



Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks

Mohammad Hossein Yaghmae^{a,*}, Donald A. Adjeroh^b

^a Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad 91775, Iran

^b Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506, United States

ARTICLE INFO

Article history:

Received 11 June 2008

Received in revised form 18 February 2009

Accepted 22 February 2009

Available online 3 March 2009

Responsible Editor: T. Melodia

Keywords:

Quality of service

Multimedia wireless sensor networks

Congestion control

Service differentiation

Transport protocols

ABSTRACT

In wireless multimedia sensor networks (WMSNs) a sensor node may have different types of sensor which gather different kinds of data. To support quality of service (QoS) requirements for multimedia applications having a reliable and fair transport protocol is necessary. One of the main objectives of the transport layer in WMSNs is congestion control. We observe that the information provided may have different levels of importance and argue that sensor networks should be willing to spend more resources in disseminating packets carrying more important information. Some applications of WMSNs may need to send real time traffic toward the sink node. This real time traffic requires low latency and high reliability so that immediate remedial and defensive actions can be taken when needed. Therefore, similar to wired networks, service differentiation in wireless sensor networks is also an important issue. We present a priority-based rate control mechanism for congestion control and service differentiation in WMSNs. We distinguish high priority real time traffic from low priority non-real time traffic, and service the input traffic based on its priority. Simulation results confirm the superior performance of the proposed model with respect to delays, delay variation and loss probability.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

A wireless multimedia sensor network (WMSN) [1] is a set of sensor nodes, whereby the nodes are equipped with multimedia devices such as cameras and microphones. Thus a WMSN will have the capability to transmit multimedia data, such as still pictures, stream video, voice, animal sounds and monitoring data. One of the most important requirements of applications in WMSNs is the low delay bounds. Some applications of WMSNs also need relative resilience to losses. WMSNs can support different types of traffic classes. The required reliability and delay guarantees can be provided by communication protocols with real-time support.

Similar to wireless sensor networks (WSNs) [2], applications of WMSNs share different characteristics, such as resource constraints, unbalanced mixture traffic, data redundancy, network dynamics and energy balance. There are many different resource constraints in WMSNs involving energy, bandwidth, memory, buffer size and processing capability. Given the physically small nature of the sensors, and that multimedia applications typically produce huge volumes of data requiring high transmission rates and extensive processing, a fundamental concern in WMSNs is the issue of power consumption. Thus, developing protocols, algorithms and architectures to maximize the network lifetime while satisfying the quality of service requirements of the applications represents a critical problem. In most WSN and WMSN applications, traffic mainly flows from a large number of sensor nodes to a base station (sink) node. Therefore, to meet the quality of service requirements and to use the network resources in a fair and efficient manner, this characteristic of WMSNs

* Corresponding author. Tel.: +98 5118830501; fax: +98 5118629530.
E-mail addresses: yaghmae@ieee.org (M.H. Yaghmae), don@csee.wvu.edu (D.A. Adjeroh).

becomes a concern, and must be considered. Furthermore, given the relatively high redundancy in the sensor data, techniques such as data compression, data fusion and aggregation are very important in maintaining robustness, while decreasing redundancy in the data. Another important characteristic of WSNs and WMSNs is the dynamic changes in topology and the unreliable nature of wireless networks. This is primarily due to changes in node mobility and/or wireless channel failure. To meet quality of service requirements, these natural characteristics of this kind of networks must be considered in designing the required protocols.

In many applications of wireless multimedia sensor networks (WMSNs), a sensor node may have different kinds of sensor which gather different kinds of data [1]. We observe that the information provided may have different levels of importance and argue that the sensor networks should be willing to spend more resources in disseminating packets carrying more important information. In heterogeneous multimedia sensor networks, a wireless node may contain different sensors including: audio sensor, video sensor and scalar sensor. As the priority of these heterogeneous traffics are not the same, it is important to consider differentiated services architecture in these networks. Similar to wired networks, in wireless multimedia sensor networks, there exist two types of traffic, real time and non-real time. Real-time traffic has hard time constraints such as delay and jitter while being more tolerant to packet losses. As this kind of traffic is too important, usually there is no transmission rate control for real-time traffic. However, to avoid congestion collapse and unfair resource sharing, there is a need for an effective rate control protocol for real-time flows. For some applications of WMSNs, there may be need to send real time traffic with low latency and high reliability toward the sink node in order to facilitate immediate remedial and defensive actions. For example, in a typical intruder detection system, data are sent to a basic station (sink node) periodically. However, whenever an important event occurs in the system, the sensor node that detected the event should send some alarm message to the sink. This alarm messages could take the form of multiple packets containing information such as the time and place of intrusion. Usually this kind of high priority traffic is bursty. That is, high priority traffic is generated only within a short period of time while low priority traffics usually exist in the network and produce thousands of packets generated periodically. Thus service differentiation is very important in sensor networks, especially in WMSNs. To provide service differentiation in WMSNs, it is necessary to assign a different priority to each traffic source.

To provide end-to-end QoS on an IP based network, the internet engineering task force (IETF) defined the differentiated services (DiffServ) model [3–5]. In this paper, starting with the priority based congestion control protocol (PCCP) [6], we develop a model for WMSNs which can provide requested quality of service for each traffic class. In practice, QoS provision is necessary in all layers of the WSN architecture. The QoS requirements in the application layer are specified by the users. Some parameters such as system lifetime, response time, data novelty, detection

probability, data reliability and data resolution are defined at the application layer. In the network layer some QoS requirements such as: path latency, routing maintenance, congestion probability, routing robustness and energy efficiency are defined. In the MAC layer the communication range, throughput, transmission reliability and energy efficiency are major QoS requirements. The main objective of the proposed algorithm is to prevent congestion in the network using exact rate allocation. The proposed algorithm is able to determine the traffic rate of each sensor nodes based on the current congestion in the network. It is clear that this mechanism is implemented in higher layer and is independent from routing and MAC layer. To provide the requested QoS in the network, all layers should contain the required mechanisms for QoS provision. In the proposed algorithm we assume that the lower layers are capable of providing the required functions. This is because the actual service deferentiation cannot be achieved unless the lower layers can support it.

The proposed model can support two major traffic classes, namely, expedited forwarding (EF) class which is assigned to real time traffic, and non real-time traffic (NRT) class. The non real-time traffics can be further divided into different levels of importance based on their requirement for network resources. We consider four types of traffic class: real time traffic (EF class), high priority non real time traffic (NRT1 class), medium priority non real time traffic (NRT2 class) and low priority non real time traffic (NRT3 class). Each traffic class is assigned a different priority. Since guaranteeing a low delay bound is an important issue for real time traffic, the proposed model guaranties a low delay bound for this type of traffic. Real time traffic is buffered in a separate queue with low buffer size, while non real-time traffic is managed using the active queue management algorithm [7].

The reminder of this paper is organized as follows. Section 2 describes congestion control in wireless sensor networks and related work. Section 3 presents the proposed congestion control and service differentiation protocol. In section 4, we use computer simulation to evaluate the performance of the proposed model. Section 5, concludes the paper.

2. Related work

Congestion control is an important issue in transport protocols. Congestion is also a difficult problem in wireless sensor networks. It not only wastes the scarce energy due to a large number of retransmissions and packet drops, but also hampers the event detection reliability. Congestion in WSNs and WMSNs has a direct impact on energy efficiency and application QoS. Two types of congestion could occur in sensor networks [8]. The first type is node-level congestion that is caused by buffer overflow in the node and can result in packet loss, and increased queuing delay. Not only can packet loss degrade reliability and application QoS, but it can also waste the limited node energy and degrade link utilization. In each sensor node, when the packet-arrival rate exceeds the packet service rate, buffer overflow may occur. This is more likely to occur at sensor nodes close

to the sink, as they usually carry more combined upstream traffic. The second type is link-level congestion that is related to the wireless channels which are shared by several nodes using protocols, such as CSMA/CD (carrier sense, multiple access with collision detection). In this case, collisions could occur when multiple active sensor nodes try to seize the channel at the same time.

To mitigate the effect of congestion in WMSNs, we need an effective way to control it. Each congestion control solution consists of three important components: congestion detection, congestion notification, and rate adjustment. In traditional TCP protocol, congestion is detected at the end nodes based on a timeout or redundant acknowledgments. In wireless sensor networks, packet loss detection and notification can be either end-to-end or link-by-link. In the end-to-end approach, the end-points are responsible for loss detection and notification. In the link-by-link method, intermediate nodes detect and notify packet loss. The end-to-end approach is not very effective for WSNs. This is because the control messages that are used for end-to-end loss detection would utilize a return path consisting of several hops, and this is not energy efficient. Furthermore the control messages travel through multiple hops and could be lost with a high probability due to either link error or congestion. In link-by-link loss detection and notification, a pair of neighbouring nodes are responsible for loss detection, and can enable local retransmission that is more energy efficient, as compared to the end-to-end approach. Link-by-link congestion detection in sensor networks is proactive, often makes use of a congestion indicator, and generally has a better performance than traditional end-to-end congestion detection. Different congestion indicators have been proposed, such as, queue length [9,10], packet service time [8], or the ratio of packet service time over packet inter-arrival time at the intermediate nodes [6].

