



## Utilizing priority-based packet scheduling based on preemptive priority scheduling in wireless sensor networks



Lethabo Mathebula \*

School Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Durban, South Africa

### ARTICLE INFO

#### Article history:

Received 17 September 2019

Received in revised form

10 January 2020

Accepted 20 January 2020

#### Keywords:

Wireless sensor networks

FCFS

Priority packet scheduling

Pre-emptive

Non-preemptive

Real-time packets

Non-real-time packets

### ABSTRACT

In this paper, each node has three levels of priority queues. Real-time packets are placed into the highest-priority queue and can preempt data packets in other queues. Non-real-time packets are placed into two other queues based on a certain threshold of their estimated processing time. Leaf nodes have two queues for real-time and non-real-time data packets since they do not receive data from other nodes and thus, reduce end-to-end delay. The priority packet scheduling scheme outperforms conventional schemes in terms of average data waiting time and end-to-end delay and also it reduces sensor energy consumption.

© 2020 The Authors. Published by IASE. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

### 1. Introduction

A priority-based packet scheduling scheme is proposed which aims at scheduling different types of data packets, such as real-time and non-real-time data packets at sensor nodes with resource constraints in Wireless Sensor Networks. Most of the existing packet-scheduling mechanisms of Wireless Sensor Networks use First Come First Served (FCFS), non-preemptive priority, and preemptive priority scheduling algorithms. These algorithms result in long end-to-end data transmission delay, high energy consumption, and deprivation of high priority real-time data packets also it results in an improper allocation of data packets to queues. Moreover, these algorithms are not dynamic to the changing requirements of Wireless Sensor Network applications since their scheduling policies are predetermined.

Wireless sensor networks are a trend of the past few years, and they involve deploying a large number of small nodes. The nodes then sense environmental changes and report them to other nodes over flexible network architecture. Sensor nodes are used in hostile environments or over large geographical areas. A wireless sensor network consists of hundreds or thousands of low-cost nodes which could either have a fixed location or be randomly deployed to monitor the environment. Due to their small size, they have several limitations, an issue that I will discuss later. Sensors usually

communicate with each other using a multi-hop approach. The flow of data ends at special nodes called base stations (sometimes they are also referred to as sinks). A base station links the sensor network to another network (like a gateway) to disseminate the data sensed for further processing. Base stations have enhanced capabilities over simple sensor nodes since they must do complex data processing; this justifies the fact that base stations have workstation/laptop class processors and of course enough memory, energy, storage, and computational power to perform their tasks well. Usually, the communication between base stations is initiated over high bandwidth links.

The biggest problems of sensor networks are power consumption, which is greatly affected by the communication between nodes. To solve this issue, aggregation points are introduced to the network. This reduces the total number of messages exchanged between nodes and saves some energy. Usually, aggregation points are regular nodes that receive data from neighboring nodes, perform some kind of processing, and then forward the filtered data to the next hop. Similar to aggregation points is clustering. Sensor nodes are organized into clusters, each cluster having a "cluster head" as the leader. The communication within a cluster must travel through the cluster head, which then is forwarded to a neighboring cluster head until it reaches its destination, the base station. Another method for saving energy is setting the nodes to go idle (into sleep mode) if they are not needed and wake up when required. Of course, the challenge is to find a pattern in which energy consumption is made evenly

\* Corresponding Author.

Email Address: [lmathebula@ukzn.ac.za](mailto:lmathebula@ukzn.ac.za)<https://doi.org/10.21833/AEEE.2020.04.001>

for all the nodes in the network (Bulut and Korpeoglu, 2007). Each sensor node operates with the help of batteries that have limited memory and limited computing power. Unlike other batteries the batteries of the sensor nodes are unchangeable and un-rechargeable, the available energy in the

batteries determines the lifetime of the sensor networks so the energy is the main parameter that has to be considered while designing the wireless sensor networks. Fig. 1 shows the network model of a Wireless Sensor Network.

