

Research Article

A Deadline-Aware and Distance-Aware Packet Scheduling Algorithm for Wireless Multimedia Sensor Networks

Lin Tang,¹ Quansheng Guan,² Shengming Jiang,³ and Bingyi Guo²

¹*School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China*

²*School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China*

³*College of Information Engineering, Shanghai Maritime University, Shanghai 210306, China*

Correspondence should be addressed to Quansheng Guan; eeqshguan@scut.edu.cn

Received 20 November 2014; Revised 12 March 2015; Accepted 19 March 2015

Academic Editor: Lillykuty Jacob

Copyright © 2015 Lin Tang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a wireless differentiated queuing service (WDQS) algorithm to meet the diverse delay requirements in wireless multimedia sensor networks (WMSNs). WDQS adopts novel latest departure time (LDT) scheduling criteria to differentiate forwarding emergency by considering the packets' lifetime, the known delay it has already experienced, and the remaining delay it will experience. We also propose an effective approach to estimate the unknown delay for the remaining journey without any message overhead by exploiting the query mechanism of the sink. We further discuss analytically the packet's lifetime setting to meet the end-to-end (e2e) delivery requirement. The simulation results verify our analytical discussion and show performance improvements in terms of e2e delay and packet drop rate.

1. Introduction

The integration of low-power wireless networking technologies with inexpensive hardware such as complementary metal-oxide semiconductor (CMOS) cameras and microphones is now enabling the development of WMSNs, where wirelessly interconnected smart devices are deployed to retrieve video and audio streams, still images, and scalar sensor data [1]. WMSNs promise a wide range of potential applications in both civilian and military areas, such as surveillance sensor networks, law-enforcement reports, traffic control systems, advanced health care delivery, automated assistance to elderly telemedicine, and industrial process control [2].

A WMSN often consists of a large number of sensors, which constantly generate all kinds of traffic, including target detection data, measurement data, and video flows. The packets of these event-driven data belong to different levels of emergencies and have their own valid periods. Thus, the packets are required to be delivered to the sink before their expiration deadlines which are their quality-of-service (QoS) requirements. There are two challenges for WMSNs to support a huge amount of data that has diverse deadlines. First, the sensor nodes have very limited resources of energy

supply, bandwidth, memory, and processing capability due to the physically small size of sensors. The limited resources require a fair and efficient utilization. Second, the packets with the same level of emergency have the same deadline requirement. The packets from the sensors which are far from the sink have to traverse more hops to reach the sink than the packets from the sensors which are close to the sink. Thus the delay requirement of remote event report packets at every intermediate node is more stringent than that of nearby event report packets.

Packet scheduling is an important scheme to address the QoS requirement issues in WMSNs. Generally, packet scheduling assigns priorities to traffic flows with QoS considerations. Then packets are queued and scheduled according to the assigned priorities, which are related to the effective periods of event reports. If a packet cannot be sent to the sink in time, it will be discarded, resulting in high packet drop rate. With the limited resource in WMSNs, the packets of the most emergent event, which have the closest deadline, should be serviced with the highest priority. On the other hand, the geographic distribution of sensors also has impact on the scheduling priorities. The delay requirement of the packets from the remote sensors is more stringent than that of nearby

event report packets, so they should be serviced with higher priority. For these considerations, packet scheduling should be both deadline-aware and distance-aware [3].

Multiple kinds of packet scheduling algorithms have been already proposed. However most of them are not both deadline-aware and distance-aware. We find that our previous proposed differentiated queueing scheduling (DQS) fits WMSNs well. DQS is first put forward for the traditional wired network in [4]. It has been extended into wireless area in [5] and been improved in [6]. DQS queues packets at intermediate node by their LDT. The smaller the LDT, the higher the priority. A packet's LDT is determined by its lifetime, the known delay which it has already experienced and the remaining delay which it will experience. A packet's deadline is determined by the difference between the known delay and its lifetime. The known delay and remaining delay correspond to the distance from its source to the sink since our study is based on geographic routing. Thus the packet's priority is related to its deadline and the distance from its source to the destination. We can see DQS is both deadline-aware and distance-aware to meet the requirements of WMSNs. But how to estimate the unknown remaining delay and how to set the packet's lifetime have not been resolved in DQS.

In this paper, we propose a new packet scheduling algorithm: wireless differentiated queueing service (WDQS) for WMSNs. WDQS inherits the enqueue principle of DQS and resolves these two problems of DQS. Our contributions include the following.

- (i) *Effective delay estimation approach.* We propose an effective delay estimation for the remaining journey of a packet to implement WDQS by exploiting the unique characteristics of WMSNs. Our approach introduces no extra message overhead and energy consumption into WMSNs.
- (ii) *Packet lifetime setting and performance analysis.* We discuss the packet's lifetime setting approach and get the general setting conditions. We also analyze the e2e performance of traffic in WMSNs when applying WDQS.

