DOI: 10.1051/matecconf/20167506004

Dynamic Multi Threshold Priority Packet Scheduling Algorithms

Tamer Dag and Sezer Uzungenc

Computer Engineering Department, Kadir Has University, 34083Istanbul, Turkey

Abstract. Packet scheduling algorithms are developed in order to use shared transmission resources efficiently. Various application packets such as real and non-real time packets might have different QoS requirements and traditional scheduling algorithms might be insufficient to respond to the applications needs. In this paper, two packet scheduling algorithms are proposed to overcome this problem: Dynamic multi threshold priority packet scheduling (DMTPS) and dynamic multi threshold priority with urgency packet scheduling (DMTPUS). The proposed algorithms aim to provide a better QoS level with a decrease in delay time and loss ratio for the low priority packets while still maintaining acceptable fairness towards high priority packets. To evaluate the performance of DMTPS and DMTPUS algorithms, they are compared with the commonly used scheduling algorithms such as first come first served (FCFS) and fixed priority. Simulation results illustrate that the dynamic multi threshold priority packet scheduling algorithms can provide a better QoS for low priority packets without decreasing the QoS levels of high priority packets.

1 Introduction

A Wireless Sensor Network (WSN) consists of tiny sensor nodes which are low-cost and low-energy and with sensing, data processing and communicating components. A WSN can be realized with many sensor nodes that are deployed closely. The feature of sensor networks is the cooperative effort of the sensor nodes. Sensor nodes are fitted with an onboard processor. Instead of sending the raw data to the nodes responsible for the fusion, they use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data [1].

WSNs are used in various areas such as military, health and home. For military applications, the rapid deployment, self- organization, and fault tolerance characteristics of WSNs make them a very promising sensing technique for military command, control, communications, computing, intelligence, surveillance, reconnaissance, and targeting systems. In health, they are used for monitoring patients and assisting disabled patients with deployment of sensor nodes. Managing inventory, monitoring product quality, and monitoring disaster areas are some other applications where WSNs are used [1].

In WSNs, sensor nodes can be mobile and heterogeneous. They are scalable to large scale of deployment. Design of a WSN is cross-layer and it is becoming an important study area for wireless communications. Cross layer design is used for improving the transmission performance which is energy efficiency, data rate and Quality of Service (QoS).

WSNs may be deployed in various environments and energy determines the lifetime of a WSN. Communication in sensor nodes is a key component and protocols and algorithms must satisfy the issues such as increasing of lifespan, self-configuration, robustness and fault tolerance. WSNs have a problem with energy consumption. Energy is the most important resource of WSN nodes [2], [3].

Scheduling is the strategy by which the system decides which task should be executed at any given time. Generally, it is used for load balance and system resources to be served effectively and with desired quality [4]. Scheduling algorithms are classified as preemptive and non preemptive. In preemptive scheduling, arrival of high priority data suspends current lower priority data. In non preemptive scheduling, the current lower priority data is processed until it is completed. Preemptive schedulers provide better performance for system utilization. Non preemptive schedulers encounter the starvation problem [5]. Non preemptive schedulers may give better performance and robustness on reduced systems [6].

The major scheduling algorithms for WSNs are FCFS and fixed priority non-preemptive scheduling. In FCFS algorithm, each priority level of data's loss ratio and delay time results are assumed to be the same since priority is not criteria in this algorithm [7]. In fixed priority non-preemptive scheduling algorithm, lower priority level data can get lost and delayed excessively and the higher priority level data are processed more quickly and there is a definite unfairness between priority levels [8].

In this work, we develop a new scheduling algorithm for WSNs which can provide a better QoS and resolve the disadvantages of the current algorithms. The proposed Dynamic Multi Threshold Priority packet scheduling algorithms try to maximize the level of QoS provided to different priority packets by reducing the loss ratio and waiting times. Also, the proposed scheduling algorithms are non preemptive due to structure of reduced system load of WSNs.

