Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 1, January 2014, pg.210 – 217

RESEARCH ARTICLE

EFFICIENT DYNAMIC MULTILEVEL PRIORITY TASK SCHEDULING FOR WIRELESS SENSOR NETWORKS

Mrs. K.Indumathi[#1], Mrs. M. Santhi*[2]

[#]M.E II year, Department of Computer Science and Engineering,

[1] mail2indumathi@gmail.com

[*]Assistant Professor, Department of Computer Science and Engineering,

Sri Subramanya College Of Engineering and Technology,

Palani, Dindigul, TamilNadu, India-624615

[2] santhivinodhan@gmail.com

Abstract

Most of the existing packet-scheduling mechanism of WSN use First Come First Served (FCFS),non-preemptive priority and preemptive priority scheduling algorithms. These algorithms incur a high processing overhead and long end-to-end data transmission delay due to improper allocation of data packets to queues in multilevel queue scheduling algorithms. Moreover, these algorithms are not dynamic to the changing requirements of WSN applications since their scheduling policies are predetermined. This paper proposes a Dynamic Multilevel Priority (DMP) packet scheduling scheme, deal with the circular wait and preemptive conditions. In proposed scheme, WSN 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. Improve the performance of task scheduling schemes in terms of end to end delay and deadlock prevention.

Keywords—WSN; packet scheduling; preemptive priority scheduling; non-preemptive priority scheduling; real-time; non-real-time; data waiting time; FCFS

I. INTRODUCTION

Most existing Wireless Sensor Network (WSN) operating systems use First Come First Serve (FCFS) schedulers that process data packets in the order of their arrival time and, thus, require a lot of time to be delivered to a relevant base station (BS). However, to be meaningful, sensed data have to reach the BS within a specific time period or before the expiration of a deadline. Additionally, real-time emergency data should be delivered to BS with the shortest possible end-to-end delay. Hence, intermediate nodes require changing the delivery order of data packets in their ready queue based on their importance (e.g., real or non-real time) and delivery deadline. Furthermore, most existing packet scheduling algorithms of WSN are neither dynamic nor suitable for large scale applications since these schedulers are predetermine and static, and cannot be changed in response to a change in the application requirements or environments. For example, in many realtime applications, a real-time priority scheduler is statically used and cannot be changed during the operation of WSN applications.

This paper proposes a Dynamic Multilevel Priority (DMP) packet scheduling scheme for WSNs in which sensor nodes are virtually organized into a hierarchical structure. Nodes that have the same hop distance from the BS are considered to be located at the same hierarchical level. Data packets sensed by nodes at different levels are processed using a TDMA scheme. For instance, 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 and classify data packets as (i) real-time (priority 1), (ii) non-real-time remote data packet that are received from lower level nodes (priority 2), and (iii) non-real-time local data packets that are sensed at the node itself (priority 3). Non-real-time data traffic with the same priority are processed using the shortest job first (SJF) scheduler scheme since it is very efficient in terms of average task waiting time.

II. RELATED WORK

The task scheduling schemes are classified based on several factors such as:

DEADLINE

Packet scheduling schemes can be classified based on the deadline of arrival of data packets to the base station (BS), which are as follows. First Come First Served (FCFS): Most existing WSN applications use First Come First Served (FCFS) schedulers that process data in the order of their arrival times at the ready queue. In FCFS, data that arrive late at the intermediate nodes of the network from the distant leaf nodes require a lot of time to be delivered to base station (BS) but data from nearby neighboring nodes take less time to be processed at the intermediate nodes. In FCFS, many data packets arrive late and thus, experience long waiting times. Earliest Deadline First (EDF): Whenever a number of data packets are available at the ready queue and each packet has a deadline within which it should be sent to BS, the data packet which has the earliest deadline is sent first. This algorithm is considered to be efficient in terms of average packet waiting time and end-to-end delay.

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, when a packet t1 starts execution, task $t1$ carries on even if a higher priority packet $t2$ than the currently running packet $t1$ arrives at the ready queue. Thus $t2$ has to wait in the ready queue until the execution of $t1$ is complete.

Preemptive:

In preemptive priority packet scheduling, higher priority packets are processed first and can preempt lower priority packets by saving the context of lower priority packets if they are already running.

The widely used operative system of WSN and classify them as either cooperative or preemptive.

