A QoS Adaptive Congestion Control in Wireless Sensor Network

Md. Obaidur Rahman, Muhammad Mostafa Monowar and Choong Seon Hong
Networking Lab, Department of Computer Engineering, Kyung Hee University, South Korea.
E-mail: {rupam, monowar} @networking.khu.ac.kr and cshong@khu.ac.kr

Abstract* — In wireless sensor networks congestion occurs in intermediate nodes while data packets travel from source to sink. Congestion causes packet loss which in turn drastically decreases network performance and throughput. As sensors are energy constraint so it is a decisive task to detect the congestion and congested regions in a network to perform congestion control. In addition to that different application i.e. real time and non-real time data in sensor network have different QoS (delay, link utilization, and packet loss) guarantee requirement. In this paper we proposed a new QoS adaptive cross-layer approach to control the congestion and support QoS guarantee for different application data in sensor network. This approach maintains two congestion control algorithm to control namely short-term and long-term congestion. To ensure real time and non-real time data flow, hop-by-hop QoS aware scheduling and QoS distributed MAC Manager are considered. The experimental outputs of this work are able to show that proposed scheme gives guaranteed QoS for different application data and gives a noticeable performance in terms of energy analysis and lifetime of the network.

***Keywords* — Congestion Control, Quality of Service (QoS), Real time data, Non-real time data.**

1. Introduction

In wireless sensor networks, handling different application data (i.e. real time and non-real time data) is a major challenge among the recent research tracks. Especially this is a complex task while providing real time service in sensor network with some QoS metrics such as packet loss, delay, link utilization. Due to shared nature of the medium all nodes in the sensor network contends for medium access. Thus observing congestion is normal and its impact on network performance: a drastic decrease in throughput and network life time. Moreover as different applications are becoming part of the sensor and it includes multimedia applications, sensing applications, file transfer etc., therefore solutions to reduce the congestion in wireless sensor networks are obligatory.

Congestion generally occurs from buffer overflow due to channel contention of sensor nodes. Two types of congestion namely: short-term and long-term are frequently occurs in the intermediate nodes between source to sink. In a multi hop communication pattern, intermediate nodes of a network carry disproportionately large amount of the traffic. Thus shortage of buffer space appeared if the nodes can not get sufficient access to the wireless medium and it radically consumes huge amount of energy as well as causes packet loss and delay. In general three steps are to be followed while designing any congestion control protocol. These are congestion detection, congestion notification and congestion control. Besides, maintaining traditional QoS metrics for different application data are also necessary.

In this paper, we proposed a new QoS adaptive cross-layer congestion control mechanism for real time and non-real time data flow. More priority is given to real time data than the non-real time data in terms of delay constraint and available link capacity. Nevertheless for non-real time data, reliability is considered using a novel queue based approach. Like PCCP [2] this paper investigates the problem of upstream congestion control in WSN.

The rest of this paper is organized as follows: section 2 presents several background studies on congestion control and avoidance techniques. Subsequently section 3 describes the system models and assumptions. Section 4 represents few design considerations like; end-to-end delay, reliability assurance. Proposed congestion control protocol is demonstrated in section 5 and section 6 has gone through some experimental validations with simulation efforts. Finally section 7 concludes this paper with few future directions.

2. Related Work

In this section, some background studies on various aspects of congestion, its effects & control techniques are explained.

CODA (Congestion Detection and Avoidance) [4] belongs to upstream congestion control. CODA attempts to detect congestion by monitoring current buffer occupancy and wireless channel load. But the upstream neighbor nodes will decrease output rate in such a way like AIMD and replay backpressure continuously, after they receive backpressure signal.

SenTCP [9] is an open-loop hop-by-hop congestion control with few special features. It jointly uses average local packet service time and average local packet inter-arrival time to estimate current local congestion degree in each intermediate node and during congestion it uses hop-by-hop congestion control.

Fusion [5] studies three congestion control techniques namely; hop-by-by flow control, source limiting scheme and

"This research was supported by the MIC under the ITRC support program supervised by the IITA" (IITA-2006-(C1090-0602-0002). Dr. Hong is corresponding author.

prioritized MAC, operate at different layers of the traditional protocol stack.