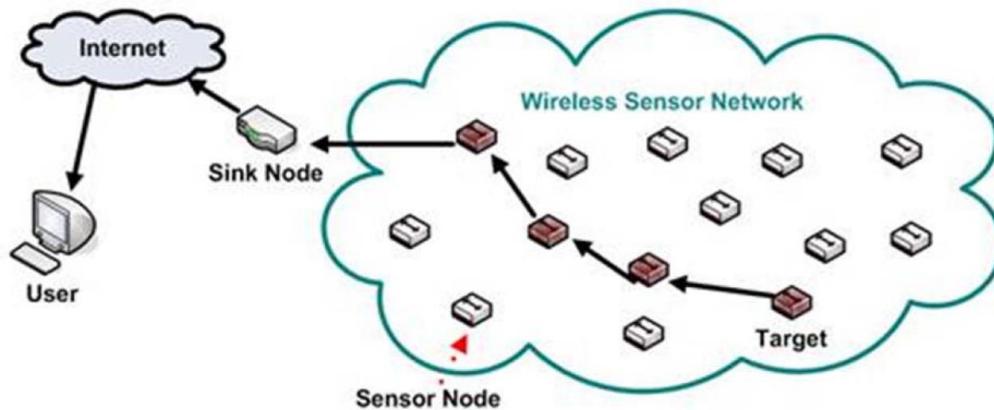


Fig. 1: Network model of wireless sensor network

In wireless communication enhancing the packet delivery through wireless links can be developed by using packet scheduling algorithms. The packet scheduling scheme is used to select which packet to be dropped or serviced and it ensures packet delivery based on priority and fairness with minimum latency, it can also be guaranteed QoS which in turn increases the transmission rate. The servicing and dropping of the packets will be based on several network parameters like bandwidth, packet arrival rate, packet deadline, and packet size. Scheduling of packets will be done in a scheduler and the scheduler will find it difficult to handle each and every packet due to the high packet rate, low bandwidth and less packet size. So the scheduler will select certain packets based on various algorithms (Jaspher and Raj, 2012).

Extensive research for scheduling the sleep-wake times of sensor nodes has been conducted (Nasser et al., 2013; Anastasi et al., 2009; Bergmann et al., 2010; Bulut and Korpeoglu, 200) and the research for packet scheduling of sensor nodes that schedules the data packets is also done (Bulut and Korpeoglu, 200; Chachra and Marefat, 2006; Liu et al., 2010; Liu et al., 2008). Most existing packet scheduling schemes of WSN are neither dynamic nor suitable for large-scale applications since these schedulers are predetermined and static and cannot be changed in response to a change in application requirements or environments (Liu et al., 2010).

The remainder of the paper is organized as follows. In section II, we discuss several existing WSN task scheduling algorithms. Section III presents the working principle of the proposed priority-based packet scheduling algorithm. Section IV evaluates the performance of the priority-based packet scheduling algorithm through simulations and compares it against existing scheduler algorithms. Finally, section V concludes the paper by defining some future research scopes.

## 2. Related works

In this section, the packet or task scheduling schemes are classified based on several factors such as Deadline, Priority, and Packet Type (Fig. 2).

First come first served (FCFS): The processing of the data in First Come First Served (FCFS) schedulers takes place in the order of their arrival times at the ready queue. In FCFS the data that are arriving late at the intermediate nodes from the nearest neighboring nodes will take less time to be processed and the data from the distant leaf nodes requires a lot of time to be delivered to the base station (BS). First Come First Served experiences long waiting times since many data packets arrive late at the intermediate nodes.

Earliest deadline first (EDF): The data packets available at the ready queue have a deadline within which they should be delivered to the base station, the data packet which has the earliest deadline is sent first. This algorithm is efficient in terms of average packet waiting time and end-to-end delay.

