Multipath Congestion Control for Heterogeneous Traffic in Wireless Sensor Network

Muhammad Mostafa Monowar, Md. Obaidur Rahman 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 and cshong@khu.ac.kr

***Abstract*—In order to achieve higher reliability and load balancing various multipath routing protocols have been proposed in Wireless Sensor Network. Moreover, wireless sensor network typically incorporates heterogeneous applications within the same network. A sensor node may have multiple sensors i.e. light, temperature, seismic etc with different transmission characteristics. Each application has different characteristics and requirements in terms of transmission rates, bandwidth, packet loss and delay demands may be initiated towards the sink. But achieving desired throughput for diverse data while disseminating through multiple paths is non trivial task as occurrence of congestion through multipath is obvious. In this paper we propose an efficient scheme to control multipath congestion so that the sink can get priority based throughput for heterogeneous data. We have used packet service ratio for detecting congestion as well as performed hop-by-hop multipath congestion control based on that metric. Finally, simulation results have demonstrated the effectiveness of our proposed approach.**

Keywords- Congestion Control, Multipath, Heterogeneous traffic, Scheduler, etc.

I. INTRODUCTION

In recent years, the use of diverse applications in sensor network is proliferating. Sensor nodes may have multiple sensors (light, temperature, seismic) with different transmission characteristics. Packets from a sensor for an application constitute its data flow. For several classes of applications a sensor node may initiate multiple flows that have diverse requirements in terms of transmission rate, reliability, delay and throughput towards the sink. In our protocol, we have designed a queuing model for generating the heterogeneous traffic within each sensor node according to the priority specified by the sink.

In WSN, usually tens or thousands of sensor nodes are deployed scattered way in an area with one or more sinks. Myriad and divergent types of traffic from simple periodic events to unpredictable bursts of messages are generated by sensor nodes. Moreover, for achieving reliability and load balancing, several multipath routing protocols have been proposed. But the limitation of these protocols is traffic overhead. Thus the occurrence of congestion in this situation is more likely. The situation becomes worse when congestion occurs in multiple paths. In order to achieve the desired rate of heterogeneous traffic by the sink the congestion control over multiple paths is indispensable.

In general, congestion control mechanism has three phases: congestion detection, congestion notification and congestion mitigation through rate control. In this paper, we propose an efficient scheme to perform multipath congestion control for heterogeneous traffic which avoids packet loss and thus enhances the probability of achieving the desired throughput of heterogeneous traffic. Our congestion detection mechanism is chosen based on packet service ratio and congestion notification is implicit.

The rest of the paper is organized as follows. Section II presents several related works on congestion control techniques. Subsequently section III describes the design considerations for our proposed scheme. Section IV represents our proposed scheme in detail. Section V details our simulation efforts and finally section VI concludes the paper with some future direction.

II. RELATED WORK

In present research train, lots of work is going on congestion control for wireless sensor network. Scenarios with multipath routing are not considered. It is not clear whether they can be directly applied to WSNs with multipath routing enabled [7]. Moreover, most of the protocols deal with homogeneous traffic. Sensor nodes may have multiple sensing devices (e.g. temperature, light, pressure etc.) and no other protocols except STCP [1] considered multiple sensing devices in the same node. But STCP has some problems: i) It doesn't consider multipath routing even it doesn't state any explicit and detailed mechanism for single path congestion control. ii) The ACK/NACK based reliability mechanism is not suitable for wireless sensor networks in terms of delay and memory use.

Recently a node priority based control mechanism PCCP [2] has been proposed for WSN. It introduces an efficient congestion detection technique addressing both node and link level for detecting congestion. PCCP prioritizes both source and transit traffic but here the limitation of handling multiple sensed data within a node also remains.

Besides these, CCF (Congestion Control and Fairness) [3] CODA [4] (Congestion Detection and Avoidance) [4], Fusion [5], Siphon [6] etc. are also remarkable congestion control technique for wireless sensor network but have the limitation

"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.

of considering only single path congestion and homogeneous traffic.

III. DESIGN CONSIDERATIONS

This section describes the design considerations of our scheme taken into account when constructing the congestion control algorithm, in particular the network and queuing model with some definitions.