CCF (Congestion Control and Fairness) [3] uses packet service time to deduce the available service rate and detects congestion. Each sensor node uses rate adjustment based on its available service rate and number of child nodes. This proposal claims simple fairness for all nodes with same throughput. Guaranteed fairness can only be maintained while each node gets same priority. But probability of existence of inactive child node with no traffic could lead to low local link utilization.

In ESRT [11], a sensor sets a congestion-notification (CN) bit in the packet header if its buffer is about to full. The sink periodically computes a new reporting rate (at which each source is supposed to report data) based on a reliability measurement, the received CN bits, and the previous reporting rate.

A node priority based congestion control mechanism PCCP [2] is proposed for sensor network. It introduces an efficient and intelligent congestion detection technique.

Siphon [6] is also a congestion mitigation scheme which detects congestion using queue length occupancy. But instead of using any rate adjustment technique it uses traffic redirection to mitigate congestion.

None of the previous congestion control scheme handled multiple application data with their requirements. In this paper we address this issue.

3. System Model & Assumptions

3.1 Node Model

According to our assumptions, the protocol layering stacks are depicted in the node model described in Figure 1. In network layer we have considered two active queues: Q_{RT} and Q_{NRT} for real time and non-real time application data respectively and one back-up queue $Q_{BACK\text{-}UP}$ for storing unacknowledged non-real time data. In case of $Q_{BACK\text{-}UP}$, it removes the non-real time data packet after getting the NACK message from its parent node. A classifier is used to classify both real time and non-real time data and a scheduler is used to send data packet to MAC layer according to their weighted priority. The scheduler schedules the queues according to the application priority. Thus it decides the service order of data packets in the queues. Let, η is the scheduling rate and if $\alpha_i = \{\alpha_{RT}, \alpha_{NRT}\}$ is the weighted priority level of real time and non-real time data then in each unit time the scheduler sends the following number of data packets from network layer to MAC layer:

$$\frac{\eta * \alpha_i}{\alpha_{RT} + \alpha_{NRT}} \tag{1}$$

Each upstream nodes store the possible routes to next hop node's id in *Next_Hop_Parent_Set*, P. We have used a QoS distributed MAC manager which measures the QoS requirement of each data packet (i.e. minimum delay requirement, minimum service rate, reliability level etc.) and send out the data packet to the appropriate next hop node via physical medium.

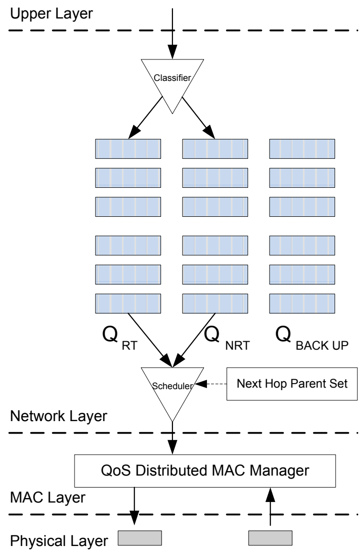

Figure 1: Node Model

3.2 Network Model

In [1], authors have shown that, network performance always improves with the increasing number of alternate routes, as expected. But the incremental improvement is very small for more than three alternative routes of each node, except when the paths are of the same length, but unfortunately it is very unlikely to occur in sensor network. Thus, in our hop-by-hop network model in Figure 2 it is considered one primary route and at least one alternative route towards the sink from each sensor node. The reason behind this idea is very much practical for wireless sensor network where notion of primary route ensures less use of network resources. Also as always sink node surrounded by a large number of sensor nodes so there will be an adequate number of alternate routes up to sink. Each sensor maintains a *Next_Hop_Parent_Set*, P. In the figure as source node has one primary route and two alternate routes, so cardinality of *Next_Hop_Parent_Set* for source node is |P| = 3.

