Prioritized Heterogeneous Traffic-Oriented Congestion Control Protocol for WSNs

Muhammad Monowar[1], Obaidur Rahman[2], Al-Sakib Khan Pathan[3], and Choong Seon Hong[2]
[1]Department of Computer Science and Engineering, University of Chittagong, Bangladesh
[2]Department of Computer Engineering, Kyung Hee University, South Korea
[3]Department of Computer Science, International Islamic University Malaysia, Malaysia

Abstract: *Due to the availability of multiple sensing units on a single radio board of the modern sensor motes, some sensor networks need to handle heterogeneous traffic within the same application. This diverse traffic could have different priorities in terms of transmission rate, required bandwidth, packet loss, etc. Because of the multi-hop transmission characteristic of this prioritized heterogeneous traffic, occurrence of congestion is very common and unless handled effectively, it could thwart the application objectives. To address this challenge, in this paper we propose a Prioritized Heterogeneous Traffic-oriented Congestion Control Protocol (PHTCCP) which performs hop-by-hop rate adjustment controlling the congestion and ensures efficient rate for the prioritized diverse traffic. This protocol also could be applied for healthcare infrastructure. We exploit cross layer approach to perform the congestion control. Our protocol uses intra-queue and inter-queue priorities along with weighted fair queuing for ensuring feasible transmission rates of heterogeneous data. It also guarantees efficient link utilization by using dynamic transmission rate adjustment. We present detailed analysis and simulation results with the description of our protocol to demonstrate its effectiveness in handling prioritized heterogeneous traffic in Wireless Sensor Networks (WSNs).*

Keywords: *Heterogeneous, congestion, inter-queue, intra-queue, scheduler, sensor.*

Received July 27, 2009; accepted May 20, 2010

1. Introduction

The sophistication of various communication protocols [9] and rapid advancements of Micro-Electro-Mechanical Systems (MEMS) technologies [23] have created a great opportunity for wide-spread utilizations of various innovative sensor network applications in near future. Today's sensors are capable of sensing more than one parameter with the aid of multiple sensor boards mounted on a single radio board. MICA2 [4] is an example of such type of sensor. ExScal mote, an extension of MICA2, also supports multiple sensing units [1, 3]. Instead of using multiple nodes with various functionalities [15], deploying such nodes might offer cost effective solutions for many applications. For example, a volcano monitoring application might require temperature, seismic, and acoustic data from same location. Several applications could even run simultaneously based on various data sent by the multi-purpose nodes. Different types of data also might have different levels of importance and accordingly their transmission characteristics might differ.

In this paper, we consider a WSN where the deployed nodes are multi-purpose nodes and they generate heterogeneous traffic destined to the Base Station (BS). Various types of data generated by the sensors have various priorities. Hence, it is necessary to ensure desired transmission rate for each type of data based on the given priority to meet the demands of BS. In such a network, the sensors could in fact generate simple periodic events to unpredictable bursts of messages. Both of these cases produce convergent data flows from source nodes to the BSs which can potentially cause congestion. Congestion becomes even more likely when concurrent data transmissions over different radio links interact with each other or when the reporting rate to the base station increases. With the increase of number of nodes in the network, congestion might occur frequently. Such congestion has a severe impact on the energy efficiency and application Quality of Service (QoS) of WSNs.

Congestion control mechanism requires the consideration of two main issues; congestion detection and efficient rate adjustment. In TCP, congestion is inferred at the receiving end based on timeout or duplicate acknowledgement while in WSN proactive methods are preferred. A commonly used mechanism is using buffer length [5, 7, 17, 20] packet service time [2], or the ratio of packet inter-arrival time and packet service time [21]. To deal with the congestion, an efficient rate control mechanism needs to be designed in order to mitigate or avoid congestion. The end-to-end [7, 14, 18] and hop-by-hop [2, 5, 17, 20, 21] strategy have been employed for the rate control in the last few years. Here, we propose a hop-by-hop rate control scheme for quick recovery of congestion at the intermediate nodes.

The rest of the paper is organized as follows. Section 2 states our motivation with relevant works. Section 3 presents network model, goals, and preliminaries, section 4 presents the details of PHTCCP. Analysis and simulation results are presented in section 5, and section 6 concludes the paper with future research directions.

2. Motivation and Related Works

A number of previous works have addressed the issue of congestion control in WSNs [22]. But most of the works have dealt with the rate control for homogeneous applications. In fact, no other work except STCP [7] has considered the use of multi-purpose sensors in the network. STCP is a generic, scalable and reliable transport layer protocol where a majority of the functionalities are implemented at the BS. The problem of STCP is twofold:

1. It takes much time for the sources to be notified of the congestion situation and thus to perform the rate reduction for congestion elimination.
2. The use of explicit acknowledgement packet is not suitable for WSN which also increases congestion. Furthermore, STCP does not provide any specific rate reduction algorithm that addresses heterogeneous traffic.

CODA [20] uses both buffer occupancy and channel load for measuring node and link level congestion in the network. It handles both transient and persistent congestions. Fusion [5] detects congestion by measuring the queue length. It controls congestion by combining three techniques, hop-by-hop flow control, source rate limiting, and prioritized MAC. Although Fusion claims to achieve good throughput and fairness at high offered load, the non smooth rate adjustment in handling transient congestion at the intermediate nodes could mess up link utilization and fairness.