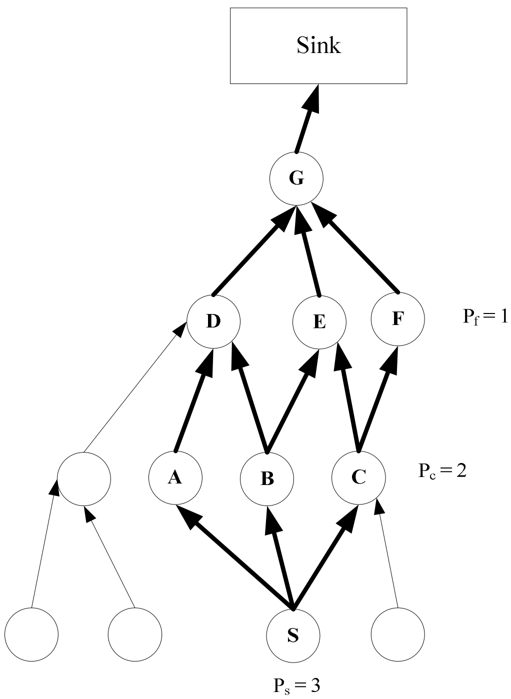

Figure 1. Network Model.

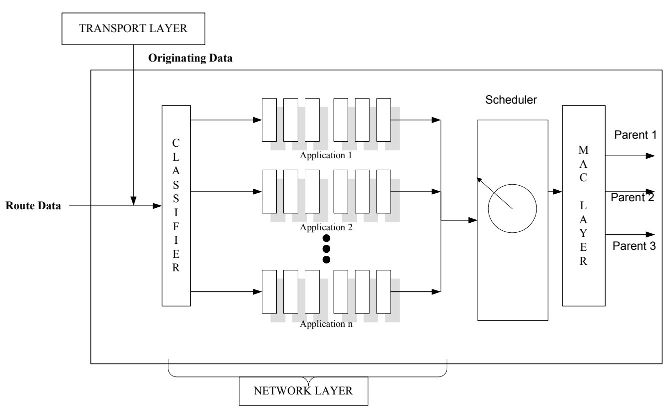

Figure 2. Queuing Model on a Particular Node.

A. Network Model

In this paper, we consider the congestion control for multi-hop multi path routing. The network model is shown in Figure 1, where each node can sense heterogeneous application data simultaneously. In our protocol we assume that every sensor node in the network has equal number and same types of sensors. All nodes are supposed to use CSMA like MAC protocol. We assume that multiple routes have been established by any multipath routing protocol and path establishment is out of the scope of this paper. In Figure 1 the highlighted links represents the established multipath route where a node is disseminating its traffic towards multiple parents (if any). We further assume that while establishing the route the sink dynamically assigns individual priority for the heterogeneous application data. Each sensor node can transmit route data of its children nodes as well as its originating data. So at any given time a sensor node may act as a source node and a forwarding node. We denote the number of the parent nodes as P_i. As shown in Figure 1 node S has 3 parent nodes, C has 2 and node F has 1.

B. Queuing Model

Figure 2 depicts the queuing model on a particular node for multipath routing. We assume that each node i has n number of equal sized queues for n types of data. For example, if a sensor node has 3 sensors for sensing temperature, light and humidity then it will have 3 separate queues for each of the sensory data. As shown in the Figure 2, a classifier has been provisioned in network layer. The purpose of this classifier is to classify the heterogeneous traffic of both originating and route data and place them in their corresponding queue. It can easily perform that by looking in the packet header which includes the type of data it carries.

The further description of queuing model proceeds through the following definitions:

1) Originating Rate: The rate at which the sensor node originates data is known as originating rate denoted as R_{or}^{i} for node i .

2) Scheduling rate: In our protocol we introduce a scheduler between network and MAC layer which maintains the queues as shown in Figure 2. The scheduling rate is denoted as R_{sch}^{i}. We can define R_{sch}^{i} as – how many packets the scheduler schedules per unit time from the priority queues. As each node may have multiple parents, the scheduling rate for a parent j of node i, can be denoted as $R_{sch}^{i,j}$. Thus, the scheduling rate will be

$$R_{sch}^{i} = \sum_{j=1}^{P_n} R_{sch}^{i,j} \qquad (1)$$

The scheduler sends the packets to the MAC layer from which these are delivered to multiple parents. In [2] the authors report that by controlling the R_{sch}^{i} , congestion control can be performed. Our scheme also performs the congestion control through adjusting R_{sch}^{i} and it does so without any change in MAC layer protocol parameters.