Upon detecting congestion in the network, the transport protocol needs to propagate congestion information from the congested node to the upstream sensor nodes or the source nodes that contribute to congestion. This can be done explicitly by sending a special control message to the other sensors, or implicitly using piggybacking technique in data packets. When a node receives a congestion notification message, it should adjust its transmission rate using a rate control techniques such as additive increase multiplicative decrease (AIMD).

Various congestion control methods have been studied for wireless sensor networks [6,8,11–17]. Among the most popular of these techniques are CCF and PCCP. Congestion control and fairness (CCF) was proposed in [8] as a distributed and scalable algorithm that eliminates congestion within a sensor network and ensures the fair delivery of packets to a sink node. CCF exists in the transport layer and is designed to work with any MAC protocol in the data-link layer. In the CCF algorithm, each node measures the average rate r at which packets can be sent from the node, divide the rate r among the number of children nodes, adjust the rate if queues are overflowing or about to overflow and propagate the rate downstream. CCF uses packet service time to deduce the available service rate. Congestion information is implicitly reported. It controls congestion in a hop-by-hop manner and each node uses

exact rate adjustment based on its available service rate and child node number. CCF is known to guarantee simple fairness.

As was shown in [6], CCF is a non-work-conserving algorithm. To explain the non-work-conserving property of CCF, suppose that a root node is connected to two nodes A and B. The non-work-conserving property of CCF implies that the root node will wait until the required number of packets has been received and transmitted from node B before considering packets from node A. This also implies that CCF cannot effectively allocate the remaining capacity, resulting in a low throughput, especially, when some nodes do not have any packet to send. Further, as was shown in [6], CCF has another major problem. The rate adjustment in CCF relies only on packet service time which could lead to low utilization when some sensor nodes do not have enough traffic or there is a significant packet error rate.

Priority based congestion control protocol (PCCP) was proposed in [6]. PCCP is an upstream congestion control protocol for WSNs which measures the congestion degree as the ratio of packet inter-arrival time to the packet service time. Based on the introduced congestion degree and node priority index, PCCP utilizes a cross-layer optimization and imposes a hop-by-hop approach to control congestion. It has also been shown that PCCP achieves efficient congestion control and flexible weighted fairness for both single-path and multipath routing.

In wireless sensor networks data are normally generated and sent to the sink periodically. However, a burst of data traffic can also be suddenly generated when an important event is triggered or detected. So, in wireless sensor networks different data packets might have different importance. For packets containing information with higher importance, the network should make more effort in delivering them. This highlights a need for having service differentiation in sensor networks. Service differentiation in wireless sensor networks is a new research area and a few methods have been proposed [18–24].

3. The proposed model

The proposed model consists of two major units, namely: Congestion control unit (CCU) and service differentiation unit (SDU). We describe each unit in detail in the following subsections.

3.1. Congestion control unit (CCU)

Our approach is motivated by the apparent limitations of existing popular schemes, such as the PCCP and CCF. The PCCP, which is a new congestion control protocol for WSNs has some major problems. The first is that it considers fixed service time which is calculated using the output rate of each node. For real wireless sensor networks, this assumption does not hold due to the variable nature of the physical link. The second problem is that PCCP performs very poorly in providing relative priority in the case of random service time. An analysis of the PCCP algorithm [6] shows that in the case of low congestion, PCCP will

increase the scheduling rate and source rate of all traffic sources without paying any attention to their priority index. In the case of high congestion, PCCP will decrease the sending rate of all traffic sources based on their priority index. The algorithm we propose solves this problem by a proper adjustment of the rate at each node. We adjust the sending rate of each traffic source based on its congestion condition and its priority index. Another problem is that PCCP considers only geographical priority and cannot discriminate between different traffic classes.

Fig. 1, shows the architecture of the proposed congestion control unit. Similar to the other congestion control protocols, it consists of three parts namely, congestion detection unit (CDU), congestion notification unit (CNU), and rate adjustment unit (RAU). The CDU is responsible for detecting any congestion in advance. The CDU measures the input rate to determine the congestion intensity. In the proposed congestion control protocol, using a measurement algorithm, each sensor node measures its input traffic load and calculates the difference between its input rate and its maximum allowable transmission rate. The output of the CDU is the difference between the input rate and the output rate which can be a positive or a negative number. In each predefined time interval, each parent node calculates the sending rate of all its child traffic sources as well as its local traffic source. Each sensor node may have different priorities since sensor nodes might be installed with different kinds of sensors in a given environment. Thus, the upstream node also considers the priority of each of its child nodes in calculating the rate of the child nodes. Based on the current congestion index and the source traffic priority, the RAU calculates the new rate for each child traffic source as well as its local traffic source. The new rate is sent to the CNU unit which is responsible for notifying all the child nodes of the new rate. To decrease energy consumption, CNU uses an implicit congestion notification by adding the new rate of each child node to the sending data of each sensor node. When a node detects any congestion, it is expected to adjust its traffic rate accordingly.

3.2. Service differentiation unit

We propose a service differentiation unit needed to support differentiated services in WMSNs.

Classifying network traffic allows us to organize traffic into traffic classes on the basis of whether the traffic

matches a specific criteria. Classifying network traffic is the foundation for enabling many quality of service (QoS) features on the network. The goal of network traffic classification is to group traffic based on user-defined criteria so that the resulting groups of network traffic can then be subjected to specific QoS treatments. The QoS treatments might include faster forwarding by intermediate nodes or reduced probability of the traffic being dropped due to lack of buffering resources. Identifying and organizing network traffic is the foundation for applying the appropriate QoS feature to that traffic, making it possible to allocate network resources to deliver optimal performance for different types of traffic. For example, high priority network traffic or traffic matching a specific criteria can be singled out for special handling, and thus, help to achieve peak application performance. In the differentiated service architecture, the network should be able to support different levels of traffic classes in the network. The number of traffic classes is an administrative task, and is application dependent. In some applications of WSNs we can define different level of quality of service for different network traffics and assign a traffic class to each of them. In the proposed model, we allocate a buffer for each traffic class, so by increasing the number of traffic classes the number of required buffers and also the hardware requirement is increased. Without loss in generality, we consider only four different traffic classes in this work. The system can be tuned to support a larger number of traffic classes.

The proposed model can support four different traffic classes namely: real time traffic class (EF class); high priority, non real-time traffic class (NRT1 class); medium priority, Non real-time traffic class (NRT2 class); and low priority, non real-time traffic class (NRT3 class). The EF class is assigned to high priority real time traffic such as alarm data or real time audio/video. Non real-time traffic is divided into three different classes. For this kind of traffic, having a low delay is not too important. High priority traffic classes need to have high throughput and low delay bound. The constraints on the throughput ($Throu.$) and delay for the EF and NRT classes are given below:

$$\begin{aligned} Throu_{.EF} &\geq Throu_{.NRT1} \geq Throu_{.NRT2} \geq Throu_{.NRT3} \\ Delay_{.EF} &\leq Delay_{.NRT1} \leq Delay_{.NRT2} \leq Delay_{.NRT3} \end{aligned} \quad (1)$$

We note that real-time traffic with low delay bound may not always require high throughput. But for simplicity, we make this assumption in our simulations.

Fig. 2 shows a typical WMSN with different sensor types. In Fig. 2, each node has different traffic classes. For example node 1 has only two types of traffic, EF and NRT1, while node 6 contains all of possible traffic classes EF, NRT1, NRT2 and NRT3.

We use separate queues for each traffic class. The queuing model of each node is shown in Fig. 3. To discriminate traffic classes from each other, the wireless node adds a traffic class identifier to its local sensor packets and puts them in the proper queue. This identifier represents the traffic class of each packet. As shown in Fig. 3, in each intermediate wireless node, arriving packets are sent to different queues according to their traffic class. We consider the single-path network, so each node has only one

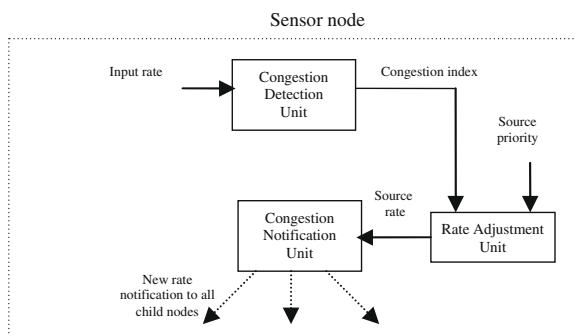


Fig. 1. Structure of the proposed congestion control unit.