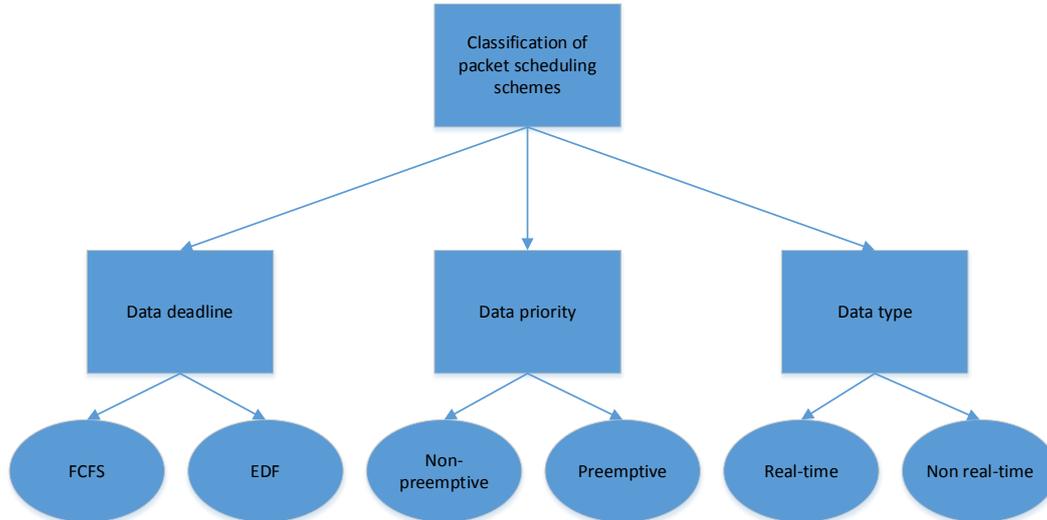
The research work done by Lu et al. (2002) proposes a real-time communication architecture that uses a priority-based scheduler. Priority is given to the data which traveled the longest distance and with the shortest deadline. This approach deduces network traffic and data processing overhead but it consumes a lot of memory and power. Mizanian et al. (2009) proposed RACE, a packet scheduling policy in which the priority queues will drop the deadline expired data packets to avoid waiting for network resources.

Priority: Packet scheduling schemes can be classified based on the priority of data packets that are sensed at different sensor nodes.

Non-preemptive: In non-preemptive priority packet scheduling, the packet after starting its execution carries out its task even if a high priority packet arrives at the ready queue, the higher priority

has to wait in the ready queue until the execution gets complete. Packet scheduling in non-preemptive

never initiates context switching.



**Fig. 2:** Classification of packet scheduling schemesDeadline: Packet scheduling schemes can be classified based on the deadline of the arrival of data packets to the base station (BS)

**Preemptive:** In preemptive priority packet scheduling, higher priority packets are processed first and it can preempt the lower priority packets by saving the context of the already running lower priority packets. The drawback of this scheduling scheme is if the deadlines of two packets are different, the shorter deadline packet would be placed into the higher-priority queue and the longer deadline packet would be placed into the lower-priority queue.

Yu et al. (2008) classify the scheduling mechanisms as cooperative or preemptive that are in Tiny OS, (Levis, 2006). Cooperative scheduling schemes are based on the earliest deadline first (EDF) and Adaptive Double Ring scheduling (ADRS) (Lin et al., 2007). The preemptive scheduling is based on Emergency Task First Rate

Monotonic (EF-RM) is a static priority scheduling, whereby the shortest- deadline job has the highest priority.

In ARDS two queues with different priorities are used in which the scheduler switches between the two queues based on the deadline, the shortest deadline packet is given the highest priority queue and the longest deadline packet is given a low priority queue.

**Packet Type:** Packet scheduling schemes can be classified based on the types of data packets, which are as follows.

**Real-time packet scheduling:** Real-time data packets are given the highest priority among all the data packets in the ready queue, so higher priority packets are processed first and then it has to be delivered to the base station (BS) with minimized end-to-end delay.

**Non-real-time packet scheduling:** The non-real-time data packets will have lower priority and so they can be delivered to the base station with little delay if there are no real-time packets in the ready

queue then the data packets use FCFS or Shortest Job First to deliver to the base station.

