A SIMPLE PRIORITY-BASED SCHEME FOR DELAY-SENSITIVE DATA TRANSMISSION OVER WIRELESS SENSOR NETWORKS

Farshad Safaei[1], Hamed Mahzoon[2], Mohammad Sadegh Talebi

[1] Faculty of ECE, Shahid Beheshti University G.C., Evin 1983963113, Tehran, IRAN
f_safaei@sbu.ac.ir
[2] Department of System Innovation Graduate School of Engineering Science
Osaka University, Osaka, Japan
hamed.mahzoon@irl.sys.es.osaka-u.ac.jp
[3] School of Computer Science Institute for Research in Fundamental Sciences (IPM)
Tehran, Iran
mstalebi@ipm.ir

ABSTRACT

In the course of last decade, wireless sensor networks (WSNs) have grabbed attention of both academic research community and industrial users. Such networks provide a broad range of applications, making them of great significance. Energy is the greatest concern of such battery-operated networks as it directly influences the lifetime of the network. As packet dropping due to congestion has a dramatically negative impact on energy consumption, congestion control is a vital issue for wireless sensor network, particularly for wireless multimedia sensor networks (WMSNs) in which bursty traffic is prevailing. In this paper, we present a simple yet efficient priority-aware congestion control scheme to support bursty data. Results achieved from our extensive simulation experiments corroborate that the proposed scheme operates well in terms of delay, throughput, and packet dropping, for both bursty and ordinary traffic.

KEYWORDS

Sensor Networks, Quality of Service, Congestion Control, Source Priority, Essential Queue

1. INTRODUCTION

A wireless sensor network consists of sensor nodes deployed over a geographic area mainly for monitoring physical phenomena like temperature, humidity, vibrations, seismic events, etc. [1]-[3]. Typically, a sensor node is a battery-operated device with a limited budget of energy, which consists of three basic components: a sensing element for data acquisition from environmental events, a processing subsystem for local data processing and storage, and a transmission device that provides wireless communication for nodes. As sensor nodes are usually deployed over environments that are almost unreachable, recharging the power sources proves cumbersome. This brings out the power consumption as a challenging issue and consequently, energy conservation is a key issue in designing WSNs [1].

Since WSN nodes are not always deployed over predetermined locations, the network needs to be designed with some levels of self-configuration capability. Once the nodes are deployed, the network is left unattended to perform monitoring, thus the network topology changes due to alternative node failures, suspensions, temporary droppings, and environmental distortion. The relatively higher number of nodes in WSNs makes its management much more complicated. A typical wireless sensor network is depicted in Figure 1, where nodes are deployed over the area referred to as sensor field. Each sensor node is capable of sensing the intended physical phenomena and routing the monitoring information toward the sink node through multipath with no infrastructure. The sink node is connected to a task manager node via Internet or satellite.

DOI : 10.5121/ijwmn.2012.4112

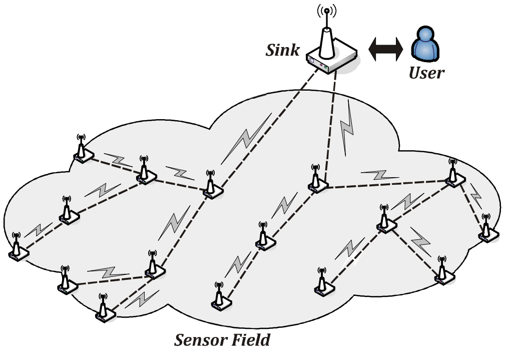


Figure 1. A wireless sensor network deployed in a sensor field

With advances in VLSI and production of MEMS, nowadays it's possible to integrate sensing, video/audio processing, and communication apparatus into a single tiny node at low prices. These advancements have been led to emergence of wireless multimedia sensor networks (WMSNs) which are missioned to sense, process, and transport image/audio/video streams both reliably and efficiently. A typical node in WMSNs is equipped with multimedia recording devices such as camera and microphones, by which two different types of data acquisition is collected: snapshot and streaming data. The first multimedia data type delivers event triggered observations obtained in a short period of time. On the contrary, the second multimedia content is generated over longer time periods. Multimedia applications typically publish great amount of data requiring high transmission rate and huge processing. Thus, compared to traditional WSNs, in such networks, the role of computation power is much more important. Characteristics that are common to sensor networks, such as resource constraints, unbalanced traffic and data redundancy, also exist in WMSNs [4]. We note, however, that for the case of WMSNs, the challenges due to these characteristics are often of greater concerns.

Supporting the preliminary requirements for providing Quality-of-Service (QoS), which is necessary for multimedia applications, is directly related to energy consumption, delay, reliability, distortion, and network lifetime. There is an inevitable correlation between levels of quality of accessible services in WMSNs and energy consumption in these networks, while obtaining any of these bases acquires the influential interaction on the other.

There exist several studies that tried to propose congestion control mechanisms that are aware of different packet priorities [5] and [6]. Such priority-aware schemes provide good performance to guarantee QoS for packets with different importance levels. However, they evince poor performance when they face bursty traffics. In this paper, we present a novel congestion control mechanism that aims to mitigate huge packet dropping in the presence of bursty traffic. It also tries to provide the required QoS for data of vital priority. To this end, the proposed mechanism leverages existing node architectures such as those proposed in [5] and [6], with only slight modifications. These modifications would only incur augmentation of a separate queue in each sensor node, thereafter referred to as essential queue. This essential queue is devoted to transfer delay-sensitive data over wireless sensor network. Thus, in the first place, all nodes send data buffered in their essential queues. If there is no packet in the essential queue, nodes will obey the rules dictated by priority-based rate adjustment schemes. We demonstrate QoS improvements asserted by extensive simulation experiments carried out in OMNET++.

The remainder of this paper is organized as follows. In the next section, a review of related works regarding QoS in WSNs is indicated. Then, a preliminary view to our scheme is mentioned in Section 3. Section 4 is devoted to our extensive experimental results. Finally, Section 5 concludes the paper.

2. RELATED WORK

This section is devoted to review some related studies. In the last few years, the networking research community has witnessed the emergence of a plethora of works concentrated on efficient communication protocol for wireless sensor networks. A large portion of such works have considered QoS guarantee in sensor network, e.g. [7]- [11]. Some other works concerned with providing light, efficient, and scalable congestion control protocols to possibly support several traffic classes in sensor networks [12]- [27].

