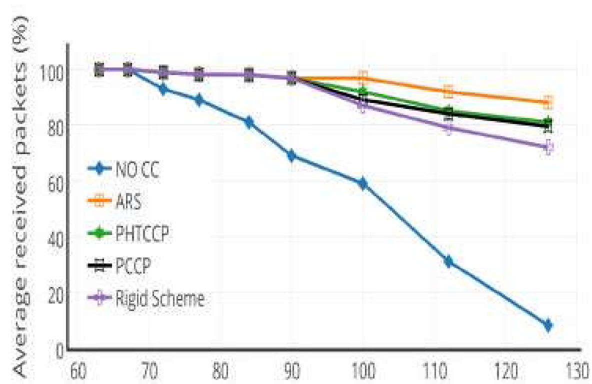

Fig. 4: Successfully Received Packets (%)

Fig. 5: Average Throughput

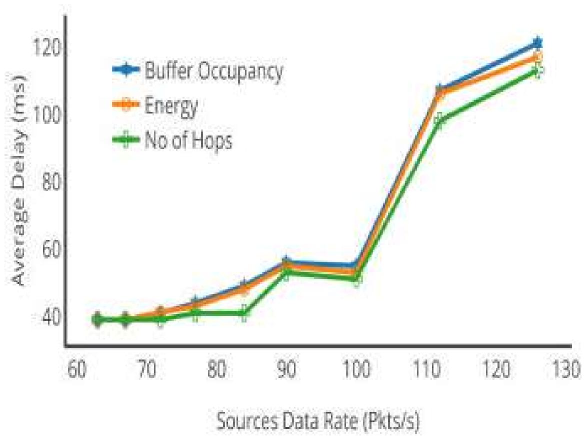

Fig. 6: Average Delay (ms)

Analysis of Priority Factor: In this section, we demonstrates the variation in the result by changing the coefficient of weighted function in equ (2). The coefficient of H_c incereses twice compared to other weighted function to calculate ARS_Delay. Likewise, we study the performance of other weighted function by doubling their coefficient.

The priority factors for each weighted function is given below,

Buffer Occupancy: PF = [(B_O* 2) + RE – H_C] * F (4)

No of Hops: PF = [B_O + RE – (H_C*2)] * F (5)

Energy: PF = [B_O + (RE *2) – H_C] * F (6)

From the Fig. 7, we can see that if the "coefficient of B_o" is doubled, then the throughput increases slightly while checking with other cases when the H_c and RE are doubled. Whenever the "no of hops" is doubled, since the network selects only the minimum hop count path irrespective of residual energy and buffer occupancy, the delay for a packet to reach BS is reduced which is shown in Fig. 6. This leads to lesser delay while considering other weighted parameters but also smaller in throughput and lifetime of the network.

Percentage of network's remaining energy is shown in Table 4. The simulation has been computed until there is no more packets transmission from source nodes to sink and 350 nodes are considered for simulation. By using ARS, we can utilize the resources maximum amount to transmit packets.

Table 4: Percentage of network's remaining energy (%)

No CC	49.11 ± 4.17
Rigid Plan	8.79± 0.91
PCCP	4.21± 0.28
PHTCCP	3.78 ± 0.09
ARS	3.68 ±0.03

Table 5: Percentage of network's remaining energy (%)

No. of Hops	5.37 ±0.06
Buffer Occupancy	3.87 ± 0.07
Energy	3.58 ±0:07

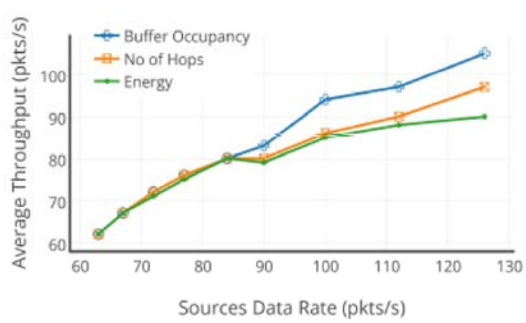

Fig. 7: Average Throughput

Likewise, the nodes which have residual energy above the threshold values are selected when the "residual energy" is doubled. Therefore, based on the application like "low delay" regardless of throughput and lifetime, it is suitable by doubling the coefficient of "no of hops". Similarly, based on the application like higher throughput, network lifetime, we can select the optimal weighted function.