The reminder of this paper is organized as follows. Section 2 presents the related works. Section 3 describes the system model. Section 4 proposes WDQS including scheduling principle, remaining delay estimating approach, packet's lifetime setting approach, and algorithm implementation. Section 5 verifies WDQS via simulations. Finally, the paper is summarized in Section 6.

2. Related Works

Multiple kinds of packet scheduling algorithms have been already proposed for traditional networks [7]. Priority Queuing (PQ) [8], Queue Length Threshold (QLT) [9], and proportional differentiation approaches such as Hybrid Proportional Delay (HPD) [10] and Waiting-time Priority (WTP) [11] are based on static priority. Earliest Deadline First (EDF) [12] assigns packets with static deadlines. It will service

the packets before their deadlines to guarantee their delay requirements. In the core stateless algorithms such as Core Stateless Fair Queuing (CSFQ) [13], all the packets are tagged by the edge nodes and the core nodes schedule the packet by its tag.

Few existing packet scheduling algorithms are dedicated to WMSNs. The literature [14, 15] applied EDF to WMSNs. The algorithms proposed in the literature [16] determine a packet's priority according to its elapsed hops. The reference [17] categorized dynamically real-time packets with high priority to preemptive data packets in the queues. Some algorithms proposed in the literature [18–20] focus on fairness and energy consumption rather than deadline-aware and distance-aware scheduling. Velocity Monotonic Scheduling (VMS) [3] assigns the priority of a packet based on its requested velocity which is the ratio of the packet's deadline and elapsed time to its Euclidean distance from the arriving node to the destination. It is one of the most suitable algorithms for WMSNs so far. It is both deadline-aware and distance-aware. However the distance is the Euclidean distance rather than the distance which a packet really passed.

3. System Model

We study packet scheduling algorithms on multitier heterogeneous architecture which is shown in Figure 1 [21], where the high tier is composed of high-end cameras, the middle tier is composed of low-end image sensors, and the bottom tier is composed of scalar sensors. Sensors from different tiers work in a coordinated way to detect and identify targets. When the scalar sensors or image sensors find some targets entering, the high-end cameras start to shoot and identify the targets through image processing.

The number of scalar and image sensors is large in WMSNs since they have simple circuit and low costs. These two types of sensors generally have the same task that is target detection. In this case, they collectively have the same traffic characteristics and requirements. Although each of them generates small amount of data and requires low bandwidth for communications, the aggregated traffic from these kinds of sensors is not ignorable. There are fewer camera sensors in WMSNs. However, they have stringent QoS requirement for their real-time video traffic. The real-time and non-real-time traffic often coexists in the same WMSNs. How to provide QoS guarantee for the mixed traffic for WMSNs is an important research issue [22].

4. Wireless Differentiated Queueing Service (WDQS)

As mentioned in Section 1, WDQS inherits the enqueue principles of DQS and resolves the problems which DQS has to face. In this section we will first describe the enqueue principles of DQS, then propose a delay estimating approach for the remaining journey of a packet, next put forward the packet lifetime setting conditions, at last describe the implementation of WDQS.

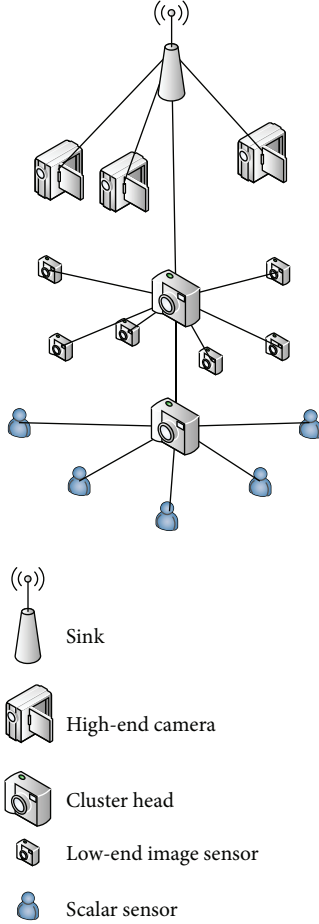


FIGURE 1: Multitier heterogeneous architecture of WMSNs.

TABLE 1: Notations for WDQS.

Sym.	Value
a	Report packet's arrival time
g	Report packet's departure time at the source
μ_{sink}	Querying packet's departure time
e	Report packet's latest departure time
D	Packet's lifetime
μ	Querying packet's e2e delay

WDQS is based on the following assumptions. Any application should have a maximum e2e delay (D). If an application can have an arbitrary D , it can be set to a certain value artificially. D should be treated as the lifetime of the associated packet. If a packet is confirmed that its lifetime has expired, it should be dropped immediately.