IFRC [17] is an interference aware rate control mechanism designed for sensor network. It detects incipient congestion at a node by observing the average queue length and performs distributed rate allocation among the nodes. IFRC would fail to ensure traffic oriented weighted fairness and maintaining a feasible transmission rate for the diverse data as it considers every flow equally. In [2], the authors propose a hop-by-hop congestion control technique, Congestion Control and Fairness (CCF), which uses packet service time to infer the available service rate and therefore detects congestion in each intermediate sensor node. CCF ensures simple fairness. However, it lacks efficient utilization of the available link capacity when some nodes do not have any traffic to send or nodes remaining in sleep mode or the nodes whose flows do not pass through the congested area.

PCCP [21] is a recent congestion control protocol for WSNs which uses hop-by-hop approach for rate control. PCCP is a node priority based congestion control protocol which allows sensor nodes to receive priority-dependent throughput. However, PCCP does not have any mechanism for handling prioritized heterogeneous traffic originated from a single node. RCRT [14] is an end-to-end rate controlled reliable transport protocol. Although this scheme supports concurrent applications, it considers heterogeneous nodes instead of heterogeneous traffic generated from a single node. Moreover, congestion detection is performed based on packet loss recovery time and rate adaptation, and rate allocation is performed by sink. We argue that the sink based congestion detection and rate control lacks quick recovery of congestion as it requires at least one RTT to detect congestion. Besides these, siphon [19] (uses traffic redirection to mitigate congestion), ESRT [18] (sink based reliable rate control protocol) etc., also address the congestion control issues but none of them consider the diverse traffic originated and routed through a single node.

Hence, the scarcity of an efficient congestion control protocol for handling diverse data with different priorities within a single node motivates us to propose PHTCCP [12].

3. Design Considerations and Preliminaries

In this section, we state various design considerations. Note that throughout the paper the terms rate control and congestion control are used interchangeably.

3.1. Network Model and Assumptions

We consider a WSN where thousands of multi-purpose nodes are deployed over a specific target area. We exclude the availability of any mobile nodes [10] as the nodes for WSN are usually static for most of the applications. All nodes are equipped with the same number of different sensor boards mounted on a single radio board. Each of the nodes can sense different types of data at the same time and sends those to BS. Figure 1 shows a model for our network depicting single path and multi-hop routing.

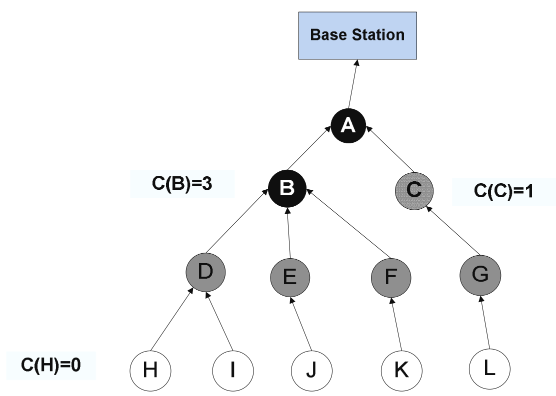


Figure 1. Network model, routing topology view.

All nodes are supposed to use Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) like MAC protocol. More about of our MAC protocol features are presented in section 4.4. We assume that the network structure and the routes to BS have been established by using some efficient routing protocol. While establishing the structure of the network, the BS dynamically assigns individual priority for each type of data. During forwarding heterogeneous data towards the BS, each sensor node transmits route data of its children nodes as well as its own generated data. So, at any given time, a sensor may act both as a source node and a forwarding node. When a sensor transmits its data to the upstream direction, it is called a child and its immediate upstream node is called its parent. Each link between any parent and child is bidirectional that is if the child gets its parent within its transmission range, the parent also gets the child within its transmission range. We denote the number of child nodes for a parent node K as $C(K)$. As in Figure 1, node B has 3 children, C has 1, and node H does not have any child. For each node in the network, there is a single path to reach to the BS. Figure 1 also shows different levels of hotspot nodes for congestion. The black nodes i.e., A and B have the highest probability of congestion as all the diverse traffic beneath these nodes in the sub-tree (including the nodes themselves) traverse through these nodes. The grey nodes might also suffer from congestion in case of burst traffic while the white nodes have the least possibility of node level congestion.

3.2. Node Model

Figure 2 depicts the node model of a particular sensor. The congestion control functionality at the transport layer has been transferred to the PHTCCP module in the network layer. We have employed cross layer functionality in designing our protocol. PHTCCP module works interacting with the MAC layer to perform congestion control function. The application layer generates originating data (if it is a source node) and the route data come from the child nodes (if it has any) and traverse through the network layer. We assume that each node i has n number of equal sized priority queues for n types of sensed data. For example, a sensor node might sense temperature, light, and humidity at the same time. In such a case, there are 3 separate queues for each type of data. The number of queues in a node depends on the application requirements. As shown in Figure 2, a classifier has been provisioned in the network layer. The purpose of putting this classifier is to classify heterogeneous traffic either generated by the same node or incoming from other nodes. Based on the type of data, they are placed in the apposite queue.

A weighted fair queue scheduler has been provisioned to schedule the diverse traffic with different priority from the priority queues. The priority of the traffic has been mapped to the queue weight. PHTCCP module does not interfere with the core functionality of the network layer. Hence, PHTCCP is independent of using any routing protocol.