The rest of the paper is organized as follows: Section 2 describes some of the most commonly used Queue Scheduling Algorithms for WSNs, Section 3 introduces the proposed dynamic multi threshold priority packet scheduling algorithms. Simulation results of the algorithms are shown in Section 4 and finally Section 5 concludes this paper.

2 Packet scheduling algorithms

In this section, the commonly used scheduling algorithms FCFS and fixed priority non-preemptive scheduling are described briefly. By comparing these algorithms against proposed DMPTS and DMPTUS algorithms, the effectiveness of the proposed algorithms is evaluated.

FCFS algorithm has a disadvantage of leading very high loss ratios and large delay times for higher priority real-time data. FCFS treats all packets the same way since data packets are ordered and transmitted according totheir arrival times only. The priority of a packet is not a criterion for the ordering process [7]. As a result, large delay times for higher priority data are experienced and delay time guarantees might not be supplied for some cases. Similarly, the loss ratios for any type of packets will be similar. Providing same level of QoS to different types of applications is the main disadvantage of FCFS.

In fixed priority non-preemptive scheduling algorithm, the packets are scheduled according to their arrival times and their priorities [8]. When a new packet arrives, it is ordered according to its priority level in the queue. However, this algorithm favors only high priority packets. Lower priority packets can be dropped excessively as they will be placed towards the end of the queue. With fixed priority non-preemptive scheduling algorithm, the QoS levels provided for high priority packets will be very high. On the other hand, lower priority packets will be in danger of starvation [9]. A steady arrival of large priority packets would result in never transmitting low priority packets, as they will always be waiting for the completion of transmission of high priority packets. Thus, the algorithm might be very unfair for some application types. Thus, the main disadvantage of fixed priority non-preemptive scheduling is unfairness.

3 Dynamic multi threshold priority packet scheduling algorithms

In this section, the proposed dynamic multi threshold priority packet scheduling (DMTPS) and dynamic multi threshold priority with urgency packet scheduling (DMTPUS) algorithms are described.

3.1 Dynamic multi threshold priority packet scheduling algorithm (DMTPS)

The first proposed algorithm is DMTPS. With DMTPS, packets are placed into the waiting queue according to their priorities and each priority level has a distinct threshold value. The threshold levels with the occupancy of the queue is used to determine the ordering of the packets. The basic threshold is shown in Fig. 1. When a new packet arrives, the priority level change decision is made based on if the queue occupancy is higher than the threshold value.

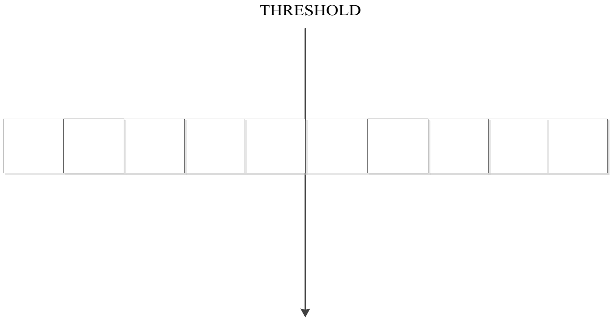

Figure 1. Basic threshold structure at the waiting queue.

Three levels of priorities are defined for the DMTPS algorithm. High, medium and low priority. The number of priority levels may be adjusted to a desired level, based on the real system but for the rest of this paper, three levels of priorities will be used.

The threshold values for each priority level are specified with different values:

For priority 1 or high priority packets, the threshold is equivalent to %k of the total capacity in the waiting queue.

For priority 2 or medium priority packets, the threshold is equivalent to %m of the total capacity in the waiting queue.

And for priority 3 or low priority packets, the threshold is equivalent to %n of the total capacity in the waiting queue.

In order to be fair towards low priority packetsbut at the same time satisfy the QoS requirements of high priority packets, the threshold values are ordered as:

$n > m > k$.

These values are also dynamic, and can be changed depending on the application types or a congestion in the network.