The notions in Table 1 are used to present our proposed WDQS.

4.1. Enqueue Principle of DQS. Suppose that parameters with subscript “ i ” denote that they are for node i . To simplify the discussion, the propagation delay is not addressed since it is a constant for a given path. Therefore, for a path consisting of

n nodes, we can have the following relation for the sojourn of a packet arriving at node i :

$$\begin{aligned}\hat{d}_i &= D - \sum_{j=1}^{i-1} \tilde{d}_j - \sum_{j=i+1}^n \tilde{d}_j \\ &= D - (a - g) - \sum_{j=i+1}^n \tilde{d}_j,\end{aligned}\quad (1)$$

where \tilde{d}_j is the actual delay that the packet experiences at node j , \hat{d}_i is the effective packet's maximum delay allowed at node i subject to its e2e delay, $\sum_{j=1}^{i-1} \tilde{d}_j$ is the delay that the packet has already experienced before it arrives at node i , and $\sum_{j=i+1}^n \tilde{d}_j$ is the delay that the packet will experience after it leaves node i . So the LDT e of this packet at node i is given by

$$\begin{aligned}e &= a + \hat{d}_i \\ &= D + g - \sum_{j=i+1}^n \tilde{d}_j.\end{aligned}\quad (2)$$

DQS will enqueue packets according to the value of e at node i . Thus, packets with smaller e will be serviced earlier. In multihop WMSNs, $\sum_{j=1}^i \tilde{d}_j$ depends on the distance from the source node to node i and $\sum_{j=i+1}^n \tilde{d}_j$ depends on the remaining distance from node i to the sink. According to (1) the longer the distance from the source to the sink, the smaller the value of e ; thus DQS is distance-aware. Also according to (2), the smaller the value of D and g , the smaller the value of e , so DQS is deadline-aware. Thus DQS is one of the most suitable packet scheduling algorithms of WMSNs.

4.2. Delay Estimation. From what was mentioned, we can see that the value of LDT of a packet (i.e., e) is critical to implement WDQS. It is difficult to implement WDQS since although the value of a , g , and D is known, the delay for the remaining journey (i.e., $\sum_{j=i+1}^n \tilde{d}_j$) can not be known in advance and is difficult to evaluate at node i . One possible approach is constantly sending probe packets and estimating the delay for the remaining journey by the replied messages [5]. This approach introduces extra message overhead and consumes more energy, which are both limited resources in WMSNs. We exploit the unique characteristics of WMSNs to estimate the delay for the remaining journey as follows.

There are some characteristics of WMSNs different from traditional wireless networks:

- (1) the destination of sensors' packets are the same sink;
- (2) the distance between sensor nodes and the sink is fixed;
- (3) the sink sends out querying packet periodically;
- (4) sensors' position information is available; thus,
- (5) geolocation based shortest path routing can be used for data transmissions.

We can evaluate the value of e more easily by taking advantage of these characteristics in WMSNs. Because of the characteristics (1), (2), and (4), the shortest path from node i to the sink is constant. According to characteristic (5), $\sum_{j=i+1}^n \tilde{d}_j$ is the delay along the remaining shortest path. According to characteristic (3), the querying packets will arrive at node i along the shortest path from the sink. So we can evaluate $\sum_{j=i+1}^n \tilde{d}_j$ by the e2e delay of the querying packet from the sink to node i .

Suppose μ_{sink} is the querying packet's departure time and μ_i is the querying packet's arrival time at node i . So the e2e delay of a querying packet from the sink to node i is given by

$$\mu = \mu_i - \mu_{\text{sink}}. \quad (3)$$

The sink sends out querying packets periodically, and the value of μ is updated accordingly. In practice, we adopt a weighted average approach to update the delay for the remaining journey μ ; that is,

$$\mu_d = (1 - \alpha) \mu_d + \alpha \mu, \quad (4)$$

where α ($0 \leq \alpha \leq 1$) is the smoothing factor. On the first arriving of the query packet, we set $\mu_d = \mu$. μ_d is updated by (4) on receiving the querying packets.

With the approach in (4), we can estimate the delay from node i to the sink as

$$\sum_{j=i+1}^n \tilde{d}_j = \mu_d. \quad (5)$$

Together with (2), the LDT for a packet is computed as follows:

$$e = D + g - \mu_d. \quad (6)$$

By (6), Packets with longer distance to the sink and later deadline will be serviced by WDQS with a higher priority. Note that the estimation of the delay for the remaining journey exploits the existing sink's query mechanism and thus introduces no extra message overhead and resource consumption to WMSNs.