In order to manage sudden transfer of data measured by sensor nodes, the authors of [13] have proposed CODA, as a congestion detection and avoidance armed with the following three mechanisms: receiver-based congestion detection; open-loop hop-by-hop back-pressure; and closed-loop multi-source regulation. He et al. [14] proposed SPEED, which is a real-time communication protocol supporting unicast, area-multicast, and area-anycast, realtime traffics. SPEED takes advantages such as being stateless, localized, scalable, and having minimal control overhead.

In [15], the authors presented a distributed and scalable algorithm to mitigate congestion occurred in a sensor network. The authors also aimed to guarantee fairness by trying to deliver equal number of packets by each node. In Yaghamaee et al. [6], a protocol is presented in which by utilizing queuing techniques and a novel weight nomination mechanism, either avoids probable network congestion, yet tries to esteem data priority. In this mechanism, for every child node of an individual node, a queue is assigned where the data transmitted from the child node flows through the corresponding queue. The data congestion evaluation is provided defining a congestion index. In [5], the authors proposed a novel congestion control protocol, dubbed Priority-based Congestion Control Protocol (PCCP), for wireless sensor networks. PCCP employs node priority index and congestion degree and enables cross-layer optimization and works under both single path and multipath routing. In [23], following Network Utility Maximization (NUM) framework [22], the authors aimed to formulate congestion control as the solution to an optimization-driven flow control problem with lifetime and link interference constraints. Similar to approaches used in NUM, the authors proposed a distributed and asynchronous flow control algorithm.

The authors of [17] have tried to improve network reliability by applying multipath data transmission. With this technique, there are multiple paths versus single shortest path to transmit data through, thus high density and congestion of data, containing multimedia records, on single shortest paths will not happen. This paper has also classified the data types and tried to perform more conscious node scheduling to moderate delay detriment. Finally, equipping nodes with multichannel transmission and exclusive transmission frequency provides a high throughput in this mechanism.

A typical sensor network has been simulated to evaluate lifetime and cost of the system by Cheng et al. [18]. In this paper, various cases are considered for different node variations like mobility, nodal structures, generated traffic types, and node energy allocation. In such cases, lifetime and cost of the network are evaluated and compared.

An evaluation model for performance of the sensor networks is presented by Chiasserini et al. [20], which compares the sensor networks with or without actuator and exhibits the differences. With the presented model, the network is simulated while the sensors are powered off. Once an event happens, the sensors are powered on via actuator nodes and start to record and transmit the phenomena.

Breaking down data into two parts, real-time and normal data, Akkaya et al. [21] innovated a technique in which real-time packets do not delay more than expected limit. Also, this mechanism tries to maximize throughput of normal packets. The nodes near to destination dynamically inform the far nodes about processing rate of real-time data so the network nodes alternatively adjust their processing rates. Calculation of approximate delay of real-time packets for different paths is led to an optimization problem, and hence, a path that minimizes the cost function is selected among different choices.

3. CONGESTION-AWARE ARCHITECTURE

In wireless multimedia sensor networks (WMSNs) different data types might be in progress to be transferred to the sink node. Most of such data are not delay-sensitive to which we refer to as ordinary data. Neither multiple transmissions of these data and dropping of excess ones impose any large data overhead to the network, nor is the received data density too high. Unlike such delay-insensitive data, others such as imaging and sonic data are almost delay-sensitive, and due to their inherent value, they are occasionally generated and transmitted in WMSNs, thus making them an essential data type. On the other hand, transmitting these data types through the network from the source node to the destination requires a great amount of energy from the nodes visited on the path. Thus, an overflow of such data in an intermediate node's queue imposes huge power waste to the sensor system. Regenerating and retransmitting lost data might add even more overhead to the network. These above mentioned challenges suggest that there should be more considerations about essential data and overhead associated to them.

It seems that classification of data can successfully handle the aforementioned challenges. We note, however, that providing this mechanism will exclusively cause loss of major amount of data types other than multimedia type in the network and thus, the network will change to an application specific one, in which the other data types can route the paths only if there is no multimedia data to be processed or transmitted. Though, managing data according to their types may cause mis-recognition of essential packets, since they might not be of multimedia ones. Meanwhile, there might be a very important such data that a user prefers to cancel receiving any other data to receive this data sooner. Thus, such data should take different priorities during the production, and in intermediate nodes different policies are pursued to apply these priorities.

In this section, we present our scheme to provide a simple yet efficient scheme for source-to-sink transmission of delay-sensitive data mentioned above. Our scheme employs the priority-based architecture proposed in [24]. However, as such a hierarchical scheme would not guarantee timeliness for delay-sensitive data delivery; we provision a separate queue, thereafter referred to as essential queue, for such data at each node. Figure 2 shows the architecture for our scheme.

In the sequel, we briefly review priority-based rate control scheme proposed in [5] and [24] that will be used for upstream rate allocation for ordinary data at each node.

3.1. Priority-Bases Rate Control

Assume that $T_s(i)$ and $\overline{T}_s(i)$ respectively show the service time of the current packet and the average service time in node *i*. The average service time is obtained by a moving average process. Then, let r_i denote the rate at which node *i* can transmit packets over the MAC layer. Then r_i is given by [24]:

$$r_i = \frac{1}{\overline{T}_s(i)} \tag{1}$$

Congestion can be quantified by congestion index according to the relevant queue size. Letting $q^k(i)$ represent the size of *k*-th queue in node *i*, the congestion index $I_X^k(i)$ is defined as a non-decreasing function of $q^k(i)$. In [24], the congestion index of *k*-th queue is defined according to a piecewise-linear model expressed by

$$I_X^k(i) = \begin{cases} \varepsilon & \text{if} \quad q^k(i) \le \min_{th}^k(i) \\ \dfrac{q^k(i) - \min_{th}^k(i)}{\max_{th}^k(i) - \min_{th}^k(i)} \max_p & \text{if} \quad \min_{th}^k(i) \le q^k(i) \le \max_{th}^k(i) \\ \max_p + \varepsilon & \text{if} \quad q^k(i) \ge \max_{th}^k(i) \end{cases} \tag{2}$$

where $\max_{th}^{k}(i)$ and $\min_{th}^{k}(i)$ denote the upper and lower thresholds for *k*-th queue of node *i*, respectively. Moreover, ε and max$_p$ are constants less than 1.

