A Hop by Hop Rate Control Based QoS Management for Real Time Traffic in Wireless Sensor Networks*

Muhammad Mostafa Monowar, Md. Obaidur Rahman, Byung Goo Choi, and Choong Seon Hong**

Networking Lab, Department of Computer Engineering, Kyung Hee University
1 Seocheon, Giheung, Yongin, Gyeonggi, 449-701 South Korea
{monowar,rupam}@networking.khu.ac.kr, cshong@khu.ac.kr

Abstract. Wireless Sensor Network typically assimilates various real time applications that must meet some QoS requirements (e.g. delay, jitter, throughput, packet loss) under severe resource limitations. Hence, quality-of-service (QoS) management in Wireless Sensor Network is an important issue yet to be investigated. Due to the high data rate and burst traffic for real time applications, occurrence of congestion is very common. Ensuring the QoS requirements under congested scenario is quite challenging. As an attempt to this direction, in this paper we propose a hop-by-hop dynamic rate control algorithm which controls the congestion as well as ensures that the real time traffic will meet the soft QoS requirements. We have introduced per-hop deadline miss ratio as the congestion indication metric and performed the rate control when this ratio goes above to a certain level. Thus by per-hop rate adjustment, soft QoS is being met locally which in turns meets the end-to-end soft QoS. Finally, simulation has demonstrated the effectiveness of our approach.

Keywords: Congestion, Soft QoS, Per-hop deadline miss ratio, Real time traffic.

1 Introduction

The recent advances in wireless sensor network communication protocols [1] and low power hardware devices such as CMOS camera and microphones have elevated the proliferation of various sensor network applications i.e. battlefield surveillance, disaster and emergency response, environmental monitoring, industrial process control etc. These applications deal with a variety of real time constraints in response to the physical world. In comparison with the traditional distributed systems, real time guarantee for sensor network is more challenging due to the diverse transmission rates, unpredictable spatiotemporal properties of the physical events in the real world and the severe resource limitations in the sensor network.

In WSN, usually tens or thousands of sensor nodes are deployed in a scattered way in an area with single or multiple sinks. Due to the high data rate and unpredictable

* This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD)" (KRF-2006-521-D00394).
** Corresponding author.

Y. Ma, D. Choi, and S. Ata (Eds.): APNOMS 2008, LNCS 5297, pp. 177–186, 2008.

burst traffic nature of the real time application, occurrence of congestion is more likely. Under congested scenario, the end-to-end meeting of deadline of the real time packets is seriously hindered. Hence, the required QoS cannot be ensured. Therefore, an efficient rate control mechanism is necessary that boost up the meeting of QoS requirements even in the congested scenario.

In this paper, we have proposed a QoS management mechanism which performs an efficient dynamic hop by hop rate adjustment to ensure the required QoS guarantee even in the congested scenario. We have used per-hop deadline miss ratio as the QoS metric as well as congestion detection metric. Lowering the per-hop deadline miss ratio, the meeting of end-to-end deadline has been ensured. We have also employed Per-hop Least Entailed Delay First (PLEDF) scheduler for reducing the deadline miss ratio. Here, we have considered the applications that require soft QoS which means it tolerates the delay and deadline misses up to a certain level.

The rest of the paper is organized as follows. Section 2 presents several related works on congestion control for real time traffic. Subsequently section 3 describes the design considerations for our proposed scheme. Section 4 represents our proposed protocol in detail. Section 5 describes the simulation and finally section 6 concludes the paper.

2 Related Works

In the present research train lots of works is going on the rate control for wireless sensor network. But a very few of them considered the QoS issues for the real time packets.

PCCP [2] is a recent congestion control protocol which takes into account the QoS of the multimedia applications. It introduces an efficient congestion detection technique addressing both node and link level congestion but can't guarantee the end-to-end deadline of the real time packets.

DART [3] is another current transport protocol for wireless sensor networks. The protocol simultaneously addresses the congestion control and timely event transport reliability. It is sink initiated protocol which controls congestion by adjusting the reporting rate that incurs extra delay for the sources far away from the sink.

