International Journal of Scientific Research in Computer Science and Engineering
Research Paper Vol-2, Issue-2 ISSN: 2320-7639

SCHEDULING TASK OF WIRELESS SENSOR NETWORK USING EARLIEST DEADLINE FIRST ALGORITHM

D.G.Harkut[1], M. S.Ali[2] and P.B.Lohiya[3*]

[1,2,3*] PRMCEAM, Badnera, Maharashtra, India

dg.harkut@gmail.com, softalis@hotmail.com, pnmlohiya127@gmail.com

Available online at www.isroset.org

Received: 02 Feb 2014 Revised: 18 Feb 2014 Accepted: 22 Mar 2014 Published: 30 Apr 2014

Abstract- This paper presents the implementation of an earliest deadline first (EDF) scheduling mechanism for Wireless Sensor Network (WSN). In this paper we are monitoring and scheduling the sensor values in real time by embedding wireless sensor nodes. We use the LPC2148 ARM7TMDI based microcontroller which is used as core control module and design the overall structure of the system according to the basic need of wireless sensor network. We design the hardware system structure and software system structure where we implement EDF algorithm for scheduling aperiodic task of the WSN in a real time way.

Index Terms- Wireless Sensor Network, EDF algorithm, Rate monotonic, Deadline Monotonic

1. INTRODUCTION

A WSN is a network of wireless interconnected sensors. These are usually small autonomous devices that monitor some physical conditions or other values with their sensors and use short range wireless link for communication between them and between higher level systems. The measured values typically are temperature, humidity, light intensity, etc [1] [2] [35]. Compare to other networks, a WSN has its own design and resource constraints. Resource constraints include a limited amount of energy, short communication range, low bandwidth, limited processing and storage in each node and also battery power, program memory available. Design constraints are application dependent and are based on the monitored environment. The environment plays an important role in determining the size of the network, the deployment scheme, and the network topology [15] [25]. The size of the network varies with the monitored environment. Consider we are building a network for indoor then fewer nodes are required to form a wireless network in a limited amount of space whereas outdoor environments may require more nodes to cover a larger area. When network is composed of hundreds or thousands of nodes then humans cannot access this deployment so better to prefer an ad hoc network. [9] [10].

Wireless sensor network is found in a great variety of applications, such as environmental monitoring, military surveillance and personal medical systems. These applications interface with the real world environments and the delivery of data is subject to certain timing constraints. It is a special type of embedded systems where deadline is one of the critical concerns. Applications in embedded systems are usually domain-specific. WSN applications consist of several tasks that are assumed to be independent, periodic and pre-emptible. The nature of a task is depending on the domain. In some applications the task may be periodic or aperiodic. They may be hard or soft [6].

To attain each and every task, there is need to use better scheduling algorithm so that no task miss their deadline. The most popular online scheduling algorithm was introduced by Liu and Layland in 1973. According to Earliest Deadline First (EDF) which is pre-emptive and dynamic priority driven, a ready job with the earliest deadline is executed first. Dertouzos proved that EDF is optimal among all scheduling algorithms on a uniprocessor machine [13] [19]. Consequently, if a set of jobs cannot be scheduled by EDF, then this set cannot be scheduled by any other algorithm. EDF scheduling policy is better to use so that we can attain the entire task as per their deadline value [10]. As part of this paper, we intend to implement an EDF scheduler for WSN.

The remainder of this paper is organized as follows: Section 2 introduces related works of different real time scheduling algorithms. Section 3 discusses the implementation of hardware. Section 4 elaborates the software implementation. Section 5 is all about the results with advantages of used technique. Finally, section 6 concludes this paper with some suggestions for further improvement.

2. RELATED WORK

2.1 Real Time System

The use of computers to control safety-critical real-time functions has increased rapidly over the past few years. As a consequence, real-time systems is a computer systems where the correctness of a computation is dependent on both the logical results of the computation and the time at which these results are produced — have become the focus of much study [14] [26] [39]. For example, in a brake-by-wire system, the computer that controls the braking systems of the car is a real-time system because when the brakes are applied, it is expected that the car stops within a specified amount of time. There are consequences when the results are not obtained at