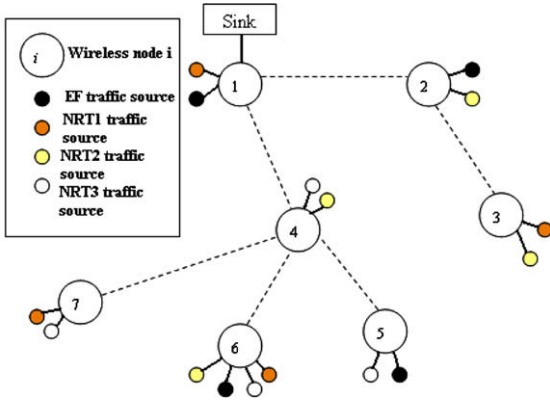


Fig. 2. A typical WMSN with heterogeneous traffic classes.

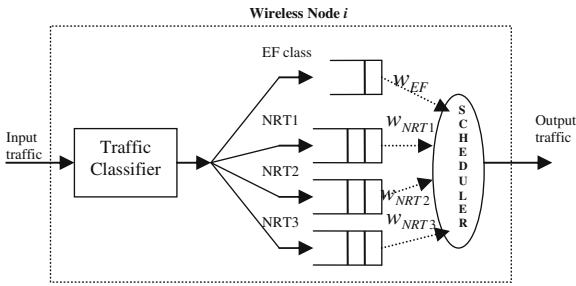


Fig. 3. Queuing model of each sensor node in the proposed model.

next hop. For non real time classes the RED protocol [7] is used which help us to achieve service differentiation with-in sensor nodes.

We use a similar congestion indicator strategy used in the RED active queue management algorithm. For this purpose, for each traffic class, a separate queue is maintained. In each queue, two different fixed thresholds are defined. When the queue length is less than a minimum threshold, there is no congestion in the queue and so the congestion index is set to zero (low congestion). In this case, based on the priority of the child node, the child transmission rate is modified. On the other hand, when queue length is greater than a maximum threshold, there is a significant congestion in the network, thus the congestion index is set to 1 (high congestion). In this case the child node should decrease its transmission rate to avoid any packet loss in the corresponding parent node. Whenever queue length is between the two thresholds the congestion index is set to a number between 0 and 1, depending on the queue length.

To provide quality of service for high priority EF flows in a wireless multimedia sensor network, we use a priority queuing (PQ) protocol which prioritizes the packet transmission process at each node.

3.3. Algorithm description

In the proposed model, each sensor node i has two different priority indexes: traffic class priority (P_{TC}^i) and Geo-

graphical priority (P_{GE}^i). Suppose each node i has different source traffic classes. Let SP_j^i denote the traffic source priority j in sensor node i . The value of source priority SP_j^i could be set manually to achieve service differentiation. For high priority traffic the value of SP_j^i should be high enough so that we can discriminate it from the other low priority traffics. In each node i the value of P_{TC}^i is equal to the sum of source priority SP_j^i and is calculated as follows:

$$P_{TC}^i = \sum_j SP_j^i \quad (2)$$

where j is the traffic class, $j \in \{EF, NRT1, NRT2, NRT3\}$.

In wireless sensor networks, sensor nodes might be either outfitted with different sensors or geographically deployed in different locations and therefore may have different importance or priority and need to gain different throughput. Therefore weighted fairness is required. To satisfy this requirement of sensor networks, we define the geographical priority, P_{GE}^i , which is set manually by the network manager.

The total priority of node i , P^i , is defined as the product of these two priorities:

$$P^i = P_{TC}^i \cdot P_{GE}^i \quad (3)$$

Let $C(i)$ be the set of i 's child nodes. Then for each node i the global priority, GP^i , is calculated as:

$$GP^i = \sum_{k \in C(i)} GP^k + P^i \quad (4)$$

If a node does not have any child, then its global priority is equal to its total priority. Note that GP^i is calculated only for active traffic sources. If a traffic source is not active, then regardless to its type of traffic class, the value of SP_j^i is set to zero. This makes the algorithm to share the existing capacity only between active nodes.

The proposed rate adjustment algorithm is described next.

3.4. Rate adjustment algorithm

A. Initialization phase:

A1: Measure the maximum output rate (r^{Sink}) at the sink node. Let T_s^{Sink} denote the service time¹ of the current packet in sink node. Using the exponential weighted sum, compute \bar{T}_s^{Sink} , the average service time:

$$\bar{T}_s^{Sink} = (1 - \alpha)\bar{T}_s^{Sink} + \alpha \cdot T_s^{Sink} \quad (5)$$

where α is a constant, $0 \leq \alpha \leq 1$.

A2: Compute r^{Sink} , the sink output rate:

$$r^{Sink} = \frac{1}{\bar{T}_s^{Sink}} \quad (6)$$

¹ The average service time \bar{T}_s^{Sink} is the time taken to successfully transmit a data packet over the MAC layer. It is measured starting from the time when the network layer first sends the packet to the MAC layer to the time when the MAC layer notifies the network layer that the packet has been transmitted.

A3: For each child node i , the sink node calculates and propagates the maximum transmission rate, r_{\max}^i , based on the child node's global priority (GP^i) and the sink's global priority (GP^{Sink}). This is done as follows:

$$r_{\max}^i = r^{\text{Sink}} \cdot \frac{GP^i}{GP^{\text{Sink}}} \quad (7)$$

where GP^{Sink} is the sum of the global priority of all the sink's child nodes. Let $C(\text{Sink})$ be the set of child nodes with sink as their parent. Then GP^{Sink} is computed as follows:

$$GP^{\text{Sink}} = \sum_{j \in C(\text{Sink})} GP^j \quad (8)$$

A4: Repeat the above steps at each network node to assign each node its initial maximum transmission rate.

B: At each periodic time interval T_{measure} (Sink node):

B1: Compute the input rate at the sink node using the rates from its child nodes:

$$r_{\text{in}}^{\text{Sink}} = \sum_{j \in C(\text{Sink})} r_{\text{out}}^j \quad (9)$$

where $r_{\text{in}}^{\text{Sink}}$ is the total input rate to the sink node and r_{out}^j is the output rate of the j th child node from the sink.

B2: Compute Δr^{Sink} , the rate difference at the sink node:

$$\Delta r^{\text{Sink}} = \beta \cdot r_{\text{in}}^{\text{Sink}} - r_{\text{in}}^{\text{Sink}} \quad (10)$$

where β is a constant close to 1.

B3: Sink node calculates and propagates the new maximum transmission rate for its i -th child node as follows:

$$r_{\text{out}}^i = r_{\text{out}}^i + \Delta r^{\text{Sink}} \cdot \frac{GP^i}{GP^{\text{Sink}}} \quad (11)$$

C: At each periodic time interval T_{measure} (Other nodes):

C1: Each parent node i calculates its total input rate using the rates from each of its child nodes:

$$r_{\text{in}}^i = \sum_{j \in C(i)} r_{\text{out}}^j \quad (12)$$

where $C(i)$ is the set of i 's child nodes, and r_{out}^j is the output rate of the j th child of parent node i .

C2: Each parent node i calculates the rate difference Δr^i :

$$\Delta r^i = \beta \cdot r_{\text{in}}^i - r_{\text{in}}^i \quad (13)$$

C3: Each parent node i calculates and propagates the new output rate of each of its child nodes j as follows:

$$r_{\text{out}}^j = r_{\text{out}}^j + \Delta r^i \frac{GP^j}{GP^i} \quad (14)$$

C4: The above steps are repeated at each network node to obtain the updated transmission rate.

In the proposed algorithm we define two different rates for each sensor node i which are the total input rate to the node (r_{in}^i) and the output rate (r_{out}^i). Each parent node i calculates its total input rate from all of its child nodes (r_{in}^i). Furthermore each parent node i calculates and propagates the new output rate of each of its child. So the output rate of each sensor node is calculated only by its parent. The rate of data generation at the parent node is calculated by its parent (the parent of the parent). For the sink node which has no parent, the transmission rate is calculated as the inverse of its average service time.

Based on the algorithm above, when some child nodes do not have enough traffic, r_{in}^i the input rate to the parent node i will be decreased. This will in turn lead to an increase in Δr^i , resulting in an increase in the transmission rate of the other child nodes. The amount of increase in the transmission rate is not the same for all the child nodes. It depends on the global priority of the child node. This means that high priority nodes will get larger increase to their transmission rates than low priority nodes. Therefore each node can scale-up its rate according to the value of its global priority and current congestion degree. On the other hand, when some child nodes produce more traffic, then input rate to the parent node i will become greater than its previous value. This makes the value of Δr^i to become negative. So to prevent any packet loss and high delay, each child node decreases its transmission rate. The amount of decrease is dependent on the global priority of each node. This leads to a guarantee in fairness and in high link utilization.

As mentioned earlier, in the proposed model, the rate assignment to each child node is based on the available resources and bandwidth at its parent node. Each parent node calculates the rate difference Δr^i as given by (13) and calculates and propagates the new output rate of each of its child j using (14). When total increment in the rate at a parent is negative, then the rates for the children are decreased. Thus, at this time the model cannot support a request for an increase in rate, given the limited network resources (bandwidth).

