I J C T A, 9(7), 2016, pp. 3307-3314 ISSN: 0974-5572

ITF Scheduling Scheme for Wireless Sensor Networks

R. Priscilla Joy* and J. Dheeba**

ABSTRACT

Most of the scheduling algorithms were concentrated on scheduling of the packets. In the packet scheduling scheme Dynamic Multilevel Priority (DMP), the packets were scheduled in three priority queue priority level 1(pr1), Priority level 2(pr2), Priority level 3(pr3) respectively. In this scheme, pr1 is scheduled using(First Come First Serve) FCFS, pr2 and pr3contains non-real time task with remote and local task respectively were scheduled using preemptive scheduling. This result in the occurrence of starvation is because of the FCFS queue. The remedy for this problem is done by proposing a novel Important Task First(ITF) Scheduling Scheme. According to the proposed scheme, pr1 containing real time data is scheduled using Important Task First (ITF). Pr2containing non-real task with remote data were scheduled using (Earliest Deadline First) EDF and pr3 containing non-real time task with local data were scheduled using preemptive FCFS. Thus in the proposed scheme the starvation which occurred in DMP gets removedand there by the performance of the system gets improved.

Key Words: task; scheduling; priority; deadlock; overhead; delay

1. INTRODUCTION

Scheduling is the process of specific sequence of tasks and determining the timing in order to carry out the planned operations. [1] Scheduling as a means to control traffic and improve performance in Wireless Sensor Networks.

Scheduling is the process in which the packets which are scheduled by using some algorithms will be scheduled accordingly one after the other.

Scheduling is carried out in many ways using the scheduling algorithms. Based on the scheduling algorithms the scheduling is carried out thus in turn will increase the life time. Because the time taken will be predefined which will minimize the time taken for each and every process to proceed.

Scheduling will arrange the process and it will put the task in the ready queue. If the resource is shared by one or more process and again another process is waiting to acquire that resource then there occurs the concept of starvation or deadlock. In general, there were four conditions for the deadlock. They are mutual exclusion, hold and wait, no preemption, circular wait.

The mutual exclusion condition is the resource which is involved should not be sharable otherwise the resource should be avoided from using it. In hold and wait condition, the resources which are used by the process will hold it even if it is waiting for another resource then the deadlock will be avoided.

In no preemption condition, the preemption condition should not be applied to the running process i.e, the process which is waiting for the resource should not take the process from the running process to complete their work. In circular wait, one process will wait for the other resource which is held by other process which in turn will hold the other resource which is needed by other process. If this condition exists then it states that there is a possibility of deadlock.[1]

* Research Scholar, Department of CSE Noorul Islam University Nagercoil, India, *Email: rpj.joy@gmil.com*

** Associate Professor Department of CSE College of Engineering Perumon Kollam, India, *Email: dheeba.jacob@gmail.com*

There are also many algorithms which are based on scheduling. They are FCFS (First Come First Serve) in which the process which is arriving first will be scheduled first, SJF (Shortest Job First) which will schedule the job which is having the shortest processing time first. EDF (Earliest Deadline First) in which the task having the deadline which is earlier than the other will be scheduled first next the other. Priority scheduling will schedule the prioritized packets first. In preemptive scheduling if a priority packet arrives when low priority packet is executing then it will be preempted scheduling the priority packet first next again continue with the current packet.[2]

2. RELATED WORKS

2.1. Scheduling Algorithms

2.1.1. First Come First Serve

In the first come first serve algorithm the task which will be arriving first will be scheduled first later on the next. It includes the preemptive FCFS also. In this type of scheduling the starvation will happen because the task with higher processing time may arrive earlier than small task then it has to wait for a long time to complete its scheduling process. This will in turn cause starvation. [2]

2.1.2. Shortest Job First

Shortest-Job-First (SJF) is a non-preemptive scheduling algorithm in which the process with small run time completion will be scheduled first moving on to the next higher level of run time completion. [3]

2.1.3. Shortest-Remaining-Time (SRT) Scheduling

The SRT is the preemptive SJF scheduling algorithm in which the process with shortest run time completion will be scheduled first next to the higher level but it will preempt if any important task arrives. This algorithm is better suitable for the time sharing environment.[3]

2.1.4. Round Robin Scheduling

Round robin scheduling is easy to implement and it contains time slicing[4] and each process in the queue will be given equal importance to process in a circular manner. So that it is free from starvation. [5]

2.1.5. EDF Scheduling

EDF is Earliest Deadline First scheduling in which the task with earliest deadline will be scheduled first later on the other. In this scheduling if the process which is scheduled does not meet its deadline then that process will be rejected.