Cooperative scheduling schemes can be based on a dynamic priority scheduling mechanism, such as EDF and Adaptive Double Ring Scheduling (ADRS) , that uses two queues with different priorities. The scheduler dynamically switches between the two queues based on the deadline of newly arrived packets. 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 one.

Preemptive scheduling can be based on the Emergency Task First Rate Monotonic (EF-RM) scheme. EF-RM is an extension to Rate Monotonic (RM), a static priority scheduling, whereby the shortest-deadline job has the highest priority. EF-RM divides WSN tasks into Period Tasks, (PT) whose priorities are decided by a RM algorithm, and non- period tasks, which have higher priority than PTs and can interrupt, whenever required, a running PT. Based on this predicted priority, the task scheduling takes place.

PACKET TYPE

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

Real-time packet scheduling:

Packets at sensor nodes should be scheduled based on their types and priorities. Real-time data packets are considered as the highest priority packets among all data packets in the ready queue. Hence, they are processed with the highest priority and delivered to the BS with a minimum possible end-to-end delay.

Non-real-time packet scheduling:

Non-real time packets have lower priority than real-time tasks. They are hence delivered to BS either using first come first serve or shortest job first basis when no real-time packet exists at the ready queue of a sensor node. These packets can be intuitively preempted by real-time packets.

NUMBER OF QUEUE

Packet scheduling schemes can also be classified based on the number of levels in the ready queue of a sensor node. These are as follows.

Single Queue:

Each sensor node has a single ready queue. All types of data packets enter the ready queue and are scheduled based on different criteria: type, priority, size, etc. Single queue scheduling has a high starvation rate.

Multi-level Queue:

Each node has two or more queues. Data packets are placed into the different queues according to their priorities and types. Thus, scheduling has two phases: (i) allocating tasks among different queues, (ii) scheduling packets in each queue. The number of queues at a node depends on the level of the node in the network. For instance, a node at the lowest level or a leaf node has a minimum number of queues whilst a node at the upper levels has more queues to reduce end-to-end data transmission delay and balance network energy consumptions.

III. PROBLEM DEFINITION

A. Existing System

Most existing Wireless Sensor Network (WSN) operating systems use First Come First Serve (FCFS) schedulers that process data packets in the order of their arrival time.

In non-preemptive packet scheduling schemes (interchangeably use as task scheduling in this paper), real-time data packets have to wait for completing the transmissions of other non-real-time data packets.

In preemptive priority scheduling, lower-priority data packets can be placed into starvation for continuous arrival of higher-priority data. In the multilevel queue scheduling algorithm, each node at the lowest level has a single task queue considering that it has only local data to process.

Most existing packet scheduling algorithms 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 the application requirements or environments.

B. Proposed System

Most of the wireless sensor network uses FCFS scheduling algorithm for scheduling the packets. These are static. It is not suitable for changing requirements of WSN. The multilevel queuing is used for static changes only. Hence now proposed to implement this DMP technique. In this technique, the tasks are scheduled to keep in the multilevel queuing based on requests and also implement without any deadlock.

The block diagram of the existing system has been shown in Figure 1.

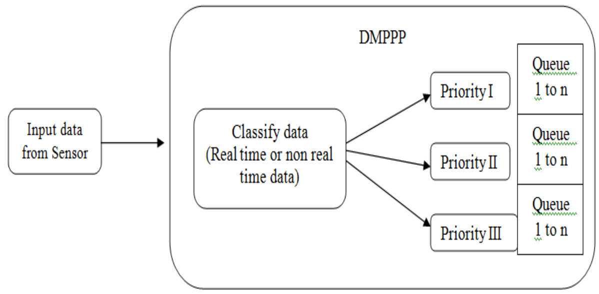

Figure 1 Block Diagram

The general working principle of the proposed DMP scheduling scheme is illustrated in Figure 2.