Each child *k* at node *i* is in possession of a queue. Thereby taking into account the traffic generated by source *i* and N_i to be the number of children of node *i*, there will be N_i +1 queues at node *i*. In order to avoid congestion, each node *i* determines the rate for each of these internal queues based on the priority of each queue and its congestion level. Using $I_X^k(i)$ defined above, for each child *k* we define

$$(3)\,\overline{I}_X^k(i)=\frac{1}{N_i}\left(1-\frac{I_X^k(i)}{\sum_{j=1}^{N_i+1} I_X^j(i)}\right);\qquad k=1,\ldots,N_i+1.$$

Nodes are assumed as having different priorities. For each node *i*, we represent the source priority and the *j*th child's priority by SP(*i*) and TP(*j*), respectively. Then, the total priority of node *i* is the sum of its source priority and the priority of its children, i.e.

$$TP(i)=\sum_{j\in C(i)} TP(j)+SP(i) \tag{4}$$

where *C*(*i*) is the set of children of node *i*. It's apparent from the above equation that for a leaf node *l*, i.e. *C*(*l*) =∅, we have TP(*l*) = SP(*l*). Based on the notion of total priority, in [24] the authors have defined the proportional priority factors for each node *i* as follows

$$p_i^1=\frac{SP(i)}{TP(i)}, \tag{5}$$

$$p_i^k=\frac{TP(k)}{TP(i)};\qquad k=2,\ldots,N_i+1. \tag{6}$$

Using this proportional priority factors, each node *i* will be able to calculate the rate for each child *k*, i.e. r_i^k , as follows

$$r_i^k=\omega_i^k r_i, \tag{7}$$

where ω_i^k is a weight factor that combines the congestion index and priority of child *k* in the following way

$$\omega_i^k=\frac{p_i^k\overline{I}_X^k(i)}{\sum_{j=1}^{N_i+1} p_i^j\overline{I}_X^j(i)} \tag{8}$$

3.2. Essential Packets and the Queue Model

Figure 2 demonstrates our proposed architecture, where a queuing unit is provided for each child node. Data of each child node flow through the corresponding queue unit, unless it is an essential data. In this case, the *essential data* flow through the essential queue. Each queue unit is equipped with a scheduler of its own, which classifies entering data and applies the local timing according to the corresponding predefined policy. Data generated the sensor node are led to a separate queue unit that classifies all generated data on traffic generator and directs them through corresponding queues. If data generated by traffic generator are also of essential type, they will flow through essential queue and will be treated as an essential packet and scheduled by the timing strategy of this queue.

The scheduler in each node prescribes the packet processing rate as follows: If essential queue is empty, i.e. there is no essential packet, the scheduler determines rate for each child node according to rate control method presented in subsection 3.1. However, if essential queue is not empty, the scheduler sends all essential packets first, and then pays attention to other packets containing ordinary data.

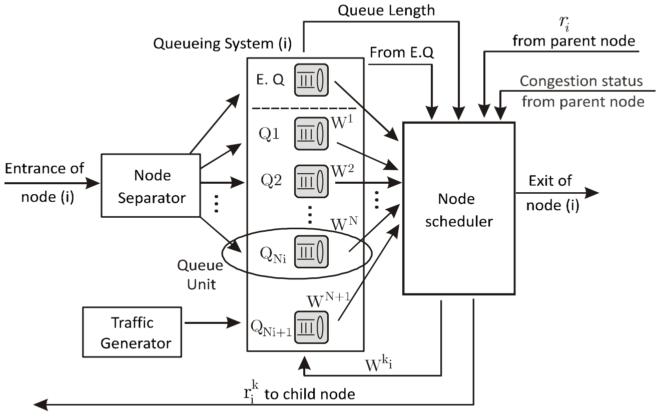


Figure 2. The proposed architecture

The priority-based rate control scheme described above assumes that child nodes generate data with the rate dictated by the parent node, and while the parent node detects crowd in a child node queue, it decreases this transmission rate to prevent dropping of data on that queue. However, in so many of WSNs, based on the usage, data might be generated in a bursty and regardless of nominated transmission rate, while most of them are essential ones. In this paper, we refer to such a deviation from nominated rate as norm-exceeding rate. If such norm-exceeding transmissions occur frequently, intermediate nodes will get in trouble since such norm-exceeding traffics are uncoordinated with the other data.

Uncoordinated data generation is referred to as data transmission with norm-exceeding rate. The first challenge is that such norm-exceeding traffics will cause congestion on the receiving buffer(s), while the ordinarily generated data fulfill the buffer, and once there is an empty room created in the buffer, again, a recently generated data occupies the available space. Thus, the later generated bursty data may find a full buffer and will be blocked. Even if the node treats these data as a high priority one, data overflow will be high over time, due to uncoordinated generation of such data types. The second challenge is that these data types are generated in a bursty and unpredictable fashion. Thus, once they are generated, the chance of dropping critical data as well as delayed adjustment of transmission rate for ordinary data due to high amount of such bursty data will increase dramatically.

In the next section, experimental results will show that how flowing such bursty data through essential queues would aim to dramatically alleviate packet dropping.

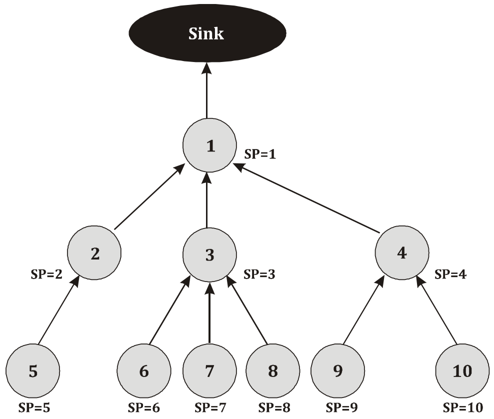


Figure 3. Topology for Simulation Experiments

4. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

In this section, we present the experimental results of the proposed architecture and compare the achieved results with those obtained from a system without using the proposed architecture. Performance of the essential queue has been explored through extensive experiments. The influence of norm-exceeding traffic on different aspects of network performance such as packet delay, packet dropping, and data throughput is also evaluated.

4.1. Simulation Setup

Simulation experiments are implemented in OMNET++ environment [28], which is a discrete event simulator. We carried out our experiments for the topology used in [24] which is a simple tree based topology shown in Figure 3. In this topology, the essential buffer length is 10 as well as the queue length of each child node. If the essential queue method is not used, the length of queues will grow up to reach the length of essential queue. If a node is equipped with the queue unit, buffers of all classes grow with the same amount of content. In the other word, unlike the case without using this scheme, when it is used the buffer for all the child nodes will grow with the same scale and this amount of growth will be equally divided among different classes in the queue unit.

4.2. Performance Assessment of Essential Queue

In this subsection, we examine utilizing the essential queue. Hence, we assume that all traffics generated by nodes are based on the feedback rate from the parent node. Thus, the essential data generation is also affected by the rate fed back from the parent node, and thus the rate of such essential data would diminish with the congestion detection and low feedback rate.