Preemption in EDF scheduling is also available. Based on this scheduling if any important task arrives when other task is in execution, then that task will be preempted and the task which is prioritized will be scheduled first later on the other task.

3. PROPOSED ITF SCHEDULING SCHEME

In Dynamic Multilevel Priority (DMP) [6] the packets were scheduled in three priority queue pr1, pr2, pr3 respectively. In this scheme, pr1 is scheduled using ITF, pr2comprises non-real time task with remote data which will be scheduled by EDF scheduling and pr3 holds non-real time task with remote data which will be scheduled by preemptive FCFS scheduling. The remedy for this problematicis ended by put forward by a novel ITF scheduling scheme (ITF), the starvation occurred in DMP scheme is eliminated by including preemptive scheduling.

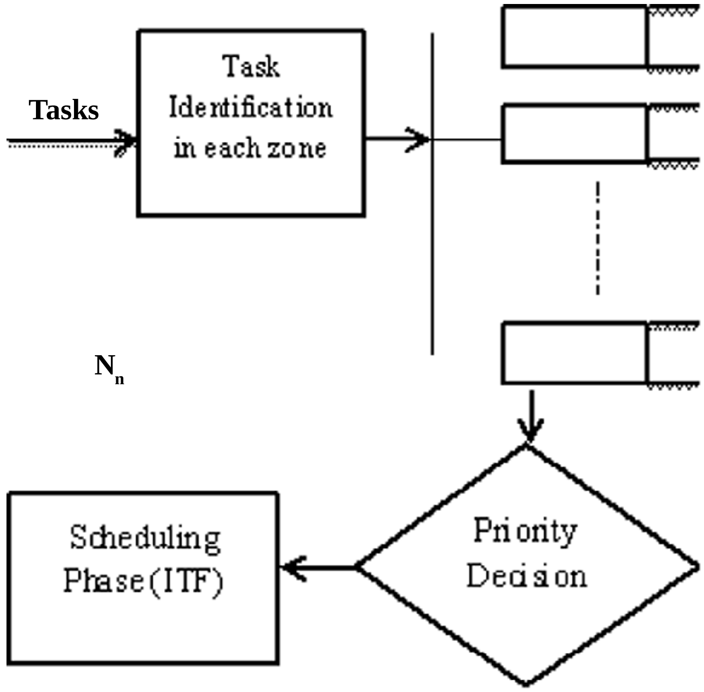


Figure 1: Overall Architecture

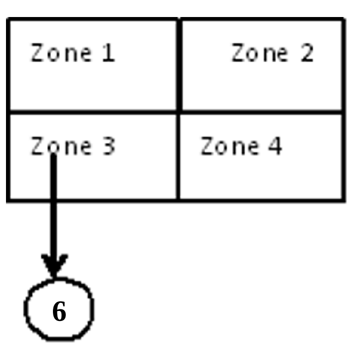
The overall architecture of the process in shown in Fig. 1, where the tasks which are entering are first subjected to the task identification module. Here the type of the task was identified and put into the node level according to the task. Then from the queue level it is directed to the priority decision module. Here according to the priorities the tasks were put into the priority level Pr1, Pr2 and Pr3accordingly. After which the tasks were scheduled in the scheduling phase based on the importance of the task. The below figure will picture about the overall architecture of the process.

4. ITF SHEDULING SCHEME

In Dynamic Multilevel Priority (DMP)the packets were scheduled in three priority queue pr1, pr2, pr3 respectively. In this scheme, pr1 is scheduled using ITF; pr2covers non-real time task with remote taskis scheduled by EDF and pr3 comprises is non-real time task with remote taskis scheduled by preemptive FCFS.