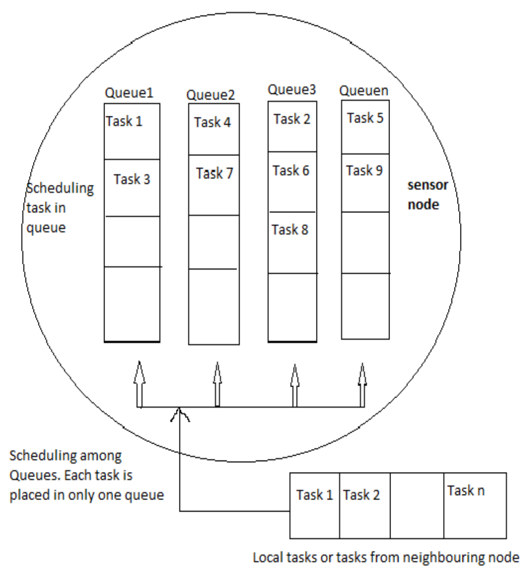

Figure 2 Scheduling data among multiple queues

Scheduling data packets among several queues of a sensor node. Data packets that are sensed at a node are scheduled among a number of levels in the ready queue. Then, a number of data packets in each level of the ready queue are scheduled; *Data*1 is scheduled to be placed in the first level, Queue1. Then, *Data*1 and *Data*3 of Queue1 are scheduled to be transmitted based of different criteria.

The proposed scheduling scheme assumes that nodes are virtually organized following a hierarchical structure. Nodes that are at the same hop distance from the base station (BS) are considered to be located at the same level. Data packets of nodes at different levels are processed using the Time-Division Multiplexing Access (TDMA) scheme. For instance, nodes that are located at the lowest level and the second lowest level can be allocated timeslots 1 and 2, respectively. Consider three-level of queues, that is, the maximum number of levels in the ready queue of a node is three: priority 1 (*pr*1), priority 2 (*pr*2), and priority 3 (*pr*3) queues. Real-time data packets go to *pr*1, the highest priority queue, and are processed using FCFS. Non-real-time data packets that arrive from sensor nodes at lower levels go to *pr*2, the second highest priority queue. Finally, non-real time data packets that are sensed at a local node go to *pr*3, the lowest priority queue. The possible reasons for choosing maximum three queues are to process (i) real-time *pr*1 tasks with the highest priority to achieve the overall goal of WSNs, (ii) non real-time *pr*2 tasks to achieve the minimum average task waiting time and also to balance the end-to-end delay by giving higher priority to remote data packets, (iii) non-real-time *pr*3 tasks with lower priority to achieve fairness by preempting *pr*2 tasks if *pr*3 tasks wait a number of consecutive timeslots.

IV. MODULE DESCRIPTION

There are 4 modules to be implemented in the system. They are:

A. Node formation and Assumption

In this module, the node refers to the sensor; the sensor will give data to through WSN. The input is taken from the sensor, and then split that into real time data and non-real time data. The real time data are the data which dynamically changes and non real time data are data which are static.

The assumptions and implement DMP packet scheduling scheme are.

- Data traffic comprises only real-time and non-real-time data, e.g., real-time health data sensed by body sensors and non-real-time temperature data.
- All data packets (real-time and non-real-time) are of same size.
- Sensors are time synchronized.
- No data aggregation is performed at intermediate nodes for real-time data.
- Nodes are considered located at different levels based on the number of hop counts from BS.
- Timeslots are allocated to nodes at different levels using TDMA scheme, e.g., nodes at the lowest level, *lk* are assigned timeslot 1. Details of timeslot allocation are explained in the "Terminologies" subsection.
- The ready queue at each node has maximum three levels or sections for real-time data (*pr*1) non-real-time remote data (*pr*2) and non-real-time local data (*pr*3).
- The length of data queues is variable. For instance, the length of real-time data queue (*pr*1) is assumed to be smaller than that of non-real-time data queues (*pr*2 and *pr*3). However, the length of the non-real-time *pr*2 and *pr*3 queues are same.
- DMP scheduling scheme uses a multichannel MAC protocol to send multiple packets simultaneously.

B. Priority identification and Task Scheduling

Priority identification means identifies the input sensor data and find out the priority of the data. The priority is based on data, if real time data occurs, it considered as high priority and placed to Priority I, if non real time data occurs, it considered as next high priority and placed to Priority II, and if local data occurs, it considered as next high priority and placed to Priority III.

C. Queue Formation for Task

Each queue is assigned for priority based on the coming tasks. Here the queue can able to hold three tasks at a time. The queue gets formed based on the incoming task. The first level queue can store the real time (priority I) task, the second level queue can store the non real time (priority II) task and the third level queue can store the local (priority III) task.

D. DMP Packet Scheduling

This module is the dynamic multilevel priority packet scheduling, it dynamically schedule the task which is sensed from the sensor. The identification of data, task scheduling and queue formation process are completed. Using this, the DMP assign the task to the particular queue of the particular priority.

V. PERFORMANCE ANALYSIS

In this section, we analyze the performance of the proposed DMP task scheduling scheme in terms of end to end delay, and total waiting time.