In a system without essential queue, it is expected that delay of essential packets exceeds that of a system with essential queue. On the other hand, it’s expected that using this technique should not affect the ordinary packets and overall packet delay. These conjectures are validated by evaluating the packet delay simulation results which illustrate that the average packet delay for a system simulated without using the essential queue is 33 ms, and the packet delay through the simulation period varies from 20 to 60 ms. The same system is simulated with the utilization of essential queue. In this case, the packets delay varies between 2 to 70 ms and the average packet delay is about 33 ms, which is similar to the corresponding average in the system without essential queue. These results reveal that utilizing essential queue in a system, hopefully does not increase the packet delay of the system.

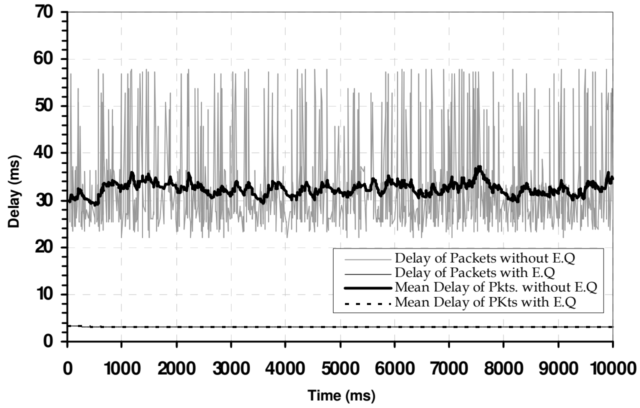
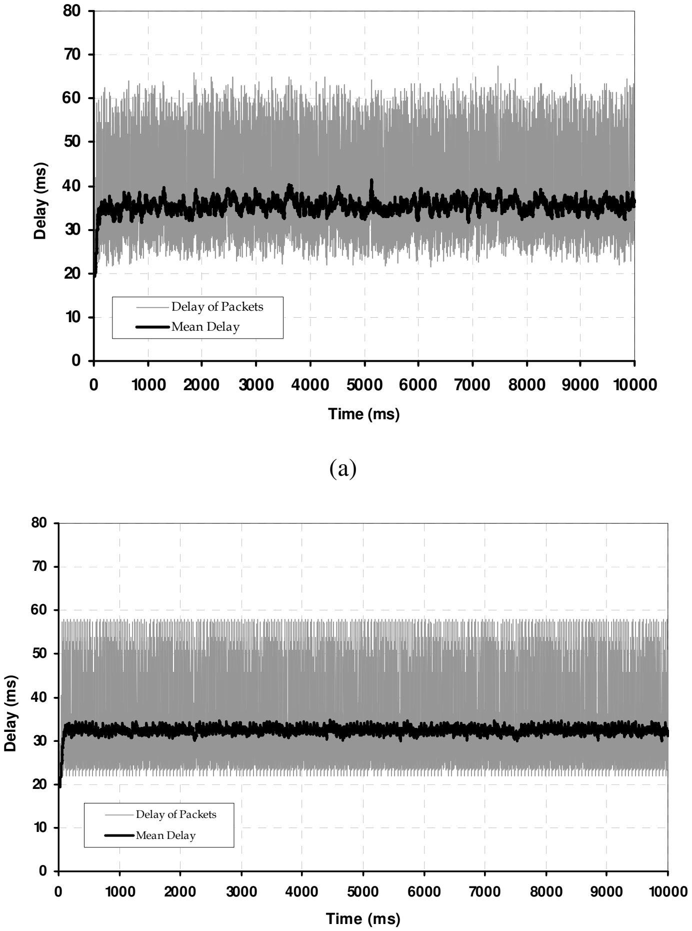


Figure 4. Essential packet delay of a system with and without utilizing essential queue

Now, we evaluate the delay of essential packets in above experiments. Figure 4 depicts the packet delay in our simulation. The curve indicated by “Without EQ” shows the system in which the essential queue is not used. The same result when the essential queue is provided in system is denoted in the curve indicated by “With EQ”. As it was expected, due to lack of a special method for essential packets, this simulation also reveals the same result as the first case, meaning that the range of delay and the average delay similarly are the same as the first case. For “With EQ” curve, however, we can observe that the delay of essential packets, when the system is equipped with the essential queue, varies between 1 to 4.6 ms with the average equal to 3.2 ms. The results of this simulation extensively differ from the system in which the essential queue is not used. Considering ordinary packets, we obtain the results denoted in Figure 5. The results of systems without and with essential queue are illustrated and in Figure 5(a) and Figure 5(b), respectively.



(a)

(b)

Figure 5. Delay of ordinary packets for a system (a) without utilizing essential queue, (b) with utilizing essential queue

Comparing these two figures, we realize that the average ordinary packet delay in the first system, i.e. 33 ms, is exacerbated up to 36 ms when the system is equipped like the second system, and this diminution is almost 9% of delay of ordinary packets. In congestion avoidance strategy, besides the status of intermediate queues, source priority should also be considered. It is expected that the source priority concept regulates the proportion of received data from different nodes. This issue is also improved using essential queue concept. Table 1 denotes the simulation results of the number of packets received by the last node of the network.

The first column of this table denotes the source priority while the second one shows the number of packets received from each priority. When there is no congestion and no bursty data received, it is expected to meet the source priority in this column. In the third column, the portion of received packets of each node to the received packets of base source is shown, while it depicts the actual priority observed in the network, and error of the calculated priority is given in the fourth column. This error is calculated based on the number of received packets of the base node, the number of received packets of the other nodes, the priority of the sources, the difference between the number of received packets, and expected number of received packets. Above evaluations have been applied to essential packets and the corresponding results are shown in columns 5 to 7. It is recognizable that due to the separate queue that has been provided for essential data, it is not expected to meet the priority of ordinary packets for essential data. Nevertheless, it seems that the essential packets are also received by the last node according to the different source priorities.

Table 1. Source priority and packet receipt with essential queue

Source Priority	Number of received packets	Proportion of received packet to base	Error (%)	Number of received essential packets	Proportion of received essential packets to base	Error (%)
1	181	1	0	17	1	0
2	362	2	0	39	2.29	14.7
3	544	3.01	0.18	65	3.82	27.5
4	725	4.05	0.13	66	3.88	-2.9
5	906	5.02	0.11	102	6	20
6	1088	6.01	0.18	101	5.94	-0.9
7	1271	7.03	0.32	127	7.47	6.7
8	1450	8.01	0.14	145	8.53	6.6
9	1633	9.03	0.25	170	10	11.1
10	1813	10.02	0.17	203	11.94	19.4

4.3. Performance of Essential Queue with Norm-Exceeding Traffic

Here, we evaluate the performance of essential queue concept while there is norm-exceeding traffic in the network. Thus, we assume that essential data are generated with no respect to nominated rate. Ordinary data are also generated with respect to the nominated rate and if there were essential data in the internal queue of a node, the essential data are transmitted and thereafter, the ordinary data are sent regarding the nominated rate. Figure 6 illustrates the mentioned traffic. Comparing “Without EQ” curve with the results of the first case discussed earlier, we see that the maximum packet delay is highly increased. Also, the average packet delay is about 500 ms, which is dramatically worse than the case with the traffic under the expected rate.