The packet scheduling mechanisms of Tiny OS cannot be applied to all the applications because certain data packets take a long execution time and there might be a chance that the real-time packets are discarded because the queue can be filled up quickly by the most frequent local data packets. To eliminate these drawbacks Zhao et al. (2009) proposed an improved priority-based soft real-time packet scheduling algorithm in which the priority for the packets is decided during the compilation phase and cannot be changed during the execution time.

### 3. Proposed priority based packet scheduling scheme

In the Priority Based (PB) packet scheduling scheme the sensor nodes are organized into a hierarchical structure. Nodes with the same hop distance from the BS are assumed to be located at the same hierarchical level. Data packets sensed by nodes at different levels are processed using a TDMA scheme. The nodes that are located at the lowest level and one level upper to the lowest level can be allocated timeslots 1 and 2, respectively. Each node maintains three levels of priority queues i.e. the maximum number of levels in the ready queue is classified as (i) real-time which is given priority 1 and it will go to the first classified queue, (ii) non-real-time remote data packets that are received from lower-level nodes are given priority2 and the data packets will be placed in the second-highest priority queue, and (iii) Non-real-time local data packets that are sensed at the node itself are given priority 3 and those data packets are placed in the third queue.

Non-real-time data traffic with the same priority is processed using the shortest job first (SJF) scheduler scheme since it is very efficient in terms of average task waiting time and First Come First

Served FCFS Modules: The modules of proposed priority based packet scheduling schemes are (i) Topology Formation (ii) Packets differentiation and (iii) PB Scheduling scheme Topology Formation: Construction of project design in NS2 should take place. Each node should send hello packets to its neighbor node which are in its communication range to update their topology.

Packets differentiation: Using the concept of three-level priority queues at each node, the proposed PB task scheduling scheme allows different types of data packets to be processed based on their priorities. Since real-time, and emergency data should be processed with the minimum end-to-end delay, they are processed with the highest priority and can preempt tasks with lower priorities located in the two other queues.

DMP Scheduling scheme: In the PB task scheduling approach, the source of a data packet is used to define the priority of data packets other than real-time. The priority of a non-real-time data packet will be more if it is sensed at a remote node rather than the current sending node. Moreover, when no real-time tasks are available, pr3 tasks can preempt pr2 tasks if they are in starvation for a long time. This allows the processing of different types of tasks with fairness. The memory is also dynamically allocated to three queues and the size of the highest-priority queue is usually smaller than the two other queues since pr1 real-time tasks do not occur frequently compared to non-real-time tasks. As the memory capacity of a sensor node is limited, this also balances memory usage. Moreover, tasks are mostly non-real-time and are processed in the pr2 and pr3 queues. Non-real-time tasks that a node  $x$  receives from the lower level nodes are known as non-real-time remote tasks and processed with higher priority (pr2) than the non-real-time local tasks that  $x$  senses. Thus, non-real-time remote tasks incur less average waiting time.

Scheduling data packets among several queues of a sensor node in which the data packets that are sensed at a node are scheduled among several levels in the ready queue. The possible reasons for choosing a maximum of three queues are to process (i) the Overall goal of Wireless Sensor networks is achieved by the scheduling real-time pr1 tasks, (ii) In order to achieve the minimum average task waiting time, and also to balance the end-to-end delay by placing Non-real-time pr2 tasks giving higher priority to remote data packets, (iii) Non-real-time pr3 tasks with lower priority to achieve fairness by preempting pr2 tasks if pr3 tasks wait a number of consecutive timeslots.