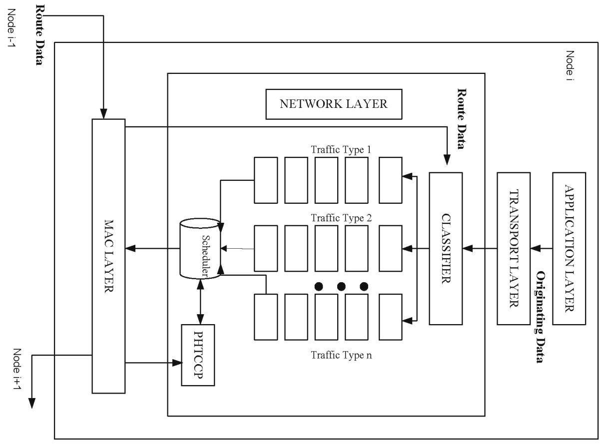


Figure 2. A sensor node model for PHTCCP.

3.3. Definitions

- *Originating Rate:* The rate at which a node originates data. Denoted as R^i_{or} for a node i.
- *Scheduling Rate (R^i_{sch})*: The scheduling rate is defined as how many packets the scheduler schedules per unit time from the queues. The scheduler forwards the packets to the MAC layer from which the packets are delivered to the next node (i.e., i+1) along the path towards the base station. The rate control function is performed by controlling the scheduling rate which is explained in detail in the next section.
- *Average Packet Service Rate (R^i_s)*: This is the average rate at which packets are forwarded from MAC layer.
- *Inter-Queue Priority*: We mentioned earlier that the base station assigns the priorities for heterogeneous traffic. Therefore, each data queue shown in Figure 2 has its own priority. This is termed as Inter-Queue priority. The scheduler schedules the queues according to the inter-queue priority. It decides the service order of the data packets from the queues and manages the queues according to their priorities. This ensures the data with higher priority to get higher service rate.
- *Intra Queue Priority*: All the queues shown in Figure 2 are priority queues. Priority queues are used for giving the route data more priority than originating data. The reason is that; as route data have already traversed some hop (s), their loss would cause more wastage of network resources than that of the originating (source) data. Hence, it

is better to forward those as soon as possible after receiving from the immediate downstream node. We term this type of priority as intra-queue priority. The classifier can assign the priority between the route data and originating data by examining the source address in the packet header.

4. Our Protocol: PHTCCP

In this section, we describe our proposed protocol. The major goals for our scheme are:

1. Generating and transmitting the heterogeneous data on priority basis.
2. Adjusting the rate while congestion occurs, and 2 to ensure efficient link capacity utilization when some nodes in a particular route are inactive or in sleep mode. PHTCCP uses Weighted Fair Queuing (WFQ) for scheduling. Here, we illustrate PHTCCP in detail using several subsections to address the issues of congestion detection, notification, and mitigation.

4.1. Congestion Detection Method

We use packet service ratio *r(i)* to measure the congestion level at each node *i*. Packet service ratio is defined as the ratio of average packet service rate (R_s^i) and packet scheduling rate (R_{sch}^i) in each sensor node *i* that is:

$$r(i) = R_s^i / R_{sch}^i \tag{1}$$

Here, the packet service rate R_s^i is the inverse of packet service time t_s^i. t_s^i is the time interval when a packet arrives at the MAC layer and when it is successfully transmitted towards the next hop. t_s^i includes packet waiting time, collision resolution, and packet transmission time at MAC layer. In equation 1, in order to obtain R_s^i, the average packet service time, t_s^i is calculated using Exponential Weighted Moving Average formula (EWMA). By using EWMA, t_s^i is updated each time a packet is forwarded as:

$$t_s^i = (1 - w_s) \times t_s^i + w_s \times inst(t_s^i) \tag{2}$$

Where, $inst(t_s^i)$ is the instantaneous service time of the packet that has just been transmitted and w_s is a constant where, $0 < w_s < 1$.

The packet service ratio reflects the congestion level at each sensor node. When this ratio is equal to 1, the scheduling rate is equal to the forwarding rate (i.e., average packet service rate). When this ratio is greater than 1, the scheduling rate is less than the average packet service rate. Both of these cases indicate the decrease of the level of congestion. When it is less than 1, it causes the queuing up of packets at the MAC layer. This also indicates link level collisions. Thus, the packet service ratio is an effective measure to detect both node level and link level congestion.

4.2. Implicit Congestion Notification

PHTCCP uses implicit congestion notification. Each node *i* piggybacks its packet scheduling rate R_{sch}^i; total number of children, $C(i)$; number of active children at time t, $A_t(C(i))$; and the weighted average queue length of its active child nodes in its packet header. Because of the broadcast nature of wireless channel, all the children of node *i* overhear the congestion notification information. Whenever the value of $r(i)$ goes below a certain threshold (application dependant), rate adjustment procedure is triggered.

4.3. Rate Adjustment

PHTCCP uses hop-by-hop rate adjustment for controlling the congestion. The output rate of a node is controlled by adjusting the scheduling rate, R_{sch}^i. We have stated earlier that the information of packet service ratio for congestion detection is piggybacked in the packet header along with other parameters. Each node *i* updates its scheduling rate if this ratio goes below the threshold or if there is any change in the scheduling rate of its parent node. The initial scheduling rate is set to r_{sch}^{init}.

Before presenting the rate adjustment algorithm, we present the notations and illustrations in Table 1. The entire rate adjustment algorithm is shown in Figure 3.