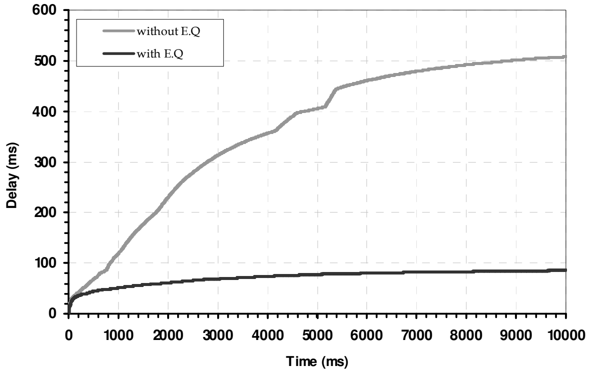


Figure 6. Packet delay of a system when norm-exceeding traffic is injected with and without utilizing essential queue

It is clearly depicted in the curve of the system utilizing essential queue that the average delay is approximately 90 ms, which is much better than the previous case. The reason for this great improvement stems from the way this system handles bursty traffics. When the essential queue is not used in the system, these essential data are generated in a bursty manner and flow through the corresponding child node. However, in the last system with the essential queue, the bursty data are directed to essential queue and they hierarchically traverse the network with essential queues toward the destination. Ordinary data then, realizing the bursty data due to the length of their queues, degrade the generating rate, and thus obviate the extra traffic. With this method the overall packet delay diminishes as well as packet dropping in the queues.

We accomplish the same evaluation on essential data and the results are portrayed in Figure 7. In the "With EQ" curve, the average packet delay is about 3 ms and the maximum delay is 6.3 ms. The significant point is that although the traffic rate is high and bursty, the packet delay is still preciously low, while the average delay is even lower than what is shown in "With EQ" curve of Figure 6. The average essential packet delay in the system without essential queue is about 900 ms with the maximum delay of 4832 ms. These results are extremely high compared with the system with the essential queue. The overall diagram is so similar to the corresponding system in Figure 6.

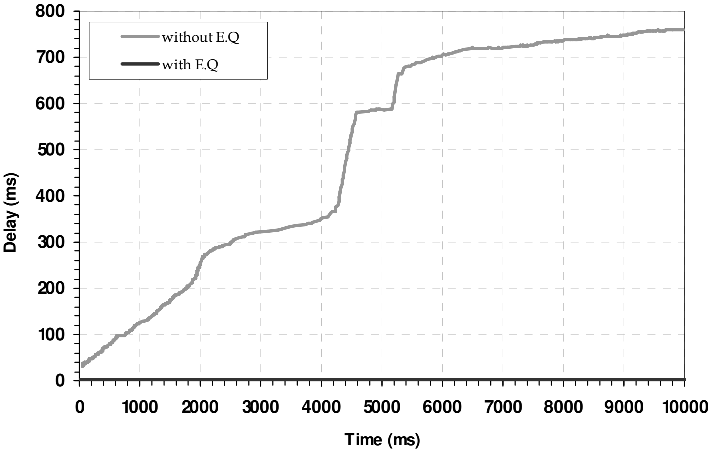
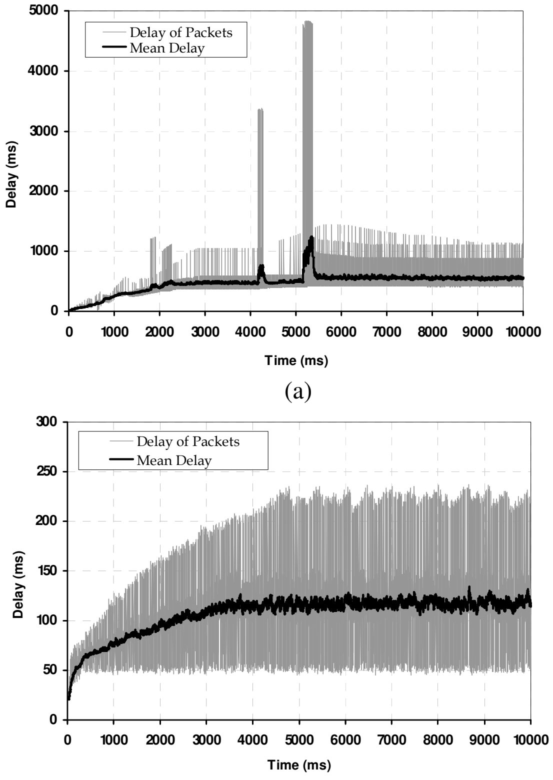


Figure 7. Essential packet delay of a system when norm-exceeding traffic is injected with and without utilizing essential queue

In Figure 8 we evaluate the results of above mentioned systems for ordinary data. It is expected that the overall shape of the diagram does not differ from the diagram for all packet types. Figure 8(a) depicts the result for the system not using the essential queue. As it seems, the average delay of ordinary packets is about 500 ms that conforms the all-type average delay. When using essential queue in this system, the results will be like Figure 8(b). The average packet delay in this case is 110 ms that is far better than that of not using essential queue. The reason of this improvement is that the ordinary data generation rate is not coordinated with the generation rate of bursty traffic. In the system without essential queue, although ordinary data justify their generation rate with the current queue status, because all the queues are crowded due to bursty traffic, the transmission rate adjustment is not appropriate regarding to the current network status. On the other hand, utilizing the essential queue in a system detaches the bursty traffic from ordinary data, and thus ordinary data are able to adjust their rate regarding the real network status. This issue will prevent from generation of traffics that sensor node is unable to support and handle.

In this part, the portion of received packets by the last node of the network is calculated. Table 2 summarizes the experiment in which essential queue has been used. Comparing these with the one without essential queue demonstrates that packet priority for essential data and ordinary data is improved by 10.9% and 4.9%, respectively. Figure 9(a)-9(b) comparatively depict the error of observing source priority in system with and without essential queue, respectively.