before a pre-specified deadline. The severity of the consequence of not meeting the deadlines further classifies real-time systems into hard and soft real-time systems.
In hard one, application may be considered to have failed, if it does not complete function within the allocated time [11]. Airplane control systems, Nuclear reactor control systems and components of pacemakers are some of the examples of hard real-time systems. In soft real-time system must process request in reasonable time. As example, to turn ON AC or audio, in a car control system [14]. As real time systems execute critical tasks, therefore it must be designed very carefully. For that, many scheduling policy has been already designed [20] [24] [32]. Scheduling is technique for allocating tasks on processors to ensure that deadlines are to meet. In traditional real-time embedded systems, deadline is one of the critical concerns. There are basic two types of scheduling policy: offline and online. Offline scheduling involves scheduling in advance of the operation. In online scheduling, the tasks are scheduled as they come into the system and tasks are assigned to processor based on their relative priority. The algorithm keeps task priority constant called static–priority algorithms. In dynamic-priority, algorithm, priority of tasks will change with time [2] [5] [7]. Figure 2.1 shows the basic scheduling algorithm function.

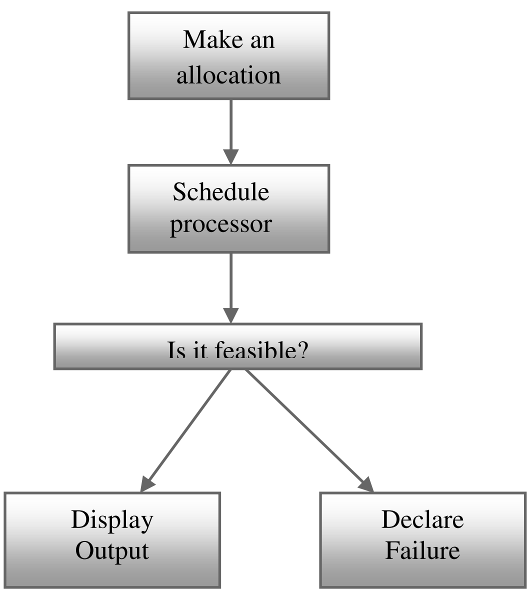

Fig 2.1: Basic Function of Scheduling Algorithm

2.2 Real Time Scheduling Algorithms

In Real-time systems scheduling algorithms are classified into two categories: Static algorithm and Dynamic algorithm. Based on execution attributes of tasks, dynamic algorithm assigns priorities at runtime [16] [19]. This algorithm allows switching of priorities between tasks. In contrast with dynamic algorithm, a static algorithm assigns priorities at design time. All assigned priorities remain fixed throughout the execution of task. Figure 2.2 gives the classification of available scheduling algorithms for real-time systems [34] [25].

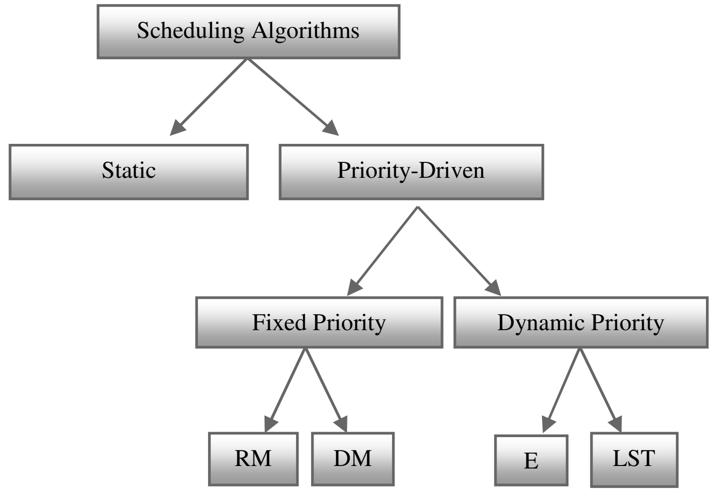

Figure 2.2: Real time scheduling algorithms

Following table 2.1 shows the comparison between the available real time algorithms.

Algorithms / Parameters	**RM**	**DM**	**LST**	**EDF**
Priority	Static	Static	Dynamic	Dynamic
CPU utilization	Average	Low	Low	High
No. of context switches	High	Less	High	Less
Optimal	No	No	No	Yes
Deadline miss chances	Average	Average	High	Less
Response time	Average	Less	Less	Average

Table 2.1: Comparison of Real time algorithms

The EDF algorithm is optimal compared with the other real-time algorithms and if task is not scheduled by EDF then all other algorithms will fail to schedule that task [17] [33] [16]. As compared to other algorithms EDF is simple to implement and gives much better utilization of processor and also EDF having dynamic nature with less number of context switches. The deadline miss chances are less in EDF algorithm.