4.3. Packets' Lifetime Setting and Performance Analysis. The real-time packets have stringent delay requirements so they must have higher priority than the scalar packets at the intermediate nodes. WDQS makes sure that all the real-time packets are delivered in time.

According to (6) only the packet's lifetime can be set artificially to affect its position in the queue of WDQS and further affect its QoS. If a real-time packet is serviced earlier than other scalar packets, its lifetime D must be set to a smaller value. However, if D is too small it might be discarded before reaching the destination. So in this section, we will discuss the approaches of lifetime setting. Then we will also analyze the forwarding ratio and e2e delay performance with these approaches. In the following discussion, we assume that the arrival process functions of the scalar packet and the real-time packet are known, and the notions in Table 2 are used.

TABLE 2: Notations for WDQS.

Sym.	Value
$a_{ci}(t)$	Scalar packet's arrival rate at node i
C_i	Forwarding rate of node i
g_T	g of real-time packets
e_T	e of real-time packets
D_T	D of real-time packets
δ_T	Forwarding ratio of real-time packets
d_T	Actual e2e delay of real-time packets
$A_{Ti}(t)$	Real-time packet's arrival process
$a_{Ti}(t)$	Real-time packet's arrival rate
L_i	Buffer length of node i
g_c	g of scalar packets
e_c	e of scalar packets
D_c	D of scalar packets
δ_c	Forwarding ratio of scalar packets
d_c	Actual e2e delay of scalar packets

4.3.1. Lifetime Setting. According to (6), e_T and e_c are given by

$$\begin{aligned} e_T &= D_T + g_T - \mu_d, \\ e_c &= D_c + g_c - \mu_d. \end{aligned} \quad (7)$$

In order to queue real-time packets always in front of scalar packets, there must be

$$e_T < e_c; \quad (8)$$

that is

$$D_T + g_T < D_c + g_c, \quad (9)$$

where $D_T < D_c$ can be set artificially. But if g_c is small enough, there may be $e_T > e_c$, and the real-time packet would be inserted into the queue behind some scalar packets. We will discuss how to set the lifetime of real-time packets and the scalar packets to meet the condition of $e_T < e_c$ in the following network scenarios.

- (1) If $a_{Ti}(t) > C_i$, the bandwidth can not satisfy the requirement of real-time communications. The network can not support this case.
- (2) If $a_{Ti}(t) + a_{ci}(t) < C_i$, there are adequate bandwidths for node i to forward all the packets. All the packets' QoS requirements are met.
- (3) Case of $a_{Ti}(t) + a_{ci}(t) \geq C_i$ but $a_{Ti}(t) < C_i$. The packet lifetime setting approach in this case is much more complicated, which is discussed in detail next.

Our lifetime setting approach for case (3) is making D_c and D_T meet the following condition expression:

$$D_c - D_T > \sum_{j=1}^n \frac{L_j}{C_j - a_{Tj}(t)}. \quad (10)$$

It is proved as follows.

Because $i \leq n$, we have

$$\sum_{j=1}^n \frac{L_j}{C_j - a_{T_j}(t)} > \sum_{j=1}^{i-1} \frac{L_j}{C_j - a_{T_j}(t)}. \quad (11)$$

If (10) is met, we get

$$D_c - D_T > \sum_{j=1}^{i-1} \frac{L_j}{C_j - a_{T_j}(t)}. \quad (12)$$

Suppose e of every real-time packet is smaller than e of all the scalar packets which are waiting in the queue; the forwarding rate for scalar packets at node i is $C_i - a_{T_i}(t)$. Suppose again that a scalar packet arrives at node i with l_i packets already in the buffer queue; the queueing delay of this scalar packet at node i will be

$$\tilde{d}_i \leq \frac{l_i}{C_i - a_{T_i}(t)} \leq \frac{L_i}{C_i - a_{T_i}(t)}. \quad (13)$$

With (12), we get

$$D_c - D_T > \sum_{j=1}^{i-1} \tilde{d}_j. \quad (14)$$

This scalar packet's experienced delay from the source to node i is $\sum_{j=1}^{i-1} \tilde{d}_j$. We can then infer its departure time at the source as

$$g_c = a - \sum_{j=1}^{i-1} \tilde{d}_j. \quad (15)$$

When a real-time packet is always serviced first, there is no queuing delay for it. So if a real-time packet arrives at node i at the same time, its departure time is $g_T \approx a$ since the propagation delay is very small. So

$$g_T - g_c \approx \sum_{j=1}^{i-1} \tilde{d}_j. \quad (16)$$

With (14) there will be $D_c + g_c > D_T + g_T$; that is, $e_T < e_c$. Thus we prove that if (10) is met, $e_T < e_c$.