3) Intra Queue Priority: The queues shown in Figure 2 are priority queues. More priority is given for route data than originating data because route data have already traversed several paths, so its loss causes more wastage of network resources. We denoted this priority as intra queue priority. The classifier can assign the priority between route data and originating data by examining the source address in the packet header. All route data will have the same priority and so forth for the originating data.

4) *Inter queue Priority:* We mentioned earlier that the sink will dynamically assign the priority for heterogeneous traffic. Therefore, every data queue shown in Figure 2 has their own priority. We denote this priority as inter queue priority. The scheduler schedules the queues according to the inter queue priority. It decides the service order of data packets in the queues and manages the queues according to their priority. This ensures the higher priority data will have the higher service rate.

IV. PROTOCOL DESCRIPTION

Our main motivation in designing this scheme is to control the congestion on multipath for heterogeneous traffic originated and transited through a single sensor node in such a way that the sink can get the priority based throughput for diverse data. In this section we discuss the detail of our scheme including the node level algorithms for congestion detection, notification and congestion mitigation.

A. *Congestion Detection*

In our proposal we have used packet service ratio, r_i to measure the congestion level at each node i. It is the ratio of average packet service rate denoted by R_s^i and packet scheduling rate R_{sch}^i in each sensor node i as follows:

$$r_i = R_s^i / R_{sch}^i \qquad (2)$$

The service rate R_s^i is the inverse of packet service time t_s^i. The packet service time, t_s^i includes the time interval when a packet arrives at MAC layer and when it successfully transmitted as shown in Figure 3. t_s^i covers packet waiting, collision resolution and packet transmission time at MAC layer. In eq. 2, in order to obtain R_s^i, the average packet service time, t_s^i is calculated using Exponential Weighted Moving Average (EWMA) formula.

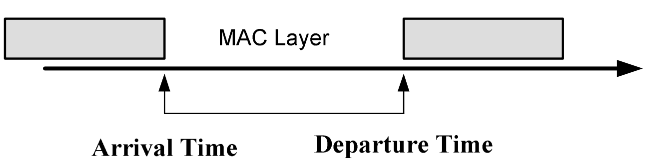

Figure 3. Packet service time, t_s^i.

By using EWMA, t_s^i is updated each time a packet is forwarded as follows:

$$t_s^i = (1 - w_s) \times t_s^i + w_s \times inst(t_s^i) \qquad (3)$$

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 $0 < w_s < 1$.

The packet service ratio is deliberately reflecting the congestion level at each sensor node. Upcoming rate of packets at the MAC layer will be equal to the forwarding rate while this ratio equal to 1. Again while this ratio is greater than 1, the upcoming rate is less than the packet service rate. Both of these cases indicate the subsiding of congestion. When it is less than 1, it means the packet incoming rate is greater than the forwarding rate which causes the queuing up of packets at the MAC layer. This also indicates the link level collisions. Thus the packet service ratio is an effective measure of detecting both node level and link level congestion. The simulation in the section V shows that the rate adjustment should be done when the ratio goes below 0.5. Our protocol tolerates up to 50% of packets to be queued up at the MAC layer.

B. *Implicit Congestion Notification*

This protocol uses implicit congestion notification. Each node i piggybacks the packet scheduling rate, R_{sch}^i; the number of child nodes, C_n and packet service rate, R_s^i in its packet header. All the child nodes of node i overhear the congestion notification information. Whenever the value of packet service ratio of parent *j* of node *i* denoted by r_i^j (lowers the threshold (0.5) or greater than 1, multipath rate control procedure (explained in the next section) is triggered.

C. *Multipath Rate Control*

Our multipath rate control scheme uses hop by hop rate adjustment for multiple paths. In this protocol, the output rate of a node is controlled by adjusting the scheduling rate, R_{sch}^i. In fact, by adjusting the scheduling rate for parent *j*, $R_{sch}^{i,j}$; the packet loss due to buffer overflow is avoided and the total scheduling rate, R_{sch}^i is adjusted automatically as scheduling rate is the sum of scheduling rate for all the parents *j* of node *i*. In section IV-A and IV-B we have stated that the information of packet service ratio is used for congestion detection and it is piggybacked in the packet header along with other parameters. Each node i updates their scheduling rate while any of its parent's packet service ratio, r_j^i goes beneath the threshold or greater than 1. The initial scheduling rate is set to r_{sch}^{init}. Figure 4 depicts the multipath rate control algorithm.

