

Hybrid Algorithm to Control Congestion in Wireless Sensor Networks

¹Neha Pant, ²Rakesh Ranjan and ¹M.P. Singh

¹*Department of Computer Science and Engineering,
NIT Patna*

²*Department of Computer Science and Engineering,
MACET Patna*

neha14pant@gmail.com, rakeshranjanbit@gmail.com, mps@nitp.com

Abstract

The Congestion in wireless sensor networks occurs when packets arrive at a fully loaded buffer of a sensor node. Due to lack of space in the buffer, packets are dropped. It also leads to packet delay, retransmission of packets, reduced QoS and throughput. A hybrid of rate control and resource control algorithms can be used to control congestion considering the priority of the packets. Congestion is detected when the service time is greater than the inter-arrival time of packets. The inter-arrival time can be increased by a certain factor computed as congestion degree, thus reducing the rate of sending packets. The congestion degree which is the ratio of service time to inter-arrival time can be sent to the sources. The sources can then adjust the rate of sending packets accordingly to control congestion in the upstream nodes. At the time of congestion, high priority packets are sent using multiple paths after increasing the time between generations of packets thus decreasing the rate of sending packets. Low priority packets are sent using a single path after decreasing the packet sending rate thus increasing the time between sending consecutive packets. The method also serves as a hybrid of congestion control and congestion avoidance.

Keywords: *Wireless Sensor Networks, congestion, congestion control, congestion avoidance, congestion notification, resource control, traffic rate control*

1. Introduction

Wireless sensor networks (WSNs) as the name suggests are networks made up of sensor nodes communicating or interacting wirelessly over a geographical network. WSNs are meant to sense the environmental [1] and physical conditions, encode the sensed data in packets and to report them to the sink node or to the base station in a cooperative manner. There is wide range of applications of wireless sensor networks like in battlefield surveillance [2], detecting fires, floods [3], earthquakes in forests, security, manufacturing, tracking objects [4], monitoring health [5] and habitat [6], *etc.*, Wireless sensor network can be used for the informing the environmental states and conditions of regions where important events and changes may take place that are important to be observed. But the sensors in wireless sensor network has limited amount of memory, processing capability, sensing capability and power. Hence they must be used carefully while transmitting and receiving data.

Node level and link level congestion are two basic kinds of congestion in WSNs. Node level congestion occurs when there is no space left in the buffer of the node where the packets arrive. This leads to the loss of data packets and hence the retransmission of packets is required. This causes extra consumption of energy. Link level congestion happens when

several neighboring nodes transmit packets simultaneously leading to collisions [7]. This reduces the link utilization and overall throughput. Congestion in WSN can be controlled using either traffic control or resource control methods. Traffic control methods attempt to decrease the traffic load whereas resource control methods try to increase the resources to manage the increasing traffic.

The remaining paper is organized in the following manner: Section II discusses some of the related work. Section III elaborates and discusses the design details of the hybrid algorithm. Section IV presents the implementation details. Section V presents the simulation results obtained using this method. Section VI analyses the result and compares it with other methods. Section VII finally concludes the paper.

2. Related Work

C.Y. Wan, S.B. Eisenman and A. T. Campbell [8] proposed Congestion Detection and Avoidance (CODA). CODA is a congestion control protocol that uses end-to-end acknowledgment based scheme for dealing with permanent congestion and open-loop backpressure strategy for resolving temporary congestion. CODA uses "current buffer occupancy" and information based on "present and past channel loading conditions" to detect congestion. On detecting congestion a node broadcasts backpressure messages. The source regulates itself if the rate of event at the source is smaller than some part of the maximum productivity of the link. However when the rate is increased above this level, then it may lead to congestion. The closed-loop congestion control is used in this situation. The source needs a continuous feedback from the sink. For persistent congestion, CODA needs explicit feedback from sink. When the acknowledgment does not arrive then the source reduces its sending rate.