A. End-to-End Delay

In the following, we formulate the average end-to-end delay of transmitting different priority data packets to the base station (BS). Again, we interchange ably use task and data to represent the data packets that are sensed at a sensor node.

Real-timePriority1QueueData:

Let us assume that a node X, residing at level lk is sensing a real-time, emergency event, e.g., fire detection. This node transmits the emergency priority1 data to BS through $lk-1$ intermediate levels. We consider the following scenario whereby every time a real data packet reaches a neighboring active node, y at an upper level, a non-real time lower priority data is being processed

while $taskk,i$ is received by $nodei$ at level k, i.e., lk **do**
if $Type(taskk,i) = real - time$ **then**
put $taskk,i$ into $pr1$ queue
else if $node\ i$ is not at lowest levels **then**
if $task\ k,i$ is not local **then**
put $task\ k,i$ into $pr2$ queue
else
put $taskk,i$ into $pr3$ queue
end if
else
put $task\ k,i$ into $pr2$ queue
end if
Assume, the duration of a timeslot at $lk \leftarrow t(k)$
Data sensing time of $nodei$ at $lk \leftarrow senseT\ imek(t)$
Remaining time after data sensing, $t1(k) = t(k) - senseT\ imek(t)$
Let total real-time tasks for $nodei$ at $lk \leftarrow nk(pr1)$
Let $procT\ imepr1(k) \leftarrow \Sigma^{nk(pr1)}_{j=1}\ procT\ ime(j)$
if $procT\ imepr1(k) < t1(k)$ **then**
All $pr1$ tasks of $nodei$ at lk are processed as FCFS
Remaining time $t2(k) \leftarrow t1(k) - procT\ imepr1(k)$
Let, total $pr2$ tasks for $nodei$ at $lk \leftarrow nk(pr2)$
Let $procT\ imepr2(k) \leftarrow \Sigma^{nk(pr2)}_{j=1}\ procT\ ime(j)$
if $procT\ imepr2(k) < t2(k)$ **then**
All $pr2$ tasks are processed as FCFS
$pr3$ tasks are processed as FCFS for the remaining time,
$t3(k) \leftarrow t2(k) - procT\ imepr2(k)$
else
$pr2$ tasks are processed for $t2(k)$ time
no $pr3$ tasks are processed
end if
else
only $pr1$ tasks are processed for $t1(k)$ time
no $pr2$ and $pr3$ tasks are processed
end if
if $pr1$ queue empty & $pr2$ tasks are processed α consecutive timeslots
since $t(k) \le procT\ imepr2(k)$ **then**
$pr2$ tasks are preempted at $\alpha + 1, \ldots, \alpha + j$ timeslots by $pr3$ tasks
if $pr1$ task arrives during any of $\alpha+1, \alpha+2, \ldots, \alpha+j$ timeslots **then**
$pr3$ tasks are preempted and $pr1$ tasks are processed
context are transferred again for processing $pr3$ tasks

end if
end if
end while

at that node. Hence, data delivery at *y* is preempted to send real-time data.

Transmission time or delay that is required to place a realtime data from a node into the medium is equal to $data_{pr1}/s_t$. The propagation time or delay to transmit data from the source to destination can be formulated as d/s_p .Considering the above mentioned scenario the end-to-end delay for sending a realtime data satisfies the following inequality.

$$delay_{pr1} \geq l_k \times (data_{pr1}/s_t + pr1_{proc}(t)) + d/s_p + (lk \times t_{overhead}) \quad (1)$$

where $data_{pr1}$ denotes the real-time data size, *st* denotes the data transmission speed, *d* is the distance from the source node to BS, where $d = \Sigma^{lk}{}_{i=1}\, d_i$, s_p denotes the propagation speed over the wireless medium, $pr1_{proc}(t)$ is the processing time of real-time tasks at each node, and $t_{overhead}$ is an overhead in terms of context switching and queuing time (including time for preemption). However, a real-time task *t*1 has to wait if there is a number, n_{pr1}, of a real-time task ahead of *t*1 at the *pr*1 queue. We assume that all real-time data have the same size.