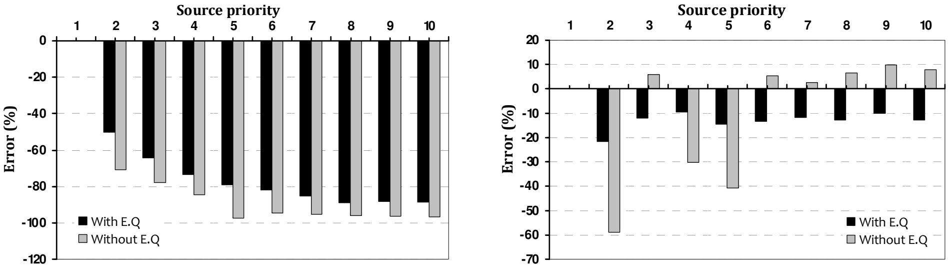
(a)

(b)

Figure 8. Delay of ordinary packets for a system with norm-exceeding traffic injection (a) without utilizing essential queue, (b) with utilizing essential queue

Table 2. Source priority and packet receiving with essential queue and norm-exceeding traffic

Source Priority	Number of received packets	Proportion of received packet to base	Error (%)	Number of received essential packets	Proportion of received essential packets to base	Error (%)
1	363	1	0	199	1	0
2	455	1.25	-37.3	198	0.99	-50.3
3	647	1.78	-40.6	214	1.07	-64.1
4	805	2.21	-44.6	211	1.06	-73.5
5	909	2.50	-49.9	208	1.05	-79.1
6	1068	2.94	-51.0	215	1.08	-81.9
7	1218	3.36	-52.1	205	1.03	-85.3
8	1319	3.63	-54.6	176	0.88	-88.9
9	1537	4.23	-52.9	208	1.04	-88.4
10	1656	4.56	-54.4	228	1.14	-88.5



(a) (b)

Figure 9. Source priority observation for norm-exceeding rate traffic: (a) for essential data, (b) for ordinary data

4.4. Packet Dropping

If all data transmission is done with the maximum rate proposed by the parent node, then packet dropping will not happen. On the contrary, when there are bursty and norm-exceeding essential data traffic in the network, packet dropping is inevitable. Here we show that how utilization of essential queue in a network alleviates packet dropping. First, we consider a systemwithout essential queue and then we simulate the same system using essential queue and denote the data dropping diagram as Figure 10. In this figure, dropping percentage is increasing at first, but after reaching the steady state, data transmission is done by dropping of 21% of data. In this figure, data dropping starts after 700 ms, and after its beginning and its bursty growth, in the time period of 800 ms to 1200 ms, dropping is restrained. This is due to increased intermediate queue lengths and consequently growth of congestion index of the queues. Thus, nodes degrade the rate of their ordinary data. Hence, data dropping is controlled in the aforementioned time period. On the other hand, because of high rate of essential data, packet dropping percentage increases up to its steady state value.

This situation changes if the essential queue is utilized in the system. As it can be seen in Figure 10, after bursty growth of dropping percentage, from 2000 ms, the transient state of the system passes and dropping percentage reaches a constant rate. The reason for such a bursty dropping at the beginning of the traffic generation and transmission is the higher priority of essential data with respect to ordinary data. The existence of essential data prevents ordinary data to be serviced. While the nodes are calculating the generation rate of their child nodes, high generation rate of essential data causes dropping of ordinary data. After the required time, when the network is in the steady state, the generation rate of nodes is specified and dropping rate reaches its permanent value.

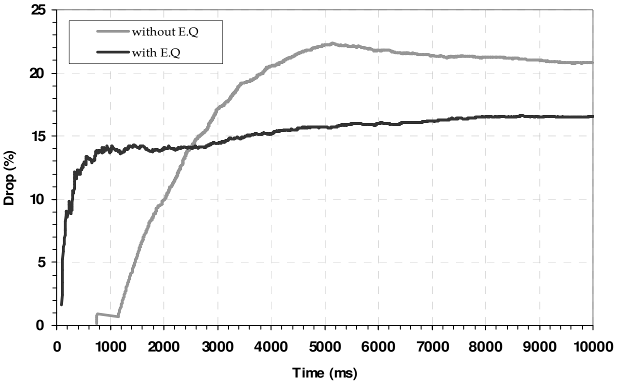


Figure 10. Packet dropping percentage when norm-exceeding traffic of parent node is used

In Figure 10, it has been shown that utilizing essential queue improves total data dropping by 5% thanks to usage of separated queues for bursty and ordinary data. This improvement is observed in the essential data (Figure 11).

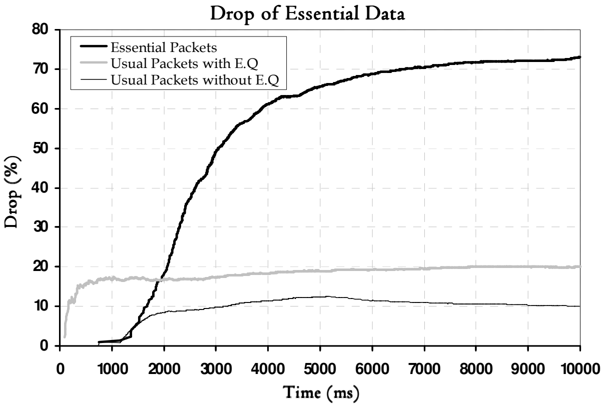


Figure 11. Packet dropping percentage when the parent traffic is norm-exceeding rate

In the system without essential queue, packet dropping rate is more than 70%. This high amount of dropping demonstrates that most of the essential data face a full queue after generation and competition for the corresponding queue. According to the coordination between the generation rate and the transmission rate of ordinary data, these data types mostly enter the queue when it is not full. Hence, utilizing essential queue in a system makes packet dropping as small as zero. It means that all droppings happened in Figure 10 for “with E.Q.”, is not related to essential data and they are all ordinary data that have been dropped. Also, in Figure 11 two curves are depicted for ordinary data dropping while utilizing and not utilizing essential queue in the system. Comparing these two curves reveals that ordinary data dropping without using essential queue is about 10%. Then, ordinary data dropping starts to diminish. Hence, the comparison between these curves illustrates that there is 10% difference of dropping of ordinary packets, whether there is essential queue provided for the system or not. This means that we can exploit essential queue in a system, and with increasing ordinary data dropping from 10 to 20%, we can degrade the essential data dropping from 70% to 0, and total data dropping decreases from 21 to 16%, which admits the efficiency of utilization of essential queue.