L. Q. Tao and F. Q. Yu [9] recommended Enhanced Congestion Detection and Avoidance (ECODA) protocol for wireless sensor networks that achieves fairness for different classes of traffic. ECODA allows cross-layer optimization. It uses two different thresholds and weighted buffer difference to detect congestion. Weighted Buffer Difference is used to decide which node has more relevant and crucial data to send. Temporary and permanent congestion are controlled in different ways. Sending implicit notification hop-by-hop in the backward direction is used to control temporary congestion. For controlling persistent congestion the congested node based traffic rate control is used. The traffic is balanced on multiple paths. There is no need of explicit acknowledgment from the sink. Hence energy wastage, bandwidth wastage and false congestion signals are minimized.

Chonggang Wang, Kazem Sohraby, Victor Lawrence, Bo Li and Yueming Hu [10] suggested Priority based congestion control protocol (PCCP) which is designed to control congestion in the upstream direction. It progresses hop-by-hop to control congestion. It reduces loss of data packets and at the same time guarantees fairness. It supports both single path and multi-path routing. It makes use of a congestion degree that is given by the ratio of packet service time to inter-arrival time of the packets. PCCP attempts to lower the occupancy of the buffer. PCCP uses Intelligent Congestion Detection (ICD), Implicit Congestion Notification (ICN) and Priority Based Rate Adjustment (PRA). ICD measures the degree of congestion which is the ratio of packet service time and packet inter-arrival time. The congestion details are piggybacked in the header of data packets during the ICN. The PRA adjusts the scheduling and source data sending rate according to the priority of the sensor node after receiving the congestion notification.

Xiaoyan Yin, Rongsheng Huang proposed Fairness Aware Congestion Control Scheme (FACC) [11] which classifies all the intermediary nodes into "near--source nodes" and "near-sink nodes". All the "near-source nodes" maintain a state for each flow. A nearly reasonable

rate is allotted to each flow. However, the "near-sink nodes" do not need to keep any per-flow state. The near sink nodes use "lightweight probabilistic" method depending on unoccupied buffer length and hit frequency to drop packets. To measure the network utilization and congestion level a "channel busyness ratio" is defined. It is calculated as the ratio of time intervals when the channel is busy because of a successful communication or collision to the entire amount of time. Resource allocation of the left over resources is done. If the packet is dropped then it becomes important to reduce the rate of that particular flow as it denotes that the particular flow is being sent at a high rate. For this warning messages and control messages are used.

J. Kang, Y. Zhang, and B. Nath recommended a Topology-Aware Resource Adaptation (TARA) [12] protocol which is a resource control method. The basic idea used in TARA is a "capacity analysis model", which is used to determine the capacity of different topologies. It makes use of spare nodes referred as the backup nodes. They are kept in sleep mode to save resources during a dormant state. Whenever congestion is detected it switches on the sleeping nodes to create new network topology that has just sufficient capacity to manage the growing traffic. It measures both the buffer occupancy and the channel load to detect congestion. When the congestion level exceeds the threshold, the node announces congestion and turns into a hot spot. Then at once two chief nodes are discovered: the distributor and the merger. The alternative path starting at the distributor node and ending at the merger node is formed.

Sergiou, V. Vassiliou and A. Paphitis suggested a Hierarchical Tree Alternative Path (HTAP) [13] which creates dynamic alternative paths to alleviate congestion in WSN. HTAP makes use of a variation of the Local Minimum Spanning Tree for monitoring the network topology [14]. Each node runs lightweight congestion detection algorithm. When the buffer reaches a threshold value, the congestion detection algorithm starts observing and counting the rate of arrival of packets at the node. If the ratio of the total receiving rate to the maximum transmission rate crosses a certain threshold, then the node sends a feedback message to the nodes that sends packets through it to find a different route. It makes use of an adaptive congestion threshold to avoid transient congestion. It produces substitute paths to the base station to deal with permanent congestion [13].

C. Sergiou and V. Vassiliou proposed a Dynamic Alternative Path Selection (DAIPaS) [15] algorithm for alleviating congestion is a performance aware, dynamic and distributed congestion control scheme in WSN. To select a different routing path in order to escape congested nodes it considers a number of factors such as the unoccupied buffer space, the remaining power of the nodes, the medium interference and the distance of the node from the sink. DAIPaS has two stages: a soft and a hard stage. A node enters into a soft stage once it obtains packets from more than one flow. The transmitting node is informed and advised that it should further transmit its packets using a next proper path starting from a different node. If these nodes continue sending the packets then the receiving nodes must continue receiving and forwarding. In the hard stage the network forces the flows to find another route on exceeding the performance threshold. The node is temporarily or permanently not able to receive any more traffic.