SUPORTS [4] is another real time traffic management protocols for Sensor Network. It is based on traffic regulation and end-to-end scheduling approach which uses hop-by-hop approach for traffic regulation and reject packets which is supposed to miss the deadline.

FLC-QM[5] is proposed as the Fuzzy logic based QoS Management in Wireless Sensor/Actuator (WSAN) Networks. It utilizes a fuzzy logic controller inside each source sensor node to adapt sampling period to the deadline miss ratio associated with data transmission from the sensor to actuator.

Besides these, CODA[6], CCF[7], FUSION[8] etc are the remarkable congestion control for wireless sensor networks but none of them explicitly considered the QoS for real time traffic.

In fact, the QoS issues considered for real time traffic in wireless sensor network mainly focuses on the routing layer solution such as SPEED [9], MMSPEED[10], Energy Aware QoS Routing [11] etc. RAP [12] is proposed to provide real time communication architecture for large scale wireless sensor networks. It utilizes MAC prioritization to provide soft guarantees by locally considering the velocity at a node which can be dynamically readjusted. Hence, the need of an efficient protocol which provides QoS guarantees even in the congested situation motivates us to design our protocol.

3 Design Considerations

This section states the network model, assumptions and node model for our proposed protocol which we have taken into account while constructing our algorithm.

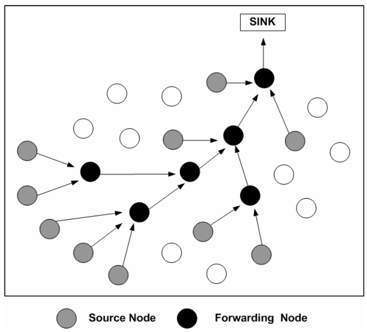

Fig. 1. Network Model

3.1 Network Model and Assumptions

In this paper, we consider the rate control for many-to-one multi hop single path routing. The network model is shown in Fig. 1 where the grey nodes function as a source node and black one as forwarding node. Source nodes transmit the real time packets periodically. All nodes are supposed to use CSMA/CA like MAC protocol. We assume that a predetermined route has been established by any routing protocol and path establishment is out of the scope of this paper. We further assume that the application will assign the deadline for every real time packet while originating. Nodes transmit its data to the downstream direction towards the sink are called child nodes and the node to which children transmits is their parent node.

3.2 Node Model

Fig. 2 depicts the node model for a particular node *i*. We have provisioned a QoS Management Module between the Network and MAC layer which ensures the lower deadline miss ratio with efficient rate adjustment even in the congested situation.

This module contains a Queue, Per-hop Least Entailed Delay First (PLEDF) scheduler, a Delay estimator and a Rate controller. The transit traffic comes from the previous node are placed into a queue after performing the scheduling by the PLEDF scheduler. Scheduling is done for each packet according to their least per-hop entailed delay with the assistance of delay estimator. In each node scheduler drops packets which are not able to meet their per hop deadline. Delay estimator is responsible to provide the information to the scheduler about average waiting time within a node and per hop entailed delay for an incoming packet. The details of measuring the average waiting time and per-hop entailed delay have been presented in the section 4.1. The rate controller carries out the rate adjustment function as described in section 4.2 interacting with the MAC layer and scheduler.