CONCLUSION

In this paper we present ARS for controlling congestion and finding a feasible path in a congested network. We are using a gentle plan to minimize the occurrence of congestion. Here we are evaluating PSR and trying to eliminate congestion and if the mechanism proves best, then congestion is avoided. On the other hand, if congestion still remains, then we opt for rigid plan, where we use the algorithm to select a new feasible path to forward data packets to the BS. ARS proved better performance when compared with PHTCCP and PCCP, which is shown in the simulation.

REFERENCES

1. Mohamed Amine Kafi, Djamel Djenouri, Jalel Ben-Othman and Nadjib Badache, 2014. "Congestion Control Protocols in Wireless Sensor Networks: A Survey, " IEEE Communications Surveys & Tutorials, 16(3), Third Quarter 2014
2. Mohamed Amine Kafi, Djamel Djenouri, Jalel Ben Othman, Abdelraouf Ouadjaout, Miloud Bagaa, Noureddine Lasla and Nadjib Badache, 2014. "Interference-Aware Congestion Control Protocol for Wireless Sensor Networks," The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014).
3. Shaikh, F.K., A. Khelil, A. Ali and N. Suri, 2010. TRCCIT: tunable reliability with congestion control for information transport in wireless sensor networks. In: Proceedings of the international wireless internet conference (WICON). Singapore.
4. Hull, B., K. Jamieson and H. Balakrishnan, 2004. Mitigating congestion in wireless sensor networks. In: Proceedings of the 2nd ACM conference on embedded net- worked sensor systems (ACM SenSys). USA, pp: 134-47.
5. Srigitha S. Nath and K. Helen Prabha, 2016. "Layer based energy efficient routing protocol for heterogeneous wireless sensor networks," Asian Journal of Information and Technology, 15(10): 1621-1627, 2016
6. Dinusha A.J. Rathnayaka, Vidyasagar M. Potdar, 2012. "Wireless Sensor Network transport protocol:A critical review, " Journal of Network Muhammad Monowar, Obaidur Rahman, Al-Sakib Khan Pathan, and Choong Seon Hong, "Prioritized Heterogeneous Traffic-Oriented Congestion Control Protocol for WSNs, '' The International Arab Journal of Information Technology, 9(1).
7. Dipti Patil, Sudhir N. Dhage, 2012. "Priority-based Congestion Control Protocol (PCCP) for Controlling Upstream Congestion in Wireless Sensor Network," 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India.
8. Chieh-Yih Wan, Shane B. Eisenman, Andrew T. Campbell, 2011. "Energy-Efficient Congestion Detection and Avoidance in Sensor Networks," ACM Transactions on Sensor Networks, 7(4), Article 32, Publication date: February 2011.
9. Kanodia, V., C. Li, A. Sabharwal, B. Sadeghi and E. Knighty, 2002. "Distributed Priority Scheduling and Medium Access in Ad Hoc Networks," Computer Journal of Wireless Networks, 8(1): 466-466.
10. Wan, C.Y., S.B. Eisenman and A.T. Campbell, 2003. "CODA: congestion detection and avoidance in sensor networks," In: Proceedings of the 1st ACM conference on embedded networked sensor systems (ACM SenSys). USA, pp: 266-79.
11. Sankarasubramaniam, Y., O.B. Akan and I.F. Akyildiz, 2003. ESRT: event-to-sink reliable trans- port in wireless sensor networks. In: Proceedings of the 4th ACM international symposium on mobile ad hoc networking and computing (ACM Mobihoc). USA, 2003. pp: 177-88.
12. Dipti Patil, Sudhir N. Dhage, 2012. "Priority-based Congestion Control Protocol (PCCP) for Controlling Upstream Congestion in Wireless Sensor Network," 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India.
13. Muhammad Monowar, Obaidur Rahman, Al-Sakib Khan Pathan and Choong Seon Hong, 2012. "Prioritized Heterogeneous Traffic-Oriented Congestion Control Protocol for WSNs," The International Arab Journal of Information Technology, 9(1).
14. Tao, L. and F. Yu, 2009. "ECODA: Enhanced congestion detection and avoidance for multiple class of traffic in sensor networks," in Proc. IEEE 15th Asia-Pacific Conf. on Commun., APCC, pp: 726-730.
15. Charalambos Sergiou, Vasos Vassiliou, Aristodemos Paphitis, 2013. "Hierarchical Tree Alternative Path (HTAP) algorithm for congestion control in wireless sensor networks," Ad Hoc Networks, 11: 257-272.
16. Sangeetha, V., Srigitha S. Nath and K. Helen Prabha, 2015. "Congestion Control using Alternative Path Selection for Heterogeneous Traffic Oriented Wireless Sensor Networks," International Journal of Applied Engineering Research, ISSN 0973-4562, 10(44).