3. Design Details of the Hybrid Algorithm

The proposed methodology is a combination of both traffic rate control and resource control technique. It also acts as a combination of congestion control and congestion avoidance algorithm. Traffic rate control or rate adjustment techniques offer to alleviate congestion by adjusting the traffic rate. Resource control methods manage and increases the resources during formation of bottleneck in the network. A hybrid algorithm may give the benefit of both the types of algorithms. During the time of generation of high priority packets

an attempt is made to increase the resources so that greater number of packets with high and essentially significant content is sent. When congestion occurs during high priority packet generation then along with increasing the resources, traffic rate reduction is also done. During the generation of the low priority packets an attempt is made to save the resources for future important events. For congestion control during the generation of low priority packets only rate control method is used. Once the source starts sensing, it creates the packets and sends them in the forward direction. If the packets generated are during the time of occurrence of important events then these high priority packets are distributed using multiple paths. High priority packets are sent using all the paths available so that more and more high priority packets reach the base station. Low priority packets are sent through a single path saving the resources for future events that may occur. This also increases the lifetime of the network. When the packets reach the forwarding node they compare the service time of the packets to the inter-arrival time of the packets to decide if a control packet is to be sent to the downstream nodes.

3.1. Rate control

The proposed solution uses rate control for congestion control during the generation of both high priority and low priority packets. Rate control method slows down the rate of the traffic flow. The algorithm measures the inter-arrival time of packets and service time of the packets and makes a comparison between them. If the service time of the packets is lesser than the inter-arrival time then it implies that no congestion is present in the node and hence no rate adjustment is needed in the downstream nodes. But when the service time of packets is greater than the inter-arrival time of the packets then congestion is bound to happen in that node. In this case the node sends a feedback or control message backwards indicating the congestion degree. Congestion degree is defined as the ratio of the service time to inter-arrival time of packets given by equation (1).

$$\text{Congestion degree} = \text{packet service time} / \text{packet inter-arrival time} \quad (1)$$

On receiving the control packets the downstream nodes must adjust their packet sending rates. For doing this, the inter-arrival time is multiplied by the congestion degree for determining the next scheduling time of packets. Equation (2) gives the scheduling time of the packets sent from the source.

$$\text{Scheduling time} = \text{current time} + (\text{inter-arrival time} * \text{congestion degree}) \quad (2)$$

3.2. Resource Control

The proposed methodology uses resource control method to increase the resources in case of the generation of high priority packets. For the flow of high priority packets during event generation it tends to increase the number of routes used to reach the base station and distributes the packets through multiple paths. Either a single path can be used for forwarding data to the next node or multiple paths can be used to forward the data leading to load balancing and balance of energy consumption over the entire network. This reduces the chances of bottleneck formation. Hence it is a good method to avoid congestion and the chances of packets being buffered grows.

Algorithm running at the source

```
If (jobcounter<numJobs && energy>=pte)
    Createthedatapacket();
    datapacket→setKind(0),datapacket→setPriority();
if (controlmessagereceived())
```

```
energy=energy-cre;
if(job→getPriority()>5)
    de=ctrlpacket→getDegree();
    if(energy>=n*pte)
Schedule At (simTime() + inter-arrival time*de);
    send the data packets through n possible paths;
    jobcounter+=n,energy = energy - n*pte;
else
    energy=energy-cre; de=j→getDegree();
if (energy>=pte)
scheduleAt(simTime() + inter-arrival time*de);
    send the data packet through a single path;
    jobcounter++, energy = energy - pte;
else
if (datapacket→getPriority()>5)
if (energy >= n*pte)
    scheduleAt(simTime() + inter-arrival time);
send the data packet through n possible path;
    jobcounter+=n, energy = energy - n*pte;
else
if (energy >= pte)
scheduleAt(simTime() + inter-arrival time);
    send the data packet through a single path;
    jobcounter++, energy = energy - pte;
```