The remedy for this problem is done by recommendinganinnovativeImportant Task First (ITF) task scheduling scheme, the starvation occurred in DMP scheme is eliminated by including preemptive scheduling.

The zone based protocol is used. Based on this the task were divided into zones and from each zone a zone head will be chosen. In that zone each and every task will be put into the queue based on the real time and non-real time task.



(Task)
Pr1real-time task-(ITF)
Pr2Non realtime remote task - (Preemptive EDF)
Pr3Non realtime local task –(Preemptive FCFS)

Figure 2: ITF Scheduling Scheme

Theperformance of the proposed ITF task scheduling scheme in terms of end-to-end delay of different types of traffic at the ready queues of active nodes is analyzed.

4.1. Pr 1 Queue Task (Real-time)

Transmission time or delay for the real time task is equal to

$$T = \frac{task\ pr\ 1}{st} \tag{1}$$

The propagation time or intervalis formulated as

$$Proc(t) = \frac{d}{sp} \tag{2}$$

The end-to-end delay for a real time task satisfies the following inequality.

$$delay\ pr1 \geq lk \times \left(\frac{task\ pr1}{st} + pr1\ Proc(t) \right) + \frac{d}{sp} + \left(lk \times overhead \right) \tag{3}$$

Where task pr1 = data size of the real-time task, *st* = data transmission speed and *d* = the distance from the source node to BS.d = $\sum_{i=1}^{lk} d_i$, *sp* = The propagation speed completed the wireless medium, pr1 proc(t).

The end-to-end delay for a real-time task *t*1 is

$$delay\ t1 \geq \sum_{i=1}^{npr1} (delay\ pr1)i \tag{4}$$

4.2. Pr 2 Queue Task (Non-real time remote):

In non-real time task preemption will be taking place when any important task arrives in the real time.

The delay or the transmission time therefore computed as

$$delay\ pr2 = task\ \frac{pr2}{st} \tag{5}$$

Thus, the total end-to-end delay is given as follows:

$$delay\ t2 = lk \times \left(\frac{task\ pr1}{st} + \frac{task\ pr2}{st} + pr1\ proc(t) + pr2\ proc(2) \right) + \frac{d}{sp} + \left(lk \times t\ overhead \right) \tag{6}$$

4.3. Pr 3 Queue Task (Non-real time local):

Here the preemption will take place in both pr1 and pr2 task. So after completion of those tasks it will be processing the pr3 tasks.

Thus, the end-to-end delay for pr3 tasks will be

$$delay\ t3 = \times t\left(k\right) + lk \times \left(\frac{task\ pr3}{st} + pr3\ proc(t) \right) + \frac{d}{sp} + \left(lk \times t\ overhead \right) \tag{7}$$

5. ITF SHEDULING ALGORITHM

The ITF(Important Task First) [7] algorithm is explained in the form as below:

do

begin

while *Type* of *task k, I is real – time* **then**

place *task k,i* into *pr*1 queue

else if *Type* of *task k, I is* non *real–time & remote* **then**

place *task k, i* into *pr*2 queue

else

place *task k,i* into *pr*3 queue

end if

Consider, the task deadline of each task in *pr*1as t_{11}…..t_{1n}

Processing time of each task as t_{11}'…..t_{1n}'

if t_{11}'> *t*11**then**

*pr*1 tasks are processed as ITF

else

drop the task

end if

Assume, the task deadline of each task in *pr*2 ast_{21}…..t_{2n}

Processing time of each task as t_{21}'…..t_{2n}'

if t_{21}'> t_{21}**then**

All *pr*2 tasks are processed as EDF

else

drop the task

end if

if*pr*1 task arrives

then

*pr*2 tasks are preempted and continue to process *pr*1

Assume *t*(*pr*1) is the total processing time of *pr*1

$t_n$as maximum task deadline of *pr*1 task

ITF as important task first

if *t*(*pr*1) >t_n preempt & & *pr*1 ≠ *ITF*

*pr*1tasks and proceed *pr2* tasks

thengoto pr1 tasks after completion of *pr*2 tasks

end if

Assume, the task deadline of each task in *pr*3 as t_{31}…..t_{3n}

Processing time of each task as t_{31}'…..t_{3n}'