3. HARDWARE IMPLEMENTATION

In this work we use hardware components like ARM LPC 2148 microcontroller, Radio Frequency (RF) modem, USB device, Temperature sensor, humidity sensor, moisture sensor, light sensor. Following figure gives the brief structure of hardware system [1].

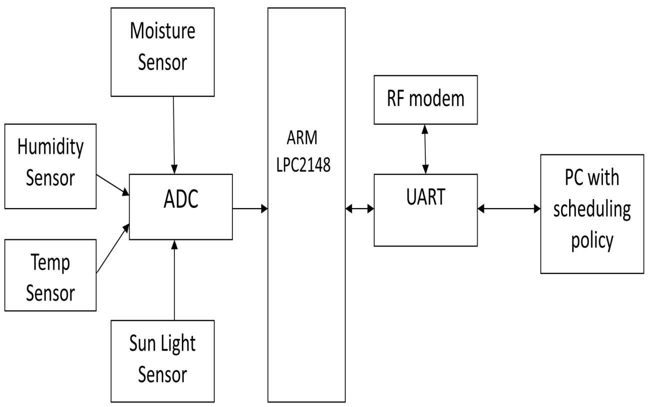

Fig 3.1: Block diagram of WSN

The above figure shows the working module of wireless sensor network. ARM LPC 2148 is the widely used Integrated Circuit (IC) from ARM-7 family. It is manufactured by Philips and it is pre-loaded with many inbuilt peripherals making it more efficient and a reliable option for the beginners as well as high end application developer. The LPC2148 microcontroller has 512KB of internal flash and 32+8K RAM. A Universal Asynchronous Receiver Transmitter (UART) is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port. Radio modem can be used for applications that need two way wireless data transmission. It transfers data wirelessly across a range of up to tens of kilo meters'. It features adjustable data rate and reliable transmission distance. The communication protocol is self controlled and completely transparent to user interface. This module works in half-duplex mode. Means it can either transmit or receive but not both at same time.

Temperature Sensor: The measurement of temperature is one of the fundamental requirements for environmental control, as well as certain chemical, electrical and mechanical controls. Temperature sensors are vital to a variety of everyday products.

Humidity Sensor: Humidity is the presence of water in air. Humidity sensing is very important, especially in the control systems for industrial processes and human comfort. Controlling or monitoring humidity is of paramount importance in many industrial & domestic applications.

Moisture Sensor: Moisture refers to the presence of a liquid, especially water, often in trace amounts. Small amounts of water may be found, for example, in the air (humidity), in foods, and in various commercial products. Moisture also refers to the amount of water vapour present in the air.

Light Sensor: Visible light is electromagnetic radiation that is visible to the human eye, and is responsible for the sense of sight. Applications include smoke detection, automatic lighting control, and batch counting and burglar alarm systems.

4. SOFTWARE IMPLEMENTATION

4.1 EDF Scheduling Algorithm

EDF scheduler still remains an efficient option for online scheduling in these new generation computing systems. EDF is an optimal scheduling algorithm on pre-emptive uniprocessor, in the following sense: if a collection of independent jobs, each characterized by an arrival time, an execution requirement and a deadline, can be scheduled (by any algorithm) in a way that ensures all the jobs complete by their deadline, the EDF will schedule this collection of jobs so they all complete by their deadline. While developing EDF algorithm some assumptions are their i.e. all the task are aperiodic task so that they can arrive at any instant of time. We implement this algorithm for uniprocessor where deadline is one of the important parameter because we assume those aperiodic tasks are hard real time.

Following are the executions steps of EDF algorithm:

Step 1: Get the arrival time of task.

Step 2: Generate random number for deadline time and execution time (Burst Time).

Step 3: Then populate array with task parameters i.e., deadline time, burst time and arrival time.

Step 4: Schedule task as per their arrival time and after each period of time check whether any task arrives having least deadline value.

Step 5: If yes, then set the priority of task by calling sorting algorithm.

Step 6: Pre-empt the running task and execute task which is having least deadline value means schedule task having higher priority.