The queue sizes differ in our proposed method based on the application requirements. Due to switching and context storage in resource constraint sensor networks, the size of the ready queue for preemptive priority schedulers is expected to be smaller than that of the pre-emptible priority schedulers in preemptive scheduling. The idea behind this is that the highest-priority real-time/emergency tasks rarely occur. They are thus

placed in the preemptive priority task queue (pr1 queue) and can preempt the currently running tasks. Since these processes are small in number, the number of preemptions will be a few. On the other hand, non-real-time packets that arrive from the sensor nodes at the lower level are placed in the preempt priority queue (pr2 queue). The processing of these data packets can be preempted by the highest priority real-time tasks and also after a certain time period if tasks at the lower priority pr3 queue do not get processed due to the continuous arrival of higher priority data packets. Real-time packets are processed in the way as First Come First served. Each packet has an ID, which consists of two fields, namely the level field and node field if two equal priority packets arrive at the ready queue at the same time, the data packet at the lower level will have higher priority. Due to this the end-to-end delay of the lower level tasks to reach the BS is reduced. When two tasks occur at the same level, the smaller task will have higher priority and the smaller task is in terms of data size and then the larger data size packet is processed. When a node senses and receives data from lower-level nodes, it process the data and forward the data within its allocated timeslot such that the probability that the ready queue at a node becomes full and dropping of the packets is low, if any data that is in the ready queue during its allocated timeslot, that data will be processed in the next allocated timeslot.

Apart from the fact that the queue lengths are different the timeslots at each level are also not fixed and it changes according to the application requirements like data sensing period, data transmission rate, and CPU speed. The time required to transmit any real-time or emergency data will be short and will not increase at the upper levels since there is no data aggregation. The time that is remaining in a timeslot of nodes at a particular level will be used to process data packets at other queues. The degradation of system performance is low since the probability of having real-time emergency data is low, but it may improve the perceived Quality of Service (QoS) by delivering real-time data fast. Moreover, if any node at a particular level completes its task before the expiration of its allocated timeslot, it goes to sleep increasing its energy efficiency (Bergmann et al., 2010). Priority-Based (PB) packet scheduling scheme ensures a tradeoff between priority and fairness, better average task waiting time, and end-to-end delay and it also ensures a tradeoff between fairness and priority. Hence the proposed PB task scheduling mechanism is highly efficient.

#### 4. Performance evaluation

A simulation model based on NS 2 is used. In order to evaluate the performance of the dynamic multilevel priority packet scheduling scheme and to come with the existing first come first serve packet scheduling scheme, the below parameters are configured in the network simulator. We assumed

that the dimension of the scenario was 1700\*1700. The number of zones used varies from 4 to 8 and we have chosen 4 zones. Nodes are distributed over the zones and the zones can hold 50 nodes here we have assumed there are 44 nodes and the protocol used is Dynamic Source Routing (DSR) and the size of the packet is 2000 bytes. In this proposed method the area is split into four zones and the 44 nodes are distributed in the different zones with the base station at the top which is numbered 0 and all the other nodes numbered 1-39 occupy the different zones.

The performance of the proposed PB task scheduling scheme in terms of energy, end-to-end delay, and packet delivery are analyzed.

**Energy:** Energy is defined as the total energy consumed by each and every node that is occupied in the total coverage area. Fig. 3 shows the comparison between the existing and proposed packet scheduling schemes in which the proposed scheme has less energy consumed when compared with the existing schemes. It demonstrates the energy of all types of data traffic over a number of zones and levels. From these results, we find that the PB task scheduling scheme is very energy efficient and also it outperforms FCFS. This is because, in the proposed scheme, FCFS will consume a large amount of energy since it processes the data in the order of their arrival time. Thus, the average energy of the proposed PB-based packet scheduling scheme is vastly reduced.