Each node *i* measures its scheduling rate by calling the *Calculate_Scheduling_Rate()* method. In this method, at first each *i* calculates its packet service ratio. When this ratio is equal to 1, it means that the incoming rate of packets to the MAC layer is equal to the average packet service rate (the rate at which packets are forwarded from MAC layer). This is the ideal case so that no congestion occurs. In this case, R_{sch}^i remains unchanged. R_{sch}^i remains unchanged as long as the packet service ratio doesn't go below the specified threshold. In fact, when the packet service ratio ($r(i)$) is less than the specified threshold value (say noted by μ), it indicates that the scheduling rate of packets is larger than the average packet service rate. In such a case, packets would be queuing up at the MAC layer buffer and might cause buffer overflow indicating congestion. To control congestion, in this case, the scheduling rate is reset (decreased) to the value of packet service rate.

Table 1. Basic notations used in the paper.

R^i_{sch}	Scheduling rate of node *i*
R^i_{or}	Originating rate of node *i*
$R^{p_i}_{sch}$	Scheduling rate of the parent of node *i*
$A_t(C(p_i))$	Total number of active child nodes of the parent of node *i* at time *t*
$C(p_i)$	Total number of child nodes of the parent of node *i*
$E(t)$	Excess link capacity at time *t*
$\varphi_i(t)$	Weight factor for node *i* at time *t*
α^i_j	Priority for the *j*th queue of node i, *where, j=1,2,..,n*
q^i_j	Current queue length for *j*th queue of node *i, where, j=1,2,..,n*
$avg^q_i(t)$	Weighted average queue length of node *i* at time *t*
N	Number of queues in node *i*

Algorithm: Rate Adjustment
Input: Each node i;
Output: Scheduling rate R^i_{sch}, Originating rate R^i_{or}
Initialization()
$R^i_{sch} = r^{init}_{sch}$; $r(i)=1$;
Calculate_Scheduling_Rate(R^i_{sch}, R^i_s)
$r(i) = R^i_s / R^i_{sch}$
If $r(i) < \mu$ then $R^i_{sch} = R^i_s$ End If
If $r(i) > 1$ then $R^i_{sch} = \beta * R^i_s$ End If
return R^i_{sch}
Dyn_Rate_Adj($R^{p_i}_{sch}, A_t(C(p_i)), C(p_i), E(t)$)
If $A_t(C(p_i)) = C(p_i)$ then $R^i_{sch} = R^{p_i}_{sch} / C(p_i)$
End If
If $A_t(C(p_i)) < C(p_i)$ then $R^i_{sch} = R^i_{sch} + \varphi_i(t)E(t)$
End If
Calc_ExcessLinkCapacity($R^{p_i}_{sch}, A_t(C(p_i)), C(p_i)$)

$$E(t) = \sum_{n=1}^{C(p_i)} R^{p_i}_{sch} / C(p_i) - \sum_{n=1}^{A_t(C(p_i))} R^{p_i}_{sch} / C(p_i)$$

return $E(t)$
Calc_NodeWeightFactor($\alpha^i_j, q^i_j, A_t(C(p_i))$)

$$avg^q_i(t) = \frac{\sum_{j=1}^{N} \alpha^i_j \times q^i_j}{N}$$

$$\varphi_i(t) = \begin{cases} \dfrac{avg^q_i(t)}{\sum_{i \in A_t(C(p_i))} avg^q_i(t)} & i \in A_t(C(p_i)) \\ 0 & otherwise \end{cases}$$

return $\varphi_i(t)$
Calculate_SourceRate(R^i_{sch}, α_i)

$$R^i_{or} = \frac{R^i_{sch}(t) * \alpha_i}{\alpha_1 + \alpha_2 + \cdots + \alpha_n}$$

return R^i_{or}

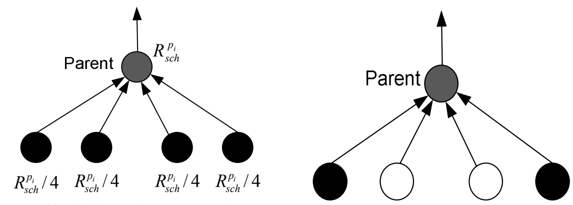
Figure 3. Rate adjustment algorithm.

When $r(i)$ reaches above 1, it indicates that the packet service rate is greater than the scheduling rate. Hence, the scheduling rate is increased using, $R^i_{sch} = \beta * R^i_s$. Here β's value is chosen to a value smaller than but close to 1. In our protocol, it is set to 0.75.

After determining the desired scheduling rate, each node i adjusts its own scheduling rate according to the scheduling rate of its parent node. This is done dynamically by calling the method *Dyn_Rate_Adj()*.

The rate adjustment depends on two cases: when node i determines that all the child nodes of its parent (including itself) are active at time *t*, as shown in Figure 4-a, $A_t(C(p_i)) = C(p_i)$), then node i makes adjustment in its scheduling rate. In this case, each node i sets its scheduling rate equal to $1/C(p_i)$th of its parent's scheduling rate. In Figure 4-a, if the scheduling rate of the parent node is $R^{p_i}_{sch}$, each child node has the scheduling rate, $R^{p_i}_{sch}/4$. This ensures that the total scheduling rate of all the child nodes is not greater than the scheduling rate of their parent node.

When node i determines that some of the child nodes of its parent (i.e., its siblings) are idle (Figure 4-b) that is when $A_t(C(p_i)) < C(p_i)$, it again adjusts its scheduling rate.



a) All child nodes are active (black colored nodes). b) Two child nodes are idle (white).