4.1.1 EDF with Array

This work considered a set of aperiodic task which arrives at any instant. Scheduler doesn't have any prior knowledge about the arrival time of task. As task arrives, scheduler schedules that task. After each interval scheduler need to check whether any new task arrives which is having less deadline as compare to previously executing task. If yes then pre-empt the running task and schedule the newly arrived task as per the EDF scheduling policy. In real time systems, the priority of task changes dynamically so after regular interval of time period scheduler sorts the deadline array and then assign priority to the task. Above situation arises when number of task already arrived at the same instance of time. So for this scheduler have to sort the deadline array with the help of sorting algorithm. Here we used three sorting algorithm listed below:

1. Quick Sort
2. Merge Sort
3. Bubble Sort

With the help of this algorithm we can sort the deadline array and then assign priority to the task as per the EDF policy i.e. task having less deadline value will get highest priority. After every time period scheduler has to keep track of deadline values and check whether any task changes its deadline value or any new task arrives. If yes then again and again call any one of the sorting algorithm and assign priorities to the task. This process of calling sorting algorithm is time consuming.

There are some disadvantages of EDF with Array Method which is given below:

1. The memory which is allocated to array cannot be increased.
2. The elements of array are stored in consecutive memory locations. So insertions and deletions are very difficult and time consuming in large number of arrays.
3. If deadline of task is get change or any new task arrives with some new deadline value then each time we have to re arrange the arrays, so again and again there is need to call sorting algorithm.

To overcome these disadvantages we implement EDF algorithm with link list.

4.1.2 EDF with Link List

In this method the deadline value is get stored in to link list where each data is randomly located. In link list we are calling sorting algorithm to arrange the deadline values. Here we used only Quick sort algorithm which takes less time to sort data [29]. After sorting we assign priorities and then schedule the task. Now consider same case in which some tasks arrive at the same instant of time period or we can say tasks are released at same time. With the help of quick sort algorithm scheduler arrange the link list and assign priority to the tasks as per their deadline and schedule the tasks. After some time period any task or few tasks can change their deadline value then we can set the flag with respect to that task so that scheduler can easily identify which task change the deadline value. According to that scheduler can insert that change value at proper position with the help of Binary search algorithm which is used here for insertion.

With plain link list we cannot perform binary search directly. Since random access on link list is O(n). Main issue, besides that we have no constant time access to the linked list elements, is that we have no information about the length of the list. In simple we have no way to cut the list in two halves. But if we have at least a bound on the link list length, problem is solvable. So, assuming that the link list is sorted and we know its length (or at least maximum length) then it's possible to implement binary search on link list.

Assume the maximum size of link list and sort the link list only once i.e., start of the scheduling and then after some time if any task newly arrived or any task changes deadline so rather than calling sorting algorithm a better approach is to call Binary search algorithm for arranging a list with minimum deadline values of task. Binary search algorithm minimizes the comparison required to arrange the link list and also time required to sort is get minimized.

5. RESULTS

5.1 EDF algorithm for aperiodic task

The experimentation result is considered after simulating more number of task sets with variation in arrival time, deadline value and execution time. As we have already defined the different cases and then generate results. Finally after scheduling set of tasks we got the graph given below. As all tasks are having aperiodic nature with hard deadline so main aim was to schedule task before deadline with minimum value of context switches, pre-emptions, miss rate and also minimized the response time of tasks. After scheduling number of task samples, we observed that in aperiodic system EDF algorithm performs well because it reduces the context-switches, pre-emptions with the miss rate.