4.3.2. Performance Analysis. Now let us look into the performance of real-time traffic. In our analysis, we study the metrics of the forwarding ratio and the e2e delay. The forwarding ratio is defined as the ratio of the forwarding rate to the arriving rate.

Obviously, when (10) is met, all the real-time packets will be serviced earlier than the scalar packets. Thus, all arriving real-time packets are forwarded and there is no queueing delay; that is, $\delta_T = 1$ and $d_T = 0$. When (10) is not met, there exists a critical queue length l_{cr} determining the enqueueing order of real-time and scalar packets. If the queue length is less than l_{cr} , the real-time packets are always queued in front of the scalar packets. Otherwise, some scalar packets will go ahead of a real-time packet. Suppose the time reaching l_{cr} is t_{cr} , when $e_T = e_c$.

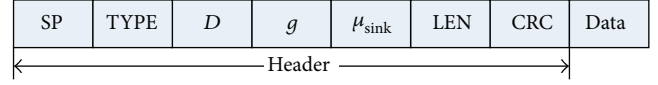


FIGURE 2: Packet format for WDQS implementation.

The queue length grows over time. At the time t_{cr} ,

$$l_{cr} = \int_0^{t_{cr}} a_{T_i}(t) dt + \int_0^{t_{cr}} a_{ci}(t) dt - C_i t_{cr}. \quad (17)$$

According to (6), when $D_c + g_c = D_T + g_T$, there would be $e_T = e_c$. It means at t_{cr} , $g_T - g_c = D_c - D_T$. What is more is that there are only scalar packets in the queue before t_{cr} . From this perspective,

$$l_{cr} = \int_0^{D_c - D_T} a_{ci}(t) dt. \quad (18)$$

Then,

$$\int_0^{t_{cr} - D_c + D_T} a_{ci}(t) dt + \int_0^{t_{cr}} a_{T_i}(t) dt - C_i t_{cr} = 0. \quad (19)$$

We can get the value of t_{cr} from the above equation by applying the specific forms of arrival rate. Based on t_{cr} , we study the following two cases.

- (i) $t \leq t_{cr}$: the performance of this case is the same as that in the case satisfying (10).
- (ii) $t > t_{cr}$: real-time packets and scalar packets will share the bandwidth in this case. The forwarding rates for real-time and scalar packets are $a_{T_i}(t)C_i/(a_{ci}(t) + a_{T_i}(t))$ and $a_{ci}(t)C_i/(a_{ci}(t) + a_{T_i}(t))$. Thus, their forwarding ratios are

$$\delta_T = \frac{C_i}{a_{ci}(t) + a_{T_i}(t)}, \quad (20)$$

$$\delta_c = \frac{C_i}{a_{ci}(t) + a_{T_i}(t)}, \quad (21)$$

respectively, subject to

$$d_T < D_T, \quad (22)$$

$$d_c < D_c. \quad (23)$$

4.4. Implementation of WDQS. In this subsection, we discuss the implementation of WDQS in WMSNs.

WDQS requires a slight modification to the packet format. Figure 2 shows the packet format expected by WDQS. The field TYPE distinguishes the packet and is one of the following: report packet and querying packet. SP refers to the position of its source node. All the report packets' destinations are the sink. Querying packets come from the sink and they are broadcasting packets. So we need not indicate the destination position in the packet header. In the report packets' header, μ_{sink} is zero, and D and g are set by sensor nodes. In querying packets' header, μ_{sink} is set by the

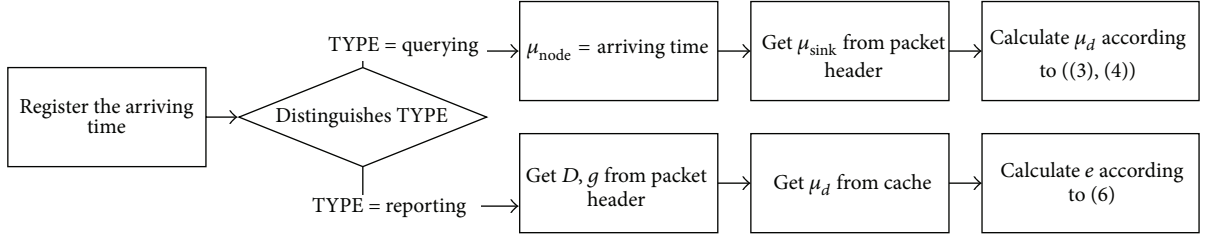
FIGURE 3: Calculation process of the LDT e at node i .

TABLE 3: Parameter settings in the simulations.

Sym.	Value
Area of sensor field	$100 * 100 \text{ m}^2$
Radio range of a sensor node	40 m
IFQ length (L)	4 kb
Receive power	0.395 W
Mean arriving rate of real-time packet (λ_T)	320 kbps
Forwarding rate (C)	480 kbps
Number of sensor nodes	200
Packet length	30 bytes
Transmit power	0.66 W
Decision interval	10 sec
Mean arriving rate of scalar packet (λ_c)	240 kbps

TABLE 4: The value of D_T .