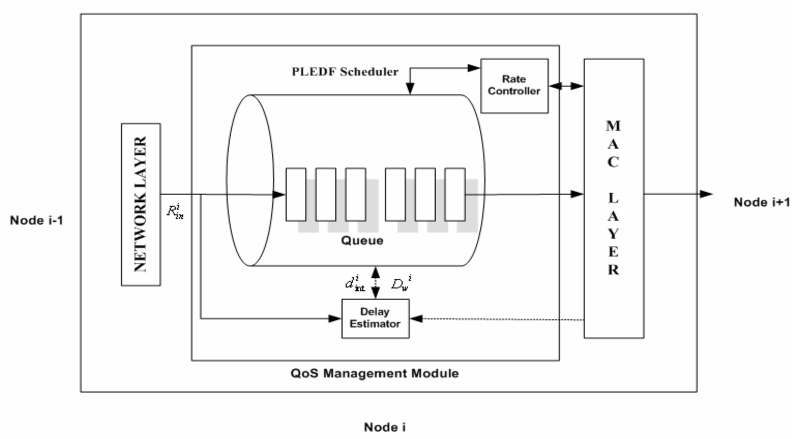

Fig. 2. Node Model

4 Proposed Protocol

Our main motivation in designing the protocol is to perform rate control for real time traffic transited through a single sensor node in such a way that a required number of packets must meet their end-to-end deadline. To accomplish this goal, every node maintains the deadline miss ratio for each of the real time packets passes through it in such a way that the deadline miss ratio will not exceed a certain level even in the congested situation. The details of the protocol are presented in the following subsections:

4.1 Determination of Per-Hop Deadline Miss Ratio

We have introduced per-hop deadline miss ratio as the basis of the rate adjustment. This is the most important metric for the soft real time system. This metric reflects the level of contention and congestion for a particular node. To verify this we have performed a simulation in ns-2 over 100 nodes using AODV as a routing protocol and we set the value of end-to-end deadline as 300 ms.. End-to-end deadline miss ratio is measured at the sink under different congested scenarios. Fig. 3 illustrates our result. Here, it shows that as the congestion increases (increasing the packet transmission rate) the deadline miss ratio also increases.

This is inevitable that the increase in congestion incurs the packet waiting delay at the buffer. Moreover, as the contention increases the packet transmission delay also increases. Hence, we have used this metric for the detection of congestion for the real time traffic on a particular node.

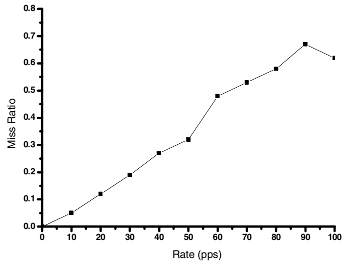

Fig. 3. Miss Ratio in Different Congested Scenario

The per-hop deadline miss ratio, d^i_{miss} is computed periodically. At the end of an epoch, *t*, the d^i_{miss} is calculated as-

$$d^i_{miss}(t) = \frac{N_{miss}(t)}{N_{sch}(t)} \tag{1}$$

Here, $N_{miss}(t)$ is the number of packets missed the per-hop deadline at the end of epoch, *t* and $N_{sch}(t)$ is the total number of packets scheduled during this epoch.

Now, the question is how we can calculate the value of $N_{miss}(t)$. To determine $N_{miss}(t)$, two information are needed; i) The projected waiting time within a node i, $D_w{}^i$ and ii) Per-hop entailed delay, $d_{ent}{}^i$.

Estimation of the waiting time for a packet within a node

The projected waiting time within a node i, $D_w{}^i$ is defined as the sum of the expected queue waiting time, $t_q{}^i$, and the average packet service time, $\overline{t^i_s}$. The queue waiting time $t_q{}^i$ is the time a packet experiences from the moment it enters into the queue until it reaches the head of the queue, and ready for transmission. We have modeled the queue at each sensor node as M/G/1 system. Hence, the expected queue waiting time, $t_q{}^i$ can be estimated by using the well known Pollaczek-Khinchin mean value formula as follows:

$$t^i_q = \frac{R^i_{in} \times \overline{t^{i^2}_s}}{2(1 - R^i_{in} \times \overline{t^i_s})} \tag{2}$$

Where, R^i_{in} is the packet incoming rate at the queue, $\overline{t^{i^2}_s} = \sigma^2 + \overline{t^i_s}^2$ and c^2 is the distribution variance.

Packet service time is defined as the time when the packet is ready for transmission until the last bit leaves from a node. It includes channel busy time, DIFS, backoff, RTS,CTS and packet transmission time. By using EWMA (Exponential Weighted Moving Average Formula), $\overline{t_s^i}$ is updated each time a packet is forwarded as follows:

$$\overline{t_s^i} = (1 - w_s) \times \overline{t_s^i} + w_s \times inst(t_s{}^i) \quad (3)$$

Where, $inst(t_s{}^i)$ is the instantaneous service time of the packet just transmitted and w_s is a constant in the range of $0 < w_s < 1$.