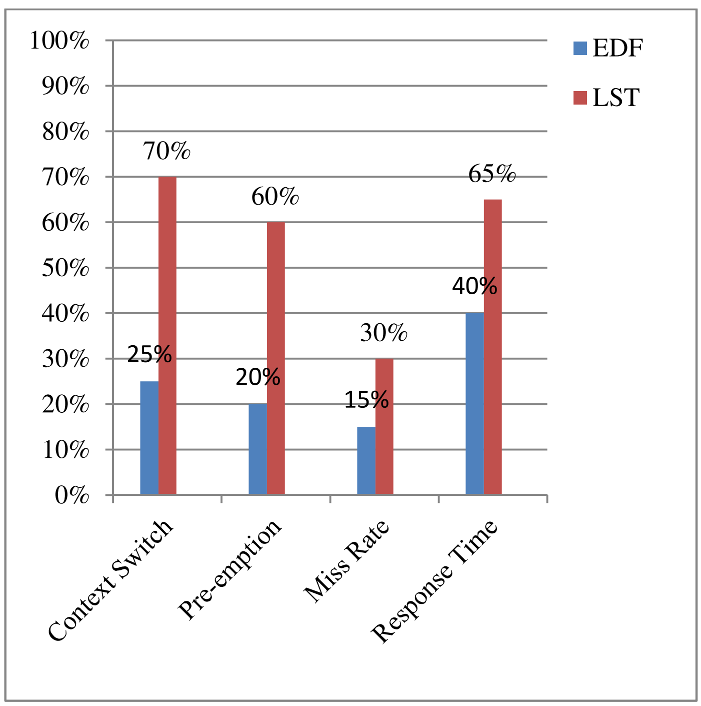

Fig 5.1: Graph shows the performance of EDF algorithm

5.2 Sorting Mechanism

In this work we have used three sorting algorithms and compare their results with respect to counter value and time required to sort. Following graph shows the comparisons with running time required and number of memory references. We found that quick sort algorithm is very efficient as compared to other sorting algorithm and also we use this quick sort with binary search so we get better results within minimum time.

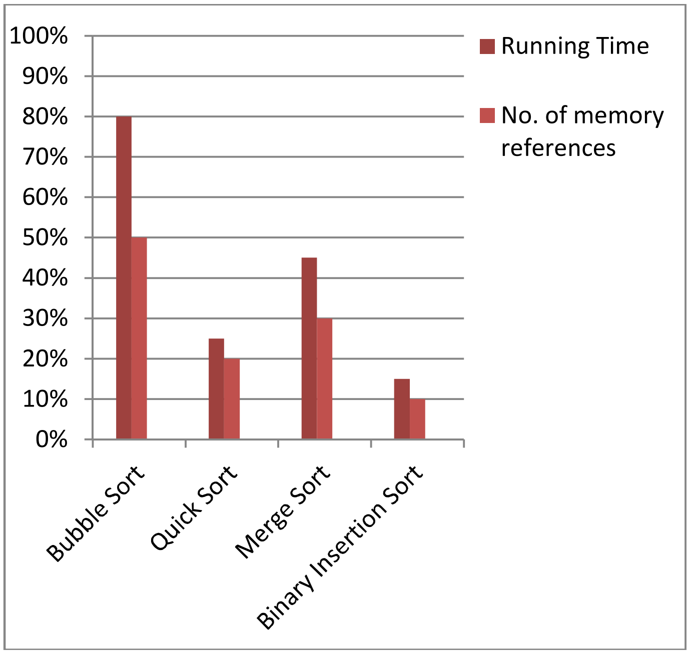

Fig 5.2: Comparison of sorting algorithm w.r.t. Running time and Counter

6. CONCLUSION AND FUTURE WORK

Hence with the help of EDF algorithm we attain the entire aperiodic task and we are scheduling those tasks as per their deadline value. EDF perform well for scheduling aperiodic task sets because it minimized the value of context switches, pre-emptions, miss rate and at some level the response time also get minimized. We got the result which proves that quick sort algorithm is better to use because it requires less time and comparisons also get minimized. With the help of binary search algorithm there is no need to sort whole link list, within minimum time we can arrange the link list by inserting change value at their proper position only if we know the length of link list. In developing the algorithm, some assumptions were made. Future work could relax these assumptions and more simulations performed to observe the performance of the algorithm. Also practical implementation issues could be a subject of future work. Further study is required to improve performance.

REFERENCES

[1] Adi Mallikarjuna Reddy, V AVU Phani Kumar, D Janakiram, and G Ashok Kumar, 2007. *Operating Systems for Wireless Sensor Networks: A Survey Technical Report*.

[2] Andre´ Rodrigues, Tiago Camilo, Jorge SaSilva, Fernando Boavida, 2012. *Diagnostic Tools for Wireless Sensor Networks: A Comparative Survey,* Springer Science+Business Media, pp. 408-452.

[3] Benitha Christinal.J, 2013. *A Survey on Priority based Packet Scheduling in Wireless Sensor Networks,* International Journal of Scientific Research in Computer Science (IJSRCS), vol. 1, Issue 4, pp. 18-21.

[4] Bhatti, S.; Carlson, J.; Dai, H.; Deng, J.; Rose, J.; Sheth, A.; Shucker, B.; Gruenwald, C.; Torgerson, H.R., 2005. *Mantis OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms*. Mobile. Network, pp. 563-579.