Figure 4. Any of the child nodes is termed as i, and the grey colored node is the parent of i.

To achieve higher link utilization by taking advantage of excess link capacity, $E(t)$ is distributed to the active child nodes according to their weight factor $\varphi_i(t)$ at a particular time, t. $\varphi_i(t)$ is determined dynamically using the *Calc_NodeWeightFactor()* method. Here the weight factor of the node depends on its weighted average queue length at time t. The weighted average queue length is calculated by using the formula:

$$avg^q_i(t) = \frac{\sum_{j=1}^{N} \alpha^i_j \times q^i_j}{N} \quad (3)$$

Here α^i_j is the priority and q^i_j is the length for queue *j* at time *t*. The weight $\varphi_i(t)$ reflects how the excess link capacity is to be allocated among the active nodes and is normalized such that:

$$\sum_{i \in A_t(C(p_i))} \varphi_i(t) = 1 \quad (4)$$

The excess link capacity is measured by using *Calc_ExcessLinkCapacity()*. It can be calculated by subtracting total scheduling rate of active child nodes from the total scheduling rate of all the child nodes. After calculating the scheduling rate, each node i updates their R_{or}^{i} according to the method *Calculate_SourceRate()*. The originating rate depends on the scheduling rate as well as on the priority for each type of data assigned by the base station.

4.4. Traffic Priority Based MAC Protocol

Our MAC protocol is mainly based on distributed CSMA with RTS/CTS collision avoidance following the strategy of DCF mode of 802.11. The prioritization of traffic can be achieved by differentiating Inter-Frame-Spacing (IFS) and back-off mechanisms. The idea is to assign short IFS and back-off to the higher priority traffic so that they can access the channel earlier than lower priority traffic [8, 16]. Hence, we adopt IEEE 802.11e [24] prioritization with some minor changes. The priority for each queue is mapped to one MAC priority class. Hence, each queue has different Arbitration Inter Frame Space (AIFS), Contention Window (CW), and Persistence Factor (PF) value according to its priority. This way, we can minimize the inter-node priority inversion such that higher priority packet in one node is not likely to be blocked by a lower priority packet in another node.

5. Simulation Results and Analysis

We performed extensive simulation to evaluate the performance of PHTCCP in ns-2 [13]. We used version 2.26 of the ns-2 simulator using the Two Ray Ground propagation model in the air and a single Omni-directional antenna commonly used with ns-2.

5.1. Simulation Parameters

Table 2 shows the simulation parameters. We used directed diffusion [6] as the routing protocol in which during the dissemination of the interest message; the BS assigns priority for each traffic class. IEEE 802.11e MAC protocol provided in ns-2 [13, 24] simulator was used. The default PHY parameters as existed in ns-2.26 for 802.11 MAC has been chosen. We used the 802.11e parameters as used by [11, 24] for the diverse traffic according to their priority. The parameter w_s is a controlling parameter and we empirically set its value to 0.1. We considered 3 different types of traffic originating from a single node and therefore each node was provisioned three queues as shown in Table 2. Traffic type 1 was given the highest priority value of 3, type 2 was given 2, and type 3 was given the value 1.

Table 2. Simulation parameters.

Parameter	Value		
Total Area	100 m X 100 m		
Number of Sensors	100		
Transmission Range	30 m		
Maximum Communication Channel Bit Rate	32 kbps		
Transmission Power	5.85e-5 watt		
Receive Signal Threshold	3.152e-20 watt		
Data Packet Size	33 bytes		
Control Packet Size	3 bytes		
Value of W_s in Eq.2	0.1		
Number of Queues	3		
Size of each Queue	10 packets		
Offered Load	4~16 packets per second (pps)		
Number of Sources	10		
Priority Values used for Queues	**High**	**Medium**	**Low**
	3	2	1
AIFSN	1	2	3
CWmin	7	10	15
Cwmax	7	31	255
Persistence Factor	2	2	2
Simulation Time	60 sec		

5.2. Simulation Results

5.2.1. Threshold of Packet Service Ratio

Figure 5 demonstrates how to determine the threshold of packet service ratio. It shows the percentage of buffer packet drops (irrespective of traffic type) for different packet service ratios considering different packet originating rates. It is noticeable that the increase in the ratio reduces the percentage of packet drops. For different packet originating rates (*pps* – packet per second), the buffer packet drop percentage gradually goes below and reaches to an almost stable state (about 2%) when the packet service ratio becomes 0.5. This is a tolerable value before notifying any congestion. Hence, we set the value of μ to 0.5.

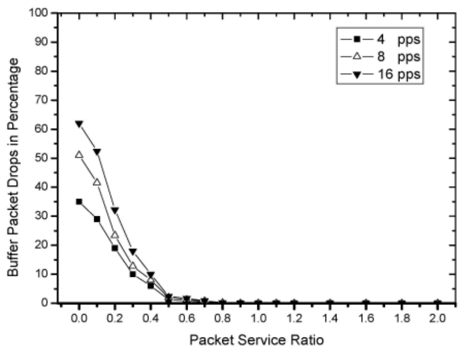


Figure 5. Percentage of packet drops vs packet service ratio for different originating rates to determine the threshold value of μ.