It is easy to analyze the computational burden on the sensor network using the above algorithm. Let $C(i)$ be the set of child nodes with node i as their parent. Let $n_i = |C(i)|$. According to (3) and (4), each sensor node i can compute its global priority using one multiplication and a few summations. The number of summations is exactly $n_i + 1$, and thus depends on the number of child nodes from sensor node i . For typical sensor networks, this will not be too much computational burden at the sensor node. In terms of theoretical message complexity analysis, we can observe that the number of messages generated at each periodic interval is $O(n)$, where n is the number of nodes in the network. Each parent node i will send n_i update messages, one to each child node. Each child node will in turn send rate update messages to its own children only. Thus, at any given periodic interval, the total number of messages over the sensor network will be linear with respect to the network size.

4. Simulation results

In this section, we use computer simulation to evaluate the performance of the proposed model under different scenarios. For this purpose, we simulated a wireless network topology as given in Fig. 4 where the values for M , N and K are set to 5, 3, 2, respectively. Each node may have different traffic classes: EF, NRT1, NRT2 and NRT3. We evaluate the performance of both the congestion control unit and the service differentiation unit. The evaluation parameters are queuing delay and normalized throughput, packet loss and achieved priority. The reported simulation results represent the average from 10 runs of the experiment, at a 95% confidence interval.

4.1. Evaluation of congestion controller

To evaluate the performance of the proposed congestion controller, we consider the general network topology shown in Fig. 4. Each sensor node has only one EF traffic source. The buffer size in each node is set to 100 packets.

In the first simulation trial, we assume that all sensor nodes have the same total priority equal to 1 ($P^i = 1, i = 1, \dots, M + N + K$). The service time for the sink node was fixed at 0.001 s. All sensor nodes start to send their data at the start of simulation time and stop at the end of simulation time. Simulation time was set to 100 s.

Table 1 shows results for both fixed and exponential service time. In the case of fixed service time all algorithms have approximately the same throughput. Both CCF and the proposed protocol have a slightly better throughput than the PCCP. The PCCP has a slightly better average queuing delay than the other algorithms. In this scenario the loss probability of all protocols is equal to zero.

To compare the performance of CCF and PCCP with that of the proposed algorithm under real-life conditions, we consider a random service time. In this case we repeated the same experiment, but assuming a sink node with an exponential service time with a mean of 0.001 s. Table 1

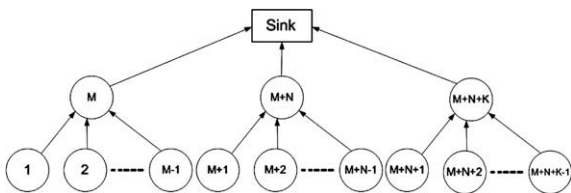


Fig. 4. Network topology used in the simulation.

Table 1 Simulation results in the case of fixed and exponential service time.

Performance metric	Fixed service time			Exponential service time		
	CCF	PCCP	Proposed	CCF	PCCP	Proposed
Normalized throughput	0.999	0.968	0.980	0.988	0.846	0.928
Loss probability	0	0	0	0.065	0.025	0.0005
Delay (s)	0.001	0.0005	0.0007	0.081	0.021	0.009

shows that, with respect to loss probability and queuing delay, the proposed protocol has a better performance than CCF and PCCP. Its throughput is also better than that of PCCP. Although the CCF has the highest throughput, it also has the highest loss probability and the longest queuing delay.

Congestion in WSNs has a direct impact on energy efficiency and application QoS: not only can packet loss degrade reliability and application QoS, but it can also waste the limited node energy and degrade link utilization. When the packet-arrival rate exceeds the packet service rate at a sensor node, buffer overflow may occur resulting in data loss. Thus, having a low loss probability is very important in WSNs. When data is lost, the sender will need to retransmit the lost data. Thus, with a high loss probability in a sensor network, the remaining energy is consumed very quickly (on retransmissions), leading to a significant reduction in the total life time of the network.

In the next trial, in the case of fixed service time, sensor node 3 was made inactive during the time interval [30 s, 70 s]. From Fig. 5a, we see that because CCF cannot effectively allocate the remaining capacity and use work-conservation scheduling algorithm, it has a significantly lower throughput in the interval [30 s, 70 s].

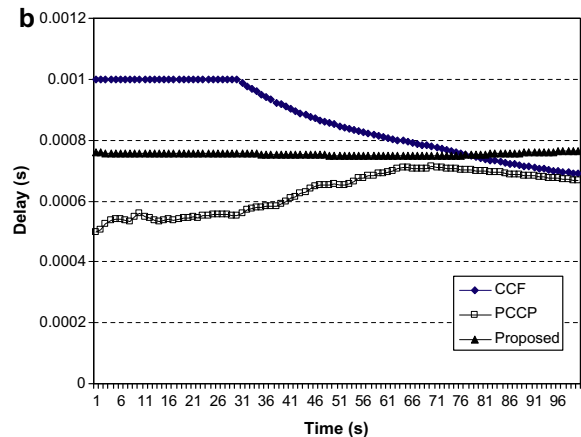
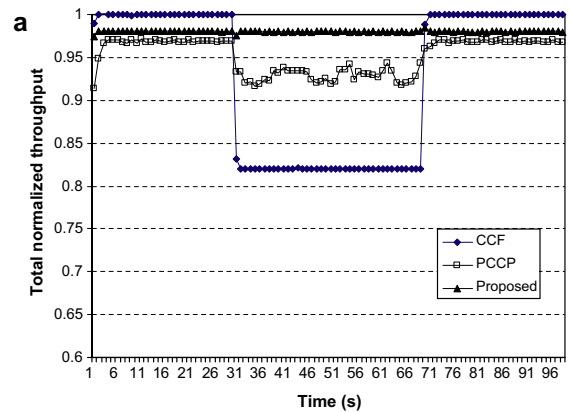


Fig. 5. Performance using a fixed service time: (a) total normalized throughput and (b) queuing delay.

Both PCCP and the proposed algorithm can use the network capacity well and thus maintained a high throughput during the interval [30 s, 70 s]. For the CCF protocol it can be seen in Fig. 5b that as the traffic load is decreased at time $t = 30$ s, the queue size and also the queuing delay are also decreased. But in the proposed protocol, when a child node becomes inactive, the total input traffic to its parent node i (r_{in}^i) is decreased. This makes the rate difference Δr^i to be increased and the new transmission rate of active child nodes is also increased. Thus, since the other

active sensor nodes can adaptively increase their sending rate, the queuing delay and throughput of the proposed protocol remain relatively stable, even when some nodes become active or inactive. All protocols have a zero loss probability in this scenario.

Fig. 6 shows the corresponding results when we repeated the previous simulation trial, but using an exponential service time. It can be seen that the proposed protocol can effectively use the remaining capacity, and provides an overall better performance.

In the next experiment, we suppose that each sensor node has a different geographical priority which is set equal to its node number. This means that sensor nodes with higher numbers have a higher priority. Under the assumption that each sensor node has the same traffic class priority equal to 1, the total priority of each sensor node will be equal to its number ($P^i = i, i = 1, \dots, M + N + K$). Since the CCF cannot support the priority of nodes, we do not consider its performance. Table 2 shows the results. The table also shows results when the same scenario is tested under an exponential service time. Unlike PCCP, the proposed protocol can assign network capacity to each sensor node based on its priority. The average loss probability and queuing delay for PCCP in this scenario were 0.08 and 0.022 s, respectively. Compare with 0 loss probability and 0.01 s queuing delay for the proposed method.

To evaluate the effectiveness of the algorithms under sudden changes in traffic load, we performed another experiment. In this scenario, sensor node 3 was made inactive (turned off) during the time interval [30 s, 70 s]. The service time was fixed. The results are shown in Fig. 7.

From Fig. 7c, when sensor node 3 goes off under the proposed algorithm, its normalized capacity ($3 / \sum_{i=1}^{10} i = 3/55 = 0.0545$) is allocated to sensor nodes 1, 2, 4, and 5. Based on the details of the proposed algorithm, each node uses the remaining capacity based on its priority. It means that node 5 which has a higher priority than nodes 1, 2 and 4, gets more capacity than the other nodes. The ideal allocation of remaining capacity for nodes 5, 4, 2 and 1 will be: $0.054(5/12) = 0.0225$; $0.054(4/12) = 0.018$; $0.054(2/12) = 0.009$; and $0.054 \cdot 1/12 = 0.0045$, respectively. Simulation results show that for the proposed algorithm the increase in normalized throughput for nodes 5, 4, 2 and 1 are 0.021, 0.018, 0.011 and 0.005, respectively. For PCCP, the