[5] Cao, Q.; Abdelzaher, T.; Stankovic, J.; He, T., 2008. *The LiteOS Operating System: Towards Unix Like Abstraction for Wireless Sensor Networks*. In Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), St. Louis, MO, USA, pp. 22–24.

[6] Chi-Tsun Cheng, Chi K. Tse, and Francis C. M. Lau, 2010. *An Energy-Aware Scheduling Scheme for Wireless Sensor Networks*, IEEE transaction on vehicular technology, vol. 59, no. 7.

[7] Dunkels, A.; Gronvall, B.; Voigt, T., 2004. *Contiki a Lightweight and Flexible Operating System for Tiny Networked Sensors*. In Proceedings of the 9th Annual IEEE International Conference on Local Computer Networks, Washington, DC, USA; pp. 455-462.

[8] Edwin Prem Kumar Gilbert, Baskaran Kaliaperumal, Elijah Blessing Rajsingh, 2012. *Research Issues in Wireless Sensor Network Applications: A Survey,* International Journal of Information and Electronics Engineering, vol. 2, no. 5, pp. 702-706.

[9] Eswaran, A.; Rowe, A.; Rajkumar, R., 2005. *Nano-RK: An Energy-Aware Resource-Centric RTOS for Sensor Networks*. In Proceedings of the 26th IEEE Real-Time Systems Symposium, Miami, FL, USA, pp. 5–8.

[10] Fengxiang Zhang, Alan Burns, 2009. *Schedulability Analysis for Real-Time Systems with EDF Scheduling* IEEE transactions on computers, vol. 58,

[11] Frederic Ridouard, Pascal Richard, Francis Cottet, 2004. *Negative results for scheduling independent hard real-time tasks with self-suspensions,* Proceedings of the 25th IEEE International Real-Time Systems Symposium, pp. 1-10.

[12] Giorgio C. Buttazzo, 2005. *Rate Monotonic vs. EDF: Judgment Day,* Real-Time Systems, Springer Science + Business Media, Inc. Manufactured in the Netherlands, pp. 5-26.

[13] Haiying Zhou, Feng Wu, Kun-mean Hou, 2008. *An Event-driven Multi-threading Real-time Operating System*

dedicated to Wireless Sensor Networks. The 2008 International Conference on Embedded Software and Systems (ICESS2008) IEEE,

[14] Jane W.S. Liu, 2001. *Real-Time Systems*, Pearson Education, India, pp. 121 & 26.

[15] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal, 2008. *Wireless sensor network survey*, Elsevier B.V. Computer Networks, pp. 2292-2230.

[16] Jinkyu Lee, Kang G. Shin, 2012. *Preempt a Job or Not in EDF Scheduling of Uniprocessor Systems,* IEEE Transaction on Computers, pp. 1-10.

[17] Karsten Albers, Frank Slomka, 2005. *Efficient Feasibility Analysis for Real-Time Systems with EDF scheduling,* Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, IEEE.

[18] Kathleen Baynes, Chris Collins, Eric Fiterman, Brinda Ganesh, Paul Kohout, Christine Smit, Tiebing Zhang, and Bruce Jacob,2003. *The Performance and Energy Consumption of Embedded Real-Time Operating Systems*,IEEE transactions on computers, vol. 52, no. 11, pp. 1454-1469.

[19] Kayvan Atefi, Mohammad Sadeghi, Arash Atefi, 2011. *Real-Time Scheduling Strategy for Wireless Sensor Networks O.S***,** International Journal of Distributed and Parallel Systems (IJDPS) vol.2, no.6, pp. 63-78.

[20] Leo Ordinez, David Donari, Rodrigo Santos, Javier Orozco, 2008. *An Application-Based Real-Time Scheduler for Wireless Sensor Networks,* XXXIV Conferencia Latinoamericana de Informática. pp. 1239-1248.

[21] Levis, P.,Madden, S. Polastre, J., Szewczyk, R.,Whitehouse, K.Woo, A. Gay, D. Hill, J. Welsh, M.Brewer, E. Culler, D.,2011. *Tinyos: An Operating System for Sensor Networks*, pp. 115-147.