Algorithm: Multipath Rate Control	
Input: Each node i; Output: Scheduling rate, R^i_{sch}, Originating rate, R^i_{or}	
Initialization()	
1.	$R^i_{sch} = r^{init}_{sch}$; $r_i = 1$, $R^{i,j}_{sch} = (\frac{R^i_{sch}}{P_n}) / C_n$, for each j∈ P_i;
RateAdjustment($R^{i,j}_{sch}$, R^j_s **)**	
2.	$r^i_j = R^j_s / R^j_{sch}$
	$r_i = R^i_s / R^i_{sch}$
3.	for $j \in P_i$ if $r^i_j < \mu$ then $R^{i,j}_{sch} = R^j_s / C_n$
	and $R^j_{sch} = R^j_s$
4	for $j \in P_i$ If $r^i_j > 1$ then $R^{i,j}_{sch} = (R^j_s \times \eta) / C_n$
	and $R^j_{sch} = R^j_s \times \eta$
5	$R^i_{sch} = \sum_{j=1}^{P_n} R^{i,j}_{sch}$
6.	return R^i_{sch}
CalcSrcRate(R^i_{sch}, α_i **)**	
7.	$R^i_{or} = \frac{R^i_{sch}(t) * \alpha_i}{\alpha_1 + \alpha_2 + \cdots + \alpha_n}$
8.	return R^i_{or}

Figure 4. Multipath Rate Control Algorithm.

The algorithm works as follows:

1) Initially, the packet service ratio for each node i will be 1, each node will have a lower scheduling rate, r^{init}_{sch} ; and the scheduling rate for each of its parents will be distributed according to line 1.

2) Each node i will adjusts its scheduling rate towards multiple parents by calling RateAdjustment() procedure. At first each node i calculates its packet service ratio as well as its parents as shown in line 2. The rate determination depends on the following cases:

a) When this ratio is equal to 1 it means the upcoming rate of packets at the MAC layer is equal to the packet service rate. This is the most desired condition as no node level and link level congestion will occur. So, the scheduling rate R^i_{sch} for each node will remain unchanged.

b) Scheduling rate R^i_{sch} will also remain unchanged until the packet service ratio, r^i_j for any parent *j* for goes beneath a threshold. In this case, some packets will be queued up at the MAC layer, but it doesn't notify any congestion which we have shown in the simulation section.

c) When the ratio, r^i_j goes below a certain threshold, μ; for parent j, it notifies congestion. Then node *i* will adjust the scheduling rate for its parent to avoid the node level buffer overflow. At that time the congested parent also adjusts its scheduling rate to make the service ratio near to 1 as in line 3.

d) When r^i_j reaches above 1, it indicates the packet service rate is greater than the upcoming rate. In this case, each node will increase the scheduling rate for its parent j to improve link utilization. At that time, the parent j will also increase its scheduling rate as shown in line 4. Here the value of η is chosen to a value smaller than but close to 1 which is used to maintain the small queue length. In our protocol, it is set as 0.75.

3) Just after calculating the scheduling rate, each node, i update their originating rate R^i_{or} according to the method *CalcSrcRate()*. The originating rate depends on the scheduling rate as well as on the priority for each application data requested by the sink.

V. SIMULATION

We have performed extensive simulations to evaluate the performance of our scheme. We determine the threshold value of packet service ratio, μ with the simulation. We also show that the proposed protocol maintains a moderate queue length to avoid buffer overflow and the sink is getting the application priority based throughput.

The simulation parameters are described as follows: 200 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 calculation of packet service time (eq 3 in section IV-A) is set to 0.1. We assume that each node has three sensing devices for three applications- temperature, pressure and seismic. Each queue length is allocated to maximum of 10 packets to hold packets from the child nodes. That is, the total queue length for a node is 30 packets where 10 packets for each queue. We also assume that a node is within the range of its parent and children to avoid the interference from node more than one hop away.