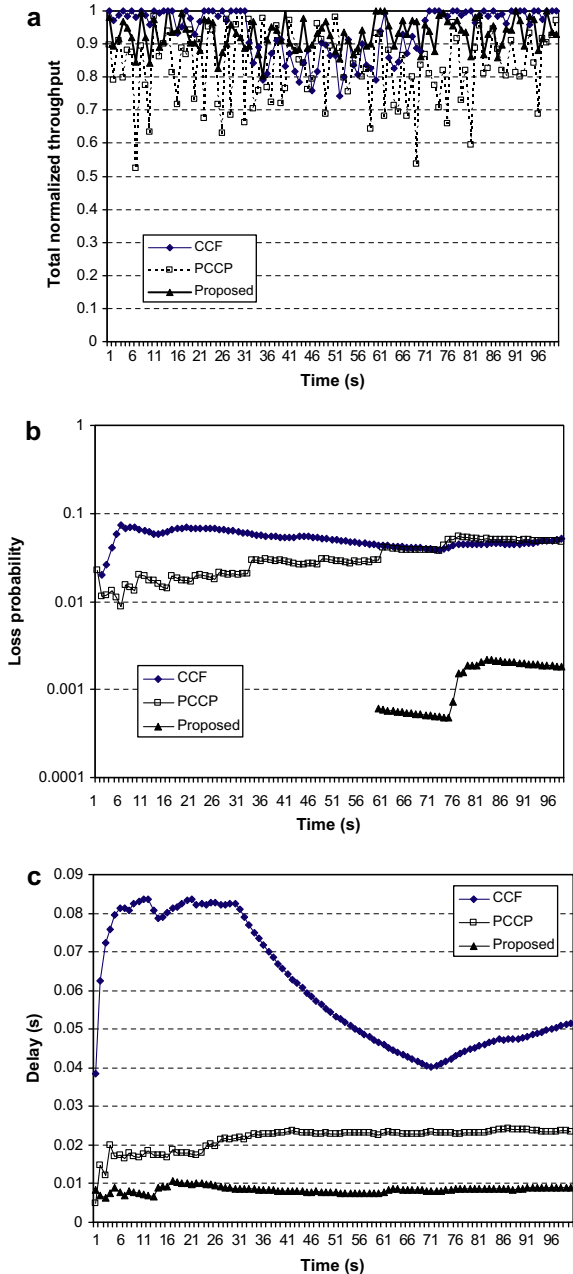


Fig. 6. Performance using a variable (exponential) service time: (a) total normalized throughput, (b) loss probability and (c) queuing delay.

Table 2

Average normalized throughput at each sensor node for PCCP and the proposed method.

Source #	Expected throughput	Fixed service time		Exponential service time	
		PCCP	Proposed	PCCP	Proposed
1	0.018	0.018	0.018	0.010	0.017
2	0.036	0.035	0.036	0.024	0.034
3	0.054	0.053	0.053	0.040	0.051
4	0.072	0.070	0.071	0.058	0.069
5	0.09	0.088	0.089	0.096	0.085
6	0.108	0.106	0.107	0.114	0.103
7	0.126	0.123	0.125	0.145	0.120
8	0.144	0.141	0.143	0.187	0.137
9	0.162	0.159	0.160	0.159	0.152
10	0.18	0.176	0.178	0.188	0.180

corresponding increase for nodes 5, 4, 2 and 1 are: 0.009, 0.008, 0.002 and 0.003, respectively. Thus, unlike the PCCP, the proposed protocol is very successful in capacity allocation based on sensor node priority. As shown in Fig. 7a, in

the case of CCF, when node 3 goes off, the other nodes cannot use the remaining capacity well and so there is a sudden drop in network throughput. Furthermore as CCF does not support node priority, all sensor nodes have the same throughput, as can be observed in the figure.

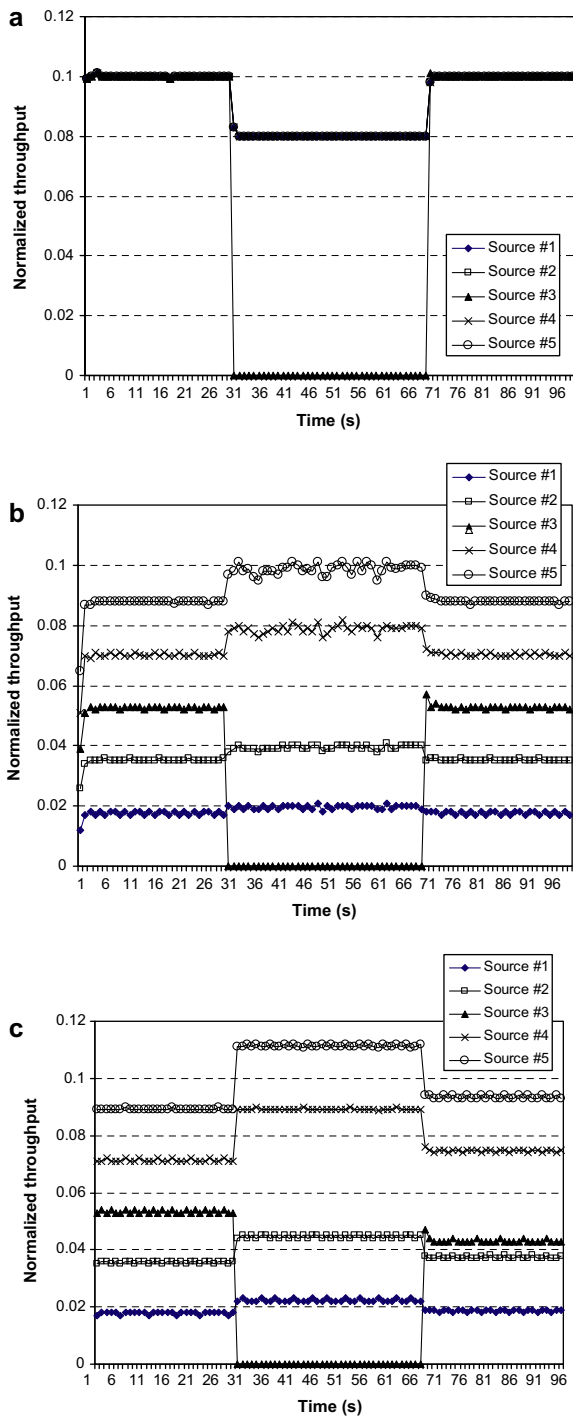


Fig. 7. Normalized throughput in the case of different priority using a fixed service time: (a) CCF protocol, (b) PCCP protocol, and (c) proposed protocol.

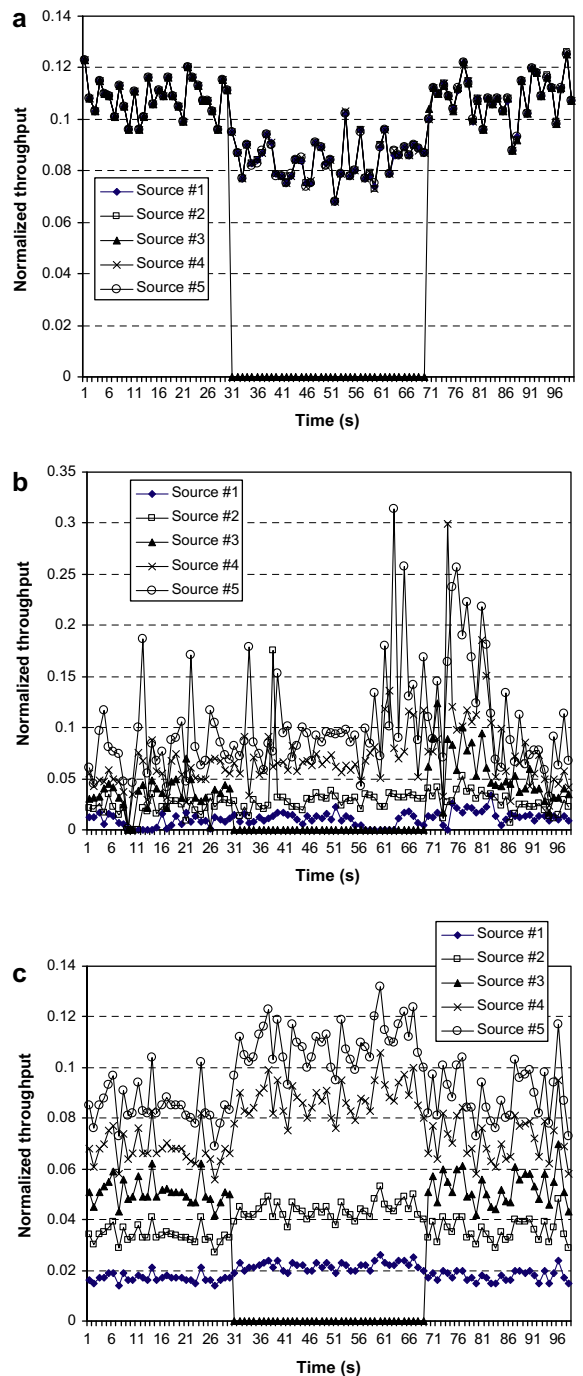


Fig. 8. Normalized throughput in the case of different priority using a variable service time: (a) CCF protocol, (b) PCCP protocol, and (c) proposed protocol.

The same experiment above was performed, but with a random service time. The results are shown in Fig. 8. The figure shows that when service time is a random variable, the proposed protocol is much more successful than the other algorithms in distributing the remaining capacity to the other nodes based on their priority.