5.2.2. Performance Analysis

Figure 6 illustrates the impact of packet service ratio over weighted average queue length, $avg_i^q(t)$ at the node closest to the sink. It shows that the weighted average queue length increases because of the increase of packet service ratio. This is because, increase in packet service ratio speeds up the packet service rate. In such case, scheduling rate should be increased in such a way that it doesn't cause any buffer overflow.

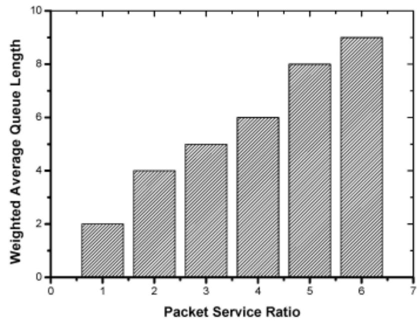


Figure 6. Weighted average queue length for different packet service ratio at the node near the sink.

By setting the value of β to 0.75, a moderate queue length could be maintained. We ran the simulation for 60 seconds and measured the weighted average queue length over time as shown in Figure 7. This figure shows that the maximum weighted average queue length reaches to 9 packets and on an average it stays in between 3 to 5 packets throughout the simulation period. This indicates that PHTCCP maintains moderate queue length to avoid overflow.

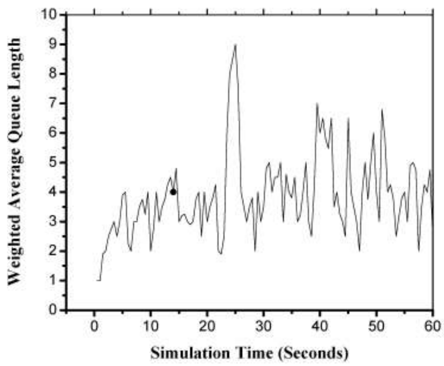


Figure 7. Weighted average queue length over time at the node near the sink reflecting the moderate queue length.

Figure 8 shows the number of different types of packets received by the BS over time. As per the priority given to diverse data, the sink received highest number of traffic type 1 packets and then traffic type 2 packets. Traffic type 3 packets were the lowest in number received throughout the simulation period.

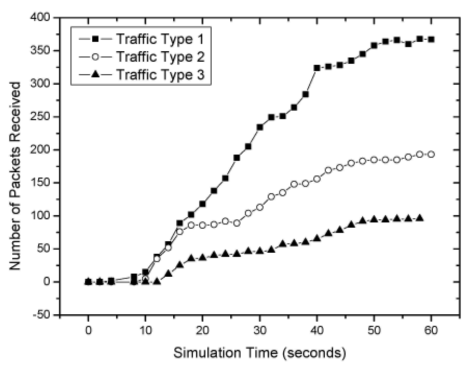


Figure 8. No. of heterogeneous data received by the BS over time.

Figure 9 depicts how the average packet latency of three different types of traffic varies with different work load. The average packet latency was measured from the time a packet originates to the moment it arrives at the BS. As the queuing delay has significant impact on the packet latency, with the increase of the offered load, packets start queuing up and latency also increases but after certain offered load due to the rate control mechanism the latency stabilizes. As in the figure, traffic type 1 suffered lowest delay due to the highest priority than the other two types of traffic which indicates the BS received traffic with diverse latency according to the priority assigned to them.

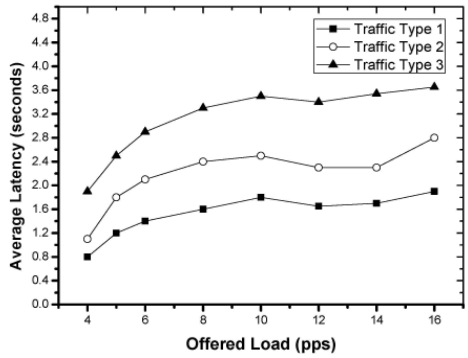


Figure 9. Average latency over different offered load.

Figure 10 compares normalized system throughput among PHTCCP, CCF, No Congestion Control, and PCCP. The system bandwidth is normalized to 1. Within the time between 30 to 50 seconds, some nodes are set idle. Within that interval, PHTCCP achieves higher system throughput than CCF since it allocates the excess link capacity to the active nodes. PCCP also has good performance during that period because of utilizing the remaining capacity but overall throughput for PHTCCP is better than PCCP as it has the efficient rate control for diverse traffic. Whenever packets are transmitted without controlling the transmission rate, the overall system throughput severely decreases which is worse during the period of 30-50 seconds.

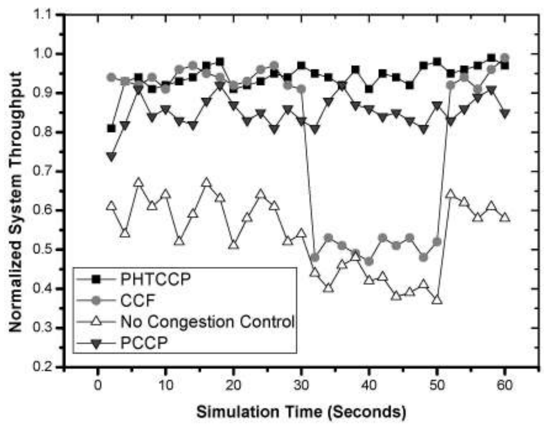


Figure 10. Normalized system throughput over time.