ift_{31}' > t_{31}then

All *pr*3 tasks are processed as preemptive FCFS

else

drop the task

end if

if *pr*1 task arrives

then

*pr*3 tasks are preempted and continue to process *pr*1

if *t*(*pr*1) >t_n preempt & & *pr*1 ≠ *ITF*

*pr*1tasks and proceed *pr*3 tasks

then go to pr1 tasks after completion of *pr*3 tasks

end if

end if

end while

6. SIMULATION RESULTS

The projected scheme can be compared with the results of two schemes: DMP and ITF. The tasks [8] are scheduled according to the priority[9]-[12]. As a result, the overhead and the delay get reduced.

In Fig. 3 the impact on overhead is taken into consideration and it is seen that the ITF is leading than the DMP. In Fig.4 the impact on delay is performed and it is seen that ITF is improved high than DMP. Thus the delay which occurred in DMP gets minimized when the ITF scheme is applied. [13] After the real time task is completed it will go back to the pr2 and pr3 task unless the starvation or deadlock appears.

The zone based protocol is used. Based on this the task were divided into zones and from each zone a zone head will be chosen. In that zone each and every task will be put into the queue based on the real time and non-real time task.

Figure 3 and 4 illustrates the • -values for the end-to-end delay in DMP and other schedulers are 0.0137. But the DMP and ITF • -values is 0.0091. Thus it validates 95% confidence level.

Table 1
Simulation Parameters and their Values

Parameters	*Values*
Size of the networks	100m × 100m
No. of nodes	Maximum 200
No. of zones	4 to 12
Position of base station	55m × 101m
Transmission energy consumptions	50 *nJoule/bit*
Free space or air energy consumption	0.01 $nJoule/bit/m^2$
Initial node energy	2 *Joule*
Transmission speed	250 *Kbps*
Propogation speed	198×10^6 *meter/sec*

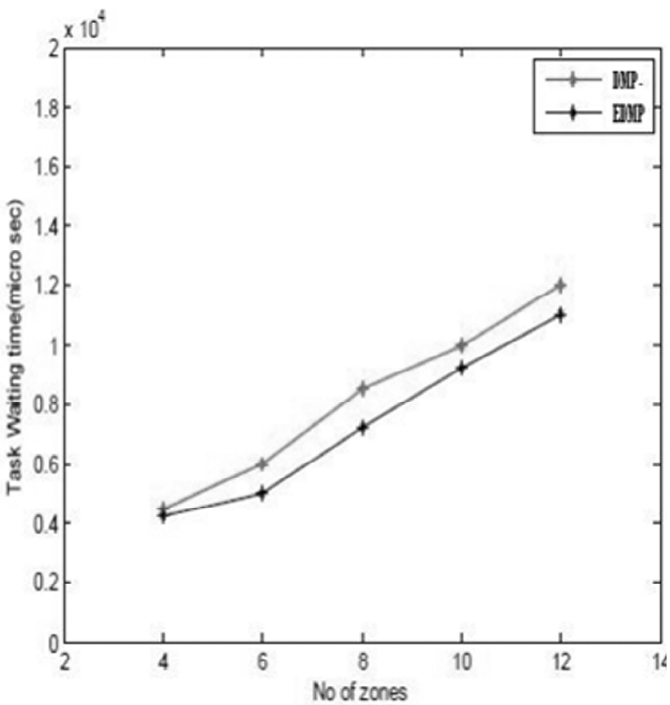


Figure 3: Number of zones compared with end-to-end delay of real-ime task.