4.2. Evaluation of the proposed service differentiation unit

To evaluate the performance of the proposed service differentiation unit, we assumed that each sensor node may have different traffic classes: EF, NRT1, NRT2 and NRT3. We evaluate the performance under two different schedulers: Weighted round-robin (WRR) scheduler and priority queuing (PQ) scheduler.

4.2.1. WRR scheduler

In the WRR scheduler, each connection i is assigned a weight w_i , i.e., it is allocated w_i slots during each round. The weighted round-robin scheduler is designed to better handle traffic classes with different service capacities. Each traffic source can be assigned a weight. Traffic sources with higher weights receive more network bandwidth than those with less weight. In the simulation, to provide service differentiation, we assign more weight to higher priority traffic classes. For this purpose the normalized weight assigned to EF, NRT1, NRT2 and NRT3 traffic classes are equal to 0.5, 0.3, 0.15 and 0.05, respectively. All sensor nodes have the same geographical priority. Fig. 9 shows the results for the case of fixed service time. From Fig. 9a, we can observe that the proposed model, can assign network bandwidth to each traffic class based on its weight. The EF class has the highest throughput while NRT3 class has the lowest throughput. Since the PCCP protocol cannot discriminate between traffic classes its total throughput is plotted which is close to 1. Fig. 9b shows that using the proposed protocol with a WRR scheduler, all traffic classes have similar queuing delay characteristics. The PCCP protocol uses a common buffer for all traffic classes, and hence cannot provide low delay bound which is necessary for high priority EF traffic class. Thus, PCCP has the highest queuing delay. In this scenario both PCCP and the proposed protocol have zero loss probability.

Fig. 10 shows the results of the same experiment, but with a random service time. We observe that the proposed service differentiation unit can effectively discriminate between the traffic classes. The proposed protocol has a better throughput, a smaller loss probability, and a shorter delay than PCCP.

The results in Figs. 9 and 10, also show that the WRR scheduler is not an ideal scheduler for high priority real time traffic class. Therefore, in the next simulations we use a priority queuing (PQ) scheduler to provide very low delay bound for high priority real time traffic.

4.2.2. PQ scheduler

Given that the delay of WRR scheduler is related to packet length and allocated weight, it does not always rep-

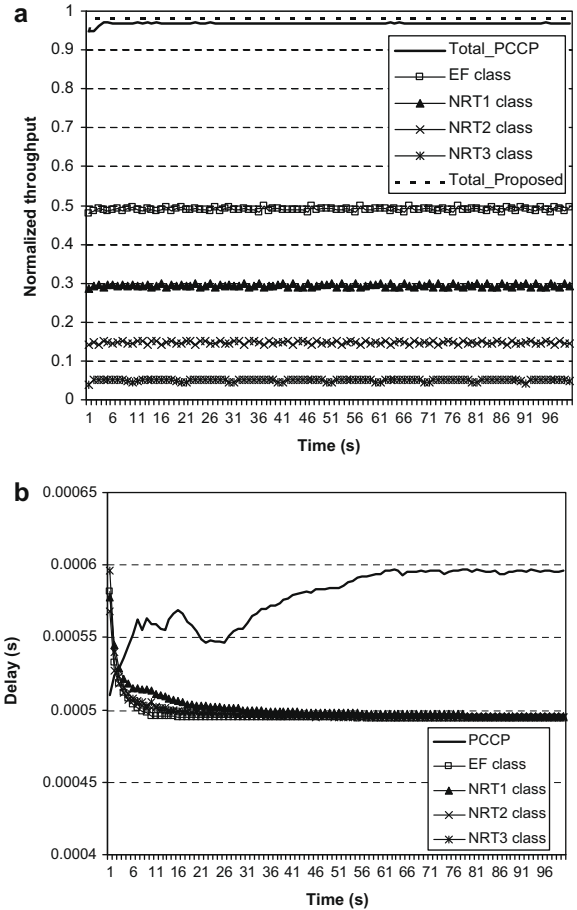


Fig. 9. Performance of the service differentiation unit using the WRR scheduler and a fixed service time: (a) normalized throughput and (b) delay.

resent a good scheduling protocol for real time traffic. Priority queues are used in a wide variety of applications including operating systems, real-time systems, and discrete event simulations. In a priority queue, each element is ordered by its associated priority. The packet with the highest priority goes first, regardless of the order of arrival. A priority queuing protocol provides low delay bound for high priority traffic.

In the next simulation trial, we use PQ protocol for high priority EF traffic class. Similar to previous simulation trials, the assigned weights to EF, NRT1, NRT2 and NRT3 classes are 0.5, 0.3, 0.15 and 0.05, respectively. Fig. 11 in the case of fixed service time, shows the normalized throughput and queuing delay for both PCCP and proposed protocol. The EF class has the highest priority, so the queuing delay for this class is always zero.

Results for the case of random service time with PQ scheduler is shown in Fig. 12. Here, the EF class has essentially zero loss probability and zero queuing delay. Since the EF class uses a PQ scheduler, the average queuing delay for NRT3, NRT2 and NRT1 classes is higher than in the earlier case with the WRR scheduler.

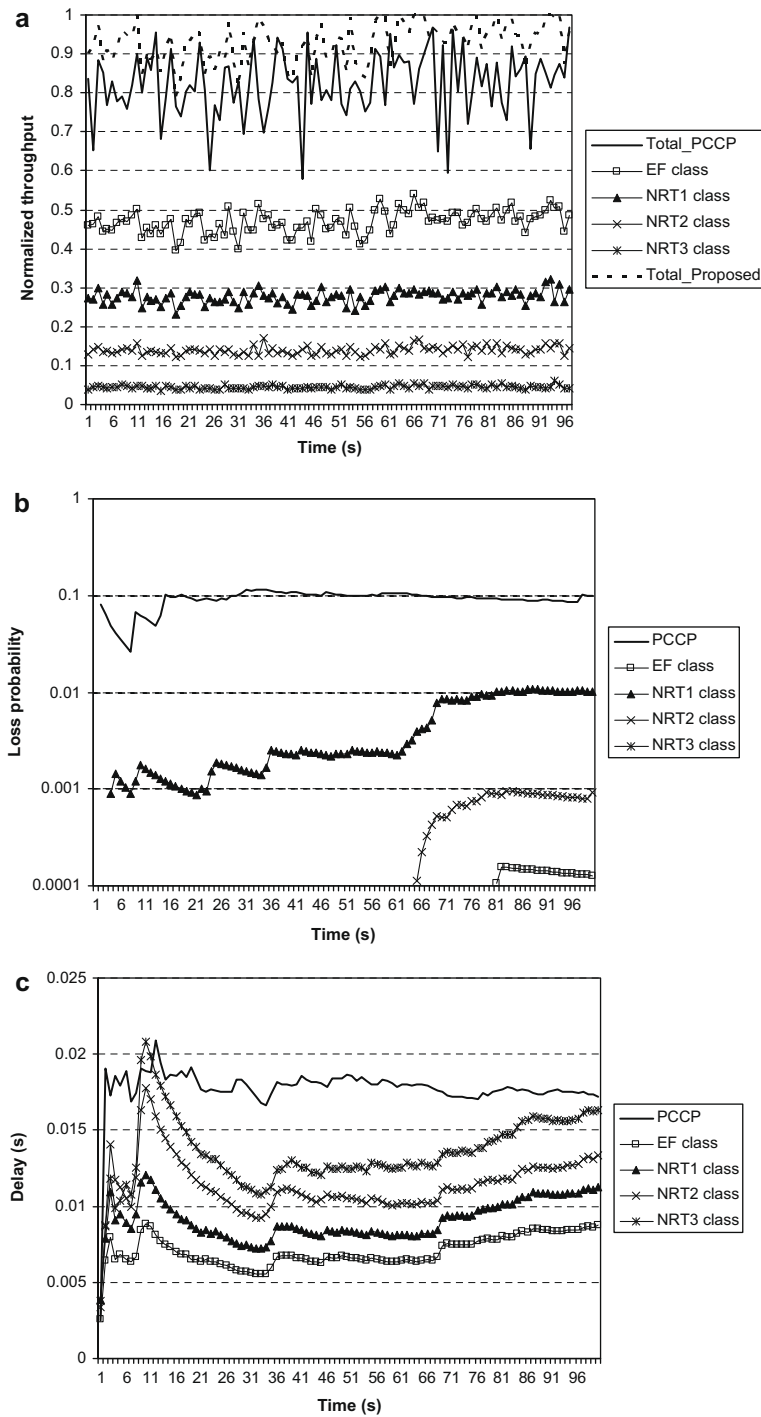


Fig. 10. Performance of the service differentiation unit using the WRR scheduler and a fixed service time: (a) normalized throughput, (b) loss probability and (c) delay.