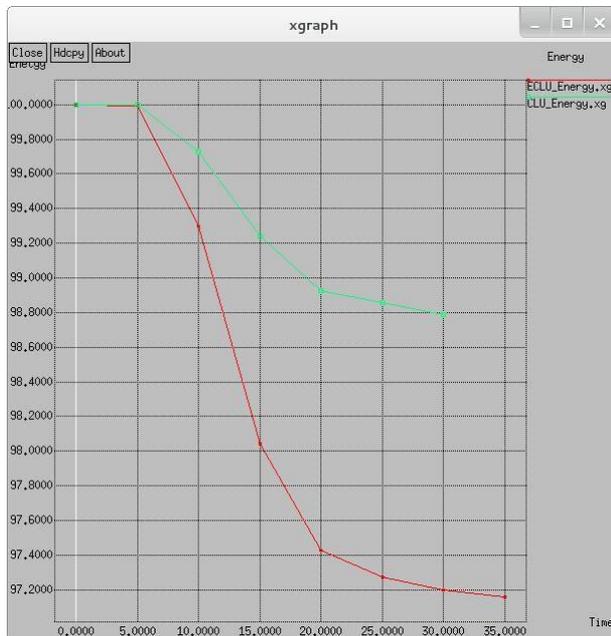


Fig. 3: Energy Vs Time plot for FCFS and PB packet scheduling schemes

**Delay:** Delay is defined as the average time taken by the packet to reach the server node from the client node. Fig. 4 shows the comparison between the existing and proposed packet scheduling schemes in which the proposed scheme has less delay when compared with the existing schemes. It demonstrates the end-to-end delay of all types of

data traffic over a number of zones and levels. From these results, we find that the PB task scheduling scheme outperforms FCFS, and scheduler in terms of end-to-end data transmission delay. This is because in the proposed scheme, the tasks that arrive from the lower level nodes are given higher priority than the tasks at the current node. Thus, the average data transmission delay is shortened.

The proposed PB scheme can be enhanced by assigning task priority based on task deadline instead of the shortest task processing time. In order to reduce processing overhead and save bandwidth, the tasks with expired deadlines from the medium are removed. The degradation of the system performance such as end-end- delay and packet delivery ratio can be done by a condition called deadlock a situation in which the resources held by the real-time task for a longer period of time such that the other tasks need to wait for an undefined period time. This deadlock situation degrades the performance of task scheduling schemes in terms of end-to-end delay. Hence, we would deal with the circular wait and preemptive conditions to prevent deadlock from occurring.

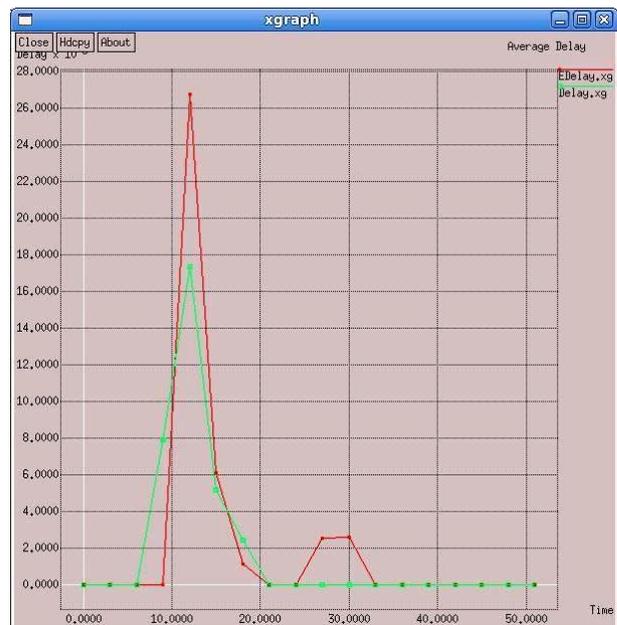


Fig. 4: Delay Vs Time plot for FCFS and PB packet scheduling schemes

**Packet delivery ratio:** Packet delivery ratio is defined as the ratio of the number of packets received to the number of packets sent by the server. Fig. 5 shows that the proposed scheme has a better packet delivery ratio and it validates our claim that the proposed PB scheme has a better packet delivery ratio when compared to the FCFS scheme.

## 5. Conclusion and future work

In this paper, the proposed Priority Based (DMP) packet scheduling scheme for Wireless Sensor Networks (WSNs) uses three-level of priority queues to schedule data packets based on their types and priorities. It ensures minimum end-to-end data

transmission for the highest priority data while exhibiting acceptable fairness toward the lowest-priority data. Experimental results show that the proposed PB packet scheduling scheme has better performance than the existing FCFS in terms of the energy end-to-end delay and packet delivery ratio.