A. Determination of μ

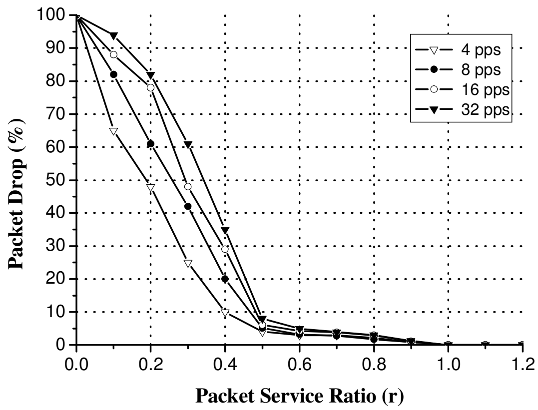

Figure 5. Choice of μ.

Figure 5 demonstrates the determination of the threshold of packet service ratio. It shows the percentage of packet drop at the MAC layer due to different packet service ratio for different packet rates. In this case we assume that at least one packet will be at the MAC layer buffer. The percentage of packet drop is the highest (about 100%) when the ratio is close to 0. The increase in the ratio reduces the packet drop percentage for each of the packet originating rate (4, 8, 16, 32 packet per second). Figure 5 shows that for different packet originating rate the packet drop percentage goes below 10 and reaches into a stable state when the packet service ratio becomes 0.5 and this is a tolerable situation before notifying any congestion. Thus we set the threshold, μ to 0.5.

B. Average Queue Length

Figure 6 illustrates the status of average queue length over time. It shows that the moderate queue length is maintained in three parents for about 30 second simulation time while all of the parents are congested in the interval [10 25]. It shows that the average queue length never exceeds the maximum queue length (10). This illustrates that the proposed scheme avoids packet loss due to buffer overflow.

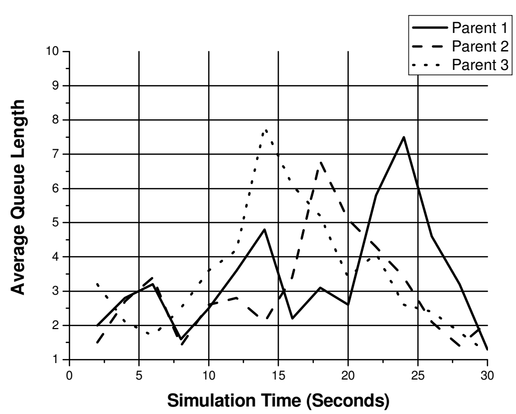

Figure 6. Average Queue Length over Time.

C. Priority Based Throughput

Figure 7 shows the number of different types of packets received by the sink over time. We assume that the temperature has the highest priority, 3; pressure has 2 and seismic belongs to 1. For about 60 seconds simulation time it shows that the sink receives almost three and two times temperature packets than the seismic and pressure packets respectively which is actually the desired goal of our proposed work.

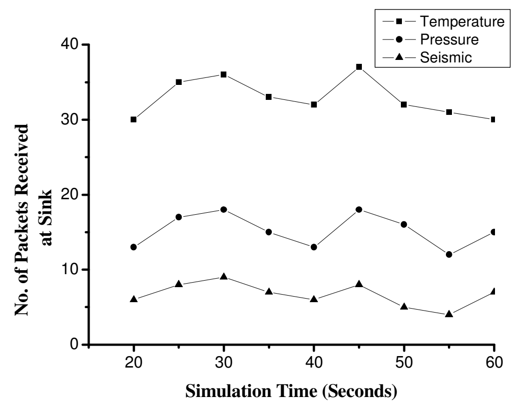

Figure 7. No of Heterogeneous Packets Received by Sink over Time.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we have presented an efficient multipath congestion control mechanism for heterogeneous data originated from a single sensor node We have demonstrated through the simulation that our scheme achieves: i) Desired throughput for diverse data according to the priority specified by the sink, ii) Moderate queue length to avoid packet loss and iii) Lower packet drop rate.

Our design of this scheme points to some directions for future works: to improve the fairness, analysis of the impact of other parameters on the proposed scheme's performance and implementing this scheme on a real sensor test-bed and compare the results with those obtained in the simulations.

REFERENCES

[1] Y.G.Iyer, S. Gandham and S. Venkatesan, "STCP: A Generic Transport Layer Protocol for Wireless Sensor Networks," Proc. IEEE ICCCN 2005. San Diego, CA, Oct. 17-19, 2005

[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] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu, "A Survey of transport protocols for wireless sensor networks," IEEE Network, vol.20, no. 3, pp. 34-40, May/June 2006.