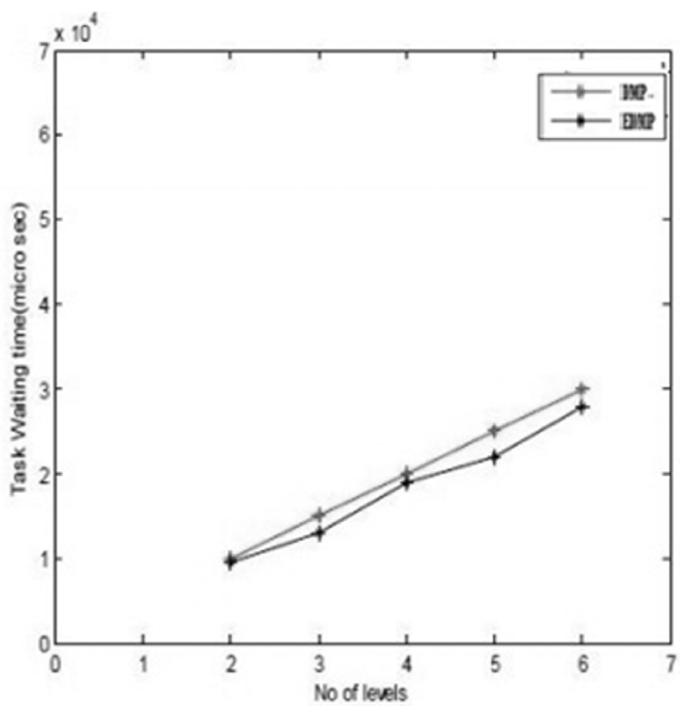


Figure 4: End-to-end delay of real-time task over a number of levels

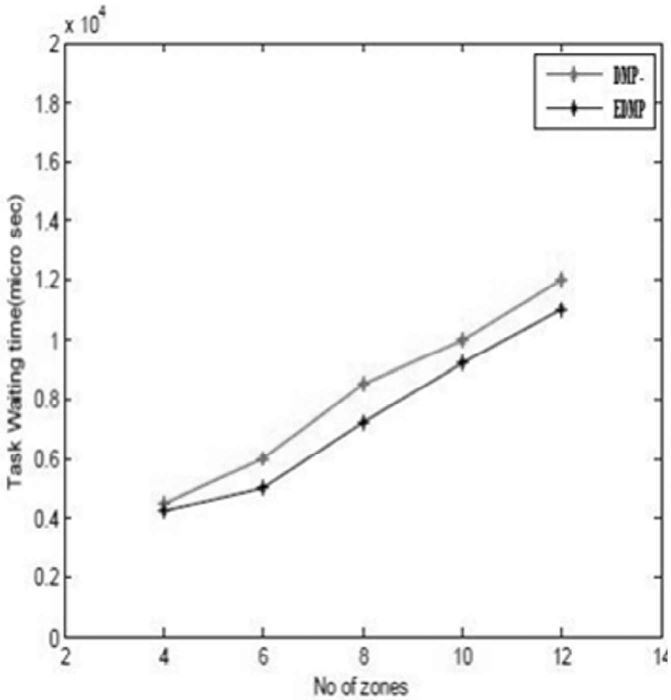


Figure 5: End-to-end delay of all categories of task over a number of zones

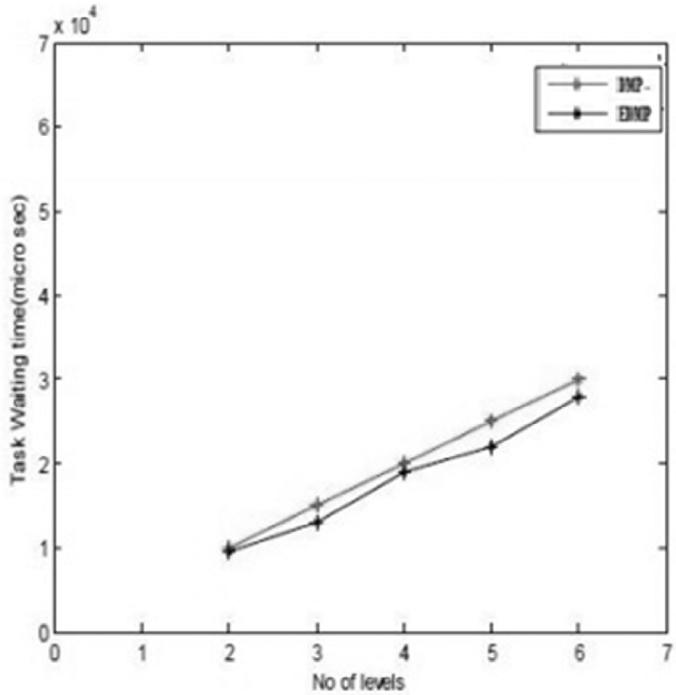


Figure 6: End-to-end delay of all types of task over a number of levels

Figure 4 and 5 are end-to-end delay of real-time tasks over number of zones and number of levels. In both the cases ITF scheme performs well than DMP.