4.5. Throughput

Now, we evaluate the performance of the network in terms of its throughput and investigate the influence of essential queue on this performance metric. We compare the essential packet throughput in a system with and without essential queue. We then study the ordinary data throughput of these systems. In Figure 12 the total data throughput is depicted while the system is equipped with and without the essential queue and norm-exceeding rate traffic is exerted. As it is shown, when the system doesn't use essential queue, in the time period of 500 to 5000 ms, throughput experiences an oscillatory behavior. Before 1000 ms, this value temporarily approaches zero. This is due to data dropping in the network because of buffer fulfillment and data generation rate adjustment by network nodes. In this figure, at the beginning, because of high packet dropping, throughput is deeply decreased and as it follows, due to adjustment of network rates to significant ones, the throughput approaches one. Finally, the average throughput in this case is about 0.9.

It is clear that from the beginning of the simulation, throughput is almost around 1 and as compared to Figure 13, here its oscillatory behavior is tolerable. From the beginning of the data transmission in the network, if there is essential data in a node, the ordinary data of that node are not sent and the corresponding queue will get a high amount of data. Thus, the congestion index of these queues will be high which will cause a quick adjustment of the generation rate of these data types. Hence, the rate adjustment of the ordinary data and even dropping of these data begin, and thus throughput will not be affected so much.

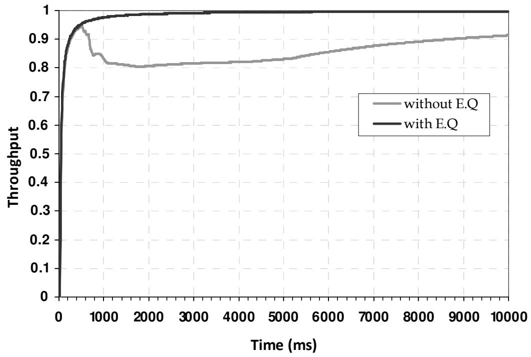


Figure 12. Data throughput for norm-exceeding rate traffic and in a system with and without essential queue

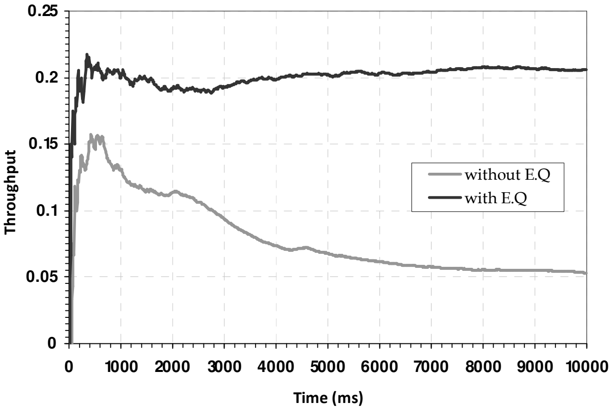
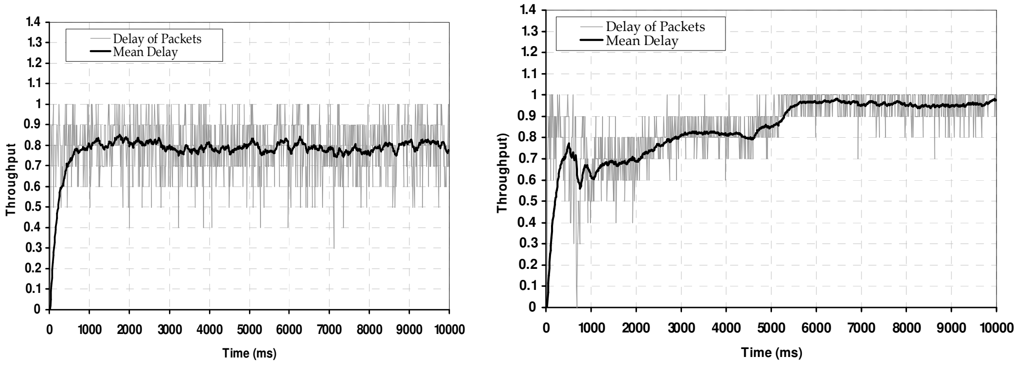


Figure 13. Essential data throughput for norm-exceeding rate traffic in a system with and without essential queue

The throughput of essential data is displayed in Figure 13. In the first figure, it is shown that although the throughput of essential data has higher values at the beginning of the simulation, but its average value is less than 0.1 and finally reaches the value 0.06 at the end.

Obviously, the throughput of essential data in such a system is increased. The steady state distribution of throughput during the simulation period illustrates that essential data transmission is done steadily. Comparing "With EQ" curves in both Figure 13 and Figure 12, we expect that ordinary data would also have high and almost constant throughput in this case. This expectation is validated with the results shown in Figure 14.

According to the throughput of total data in a system without essential queue, we expect that ordinary data throughput is also affected by the data whose generation rate is not coordinated with the rate proposed by the parent node. The result for such a system is displayed in Figure 14(a). As we expected, the ordinary data has the highest throughput too, which is validated due to non-existence of dropping that is seen in Figure 12. The throughput distribution is smooth with average of 0.8; thereby affection of bursty traffic is seen there. Thus, we conclude that utilizing essential queue not only has no negative effect on ordinary data, but also the process applied for essential data also improves the status of ordinary data in terms of dropping, delay, and throughput.



(a)

(b)

Figure 14. Ordinary data throughput for norm-exceeding rate traffic in a system (a) without essential queue, (b) with essential queues

In Figure 14(b) the above system is implemented without essential queue. As it was expected from the "Without EQ" curve of Figure 12 and Figure 13, in this case ordinary data are affected by the bursty generation of essential data and their throughput have been degraded precipitately. While, according to the "With EQ" curve of Figure 13, the throughput of essential data is of little variation, due to high amount of essential data generation and also dropping of ordinary data, their throughput catch extreme variations around the time period of 600 to 1400 ms. In the long run, when the network reaches steady state regime for data generation rate and dropping, ordinary data throughput tends to 86%.

5. CONCLUSION

In this paper we presented a novel priority based rate adjustment scheme for wireless sensor networks to manage the priority of delay-sensitive traffics in wireless sensor networks. Our scheme is based on the existing priority based schemes with an extra queue unit to serve delay-sensitive data that might be generated in a bursty manner. We introduced the essential queue to propose priorities for the packets and to recognize the essential data from others. Then, to evaluate the performance of the proposed scheme, we simulated an illustrative wireless sensor network with tree topology. Our experimental results showed that augmentation of such a simple queue unit to each node for supporting essential data, can greatly improve delay, throughput, and packet dropping rate of such delay-sensitive data. On the other hand, it would incur slight impact on such performance metrics for ordinary data which obey priority-based rate assigned by their parents.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci (2002), "Wireless sensor networks: a survey," *Elsevier Computer Networks*, vol. 38, no. 4, pp. 393–422.