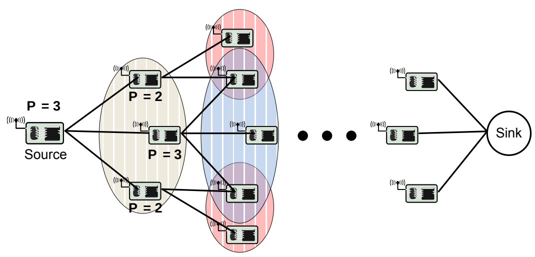

Figure 2: Illustration of Basic Hop-by-Hop Network Model

4. Design Considerations

4.1 End-to-End Delay

As real time applications have strict QoS requirement on the packet delay, therefore packet retransmission is not allowed in our protocol. In case of congestion in an intermediate node, sensor network supposes to maintain a continuous flow of real time data as they got higher priority in the network. In our protocol while congestion occurs in an intermediate node continuous flow is maintained for real time data via least delay-constrained next hope node. Least delay constraint feasible next hop nodes are to be recognized by a child node based on end-to-end delay of their primary route. Thus we estimated end-to-end delay constraint of each node *i*'s primary route considering the following parameters:

Table 1: Parameters Considered

End-to-end queuing delay up to sink	D_q
End-to-end propagation delay up to sink	Dp
Real time data rate	λ^{RT}
Queuing delay for real time data	D_q^{RT}
Service rate for real time data	r_s^{RT}

So, for real time data the total queuing delay on node *i* is:

$$D_q^{RT} = \lambda^{RT} / r_s^{RT} \tag{2}$$

Thus the end-to-end queuing delay up to sink for the primary path of node *i*:

$$\begin{aligned} D_q &= \sum_{i,\sin k \in path} D_q^{RT} \\ &= \sum_{i,\sin k \in path} \lambda^{RT} / r_s^{RT} \end{aligned} \tag{3}$$

Again, the end-to-end propagation delay up to sink for the primary path of node *i*:

$$\begin{aligned} & D_{end-end} = D_q + D_p \\ \Rightarrow & \sum_{i,\sin k \in path} \lambda^{RT} / r_s^{RT} + \sum_{i,\sin k \in path} C \times Dist_{i,\sin k} \end{aligned} \tag{4}$$

4.2 Reliability Assurance

Based on the criticality of different application data inside a packet, different priority levels are assigned for a data packet. Each priority levels maps to a desired reliability for data delivery. In WSN reliability can not be treated as a single issue for QoS concern [7], therefore we tried to assure the reliability of real time data by other QoS parameter like packet loss, packet delay, energy per packet. But in case of non-real time data we maintained a desired reliability r from source node to sink node separated by *h* hops.

Non-real time packets are routed through the primary route of the source node. We used sequence number in the header of each data packet and copy of each unacknowledged non-real time data packets are stored in $Q_{BACK\text{-}UP}$ queue. Upon receiving a negative acknowledgement (NACK) a node discards the copy of the packet and retransmits the lost packets.

Using the hop-by-hop methods, it is required $r_i = r^{1/h}$ reliability at each hop so that on a primary route from node *i* the total reliability can be $\Pi\, r_i = r$. Again the presence of error probability will reduce the service rate at MAC layer causing retransmission of non-real time data packet. Thus we consider hop-by-hop error probability is *e* and try to find out the impact of retransmitting packet due to high error probability. Let, probability that at least one of the copies of the non-real time packet will be received at the next hope node should be $r^{1/h}$, which in turn reflects the hop-by-hop reliability of that packet. Therefore,

$$\begin{aligned} & r^{1/h} = 1 - e^{E[P]} \\ \Rightarrow & E[P] = \frac{\log(1 - r^{1/h})}{\log(e)} \end{aligned} \tag{5}$$

Where, *E[P]* is the expected number of copies of a non-real time packet necessary to retransmit to meet the hop-by-hop reliability of $r^{1/h}$.

4.3 Node Price Estimation

As sensor nodes are resource constraint so there is some limitation in case of capacity of wireless sensor network. Therefore, achieving best utilization of the resources can maximize the network capacity. To meet this goal we estimate each node *i*'s weighted price denoted by ϕ_i . Obviously node price is determined by considering the downstream condition of a network. It includes end-to-end delay from a node *i* up to sink, reliability assurance in terms of expected number of retransmission at each hop. Moreover channel error probability and residual energy are also under consideration to estimate node price.