Figure 5 and 6 illustrates the • -values for the task in DMP and other schedulers are 0.01156. But the DMP and ITF • -values is 0.00034. Thus it validates 95% confidence level. The table below illustrates the simulation parameters and their values.

Thus in EDMP the end to end delay of the task decreased over the DMP. Also the waiting time decreases. Thus the overall performance of the system gets improved.

7. CONCLUSION AND FUTURE ENHANCEMENT

In ITF, the real time task is subjected to ITF (Important Task First). After completion of the tasks which are before the task deadline the pr2 tasks were scheduled using Earliest Deadline First. After the completion of the task the pr3 tasks were scheduled Serve.

If pr1 task arrives when processing both pr2 and processed using preemptive First Come Firstand pr3, the tasks will be preempted and pr1 task will be processed using circular wait. In all pr1, pr2 and pr3 tasks if the processing time exceeds the task deadline then the task will be dropped. In future the dropping of task which is based on the task deadline can be minimized by upgrading the task. So that the overall performance can be increased.

REFERENCES

[1] Remzi H. Arpaci-Dusseau; Andrea C. Arpaci- Dusseau “Chapter 7: Scheduling: Introduction, Section 7.6: a new metric: response time. Operating systems: three easy pieces (PDF). pp. 6. retrieved February 2, 2015.

[2] Abraham Silberschatz, Peter Baer galvin and Greg Gagne Operating system concepts 9. John Wiley & Sons, Inc. ISBN 978-1-118-06333-0, 2013.

[3] Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts”, Sixth Edition.

[4] careerride.com.n.d “what is pre-emptive and non- preemptive scheduling”[online] Available at: http:// careerride.com/OS-preemptive-scheduling.aspx, [accessed: November 2012]

[5] Pierre G. Jansen, Sape J. Mullender, Paul J.M. Havinga and Hans, “Light weight EDF Scheduling with Deadline Inheritance”. Cite Seer X, Citation 11-3, 2013

[6] Nidal Nasser, Lutful Karim, and Tarik Taleb, “Dynamic Multilevel Priority Packet Scheduling Scheme for Wireless Sensor Network”, IEEE Transactions on Wireless Communications, Vol. 12, No. 4, April 2013.

[7] Yong Zhao, Chengdu, Liang Chen, Youfu Li and Wenhong Tian, “Efficient task scheduling for Many Task Computing with resource attribute selection” IEEE transaction on Communications , Vol. 11, Issue: 12 , pp. 125–140, Dec. 2014.

[8] Yan Wang, Kenli Li, Hao Chen, Ligang He, and Keqin Li, “Energy-Aware Data Allocation and Task Scheduling on Heterogeneous Multiprocessor Systems With Time Constraints”, IEEE transactions on emerging topics in computing, January 2014.

[9] Lutful Karim, Nidal Nasser, Tarik Taleb, and Abdullah Alqallaf, “An Efficient Priority Packet Scheduling Algorithm for Wireless Sensor Network” IEEE transaction on Ad-hoc and Sensor Networking, 2012.

[10] Y. H. Wang, Y. L. Wu, and K. F. Huang, “A power saving sleepscheduling based on transmission power control for wireless sensornetworks,” in Proc. 2011 International Conf. Ubi-Media Comput., pp.19–24.

[11] E. Karimi and B. Akbari, “Improving video delivery over wireless multimediasensor networks based on queue priority scheduling,” in Proc.2011 International Conf. Wireless Commun., Netw. Mobile Comput., pp.1–4.

[12] L. Karim, N. Nasser, and T. El Salti, “Efficient zone-based routingprotocol of sensor network in agriculture monitoring systems,” in Proc.2011 International Conf. Commun. Inf. Technol., pp. 167–170.