Experiment	1	2	3	4	5	6	7	8	9
Value of D_T (ms)	60	80	100	140	160	180	280	300	320

sink. LEN is the length of the entire packet and CRC is the checksum.

Figure 3 shows the calculating process of e when a packet arrives at node i .

5. Simulation

We implement and study WDQS in NS2. 200 sensor nodes were randomly positioned in a 100×100 sensor field. Node parameters such as radio range were carefully chosen to mirror typical sensor mote values [23]. One of these nodes was chosen as the sink to which all source data was sent. Real-time packets' source nodes and scalar packets' source nodes were randomly chosen and generate Poission traffics. In order to communicate source data to the sink, we employed a simple CSMA/CA based MAC protocol and Geographic and Energy Aware Routing (GEAR) [24]. The parameter settings are shown in Table 3. According to the routing mechanism of GEAR, the area of sensor fields, and the radio range of a sensor node, the number of nodes of the longest path from source to sink is $n = 3$. With the parameters in Table 3, we have the following conditional expression according to (10):

$$D_c - D_T > 75 \text{ ms.} \quad (24)$$

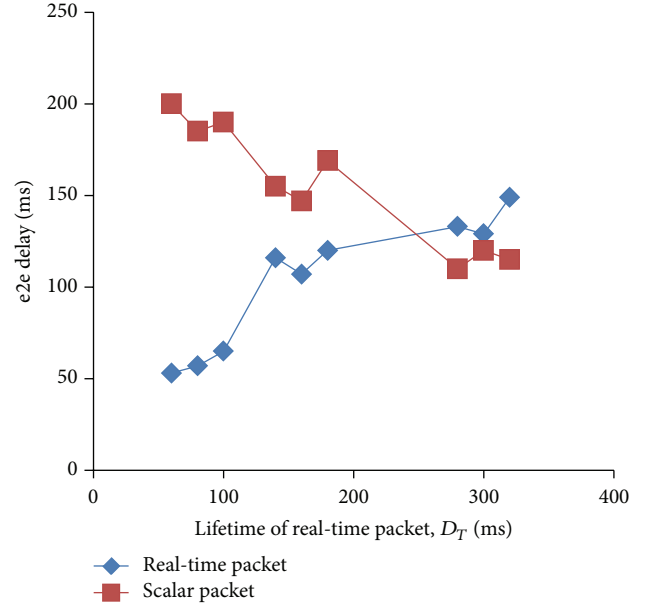


FIGURE 4: Packet's e2e delay.

We conduct two sets of experiments. The first set of experiments verify the effectiveness of lifetime setting condition. The second set of experiments compare WDQS with some famous packet scheduling algorithms.

5.1. Experiment Set 1. In this set of experiments D_c is fixed and $D_c = 200$ ms. We conduct 9 experiments to study packets' e2e delay and drop rate with the change of D_T . The value of D_T of every experiment is shown in Table 4. In the table, the first three D_T meet (10), the middle three $D_T < D_c$ but do not meet (10), and the last three D_T are much larger than D_c . The results are shown in Figures 4 and 5. Figure 4 shows the e2e delay of real-time packets and scalar packets. Figure 5 shows the packet drop rate. From the simulation results we find the following.

- (i) In the first three experiments, the e2e delay of real-time packets is very small and that of scalar packets is much larger. The drop rate of real-time packets is much smaller than that of scalar packets. That is because when (10) is met, real-time packets are all forwarded with higher priority.

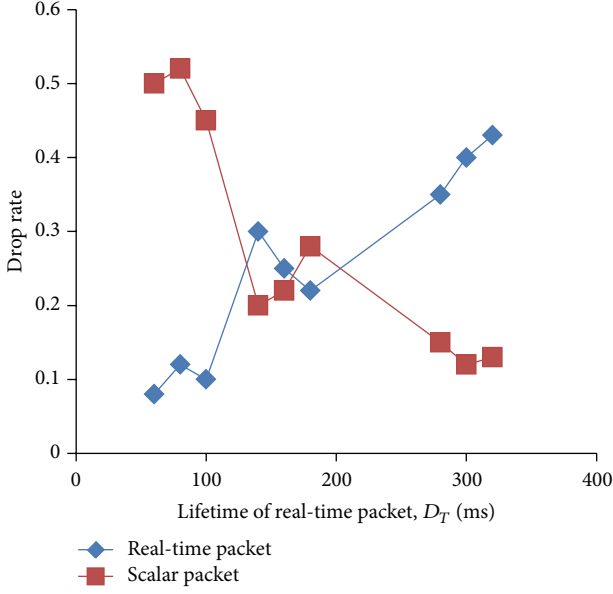


FIGURE 5: Packet drop rate.