In our simulation we define energy efficiency as: *T/RH* where *T* is the number of bytes transmitted in the whole network during a period of time, R is the number of data bytes received by the BS during the same time and H is the average number of hops a delivered packet travels. *A smaller value indicates better efficiency.* This measurement includes the actual transmission of data, the energy waste due to collision, and the energy waste due to packet drops. In comparison with the four schemes, Figure 11 shows that PHTCCP achieves much better energy efficiency than CCF and PCCP.

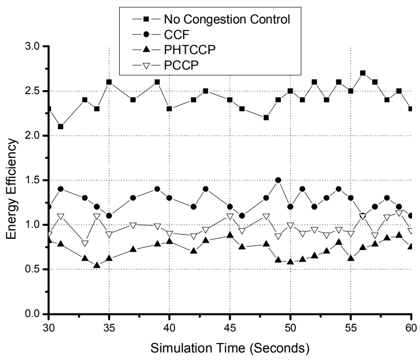


Figure 11. Normalized system throughput over time.

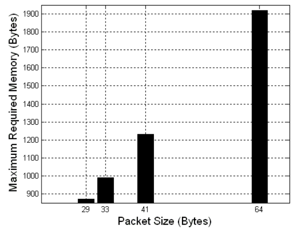
5.2.3. Memory Analysis

Figure 12-a show the maximum memory requirements for different packet sizes (considering 29 byte, 33 byte, 41 byte, and 64 byte packets). The memory requirements can be calculated by using the following equations:

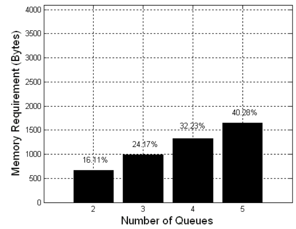
$$M_r = N \times p_l \times q_l \tag{5}$$

Where, p_l is the packet length, N is the total number of queues, and q_l is the size of each queue. As we have considered three queues in total and each queue can contain maximum 10 packets, the memory requirements are 870, 990, 1230, and 1920 bytes for packet sizes of 29, 33, 41, and 64 bytes respectively. Thus it shows that for packet size of 64 bytes, which is long enough for a sensor network application, the memory requirement is less than 2 KB. Hence, if a sensor mote has at least 4KB (4096 Bytes) onboard memory, the maximum memory occupancy would be less than 50% and on an average it is less than 30% which proves that our protocol could well be supported with current specifications of motes.

Figure 12-b shows the memory requirements for different number of queues considering 33 byte packets. With 33-byte packet size, even if we have simultaneously 5 different sensing units (5 different queues), the protocol has 41% memory occupancy if the mote has at least 4 KB onboard memory. When the number of queues is 3, the occupancy is about 25% of total available onboard memory.



a) Maximum memory requirements considering different packet sizes.



b) Memory requirement (bytes) and percentage of memory allocation for different number of queues.

Figure 12. Memory analysis.

6. Conclusions and Future Works

In this paper, we have presented PHTCCP, an efficient congestion control mechanism for heterogeneous data originated from multipurpose sensor nodes. We have demonstrated through simulation results and analysis that it achieves:

1. Desired throughput for diverse data according to the priority specified by the BS.
2. High link utilization.
3. Moderate queue length to reduce packet loss.
4. Relatively low packet drop rate.

Therefore, PHTCCP is energy efficient and provides lower delay. It is also feasible in terms of memory requirements considering the configurations of today's multi-purpose motes.

As our future work, we would like to work on integrating end-to-end reliability mechanism and improving fairness for PHTCCP.

Acknowledgement

This research was supported by the MKE, Korea, under the ITRC support program supervised by the NIPA" (NIPA-2010-(C1090-1021-0003)). Dr. CS Hong is the corresponding author.

References

[1] Dutta P., Grimmer M., Arora A., Bibyk S., and Culler D., "Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral Events," *in Proceedings s of the 4th International Symposium on Information Processing in Sensor Networks*, California, pp. 497-502, 2005.

[2] Ee C. and Bajcsy R., "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," *in Proceedings s of the 2nd International Conference on Embedded Networked Association for Computing Machinery Sensor Systems*, USA, pp. 148-161, 2004.

[3] Gu L., Jia D., Vicaire P., Yan T., Luo L., Tirumala A., Cao Q., He T., Stankovic J.,

Abdelzaher T., and Krogh B., "Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments," *in Proceedings of the 3rd International Conference on Embedded Networked Association for Computing Machinery Sensor Systems*, USA, pp. 205-217, 2005.

[4] MICA2-Datasheet, available at: http://www.xbow.com/ Products/Product-pdf-files/Wireless-pdf/MICA2-Datasheet.pdf, last visited 2007.

[5] Hull B., Jamieson K., and Balakrishnan H., "Mitigating Congestion in Wireless Sensor Networks," *in Proceedings of the 2nd International Conference on Embedded Networked Association for Computing Machinery Sensor Systems*, USA, pp. 134-147, 2004.

[6] Intanagonwiwat C., Govindan R., and Estrin D., "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *in Proceedings of 6th Mobicom*, USA, pp. 56-67, 2000.

[7] Iyer Y., Gandham S., and Venkatesan S., "STCP: A Generic Transport Layer Protocol for Wireless Sensor Networks," *in Proceedings of IEEE International Conference on Computer Communications and Networks*, USA, pp. 449-454, 2005.

[8] Kanodia V., Li C., Sabharwal A., Sadeghi B., and Knighty E., "Distributed Priority Scheduling and Medium Access in Ad Hoc Networks," *Computer Journal of Wireless Networks*, vol. 8, no. 1, pp. 466-466, 2002.