4.3. Effect of parameter β (traffic load controller)

We also evaluated the effect the traffic load parameter β on delay and throughput. From (10) and (13), changing the parameter β will affect the channel rate as well as the

source rate of all traffic classes. We evaluated the performance with respect to delay and throughput at different values of β for both WRR scheduler and PQ scheduler. Fig. 13 shows the normalized throughput and queuing delay for all traffic classes plotted against β .

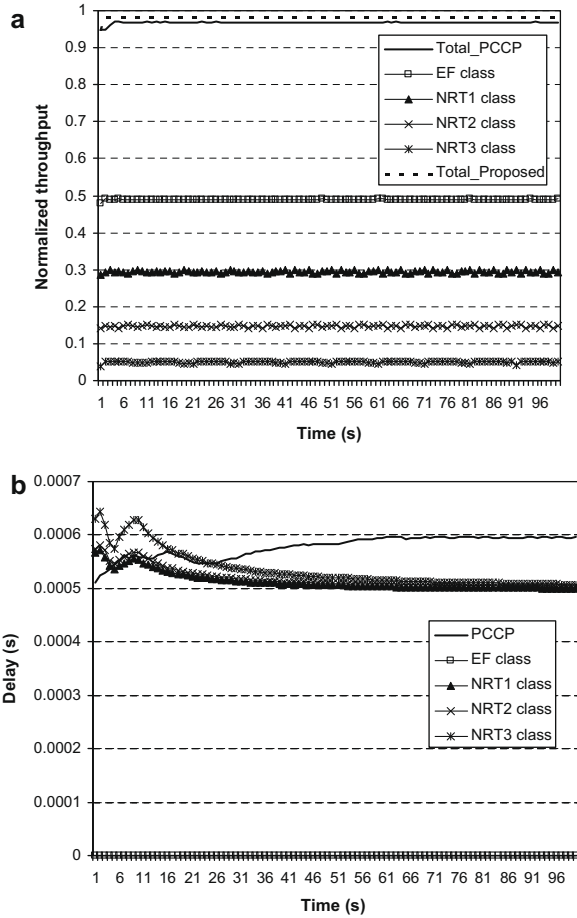


Fig. 11. Performance of the service differentiation unit using the PQ scheduler and a fixed service time: (a) normalized throughput and (b) delay.

When $\beta < 0.85$, the traffic load is low and all queues are always empty. So the queuing delay of all traffic classes is close to zero. When β is increased to 1, the traffic load in the network is also increased, thus the queues are no longer empty. For both WRR and PQ schedulers, the EF class has the lowest delay. In the case of PQ scheduler, the change in traffic load, does not affect the delay performance of the EF class. The EF class always has a zero queuing delay. Fig. 13c shows that, by increasing the value of β , the traffic load is also increased so the throughput of each traffic class is increased.

4.4. Effect of node priority

In all previous simulation trials, we assumed that all wireless nodes have all four types of traffic classes EF, NRT1, NRT2 and NRT3. In the next simulation trial, we consider a wireless sensor network where each node could have a different combination of traffic classes. This combination of traffic classes is shown in Table 3. We assume all sensor nodes have the same geographical priority equal to 1. Suppose the assigned priority for EF, NRT1, NRT2 and

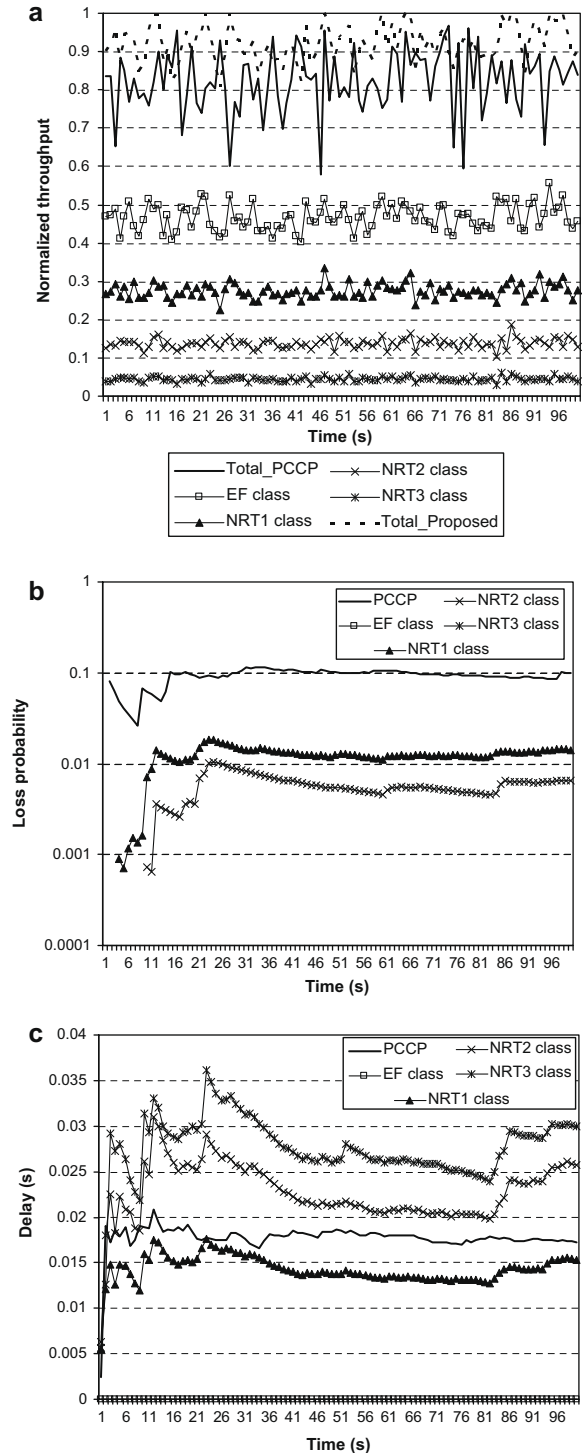


Fig. 12. Performance of the service differentiation unit using the PQ scheduler and a variable service time: (a) normalized throughput, (b) loss probability and (c) delay.

NRT3 classes are equal to 10, 6, 3 and 1, respectively. In this case, for sensor node 1 which has all traffic classes, the traffic class priority, P_{TC}^1 , is equal to $10 + 6 + 3 + 1 = 20$, while

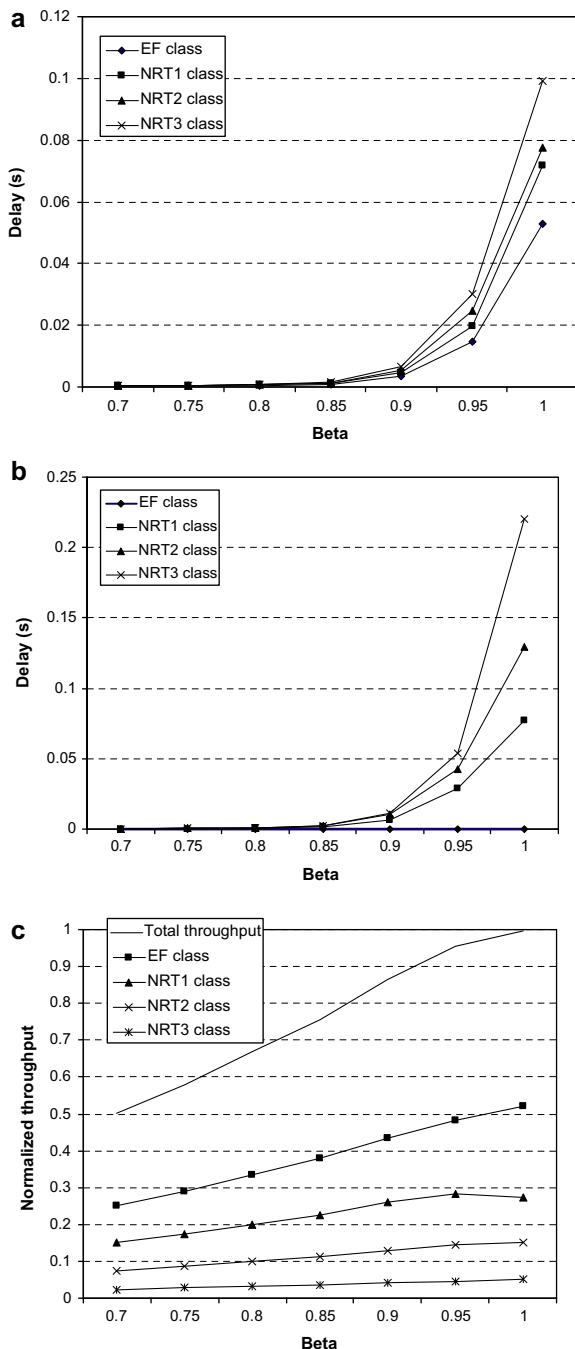


Fig. 13. Effect of traffic load parameter β on delay and throughput: (a) effect on delay (WRR scheduler), (b) effect on delay (PQ scheduler), and (c) effect on throughput (WRR scheduler).

for sensor node 10 which has only NRT1 and NRT3 traffic classes, the traffic class priority, P_{TC}^{10} , is equal to $6 + 1 = 7$. The normalized throughput of each sensor node is given in the last column of Table 3. It can be seen that the proposed protocol can allocate network capacity to each node based on its priority. In this case node 1 which has the highest priority (20) also has the highest throughput

Table 3

The state of traffic classes in each sensor node.