Algorithm running at the intermediate nodes

```
Receive and send the data packet to the upstream nodes;
Energy=energy - pre;
sertime=datapacket→getTotalServiceTime();
createcontrolpacket();
if (sertime > inter-arrival time)
    ctrlpacket→setKind(1);
    deg=sertime/inter-arrival time;
    ctrlpacket→setDegree(deg);
    Send the control packet to n downstream nodes;
energy = energy - n*cte;
```

where,

```
cte = energy consumed on control packet transmission
cre = energy consumed on receiving control packet
pte = energy consumed on data packet transmission
pre = energy consumed on receiving data packet
```

4. Implementation

The simulation is done in OMNET++. It is assumed that the network is static in nature. It has also been assumed that the packets lost are not retransmitted and communication takes place in only one direction that is from the source to the sink. Control packet length is assumed to be 1 byte and data packet is of 2 bytes. Transmit energy is $0.720\mu\text{J}$ for control

packet and 1.44 μ J for data packet. Receiving a control packet requires 0.4 μ J and receiving a data packet requires 0.8 μ J. Transmitting range is taken as 20m.

5. Simulation Results

The hybrid algorithm works for various set of parameters a few of which are given in table 1. These parameters are only serving as an example from a wide range of parameters. Table 1 enumerates three set of parameters for which the result is shown as a graph in Figure 1 to Figure 6. Figure 1-Figure 3 illustrates the number of packets received at the sink after using hybrid algorithm as compared to No Congestion Control (NCC) and backpressure algorithm. Figure 4-Figure 6 illustrates lifetime of the network after using hybrid algorithm as compared to NCC and backpressure algorithm.

Table 1. Initial Values of Various Parameters Used

	Parameter Set 1	Parameter Set 2	Parameter Set 3
Source1.numJobs	100	50	150
Source2.numJobs	10	10	20
Source3.numJobs	10	10	20
Source1.numJobs	100	50	150
Service Time	4s	4s	4s
Queue2.capacity	30	30	30
Queue1.capacity	30	30	30
Queue3.capacity	100	100	100
Queue4.capacity	60	60	80
InterArrivalTime	2s	2s	2s
Queue2.Energy	30 μ J	30 μ J	30 μ J
Queue1.Energy	30 μ J	30 μ J	30 μ J
Queue3.Energy	500 μ J	300 μ J	400 μ J
Queue4.Energy	600 μ J	300 μ J	300 μ J
Source1.Energy	500 μ J	300 μ J	300 μ J
Source2.Energy	500 μ J	300 μ J	300 μ J
Source3.Energy	500 μ J	300 μ J	300 μ J
Source4.Energy	500 μ J	300 μ J	300 μ J

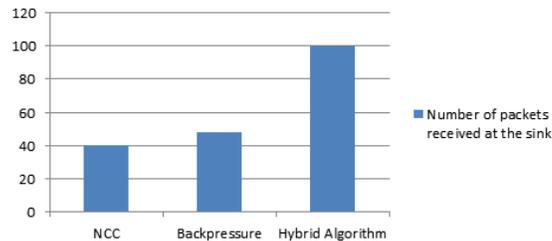


Figure 1. Graph Comparing Number of Packets Received at the Sink using NCC, Backpressure and Hybrid Algorithm using Parameter Set 1

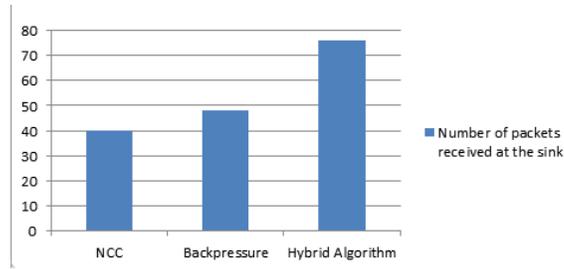


Figure 2. Graph Comparing Number of Packets Received at the Sink using NCC, Backpressure and Hybrid Algorithm using Parameter Set 2