[9] Karl H. and Willig A., *Protocols and Architectures for Wireless Sensor Networks*, Wiley, 2006.

[10] Kim B. and Kang S. "IEEE 802.15.4 MAC-Based Location-ID Exchange Protocol for Realizing Micro-Cell Connectionless Location-Awareness Services," *Computer Journal of Science and Engineering*, vol. 2, no. 4, pp. 412-427, 2008.

[11] Mangold S., Choi S., May P., Klein O., Hiertz G., and Stibor L., "IEEE 802.11e Wireless LAN for Quality of Service," *in Proceedings of the European Wireless*, Italy, pp. 32-39, 2002.

[12] Monowar M., Rahman M., Pathan A., and Hong C., "Congestion Control Protocol for Wireless Sensor Networks Handling Prioritized Heterogeneous Traffic," *in Proceedings of SMPE'08 with MobiQuitous*, Ireland, pp. 21-25, 2008.

[13] NS-2, available at: http://www.isi.edu/nsnam/ns, last visited 2005.

[14] Paek J. and Govindan R. "RCRT: Rate-Controlled Reliable Transport for Wireless Sensor Networks," *in Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, Australia, pp. 305-320, 2007.

[15] Pathan A., Heo G., and Hong C., "A Secure Lightweight Approach of Node Membership Verification in Dense HDSN," *in Proceedings of the IEEE Military Communications Conference*, USA, pp. 29-31, 2007.

[16] Pattara-Atikom W., Krishnamurthy P., and Banerjee S., "Distributed Mechanisms for Quality of Service in Wireless LANs," *IEEE Wireless Communication*, vol. 10, no. 3, pp. 26-34, 2003.

[17] Rangwala S., Gummadi R., Govindan R., and Psounis K., "Interference-Aware Fair Rate Control in Wireless Sensor Networks," *in Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Italy, pp. 63-74, 2006.

[18] Sankarasubramaniam Y., Akan O., and Akyildiz I., "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," *in Proceedings of 4th Association for Computing Machinery MobiHoc*, USA, pp. 177-188, 2003.

[19] Wan C., Eisenman S., Campbell A., and Crowcroft J., "Overload Traffic Management for Sensor Networks," *Association for Computing Machinery ToSN*, vol. 3, no. 4, pp. 4-18, 2007.

[20] Wan C., Eisenman S., and Campbell A., "CODA: Congestion Detection and Avoidance in Sensor Networks," *in Proceedings of Association for Computing Machinery Sensor Systems*, USA, pp. 266-279, 2003.

[21] Wang C., Li B., Sohraby K., Daneshmand M., and Hu Y., "Upstream Congestion Control in Wireless Sensor Networks Through Cross-Layer Optimization," *IEEE on Selected Areas in Communication*, vol. 25, no. 4, pp. 786-795, 2007.

[22] Wang C., Li B., Sohraby K., Daneshmand M., and Hu Y., "A Survey of Transport Protocols for Wireless Sensor Networks," *IEEE Network*, vol. 20, no. 3, pp. 34-40, 2006.

[23] Warneke B. and Pister K., "MEMS for Distributed Wireless Sensor Networks," *in Proceedings of 9th IEEE International Conference on Electronics, Circuits and Systems*, USA, pp. 291-294, 2002.

[24] Wietholter S. and Hoene C., "Design and Verification of an IEEE 802.11e EDCF Simulation Model in ns-2.26," *Technical Report,* Telecommun Network Group, Technische Universitat Berlin, 2003.

Muhammad Monowar received Ph.D. degree in Computer Engineering in 2011 from Networking lab, Department of Computer Engineering, Kyung Hee University (KHU), South Korea. He received B.Sc. degree in Computer Science and Information Technology (CIT) from Islamic University of Technology, Bangladesh (IUT) in 2003. He is currently an Assistant professor at Computer Science & Engineering department, University of Chittagong, Bangladesh. His research interest includes Wireless Ad hoc and Sensor Networks.

Obaidur Rahman is a lecturer (study leave) at CSE Department, Dhaka University of Engineering and Technology, Bangladesh. Currently, he is pursuing his PhD in computer engineering at KHU, Korea. He received his BSc in Computer science and Information Technology (CIT) from IUT, Bangladesh in 2003, and his MS in computer engineering from KHU, Korea in 2008.

Al-Sakib Khan Pathan received his PhD degree in computer engineering in 2009 from Kyung Hee University, South Korea. He received his BSc degree in computer science and information technology from Islamic University of Technology (IUT), Bangladesh in 2003. He is currently an assistant professor at Computer Science Department in the International Islamic University Malaysia (IIUM), Malaysia. His research interest includes wireless sensor networks, network security, and e-services technologies. He is the editor of several journals and books, and organizing committee member of multiple conferences/workshops. He is a member of IEEE, IEEE ComSoc Bangladesh Chapter, and several other international organizations.

Choong Seon Hong received his BS and MS in electronic engineering from Kyung Hee University, Korea, in 1983, 1985, respectively. He received his PhD at Keio University in 1997. He worked for the Telecommunications Network Lab, KT as a senior member of technical staff and as a director of networking research team until August 1999. Since 1999, he has been working as a professor of the School of Electronics and Information, KHU. He has served as a PC and an organizing committee member for International Conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN, and TINA. His research interests include ad hoc networks, network security and network management. He is a member of IEEE, IPSJ, KIPS, KICS, and KIISE.