Sensor #	EF (w = 10)	NRT1 (w = 6)	NRT2 (w = 3)	NRT3 (w = 1)	Global priority	Normalized throughput
1	ON	ON	ON	ON	20	0.13571
2	ON	ON	ON	OFF	19	0.12928
3	ON	ON	OFF	ON	17	0.11544
4	ON	ON	OFF	OFF	16	0.10901
5	ON	OFF	ON	ON	14	0.09489
6	ON	OFF	ON	OFF	13	0.09009
7	ON	OFF	OFF	ON	11	0.0762
8	OFF	ON	ON	ON	10	0.06917
9	OFF	ON	ON	OFF	9	0.06197
10	OFF	ON	OFF	ON	7	0.0482

(0.13571) while node 10 which has the lowest priority (7) also has the lowest throughput (0.0482).

5. Conclusion

New applications made possible by rapid improvements and miniaturization in hardware has motivated recent developments in wireless multimedia sensor networks (WMSNs). To provide the required quality of service for multimedia applications in WMSNs, congestion control is necessary. Each congestion control protocol should be able to detect congestion in advance, and allocate available rates to the sensor nodes accordingly. For some applications, there is a need to send real time traffic toward the sink node with low latency and high reliability so that immediate remedial and defensive actions can be taken, as appropriate. Further, when an important event occurs in the system, the sensor node that detected the event should send some alarm message to the sink. Usually this kind of high priority traffic is bursty. This means that high priority traffic is generated only for a short period of time while low priority traffic usually exists in the network and produce thousands of packets generated periodically. For such environments, service differentiation in wireless multimedia sensor networks becomes an important problem. To provide service differentiation in WMSNs, it is necessary to consider a different priority for each traffic source.

In this paper we presented a model for congestion control and service differentiation in WMSNs. The proposed congestion control protocol can adjust the source traffic rates based on current congestion in the upstream nodes and the priority of each traffic source. The proposed model can support four different traffic classes namely, real time traffic class (EF class); high priority, non real-time traffic class (NRT1 class); medium priority, non real-time traffic class (NRT2 class); and low priority, non real-time traffic class (NRT3 class). We evaluated the performance of the proposed model on different cases and with different scheduling protocols. Simulation results show that the proposed congestion control protocol can achieve low packet loss probability. Using the proposed service differentiation model with a priority queuing scheduling protocol, it is possible to provide low queuing delay and guaranteed bandwidth for high priority real time traffic.

References

- [1] I.F. Akyildiz, T. Melodia, T.R. Chowdhury, A survey on wireless multimedia sensor networks, *Computer Networks* 51 (2007) 921–960.
- [2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, *Computer Networks* 38 (4) (2002) 393–422.
- [3] S. Blake, M. Carlson, E. Davies, Z. Wang, W. Weiss, An architecture for differentiated services, IETF RFC 2475 (1998).
- [4] K. Nichols, V. Jacobson, L. Zhang, A two-bit differentiated services architecture for the internet, IETF RFC 2638 (1999).
- [5] T. Li, Y. Rekhter, A provider architecture for differentiated services and traffic engineering (PASTE), IETF RFC 2430 (1998).
- [6] C. Wang, Member, K. Sohraby, M. Daneshmand, Y. Hu, Upstream congestion control in wireless sensor networks through cross-layer optimization, *IEEE Journal on Selected Areas in Communications* 25 (4) (2007) 786–795.
- [7] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Transactions on Networking* 1 (4) (1993) 397–413.
- [8] C.-T. Ee, R. Bajcsy, Congestion control and fairness for many-to-one routing in sensor networks, in: *Proceedings of ACM Sensys*, November 2004.
- [9] Y.G. Iyer, S. Gandham, S. Venkatesan, STCP: a generic transport layer protocol for wireless sensor networks, in: *Proceedings of IEEE ICCCN 2005*, San Diego, CA, October 17–19, 2005.
- [10] B. Hull, K. Jamieson, H. Balakrishnan, Mitigating congestion in wireless sensor networks, in: *Proceedings of ACM Sensys'04*, Baltimore, MD, November 3–5, 2004.
- [11] C.-Y. Wan, S.B. Eisenman, A.T. Campbell, CODA: congestion detection and avoidance in sensor networks, in: *Proceedings of ACM Sensys'03*, Los Angeles, CA, November 5–7, 2003.
- [12] Woo, D.C. Culler, A transmission control scheme for media access in sensor networks, in: *Proceedings of ACM Mobicom'01*, Rome, Italy, July 16–21, 2004.
- [13] C.-Y. Wan, A.T. Campbell, PSFQ: a reliable transport protocol for wireless sensor networks, in: *Proceedings of ACM WSNA'02*, Atlanta, GA, September 28, 2002.
- [14] C.-Y. Wan et al., Siphon: overload traffic management using multi-radio virtual sinks in sensor networks, in: *Proceedings of ACM SenSys'05*, San Diego, CA, November 2–4, 2005.
- [15] Levis et al., Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks, in: *Proceedings of First Symposium on Networked Systems, Design and Implementation*, San Francisco, CA, March 29–31, 2004.
- [16] F. Stann, J. Heidemann, RMST: reliable data transport in sensor networks, in: *Proceedings of IEEE SNPA'03*, Anchorage, AK, May 11, 2003.
- [17] H. Zhang et al., Reliable bursty convergecast in wireless sensor networks, in: *Proceedings of ACM Mobicom'05*, Urbana-Champaign, IL, May 25–28, 2005.
- [18] J. Wang, V.M. Vokkarane, Optimal remote homing for providing service differentiation in information-aware multi-layered wireless sensor networks, in: *Proceedings of IEEE International Conference on Communications*, vol. 1, June 2006, pp. 361–366.
- [19] E. Felemban, Chang-Gun Lee, E. Ekici, MMSPEED: multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks, *IEEE Transactions on Mobile Computing* 5 (6) (2006) 738–754.
- [20] H.W. Oh, I.T. Han, K.R. Kwang, S.H. Kim, An enhanced multi-path scheme for QoS guarantee in wireless sensor network, in: *Proceedings of IEEE International Symposium on Consumer Electronics*, 2007 (ISCE 2007), 20–23 June 2007, pp. 1–5.
- [21] A. Boukerche, R.B. Araujo, L. Villas, A wireless actor and sensor networks QoS-aware routing protocol for the emergency preparedness class of applications, in: *Proceedings of 31st IEEE Conference on Local Computer Networks (LCN 2006)*, November 2006, pp. 832–839.
- [22] J. Grewal, J.M. DeDoutre, A framework for quality of service in wireless networks, in: *Proceedings of the Third Communication Networks and Services Research Conference*, 16–18 May 2005, pp. 231–236.
- [23] J. Xu, R. Li, A new in-network differentiated services protocol in wireless sensor networks, in: *Proceedings of International Conference on Communication Technology (ICCT'06)*, November 2006, pp. 1–3.
- [24] W.L. Tan, O.-C. Yue, W.C. Lau, Performance evaluation of differentiated services protocols over wireless sensor networks, in: *Proceedings of 64th IEEE Vehicular Technology Conference (VTC-2006)*, September 2006, pp. 1–5.



Mohammad Hossien Yaghmaee was born on July 1971 in Mashad, Iran. He received his B.S. degree in Communication Engineering from Sharif University of Technology, Tehran, Iran in 1993, and M.S. degree in communication engineering from Tehran Polytechnic (Amirkabir) University of Technology in 1995. He received his Ph.D. degree in communication engineering from Tehran Polytechnic (Amirkabir) University of Technology in 2000. He has been a computer network engineer with several networking projects in Iran

Telecommunication Research Center (ITRC) since 1992. From November 1998 to July 1999, he was with Network Technology Group (NTG), C&C Media research labs., NEC Corporation, Tokyo, Japan, as visiting research scholar. From September 2007 to August 2008, he was with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, USA as a visiting associate professor. He is the author of 3 books all in Farsi language. He has published more than 85 international conference and journal papers. His research interests are in Wireless Sensor Networks (WSNs), traffic and congestion control, high speed networks including ATM and MPLS, Quality of Services (QoS) and fuzzy logic control.



Donald A. Adjeroh received the Ph.D. degree in computer science from the Chinese University of Hong Kong in 1997. He joined the Lane Department of Computer Science and Electrical Engineering, West Virginia University (WVU), Morgantown, in 2000, where he is currently an Associate Professor and Coordinator of the graduate program in computer science. Before joining WVU, he was a faculty member at the Department of Computer Science, University of Canterbury, New Zealand. His general research interests are in image and video processing, distributed multimedia systems, computational aspects of vision, pattern matching, and computational biology. His work has been supported by grants from the National Science Foundation, NASA, and WV EPSCoR. He received the DOE CAREER award in 2002.