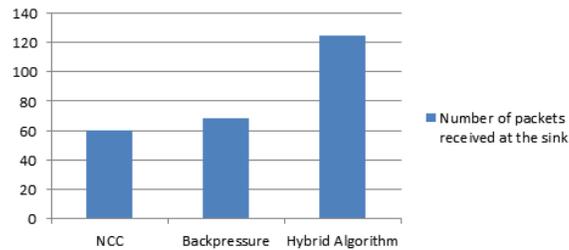


Figure 3. Graph Comparing Number of Packets Received at the Sink using NCC, Backpressure and Hybrid Algorithm using Parameter Set 3

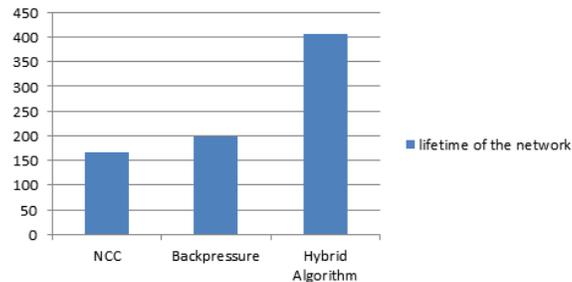


Figure 4. Graph Comparing the Lifetime of the Network in Seconds using NCC, Backpressure and Hybrid Algorithm using Parameter Set 1

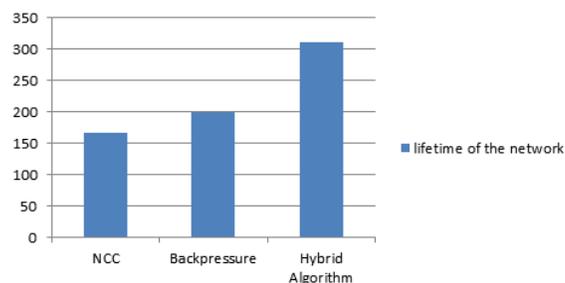


Figure 5. Graph Comparing the Lifetime of the Network in Seconds using NCC, Backpressure and Hybrid Algorithm using Parameter Set 2

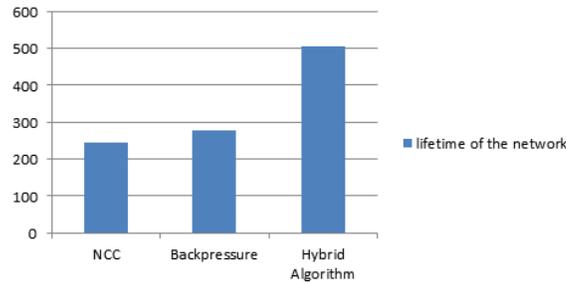


Figure 6. Graph Comparing the Lifetime of the Network in Seconds using NCC, Backpressure and Hybrid Algorithm using Parameter Set 3

6. Analysis and Comparison

No Congestion Control (NCC) keeps sending the data packets at the same rate without taking into consideration the congestion that is taking place in the upstream direction. It does not impose any congestion alleviation method to stop the dropping of the packets. Backpressure algorithm on the other hand uses rate control algorithm to alleviate congestion for both the low priority and high priority packets. It uses congestion degree to reduce the rate of sending packets. Congestion degree is defined as the ratio of packet service time to packet inter-arrival time of the packets at the intermediate nodes. The packets at the source nodes are sent after increasing the time between sending consecutive packets by a factor of congestion degree. It deals with the high priority and low priority packets in the same manner. On the other hand the hybrid algorithm handles the high priority packets and the low priority packets differently. The hybrid algorithm uses backpressure algorithm alone for low priority packets. It uses multiple paths along with backpressure for sending high priority packets.

High priority packets are sent using multiple paths to avoid congestion at a particular node. Also at the time of congestion it sends the high priority packets with the increase in the inter-sending time by a factor of congestion degree. Only the backpressure algorithm is used for the low priority packets. This saves the resources for the high priority packets that may be generated when important events may occur in the future. Although the hybrid algorithm may reduce the number of low priority packets that are delivered to the base station but it increases the percentage of high priority packets. The hybrid algorithm reduces the percentage of packet drops and in a way makes an effort to balance the energy usage uniformly over the entire network. This helps in a manner that it does not let the network be cut into parts and increases the number of packets reaching the base station by avoiding congestion.