5. Proposed Congestion Control Mechanism

5.1 Congestion Detection

As it is said earlier generally three steps are to be followed while designing congestion control. These are congestion detection, congestion notification and congestion control. Thus, to detect congestion, we proposed a congestion detection mechanism which is close to the previous work PCCP [2]. In order to measure local congestion level at each intermediate node PCCP proposes intelligent congestion detection that detects congestion based on the ratio between packet inter-arrival (t^i_a) time and packet service times (t^i_s) at the MAC layer. But we define two new metrics: packet inter-arrival rate (r^i_a) and packet service rate (r^i_s) at MAC layer. To control the congestion precisely by adapting rate adjustment this two new metrics give more flexibility in our proposal than that of PCCP. Packet service rate is determined by the processing time per packet [8] at the MAC layer which includes the time interval between the time a packet arrives at MAC layer and the time it successfully departed MAC layer. This MAC layer processing also covers packet waiting time, collision resolution and packet transmission time at MAC layer. Based on this two new metrics we measured an index naming *Congestion Scale; S(i)* is defined as the ratio between mean packet service rate and mean packet inter-arrival rate. This Congestion Scale is calculated over a pre-specified timer interval in each sensor node *i* by the following equation:

$$S(i) = r_s^i / r_a^i \tag{6}$$

As calculation of congestion scale is a continuous process, therefore each node periodically calculates its own mean packet inter-arrival rate and mean packet service rate.

According to queuing theory, let A_t be the number of arrivals of the packet at the MAC layer of a node from time zero up to time *t*. An intuitive definition of the arrival rate λ is:

$$\lambda = \lim_{t\to\infty} \frac{A_t}{t} \tag{7}$$

Now suppose t_1, t_2, t_3 are the length of time interval between arrivals. The arrival rate up to the time when k-th packet arrived at MAC layer is:

$$\frac{k}{t_1 + t_2 + t_3 \cdots + t_k} \tag{8}$$

The average inter-arrival time (time between intervals) up to the time when k-th packet arrives is:

$$\frac{t_1 + t_2 + t_3 \cdots + t_k}{k} \tag{9}$$

Therefore using equation 7, 8 and 9 we get:

$$\lambda = \lim_{k\to\infty} \frac{k}{\sum_{i=1}^{k} t_k} \Rightarrow \lim_{k\to\infty} \frac{1}{\sum_{i=1}^{k} t_k / k} = \frac{1}{\lambda^{-1}} \tag{10}$$

Shows that, the mean inter-arrival rate r^i_a is λ^{-1}. Each node periodically calculates the mean inter-arrival rate using the equation 10.

Again Packet service rate is updated using the Exponential Weighted Moving Average (EWMA) formula:

$$r_s^i = (1 - w) * prev(r_s^i) + w * (1/inst(t_s^i)) \tag{11}$$

Here *w* is a constant in the range between 0 and 1 and inst(t^i_s) is the instantaneous service time of the packet just transmitted prev(r^i_s) is the immediate service rate.

When the mean packet inter-arrival rate gets larger than the mean packet service rate then the queue length increases and this criterion can be used for detecting congestion. Thus, in proposed mechanism the following three cases can be considered for indicating congestion state:

$$\begin{array}{ll} Case\,1: & if\ (S(i) > 1),\quad No\ Queuing \\ Case\,2: & if\ (S(i) < 1),\quad Queuing \\ Case\,3: & Otherwise,\quad No\ Queuing\ as\ r_a^i = r_s^i \end{array}$$

The *Congestion Scale* in a node can deliberately replicate the congestion level of itself. Queuing up of packets also indicates link level collisions. Thus Congestion scale is an effective measure of detecting both node level and link level congestion.

5.2 Congestion Notification

To notify upstream nodes regarding the congestion status, we assumed, ICN (Implicit Congestion Notification), avoiding the additional control messages.