Thus, the estimated waiting time within a node i is,

$$D_w{}^i = t_q{}^i + \overline{t_s^i} \quad (4)$$

The calculation of the estimated waiting time within a node is performed each time when a packet arrives at the queue.

Estimation of the per-hop entailed delay for a packet

We have adopted the estimation of per-hop entailed delay for a packet as the procedure described in [4]. The per-hop entailed delay, d_{int}^i works as a threshold in order to determining the per-hop deadline miss ratio for a packet. It can be defined as the maximum allowable time a packet can stay in a node from the moment it arrives until the transmission ends. Hence, per-hop entailed delay, d_{int}^i can be computed as

$$d_{\text{int}}^i = t_{deadline} - t_{elapsed} - t_{e2e}^i \quad (5)$$

Where, $t_{deadline}$ is the deadline for a packet, $t_{elapsed}$ is the elapsed time for a packet (the elapsed time will be piggybacked in the packet by the upstream node) since it was initiated from the source and t_{e2e}^i is the projected end-to-end (node i to sink) delay which can be calculated as,

$$t_{e2e}^i = D_w^i + \sum_{j=i+1}^{M} t_{e2e}^j \quad (6)$$

Here, M stands for the number of hops up to the sink from node i. Each node maintains and feedbacks to the previous hop regarding expected delay, t_{e2e}^i from itself to the sink. And this feedback information can be propagated through the per frame acknowledgement packet used in 802.11.

When $D_w{}^i$ and d_{int}^i are available to node *i* for an incoming packet, the packet will be treated as it will miss the per-hop deadline when the relation, $d_{\text{int}}^i < D_w^i$ becomes true. As the calculation of entailed delay includes the estimated end-to-end delay from a particular node *i*, it guarantees that the packet which misses per-hop deadline at node i will certainly going to miss the end-to-end deadline.

The scheduler will maintain a counter to determine the number of the per-hop deadline missed packets during epoch duration as well as counts the total scheduled packets in that epoch. These parameter values are fed to the rate controller in order to calculate d_{miss}^i according to equation 1.

4.2 Hop by Hop Dynamic Rate Adjustment

The hop by hop dynamic rate adjustment is performed by the child nodes of particular node *i*, through the overhearing of the parent's packet. Each node *i* will piggyback its current average packet service time, $\overline{t_s^i}$, total number of child nodes, C_p and deadline miss ratio d_{miss}^i value in its packet header. Because of the broadcast nature of wireless channel, all the child nodes of node *i* overhear the packet header parameters. The update of their transmission rate occurred based on the value of d_{miss}^i of parent node. The rate adjustment algorithm is shown in Fig. 4.

Algorithm:	Hop by Hop Rate Adjustment
Input:	Each node i
Output:	Packet Transmission Rate, R_{trans}^i
Initialization()	
$R_{trans}^i = R_{trans}^{init}$, $d_{miss}^i = 0$	
Calculate_Rate ($\overline{t_s^{p^i}}$, $C_{p,}$ $d_{miss}^{p^i}$)	
If $d_{miss}^{p^i}$ <. 0.1	
then $R_{trans}^{p^i} = \frac{1}{\overline{t_s^{p^i}}}$, $R_{trans}^i = \frac{R_{trans}^{p^i}}{C_p}$	
End if	
If $d_{miss}^{p^i} > 0.1$	
then $R_{trans}^i = \frac{R_{trans}^{p^i}}{C_p} \times (1 - d_{miss}^{p^i})$	
End if	
Return R_{trans}^i	

Fig. 4. Rate Adjustment Algorithm

The algorithm works as follows:

Initially, every source node will set their PHY/MAC transmission rate to a small value R_{trans}^{init} and the d_{miss}^i is set to 0.