[22] Liang Dai, Yilin Chang, Zhong Shen, 2011. *An Optimal Task Scheduling Algorithm in Wireless Sensor Networks,* Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 vol.6, no. 1, pp. 101-112.

[23] LiteOS. *LiteOS* [online], 2011[cit.2012-05-28]. Available from: http://www.liteos.net

[24] M. Sirisha, S.Swetha, 2013. *A Survey On WSN OS Using Real-Time Scheduling Strategy,* International Journal of Technology Enhancement and Emerging Engineering Research, vol. 1, no.5, pp. 13-19.

[25] M.Kaladevi, Dr.S.Sathiyabama, 2010. *A Comparative Study of Scheduling Algorithms for Real Time Task*, International Journal of Advances in Science and Technology, vol. 1, no. 4, pp. 8-14.

[26] Maryline Chetto, 2013. *Optimal Scheduling for Real-Time Jobs in Energy Harvesting Computing Systems,* IEEE Transactions on Emerging Topics in Computing, pp. 1-13.

[27] Myunggwon Hwang, Dongjin Choi, Pankoo Kim, 2011. *Least Slack Time Rate First: an Efficient Scheduling Algorithm for Pervasive Computing Environment,* Journal of Universal Computer Science, vol. 17, no. 6, pp. 912-925.

[28] Octav Chipara, Chenyang Lu, Gruia-Catalin Roman, 2013. *Real-Time Query Scheduling for Wireless Sensor Networks*, IEEE Transactions on Computers, vol. 62, no. 9, pp. 1850-1865.

[29] Pankaj Sareen, 2013. *Comparison of Sorting Algorithms (On the Basis of Average Case),* International Journal of Advanced Research in Computer Science and Software Engineering,

[30] Rowe, A.; Lakshmanan, K.; Yhu, H.; Rajkumar, R., 2008. *Rate-Harmonized Scheduling for Saving Energy*, In Proceedings of the 29th IEEE Real-Time Systems Symposium, Barcelona, Spain,

[31] Rym Chéour, Sébastien Bilavarn, Mohamed Abid, 2011. *Exploitation of the EDF Scheduling in the Wireless Sensors Networks,* International Journal of Measurement Technologies and Instrumentation Engineering, 1(2), pp.14-27.

[32] Sangho Yi, Hong Min, Junyoung Heo, Boncheol Gu, Yookun Cho, Jiman Hong, Jinwon Kim, Kwangyong Lee, and Seungmin Park, 2006. *Performance Analysis of Task Schedulers in Operating Systems for Wireless Sensor Networks* M. Gavrilova et al., LNCS 3983, Springer-Verlag Berlin Heidelberg, pp. 499–508.

[33] Sumit Kumar, Siddhartha Chauhan, 2011. *A Survey on Scheduling Algorithms for Wireless Sensor Networks,* International Journal of Computer Applications (0975 – 8887), vol. 20, no.5, pp. 7-13.

[34] Swati Pandit, Rajashree Shedge, 2013. *Survey of Real Time Scheduling Algorithms*, OSR Journal of Computer Engineering, vol. 13, Issue 2, pp. 44-51.

[35] Tarek M. Salem, Sherine M., Abd E-kader, 2013.*Mapping Wireless Sensor Network Applications Requirements to Existing Operating Systems,* IJCSI International Journal of Computer Science vol. 10, Issue 5, no 1, pp. 258-267.

[36] *TinyOS Network Working Group*; Tutorials, Network Protocols (accessed on 17 April 2011).

[37] TinyOS [EB/OL], http:// www.tinyos.net, 2007-6-1. vol 3, Issue 3, pp. 522-532.

[38] Wei Dong, Chun Chen, Xue Liu, Yunhao Liu, Jiajun Bu, Kougen Zheng, 2011. *SenSpire OS: A Predictable, Flexible, and Efficient Operating System for Wireless Sensor Networks*, IEEE Transactions on Computers, vol. 60, no. 12, pp. 1788-1800.

[39] Yingwei Yao, Georgios B. Giannakis, 2005. *Energy-Efficient Scheduling for Wireless Sensor Networks,* IEEE Transactions on Communications, vol. 53, no. 8, pp. 1333-1342.

[40] Zhao Zhi bin, Gao Fuxiang, 2009. *Study on Preemptive Real-Time Scheduling Strategy for Wireless Sensor Networks*, Journal of Information Processing Systems, vol.5, no.3. pp. 135-144.