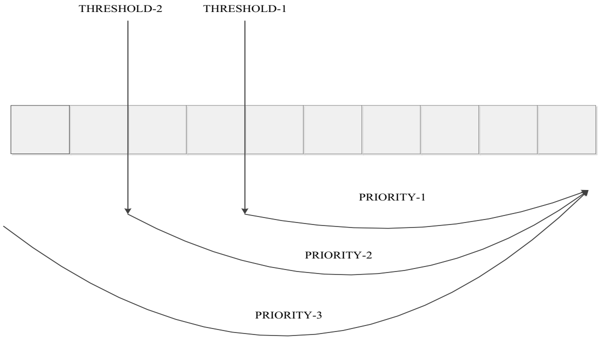

Figure 2. Three levels of priority with their corresponding threshold values.

Fig. 2 shows an example of the representation of initial values of threshold values. For this figure, it is assumed that for low priority packets, the threshold level is %100.

The basic working principle of DMTPS algorithm is as follows:

When a packet arrives and if the queue occupancy is less than the threshold value for that packet's priority level, the packet is placed at the end of the queue just like in FCFS. The newly arriving packet will not be placed in front of lower priority packets as it is still possible to satisfy its QoS requirements. This case allows, low priority packets to be transmitted without starvation.

When a packet arrives and if the queue occupancy is greater than the threshold value for that packet's priority level, the new packet needs to be placed somewhere in the queue between its threshold and the head of the queue to satisfy its QoS requirements. If a lower priority packet is available between the corresponding threshold value and the head of the queue, it is replaced with the newly arriving packet. If the queue is full and a lower priority packet is available when a new packet arrives, the newly arriving packet will take the place of the lower priority packet, but in this case, lower priority packet will need to be dropped.

When a packet arrives and if the queue occupancy is greater than the threshold value for that packet's priority level or if the queue is full, and there is no lower priority packet between its threshold and the head of the queue, the new arrival will be treated as in FCFS.

3.2 Dynamic multi threshold priority with urgency packet scheduling algorithm (DMTPUS)

The main objective of the DMTPUS algorithm is similar to the DMTPS algorithm. The multi threshold structure and packet ordering is the same as DMTPS. However, the major difference is the urgent status of the packets. An urgent status packet needs to be processed as quickly as possible with DMTPUS.

DMTPUS is an efficient technique for scheduling of packets which are urgent and should be processed immediately. Urgent status is defined as a field (*u*) in the packet and if a packet is marked as urgent it is upgraded. Table 1 shows the priority levels of packets with or without urgent status.

Table 1. Urgent status

Priority Level	**Urgent**	**Not Urgent**
Priority 1	Marked as the Most Valuable Packet (The highest priority level and higher than Priority 1)	Priority 1
Priority 2	Priority 1	Priority 2
Priority 3	Priority 2	Priority 3

Packets which have urgent statuses are evaluated as follows:

A priority 1 or the highpriority level packet with urgent status is considered to be the most urgent packet in the system. Such a packet needs to be delivered immediately. A packet in this case is directly added to head of the waiting queue because it should be processed immediately. All remaining packets in the queue will shift one step back with the arrival of such a packet. Fig. 3 shows an example of the behavior of the system when a high priority level packet with urgent status arrives.

A priority 2 or the medium priority level packet with urgent status is upgraded and marked as a priority 1 packet. This packet afterwards is treated as a priority 1 packet in the DMTPS algorithm.

A priority 3 or the low priority level packet with urgent status is upgraded and marked as a priority 2 packet. This packet afterwards is treated as a priority 2 packet in the DMTPS algorithm.