During the normal operation of the network, as long as the parent's per-hop deadline miss ratio, $d_{miss}^{p^i}$ remains less than 0.1, each child node i, updates their transmission rate as,

$$R_{trans}^i = \frac{R_{trans}^{p^i}}{C_p}$$

Where, parent's transmission rate, $R_{trans}^{p^i}$ will be measured as the inverse of its packet service time, $\overline{t_s^{p^i}}$. This is performed in order to maintain the fairness among all the child nodes.

Each of the child node i, will update their transmission rate when $d_{miss}^{p^i} > 0.1$, (per-hop deadline miss ratio becomes greater than 10%) as follows:

$$R_{trans}^{i} = \frac{R_{trans}^{p^i}}{C_p} \times (1 - d_{miss}^{p^i})$$

This indicates the rate decrease proportional to the per-hop deadline miss ratio. As updated value of deadline miss ratio above a certain level (10%) indicates congestion, the proportional decrease rate lowers the incoming rate to the parents. This certainly reduces the waiting time of a packet. Therefore, the per-hop deadline miss ratio will decrease which indicates the subsiding of congestion.

But truly, the congestion detection threshold depends on the QoS requirements of the corresponding application. Application with the requirement of very lower deadline miss ratio may lower the threshold.

5 Simulation

We have performed extensive simulations using ns-2 [13] to evaluate the performance of our protocol. The simulation parameters are described as follows: 100 sensors are randomly deployed in 100x100 m^2 sensor field. The transmission range of the sensors is 30 m. The maximum communication channel bit rate is 32 kbps. We assume each packet size is 30 bytes. The weight used in the exponential weighted moving average formula for packet service time (eq 3 in section 4.1) is set to 0.1. The epoch size is set to 100 ms and the maximum queue size is set as 30 packets. Throughout the simulation, we used a fixed workload that consists of 10 sources and 1 sink. The initial transmission rate was set to 4 pps (packets per second).

The IEEE 802.11 DCF has been used as MAC protocol and as a routing protocol we have used AODV in the simulation. We have run the simulation for 60 seconds.

We have compared our protocol with the recent rate control protocol DART [3] and SUPORTS [4]. Fig. 5 depicts the miss ratio under different offered load. We set the end-to-end deadline as 200 ms. It shows that as offered load increases the end-to-end miss ratio for our proposed scheme remains within 10% as compared to DART and SUPORTS and for their protocol it is much higher after a certain offered load. The miss ratio for multiple deadlines under different congestion for our proposed protocol is shown in Fig. 6. For three deadlines (100, 200 and 300 ms), it shows the higher the deadline the lower the miss ratio for several offered load. Here, only the packets with 100 ms deadline failed to maintain the miss ratio below 10%. So, for the application with much lower deadline the congestion detection threshold need to be changed. The end-to-end delay for different offered load is shown in Fig. 7. In this case, our proposed scheme maintains moderate E2E delay than the other two schemes. Fig. 8 illustrates the effective throughput for diverse offered load. The effective throughput is defined as the fraction of the packets received within deadline over the load offered in the network. Our proposed scheme upholds about 80% effective throughput for different load. Due to the hop by hop adjustment of rate our proposed scheme and SUPORTS works well than sink initiated DART protocol.