5.3 Mitigating Congestion

In order to mitigate congestion we proposed two congestion control mechanism based on the above mentioned network model where primary routes and alternative routes have already been established between source to sink. The underlying assumption of our proposal is that, interference among the alternative paths and primary path is negligible.

- **Short Term Congestion Control**

In wireless sensor network with multi hop communication, congestion usually occurs in the intermediate nodes of the network. Fortunately, the connectivity of the wireless sensor networks is rich enough to allow routing the packets on alternate routes that avoid the congested areas. To remove the short-term congestion, the idea of our proposed algorithm is straightforward. When a node experiences congestion, its immediate child node split the real time traffic on to its alternate parent (route) in a proportion to their weight factor w_i. This weight factor w_i is solely depends on the end-to-end path delay from the alternative parent (route) node to the sink. This approach will eventually carry the newly created real time data flows at a slower rate along the primary route, allowing the congested node to be relieved and thus alleviate congestion.

- **Long Term Congestion Control**

In our proposed protocol we have considered short-term congestion control as a temporary solution to avoid the transient congestion. The assumption is quite practical when short-term congestion control can not successfully support for congestion avoidance in intermediate nodes due to huge flow of traffic. Therefore intermediate nodes on a primary route periodically send the ICN as a backpressure message as an indication of congestion. If the back pressure message reaches the source node then upon receiving that message, source node initiates long-term congestion control. Source node sends proportionate real time traffic as the similar way of short-term congestion control along its alternate routes (parents) based on their weight factor w_i. Moreover, source node dynamically adjusts to the changing conditions and selects the best node (parent) as its primary route to send further packets. As a result subsequently both the real time and non-real time data flows will follow the changed or updated primary route.

6. Experimental Results

We have performed extensive simulations to evaluate the performance of our proposed scheme. We also study the over all network throughput, average queue occupancy and energy impact of the proposed protocol.

Static single path routing is used as the routing protocol in our simulations. The simulation parameters are described as follows: 50 sensors are randomly deployed in 1000 x 1000m^2 sensor fields. The maximum transmission rate is considered as 512 kbps for each sensor node. Initially due to limited energy supply, we consider a sensor with a maximum sustainable rate of 10 packets per second and it may generate at a lower rate due to rate control over time. Each data packet size is 30 bytes. Each data source generates new data at an initial rate of 4 packets per second.

In simulations, we only use the CSMA MAC [10] protocol to ensure even access opportunities for each sensor node. We also assume that a node is within the range of its parent and children. The following congestion control schemes are implemented to deduce the comparison with proposed protocol:

- **Proposed Scheme:** Scheme proposed in this paper.
- **Congestion Control and Fairness (CCF)** [3]**:** This scheme controls congestion in a hop-by-hop manner and each node uses rate adjustment based on its available service rate and number of child node.
- **No Congestion Control (NCC):** It is the scheme where no action is taken in to account while congestion occurs in any intermediate nodes.

6.1 Impact on Average Queue Length

Simulated dynamics of in Figure 3 represents one sensor node's stable queue occupancy state with respect to simulation time. The average queue occupancy remains below 2 during 0 to 7 seconds and gradually it fluctuates with in the same range. Figure 3 shows few exceptions for the average queue length during the simulation time; in between 25 to 40 seconds. Actually this exception occurs due to heavy load and thus incurs high level of queue occupancy. As the initial source rate increases over time therefore high average queue occupancy is logistic and it becomes stable afterwards (40 to 60 seconds). Simulated results also show that, queues of other nodes have very similar trends under the same network scenario.