The hybrid algorithm is a combination of proactive and reactive way of dealing with congestion, thus reducing the probability future congestion. The hybrid algorithm also increases the lifetime of the network. Also the Hybrid algorithm makes best use of the resources in terms of buffer capacity and energy of those nodes.

7. Conclusion

Congestion is a crucial factor that affects the performance of a wireless sensor network. The Hybrid congestion control algorithm increases the total number of packets that are finally delivered to the base station thus increasing the throughput. Along with this it also increases the lifetime of the network and reduces the number of dropped packets. The percentage of the higher priority packets received at the base station also increases. The hybrid congestion control algorithm increases the number of higher priority packets received at the base station.

It can be improved if the number of control packets sent is reduced. If the application requires the packets to be sent in both the directions then implicit congestion notification can be used to further combat energy loss due to the sending of the control packets. Different congestion control protocols must be designed and developed taking into consideration different types of environment, working conditions, fairness and energy usage. Also methods can be developed considering multiple sinks. A balance between stability and fairness must also be kept in mind during the development of congestion alleviating methods [16].

References

- [1] J. Stanley and T. Ozkan-Haller, "Applying video sensor networks to nearshore environment monitoring," *IEEE Pervasive Computing*, (2003), pp. 14-21.
- [2] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J.Sallai and K. Frampton, "Sensor network-based countersniper system," *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (Sensys)*, Baltimore, MD, (2004).
- [3] M. Castillo-Effen, D. Quintela, R. Jordan, W. Westho and W. Moreno, "Wireless sensor networks for flash-flood alerting," *Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits and Systems*, Dominican Republic, (2004).
- [4] B. Brooks, P. Ramanathan and A. Sayeed, "Distributed target classification and tracking in sensor networks," *Proc. IEEE*, vol. 91, (2003) August, pp. 1163-1171.
- [5] L. Schwiebert, S. Gupta and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," *ACM MobiCom*, Rome, Italy, (2001) July, pp. 151-165.
- [6] A. Cerpa, J. Elson, M. Hamilton and J. Zhao, "Habitat monitoring: application drive for wireless communications technology," *Proc. ACM SIGCOMM Workshop Data Commun. Latin Amer, Caribbean*, San Jose, Costa Rica, (2001) April.
- [7] D. Flora, V. Kavitha and M. Muthuselvi, "A survey on congestion control techniques in wireless sensor networks," *Proceedings of ICETECT*, (2011).
- [8] C. Wan, S. Eisenman and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks," in *Proc. ACM SenSys*, (2003) November.
- [9] L. Tao and F. Yu, "ECODA: Enhanced congestion detection and avoidance for multiple class of traffic in sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, (2010) August.
- [10] C. Wang, K. Sohraby, V. Lawrence, B. Li and Y. Hu, "Priority-based congestion control in wireless sensor networks", *The IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)* 0-7695-2553-9/06, (2006).
- [11] X. Yin and R. Huang, "A fairness-aware congestion control scheme in wireless sensor networks," *IEEE Transactions on vehicular technology*, vol. 58, no. 9, (2009) November, pp. 5225-5234.
- [12] J. Kang, Y. Zhang and B. Nath, "TARA: Topology-aware resource adaptation to alleviate congestion in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, (2007), pp. 919-931.
- [13] S. V. Vassiliou and A. Paphitis, "Hierarchical tree alternative path (HTAP) algorithm for congestion control in wireless sensor networks," *Ad Hoc Netw.*, vol. 11, no. 1.
- [14] N. Li, J. Hou, and L. Sha, "Design and analysis of an mst-based topology control algorithm," *INFOCOM, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, IEEE Societies, vol. 3, (2003).
- [15] Sergiou and V. Vassiliou, "DAIPaS: A performance aware congestion control algorithm in wireless sensor networks," *18th International Conference on Telecommunications (ICT)*, (2011) May, pp. 167-173.
- [16] J. Zhao, L. Wang, S. Li, X. Liu, Z. Yuan and Z. Gao, "A survey of congestion control mechanisms in wireless sensor networks," *Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, (2010) October, pp. 719-722.