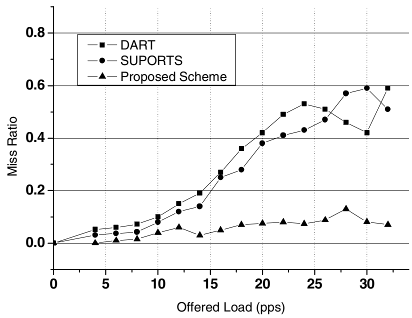

Fig. 5. Miss Ratio Under Different Offered Load

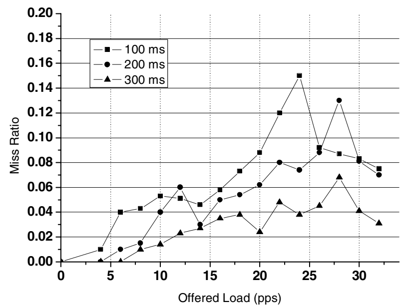

Fig. 6. Miss Ratio Under Multiple Deadlines

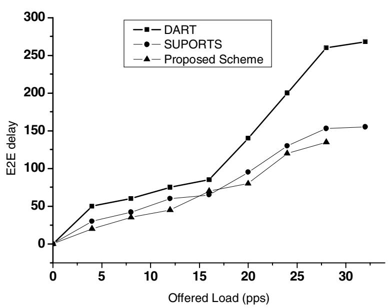

Fig. 7. E2E delay Under Different Offered Load

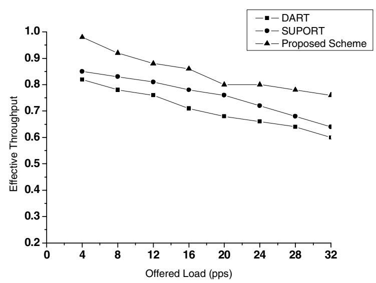

Fig. 8. Effective Throughput Under Different Offered Load

6 Conclusion

In this paper, we have presented an efficient QoS management scheme for providing better QoS under congestion situation through adjusting the traffic rate. We have demonstrated through the simulation that our proposed scheme achieves, i) Lower End-to-End Miss ratio, ii) A good Effective Throughput and iii) Lower End-to-end delay in diverse congestion situation. In future, our goal is to implement our protocol in real sensor test bed scenario.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: A survey. Computer Networks J. 38, 393–422 (2002)
2. Wang, C., et al.: Upstream Congestion Control in Wireless Sensor Networks Through Cross-Layer Optimization. IEEE Journal on Selected Areas in Communications 25(4) (May 2007)
3. Gungor, V.C., Akan, O.B.: Delay Aware Reliable Transport in wireless sensor networks. IEEE Journal of Communication Systems 20, 1155–1177 (2007)
4. Kareons, K., Kalogeraki, V.: Real Time Traffic Management in Sensor Networks. In: Proc of IEEE International Real-Time Systems Symposium (2006)
5. Xia, F., Zhao, W., Sun, Y., Tian, Y.: Fuzzy Logic Control Based QoS Management in Wireless Sensor/Actuator Networks, Sensors 7, pp. 3179–3191 (December 2007)
6. Wan, C.-Y., Eisenman, S.B., Campbell, A.T.: CODA: Congestion Detection and Avoidance in Sensor Networks. In: Proceedings of ACM SenSys., vol. 5-7, pp. 266–279 (November 2003)
7. Ee, C.T., Bajcsy, R.: Congestion Control and Fairness for Many-to-One Routing in Sensor Networks. In: Proceedings of the ACM SenSys., vol. 3-5, pp. 148–161 (November 2004)
8. Hull, B., Jamieson, K., Balakrishnan, H.: Mitigating Congestion in Wireless Sensor Networks. In: Proceedings of ACM SenSys., vol. 3-5, pp. 134–147 (November 2004)
9. He, T., Stankovic, J.A., Lu, C., Abdelzaher, T.F.: A spatiotemporal communication protocol for wireless sensor networks. IEEE Trans. Parallel Distr. Syst. 16(10), 995–1006 (2005)
10. Felemban, E., Lee, C.-G., Ekici, E.: MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. IEEE Trans. Mobile Comput. 5(6), 738–754 (2006)
11. Akkaya, K., Younis, M.: An energy-aware QoS routingprotocol for wireless sensor networks. In: Proc. of Intl. Conf. on Distributed Computing Systems Workshops (ICSDSW), Washington, DC (2003)
12. Lu, C., Blum, B.M., Abdelzaher, T.F., Stankovic, J.A., He, T.: RAP: A real-time communication architecture for large-scale wireless sensor networks. In: IEEE Real-Time and Embedded Technology and Applications Symposium, San Jose, CA, pp. 55–66 (September 2002)
13. Network Simulator NS-2, `http://www.isi.edu/nsnam/ns`