[2] J. Yick, B. Mukherjee, and D. Ghosal (2008), "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330.

[3] C. Y. Chong and S. P. Kumar (2003), "Sensor Networks: Evolution, Opportunities, and Challenges," *Proc. of the IEEE*, vol. 91, no. 8, pp. 1247–1256.

[4] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury (2007), "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960.

[5] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu (2007), "Upstream congestion control in wireless sensor networks through cross-layer optimization," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 786–795.

[6] M. H. Yaghmaee and D. Adjeroh (2008), "A New Priority Based Congestion Control Protocol for Wireless Multimedia Sensor Networks," *Proc. of WoWMoM 2008*, pp. 23-28.

[7] E. Felemban, C. -G. Lee, E. Ekici, R. Boder, and S. Vural (2005), "Probabilistic QoS Guarantee in Reliability and Timeliness Domains in Wireless Sensor Networks," *Proc. of IEEE INFOCOM 2005*, vol. 4, pp. 2646–2657.

[8] Y. Wu, S. Fahmy, and N. B. Shroff (2010), "Sleep/wake scheduling for multi-hop sensor networks: Nonconvexity and approximation algorithm," *Ad Hoc Networks*, vol. 8, no. 7, pp. 681–693.

[9] S. Tang and W. Li (2006), "QoS supporting and optimal energy allocation for a cluster based wireless sensor network," *Computer Communications*, vol. 29, no. 13–14, pp. 2569–2577.

[10] M. M. Alam, M. Mamun-Or-Rashid, and C. S. Hong (2006), "QoS-Aware Routing for Sensor Networks Using Distance-Based Proportional Delay Differentiation (DPDD)," *Int'l Conf. on Next-Generation Wireless Systems (ICNEWS)*, Dhaka, Bangladesh.

[11] H. O. Sanli, H. Cam, and X. Cheng (2004), "EQos: An Energy Efficient Qos Protocol for Wireless Sensor Networks," *Proc. of 2004 Western Simulation MultiConference*, San Diego, CA.

[12] S. Borasia and V. Raisinghani (2011), "A Review of Congestion Control Mechanisms for Wireless Sensor Networks," *Springer Journal of Technology Systems and Management*, pp. 201–206.

[13] C. Y. Wan, S. B. Eisenman, A. T. Campbell (2011), "Energy-efficient congestion detection and avoidance in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 4, pp. 1–32.

[14] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher (2003), "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," *Proc. of IEEE International Conference on Distributed Computing Systems*, pp. 46–55.

[15] C. T. Ee and R. Bajcsy (2004), "Congestion control and fairness for many-to-one routing in sensor networks," *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 148–161.

[16] S. K. Dhurandher, S. Misra, H. Mittal, A. Agarwal, I.Woungang (2011), "Using ant-based agents for congestion control in ad-hoc wireless sensor networks," *Springer Cluster Computing*, vol. 14, no. 1, pp. 41–53.

[17] M. A. Hamid, M. M. Alam, and C. S. Hong (2008), "Design of a QoS-aware Routing Mechanism for Wireless Multimedia Sensor Networks," *IEEE GLOBECOM 2008*, New Orleans, LO, pp. 1–6

[18] Z. Cheng, M. Perillo, andW. B. Heinzelman (2008), "General Network Lifetime and Cost Models for Evaluating Sensor Network Deployment Strategies," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 484–497.

[19] I. Politis, M. Tsagkaropoulos, and S. Kotsopoulos (2008), "Optimizing Video Transmission over Wireless Multimedia Sensor Networks," *IEEE Global Telecommunications Conference (GLOBECOM)*, New Orleans, LO, pp. 1–6.

[20] C. F. Chiasserini and M. Garetto (2004), “Modeling the Performance ofWireless Sensor Networks,” *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*.

[21] K. Akkaya and M. Younis (2003), “An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks,” *Proc. of 23rd International Conference on Distributed Computing Systems*, pp. 710-715.

[22] M. Chiang, S. Low, A. R. Calderbank, and J. C. Doyle, “Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures,” In Proc. IEEE, vol. 95, no. 1, pp. 255–312, Jan. 2007.

[23] J. Chen, W. Xu, S. He, Y. Sun, P. Thulasiraman, X. Shen (2010), “Utility-based asynchronous flow control algorithm for wireless sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1116–1126.

[24] M. H. Yaghmaee and D. A. Adjeroh (2008), “A Model for Differentiated Service Support inWireless Multimedia Sensor Networks,” *Proc. of ICCCN 2008*, pp. 881–886.

[25] M. I. Khan, W. N. Gansterer, and G. Haring (2007), “Congestion Avoidance and Energy Efficient Routing Protocol for Wireless Sensor Networks with a Mobile Sink,” *Journal of Networks*, vol. 2, no. 6, pp. 42–49.

[26] E. Toscano, O. Mirabella, and L. L. Bello (2007), “An Energy-efficient Real-Time Communication Framework for Wireless Sensor Networks,” *Int’l Workshop on Real Time Networks (RTN)*, Pisa, Italy.

[27] F. Ren, T. He, S. Das, and C. Lin (2011), “Traffic-Aware Dynamic Routing to Alleviate Congestion in Wireless Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems*, no. 99, pp. 1585–1599.

[28] A. Varga (2001), “The OMNeT++ discrete event simulation system,” *Proceedings of the European Simulation Multiconference (ESM’01)*, pp. 319–324.

Authors

Farshad Safaei received the B.Sc., M.Sc., and Ph.D. degrees in Computer Engineering from Iran University of Science and Technology (IUST) in 1994, 1997 and 2007, respectively. He is currently an assistant professor in the Department of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran. His research interests are performance modelling/evaluation, Interconnection networks, computer networks, and high performance computer architecture.

Hamed Mahzoon received the Bachelor’s degree in computer engineering (hardware) from Iran University of Science and Technology, Tehran, Iran, in 2007, the Master’s degree in computer engineering (computer architecture) from Shahid Beheshti University, Tehran, Iran, in 2010. He is currently Ph.D. student in Intelligent Robotics Laboratory -Ishiguro laboratory-, Osaka University, Osaka, Japan. His research interest is in Sensor Networks, Quality of Service Control and Autonomous Systems.

Mohammad Sadegh Talebi received his BSc degree in electrical engineering from Iran University of Science and Technology (IUST), Iran in 2004, and MSc degree in electrical engineering Sharif University of Technology, Iran, in 2006. He is currently a research assistant in School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics (IPM), Iran. His research interests are optimization and control for wired and wireless data networks.