Therefore, the end-to-end delay for a real-time task *t*1 considering that *t*1 has n_{pr1} number of real-time tasks ahead of it,

$$delay_{t1} \geq \Sigma^{npr1}{}_{i=1}(delaypr1)_i \quad (2)$$

Non-real time Priority 2 Queue Data:

Tasks at *pr*2 queue can be preempted by real-time ones. The transmission time or delay to place *pr*2 data from a node into the medium can be therefore computed as $data_{pr2}/s_t$. Thus, the total end-to-end delay for a *pr*2 task that can be processed in the same timeslot exceeds

$$l_k \times (data_{pr1}/s_t + data_{pr2}/s_t + pr1_{proc}(t) + pr2_{proc}(t)) + d/s_p + (l_k \times t_{overhead}) \quad (3)$$

Non-real time Priority 3 Queue Data:

In the best case, when no task is available at the *pr*1 and *pr*2 queues, the end-to-end delay of the *pr*3 tasks will be almost equal to that of the *pr*1 queue tasks (Equation 1) although it can differ slightly based on the size of the *pr*3 queue task. We assume that the *pr*3 queue tasks are processed by preempting *pr*2 queue tasks if for α consecutive timeslots there is no task at the *pr*1 queue but there are tasks available at the *pr*2 queue. Let *tk* denote the length of a timeslot of nodes at level l_k. The transmission time or delay to place *pr*3 data from a node into the wireless medium is equal to $data_{pr3}/s_t$. However, during the processing of the *pr*3 queue tasks, these tasks can be preempted by realtime tasks. They are processed again after the completion of real-time tasks. Thus, the end-to-end delay for processing *pr*3 tasks will be exceeding

$$\alpha \times t(k) + l_k \times (data_{pr3}/s_t + pr3_{proc}(t)) + d/s_p) + (lk \times t_{overhead}) \quad (4)$$

VI. CONCLUSION

Thesis, DMP 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 highest priority data while exhibiting acceptance fairness towards lowest priority data. Experimental results show that the proposed DMP packet scheduling scheme has better performance than the existing FCFS and Multilevel Queue Scheduler in terms of the average task waiting time and end-to-end delay, deal with the circular wait and preemptive conditions to prevent deadlock from occurring and would also validate the simulation result using a real test-bed.

ACKNOWLEDGEMENT

We thank the Sri Subramanya College of Engineering & Technology for the motivation and encouragement for giving the opportunity to do this research work as successful one.

REFERENCES

[1] G. Anastasi, M. Conti, and M. Di Francesco, “Extending the lifetime of wireless sensor networks through adaptive sleep,”IEEE Trans. Industrial Informatics, vol. 5, no. 3, pp. 351–365, 2009.

[2] E. Bulut and I. Korpeoglu, “DSSP: a dynamic sleep scheduling protocol for prolonging the lifetime of wireless sensor networks,” in Proc. 2007 International Conf. Advanced Inf. Networking Appl., vol. 2, pp. 725–730.

[3] S. Chachra and M. Marefat, “Distributed algorithms for sleep scheduling in wireless sensor networks,” in Proc. 2006 IEEE International Conf. Robot. Autom., pp. 3101–3107.

[4] P. Guo, T. Jiang, Q. Zhang, and K. Zhang, “Sleep scheduling for critical event monitoring in wireless sensor networks,” IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 2, pp. 345–352, Feb. 2012.

[5] D. Shuman and M. Liu, “Optimal sleep scheduling for a wireless sensor network node,” in Proc. 2006 Asilomar Conf. Signals, Syst. Comput., pp. 1337–1341

[6] G. Bergmann, M. Molnar, L. Gonczy, and B. Cousin, “Optimal period lengthfortheCQSsensornetworkscheduling algorithm,”in Proc.2010 International Conf. Netw. Services, pp. 192Ð199.

[7] Y. Zhao, J. Wu, F. Li, and S. Lu, “VBS: maximum lifetime sleep scheduling for wireless sensor networks using virtual backbones,” in Proc. 2010 IEEE INFOCOM, pp. 1–5.

[8] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, “RAP: a real-time communication architecture for large-scale wireless sensor networks,” in Proc. 2002 IEEE Real-Time Embedded Technol. Appl.Symp., pp. 55–66.

[9] Y. Wang, D. Wang, W. Fu, and D. P. Agrawal, “Hops-based sleep scheduling algorithm for enhancing lifetime of wireless sensor networks,” in Proc. 2006 IEEE International Conf. Mobile Adhoc Sensor Syst., pp. 709–714.

[10] X. Xu, Y. H. Hu, J. Bi, and W. Liu, “Adaptive nodes scheduling approach for clustered sensor networks,” in Proc. 2009 IEEE Symp. Comput.Commun.,
pp. 34–39.