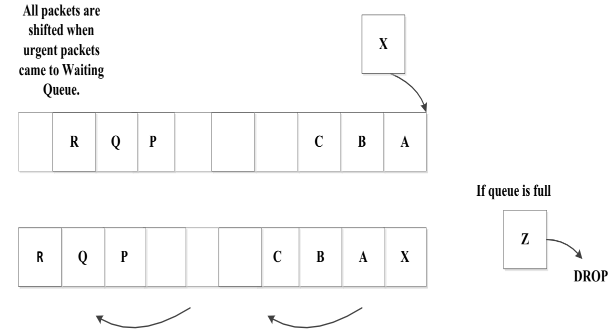
All packets are shifted when urgent packets came to Waiting Queue.
X
R
Q
P
C
B
A
If queue is full
Z
DROP
R
Q
P
C
B
A
X

Figure 3. Arrival a packet X, priority level 1 with urgent status

4 Simulation results

In this section, the performance of the proposed DMTPS and DMTPUS algorithms are analyzed through simulation studies. The simulations are performed using discrete event system simulation technique in Java. For the simulations, two important QoS metrics are evaluated: The loss ratio and the average delay. These metrics for the proposed algorithms are compared with FCFS and fixed priority non-preemptive scheduling algorithms.

The utilization of the queue is a factor that has impact on the value of the loss ratio and the delay time results. The value of both loss ratio and the delay time increases with increased utilization. The capacity of the waiting queue is also an important factor that affects the QoS metrics. The loss ratio seen in the system decreases with the increase of capacity.

For the simulations, in order to observe the operation of the algorithms under heavy traffic, the utilization of the queue is assigned as 0.99. Capacity of the waiting queue is assigned with values between 20 and 100.

The threshold level values described in Section 3.1 are chosen as 60, 80 and 100 for the k*, m* and *n* values respectively. The urgent status described in Section 3.2 for the DMTPUS algorithm is applied randomly to %20 of the packets of any priority level.

For DMTPS, DMTPUS, FCFS and priority scheduling algorithms, high, medium, and low priority results are plotted separately.

4.1 Loss ratio

Loss ratio results of all algorithms within each priority levels are shown in Fig. 4.

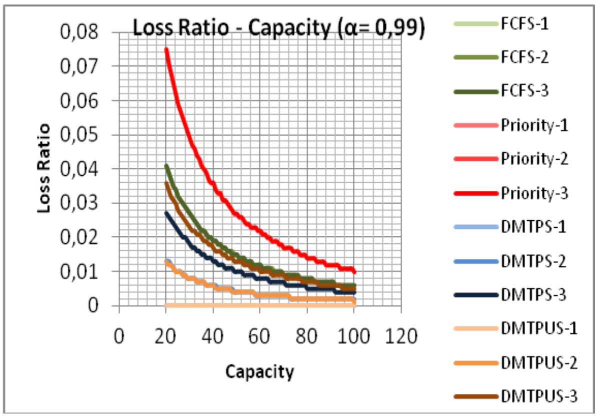

Figure 4. Loss ratio vs. capacity for DMTPS, DMTPUS, FCFS and priority scheduling.

In FCFS algorithm, all loss ratio results of priority levels are similar with each other. Because, priority value of packet is not important for FCFS algorithm. In fixed priority non-preemptive scheduling algorithm, the higher priority packets whose priorities are 1 and 2 level's loss ratio are less than packets which have the lowest priority level. Because priority is important at the fixed priority non-preemptive scheduling algorithm and order is specified according to packet's priority. However, the lowest priority packets loss is very high.

Dynamic multi threshold priority packet scheduling algorithms are given higher QoSlevels for the lowest priority when compared with fixed priority non-preemptive scheduling algorithm.

DMTPS and DMTPUS algorithms have similar loss ratio results. Loss ratio results of packets which are higher priority level (Priority-1 and Priority-2) are similar to the fixed priority non-preemptive scheduling algorithm and the lowest priority level packets loss ratio results are similar with the FCFS algorithm. Threshold algorithms improve loss ratio results and these algorithms provide efficiency to all priority levels. The significant alteration is observed at the lowest priority level packets loss ratio results.

4.2 Average delay time

Average delay time results of all algorithms within each priority levels are shown at Fig. 5.

In FCFS algorithm, total delay time and average delay time results are similar and equal to each other for all priority levels.