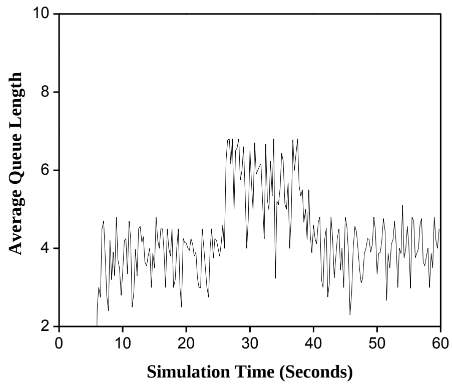

Figure 3: Average Queue Occupancy over Time

6.2 Energy Analysis

In what follows, we study the per node average residual energy for the proposed scheme in terms of its energy impact. We trace out the residual energy of each node and at the end of simulation; plot the average residual energy distribution of the network. We also analyzed the simulated result and make a comparison with CCF [3] and no congestion control approach. Figure 4 demonstrate that there is a rapid decrement (a sharp slope) in average residual energy in the network for no congestion control mechanism; where as proposed scheme shows a slow decrement in terms of energy loss. Moreover, proposed scheme gives better energy utilization than CCF; which is very important as it is a rate based congestion control technique. The average residual energy of the proposed approach lies around 75% till the end of simulation. Residual energy is calculated as:

$$Residual\ Energy = (Remaining\ Energy)/\ Initial\ Energy \qquad (12)$$

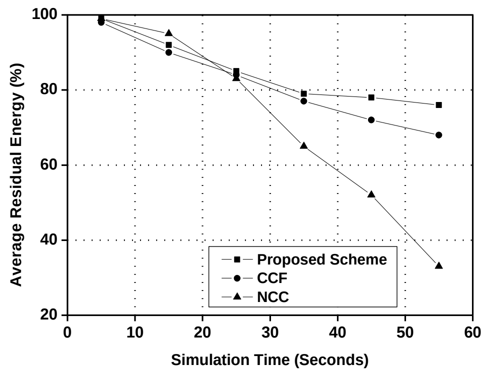

Figure 4: Average Residual Energy of Overall Network

7. Conclusion

In this paper, several properties of sensor networks are studied considering the congestion state in the intermediate nodes. Due to the random access of the medium, queue length varies significantly over time. In sensor networks, a couple of adjacent nodes will have their queues building up when congestion expected to occur. This paper addresses this phenomenon to control the congestion based on the observation of immediate downstream node. Moreover, priority based application are used for both real time and non-real time data using node priority depending weight factor. Implicit piggyback message carrying information of different parameter helped the upstream node in order to take proper actions to avoid congestion. Extensive simulations are done here and it shows better performance than two other existing schemes.

The design consideration of proposed scheme points to some future directions. Scopes of integrating end-to-end reliability, complete probabilistic analysis over its performance are still alive. In near future we will implement the proposal on a real sensor test-bed and compare the test-bed results with simulations.

REFERENCES

[1] A. Nasipuri, R. Castaeda, and S.R. Das, "Performance of Multipath routing for on-demand protocols in mobile ad hoc networks ", Mobile Netw. Appl., vol. 6, no. 4, pp.339-349, Aug. 2001.

[2] C. Wang et al., "Upstream Congestion Control in Wireless Sensor Networks Through Cross-Layer Optimization," IEEE Journal on Selected Areas in Communications, Vol.25, No.4, May 2007.

[3] C.T. Ee and R. Bajcsy, "Congestion control and fairness for Many-to-One routing in sensor networks," in Proc. ACM Sensys, Nov. 2004.

[4] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks," in Proc. ACM SenSys, Nov.2003.

[5] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks," in Proc. ACM Sensys, Nov. 2004.

[6] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: Overload Traffic Management Using Multi-Radio Virtual Sinks in Sensor Networks," in Proc. ACM SenSys, Nov. 2005.

[7] Chen. Dazhi, K. Varshney. Pramod, "QoS Support in Wireless Sensor Networks: A Survey", Proc. of the 2004 International Conference on Wireless Networks (ICWN 2004), Las Vegas, Nevada, USA, June 21-24, 2004.

[8] http://oakham.cs.ucl.ac.uk/pipermail/iccrg/2006-February/000034.html

[9] C. Wang, K. Sohraby, and B. Li, "SenTCP: A hop-by-hop congestion control protocol for wireless sensor networks," in Proceedings of IEEE INFOCOM 2005 (Poster Paper), Miami, Florida, USA, Mar. 2005.

[10] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications"; ANSI/IEEE Std. 802.11 1999 Edition, 1999.

[11] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," Proc. Fourth ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '03), June 2003.