Fig. 5: PDR Vs Time plot for FCFS and PB packet scheduling scheme

## Compliance with ethical standards

## Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## References

- Nasser N, Karim L, and Taleb T (2013). Dynamic multilevel priority packet scheduling scheme for wireless sensor network. *IEEE Transactions on Wireless Communications*, 12(4): 1448-1459. <https://doi.org/10.1109/TWC.2013.021213.111410>
- Anastasi G, Conti M, and Di Francesco M (2009). Extending the lifetime of wireless sensor networks through adaptive sleep. *IEEE Transactions on Industrial Informatics*, 5(3): 351-365. <https://doi.org/10.1109/TII.2009.2025863>
- Bergmann G, Molnár M, Gönczy L, and Cousin B (2010). Optimal period length for the CGS sensor network scheduling algorithm. In 2010 6<sup>th</sup> International Conference on Networking and Services, IEEE, Cancun, Mexico: 192-199. <https://doi.org/10.1109/ICNS.2010.33>
- Bulut E and Korpeoglu I (2007). DSSP: A dynamic sleep scheduling protocol for prolonging the lifetime of wireless sensor networks. In 21<sup>st</sup> International Conference on Advanced Information Networking and Applications Workshops, IEEE, Niagara Falls, Canada, 2: 725-730. <https://doi.org/10.1109/AINAW.2007.157>
- Chachra S and Marefat M (2006). Distributed algorithms for sleep scheduling in wireless sensor networks. In Proceedings 2006 IEEE International Conference on Robotics and Automation, IEEE, Orlando, USA: 3101-3107. <https://doi.org/10.1109/ROBOT.2006.1642173>
- Liu F, Tsui CY, and Zhang YJ (2010). Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks. *IEEE Transactions on Wireless Communications*, 9(7): 2258-2267. <https://doi.org/10.1109/TWC.2010.07.090629>
- Liu J, Gu N, and He S (2008). An energy-aware coverage based node scheduling scheme for wireless sensor networks. In the 9<sup>th</sup> International Conference for Young Computer Scientists, IEEE, Hunan, China: 462-468. <https://doi.org/10.1109/ICYCS.2008.334>
- Mizanian K, Hajisheykhi R, Baharloo M, and Jahangir AH (2009). RACE: A real-time scheduling policy and communication architecture for large-scale wireless sensor networks. In 2009 7<sup>th</sup> Annual Communication Networks and Services Research Conference, IEEE, Moncton, Canada: 458-460. <https://doi.org/10.1109/CNSR.2009.84>
- Yu M, Xiahou S, and Li X (2008, October). A survey of studying on task scheduling mechanism for TinyOS. In 2008 4<sup>th</sup> International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, Dalian, China: 1-4. <https://doi.org/10.1109/WiCom.2008.960>
- Levis PA (2006). Tinyos: An open operating system for wireless sensor networks (invited seminar). In 7<sup>th</sup> International Conference on Mobile Data Management, IEEE, Nara, Japan: 63-63. <https://doi.org/10.1109/MDM.2006.151>
- Lin K, Zhao H, Yin ZY, and Bi YG (2007). An adaptive double ring scheduling strategy based on tinyos. *Journal-Northeastern University Natural Science*, 28(7): 985.-988.
- Jaspher WK and Raj A (2012). Packet scheduling algorithms in different wireless networks a survey. *International Journal of Engineering Research and Technology*, 1(8): 1-6.
- Zhao Y, Wang Q, Wang W, Jiang D, and Liu Y (2009). Research on the priority-based soft real-time task scheduling in TinyOS. In 2009 International Conference on Information Technology and Computer Science, IEEE, Kiev, Ukraine: 1: 562-565. <https://doi.org/10.1109/ITCS.2009.122>