- (ii) In the middle three experiments, the e2e delay of real-time packets increases and that of scalar packets decreases. Drop rates of these two kinds of packets become close to each other. That is because when $D_T < D_c$ but do not meet (10), packets are fair forwarded, real-time packets and scalar packets share the bandwidth equitably.
- (iii) In the last three experiments, the e2e delay of real-time packets increases further and that of scalar packets drops further. The drop rate of real-time packets becomes much larger and that of scalar packets becomes smaller. That is because when D_T is much larger than D_c , scalar packets are forwarded with higher priority while real-time packets only take the remaining bandwidth.

From the simulation results, we can see the conditional expression (24) is reasonable. That is to say the conditional expression (10) is suitable for WMSN applications.

5.2. Experiment Set 2. In this set of experiments we compare WDQS with FCFS, EDF, DM, and VMS. FCFS is selected as an opponent in order to show the great improvements of network performances made by WDQS compared with nonpriority scheduling algorithms. EDF and DM are famous scheduling algorithms, priorities of which are all related to the packet's lifetime. VMS is one of the most suitable algorithms for WMSNs so far as mentioned in Section 2, priorities of which are also related to the packet's lifetime. We compare with these algorithms in order to show the performance improvements compared with the same type of algorithms. We let $D_c = 200$ ms and $D_T = 100$ ms which meet (10). We simulate three times to study e2e delay and drop rate with the change of λ_T . Other parameters and experiment topology are the same as the first set of experiments. The results are shown

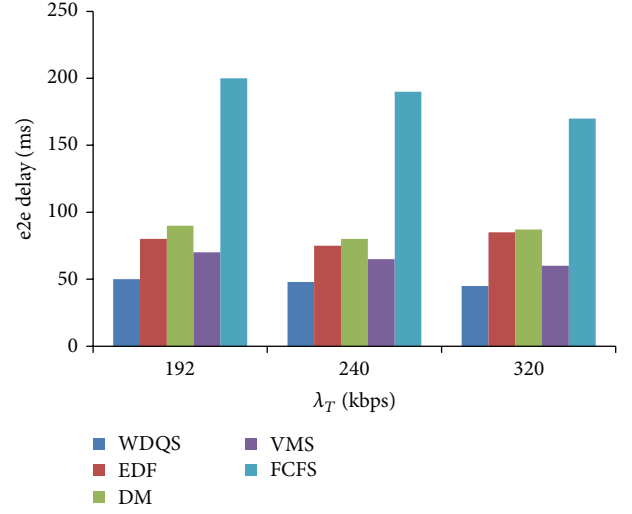


FIGURE 6: Packet's e2e delay comparison.

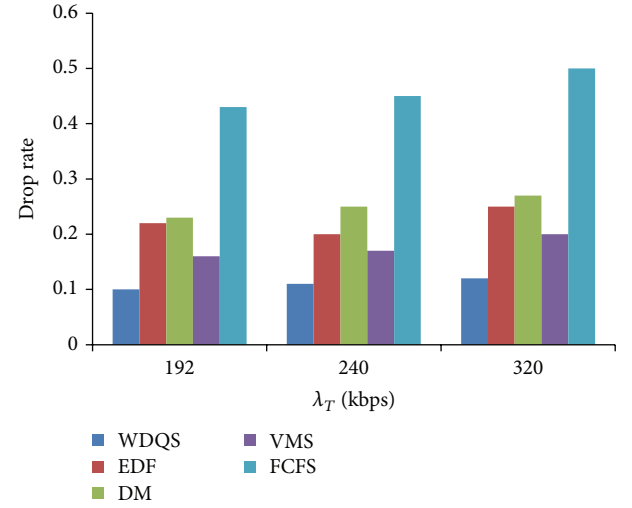


FIGURE 7: Packet drop rate comparison.

in Figures 6 and 7. From the simulation results we find that the e2e delay and the drop rate of WDQS are the smallest. Packet's e2e delay and drop rate of FCFS are much larger than those of WDQS. That is to say WDQS improves the network performance greatly and it has better performance than other packet scheduling algorithms which are chosen.

All in all, the simulation results in this section show that our proposed WDQS improves the network performances in WMSNs and the packet lifetime setting approach is effective.

6. Conclusion

We have studied the QoS issues in WMSNs and propose a WDQS algorithm for packet scheduling to support the QoS requirements of WMSN applications. We put forward an estimating approach of remaining delay and a conditional expression of lifetime setting in order to calculate the LDT of a packet which affects the packet's position in the queue.