In fixed priority non-preemptive scheduling algorithm, total delay time level and average delay time of each priority is associated with loss ratio results. Delay time results of threshold algorithms are similar. Comparing threshold algorithms with fixed priority non-preemptive scheduling algorithm and FCFS algorithm have different results at average delay time analysis. The lowest priority level packets are delayed more than threshold algorithms, but the higher priority level packets are delayed less than threshold algorithms. FCFS algorithm has consistent results on all priority levels.

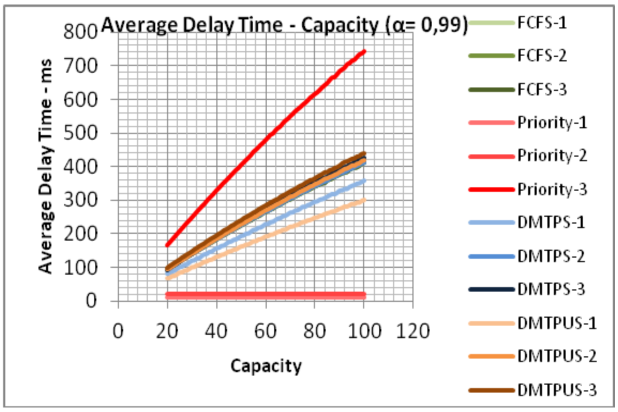

Figure 5. Average delay time vs. capacity for DMTPS, DMTPUS, FCFS and priority scheduling.

Threshold algorithms aim at the lowest priority packets delay less than fixed priority non-preemptive scheduling algorithm and the higher priority packets delay less than FCFS algorithm. The efficiency on the average delay time is reached with dynamic threshold algorithms. DMTPS and DMTPUS algorithms provide improvement on all priority level's loss ratio and delay time results.

5 Conclusions

All algorithms are experimented and analyzed for improving various packet scheduling algorithm's QoS. In this paper, we propose dynamic multi threshold priority packet scheduling algorithms for WSNs. These algorithms ensure a decrease in delay time and loss ratio for the lower priority levels with acceptable fairness towards higher priority data. Dynamic multi threshold priority packet scheduling algorithms have better performance than the existing FCFS and fixed priority non-preemptive scheduling algorithm in terms of the average delay time and loss ratio. All algorithms are analyzed according to contribution of QoS. The results show that proposed algorithms improve the QoS of selected attributes. DMTPS algorithm provides higher QoS than DMTPUS algorithm which algorithm also has a good QoS in decreasing of average delay time. Urgency and threshold optimization factors are another factor that can be improved for future works.

References

1. I. F. Akyildiz, W. Su, Y.Sankarasubramaniam, E. Cayirci, A Survey on Sensor Networks**,** Georgia Institute of Technology, IEEE Communications Magazine, August (2002).
2. M. Bernard T. and Fouchal H., A low energy consumption MAC protocol for WSN, IEEE Communication Conference, ICC, (2012).
3. S. Kim, Y. Kim, An energy efficient priority-based MAC protocol for wireless sensor networks,

International Conference ICT Convergence ICTC, (2012).

4. K. Ramamritham, J.A. Stankovic, Scheduling Algorithms and Operating Systems Support for Real-Time Systems, University of Massachusetts, Amherst, (1994).
5. A. Iqbal, A. Zafar, B. Siddique, Dynamic Queue Deadline First Scheduling Algorithm for Soft Real Time System, IEEE International Conference on Emerging Technologies, (2005).
6. Aloysius K. Mok, Wing-Chi Poon, Non-Preemptive Robustness under Reduced System Load, Proceedings of the 26th IEEE International Real-Time Systems Symposium, (2005).
7. L. XinYan, L. PingPing, Research and Improvement of Real-Time Queue Scheduling Algorithm, International Forum on Information Technology and Applications, (2010).
8. C. Taddia, G. Mazzini, On the Jitter Performance of FIFO and Priority Queues mixture, The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, University of Ferrara, Italy, (2006).
9. N. Nasser, L. Karim, T. Taleb, Dynamic Multilevel Priority Packet Scheduling Scheme for Wireless Sensor Network, IEEE Transactions on Wireless Communications, **12**, No.4, (2013).