WDQS is deadline-aware and distance-aware by exploiting the unique characteristics of WMSNs to guarantee the end-to-end delay of packets with diverse QoS requirements in WMSNs. Simulation results show the performance improvements in provisioning e2e QoS in WMSNs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the 973 Programs of China (no. 2011CB707003), the NSFC (nos. 61302058 and 61201256), the Fundamental Research Funds for the Central Universities of China (no. 2014ZZ0030), the Key Grant Project of Chinese Ministry of Education (no. 313021), and the NCET Program (no. NCET-12-0196).

References

- [1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "Wireless multimedia sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32–39, 2007.
- [2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "Wireless multimedia sensor networks: applications and testbeds," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1588–1605, 2008.
- [3] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "RAP: a real-time communication architecture for large-scale wireless sensor networks," in *Proceedings of the 8th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '02)*, pp. 55–66, September 2002.
- [4] S. Jiang, "Granular differentiated queueing services for qos: structure and cost model," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 2, pp. 13–22, 2005.
- [5] X. Teng, S. Jiang, G. Wei, and G. Liu, "A cross-layer implementation of differentiated queueing service (DQS) for wireless mesh networks," in *Proceedings of the IEEE Vehicular Technology Conference (VTC '08)*, pp. 2233–2237, IEEE, Singapore, May 2008.
- [6] S. Jiang, "Differentiated queueing service (DQS) for granular QoS," in *Future Wireless and Optical Networks*, pp. 73–87, Springer, Berlin, Germany, 2012.
- [7] G. Hasegawa, T. Matsuo, M. Murata, and H. Miyahara, "Comparisons of packet scheduling algorithms for fair service among connections on the Internet," *Journal of High Speed Networks*, vol. 12, no. 1, pp. 1–27, 2003.
- [8] V. Jacobson, K. Nichols, K. Poduri et al., "An expedited forwarding phb," RFC 2598, 1999.
- [9] A. K. Choudhury and E. L. Hahne, "Dynamic queue length thresholds for shared-memory packet switches," *IEEE/ACM Transactions on Networking*, vol. 6, no. 2, pp. 130–140, 1998.
- [10] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional differentiated services: delay differentiation and packet scheduling," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 109–120, 1999.
- [11] C. Dovrolis and D. Stiliadis, "Relative differentiated services in the internet: issues and mechanisms," *ACM SIGMETRICS Performance Evaluation Review*, vol. 27, no. 1, pp. 204–205, 1999.
- [12] M. Andrews, "Probabilistic end-to-end delay bounds for earliest deadline first scheduling," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 2, pp. 603–612, Tel Aviv, Israel, March 2000.
- [13] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: a scalable architecture to approximate fair bandwidth allocations in high-speed networks," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 33–46, 2003.
- [14] M. Yu, S. J. Xiahou, and X. Y. Li, "A survey of studying on task scheduling mechanism for TinyOS," in *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 1–4, Dalian, China, October 2008.
- [15] K. Mizanian, R. Hajisheykhi, M. Baharloo, and A. H. Jahangir, "Race: a real-time scheduling policy and communication architecture for large-scale wireless sensor networks," in *Proceedings of the 7th Annual Communication Networks and Services Research Conference (CNSR '09)*, pp. 458–460, May 2009.
- [16] E.-M. Lee, A. Kashif, D.-H. Lee, I.-T. Kim, and M.-S. Park, "Location based multi-queue scheduler in wireless sensor network," in *Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT '10)*, vol. 1, pp. 551–555, February 2010.
- [17] N. Nasser, L. Karim, and T. Taleb, "Dynamic multilevel priority packet scheduling scheme for wireless sensor network," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1448–1459, 2013.
- [18] F. Tirkawi and S. Fischer, "Adaptive tasks balancing in wireless sensor networks," in *Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA '08)*, pp. 1–6, April 2008.
- [19] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Transactions on Communications*, vol. 60, no. 1, pp. 220–230, 2012.
- [20] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in LTE cellular networks: key design issues and a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 678–700, 2013.
- [21] T. Melodia and I. F. Akyildiz, "Research challenges for wireless multimedia sensor networks," in *Distributed Video Sensor Networks*, pp. 233–246, Springer, London, UK, 2011.
- [22] M. H. Yaghmaee and D. A. Adjeroh, "Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks," *Computer Networks*, vol. 53, no. 11, pp. 1798–1811, 2009.
- [23] Y. Sankarasubramaniam, Ö. B. Akan, and I. F. Akyildiz, "ESRT: event-to-sink reliable transport in wireless sensor networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pp. 177–188, ACM, June 2003.
- [24] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks," Tech. Rep. ucla/csd-tr-01-0023, UCLA Computer Science Department, 2001.

