



An Efficient Packet Scheduling Scheme in Wireless Sensor Networks to Prevent the Deadlock

J.Nivetha¹, A.Deepa²

Assistant Professor, Department of CSE, Apollo Engineering College, Chennai, Tamilnadu, India¹

Student, M.E (CSE), Ponnaiyah Ramajayam Institute of Technology, Thanjavur, Tamilnadu, India²

ABSTRACT:- Scheduling different types of packet in wireless sensor network is highly important since it ensures delivery of different types of data packets based on their priority and fairness with a minimum latency. In the existing Dynamic Multilevel Priority (DMP) packet scheduling scheme the sensor nodes are virtually organized into a hierarchical structure. Each node maintains three levels of priority queues. We classify as (i) real-time (priority 1), (ii) non-real-time remote data packet that are received from lower level nodes (priority2), and (iii) non-real-time local data packets that are sensed at the node itself (priority 3). If a real-time task holds the resources for a longer period of time, other tasks need to wait for an undefined period time, causing the occurrence of a deadlock. This deadlock situation degrades the performance of task scheduling schemes in terms of end-to-end delay. In the proposed scheme is to provide an improvement over other deadlock prevention algorithms. A process request resources; and if the resources are not available at that time, it enters a wait state and may never execute. This paper aims to present a resource allocation algorithm that ensures that the requested resources are made available to the processes while incurring lower overhead for deciding about the resource allocation.

KEYWORDS: Wireless sensor network, packet scheduling, real-time, non-real-time, data waiting time, Deadlock, Resources.

I.INTRODUCTION

Wireless Sensor Networks (WSN) has gained a great value and importance due to its flexibility, cheaper implementation cost, mobility etc. The sensor networks are expected to play increasingly important role in future especially in monitoring and military applications on large scales and it consists of small and inexpensive sensor nodes that have limited memory, limited computing power, and that operate using batteries. The sensor network consists of a group of sensor nodes that perform distributed sensing and acting tasks. Sensors are able to sense environmental information and actors are able to act upon environment. These nodes communicate wirelessly and dynamically to monitor and control environment. The existing 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 time slots 1 and 2, respectively. Each node maintains three levels of priority queues. This is because we classify data packets as, if the data packets are real-time priority packet then placed on the priority queue 1. Non-real-time remote data packet that are received from lower level nodes then the data packets are placed on priority queue 2 and non-real-time local data packets that are sensed at the node itself are placed on priority queue 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 but the a real-time task holds the resources for a longer period of time, other tasks need to wait for an undefined period time, causing the occurrence of a deadlock. 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.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

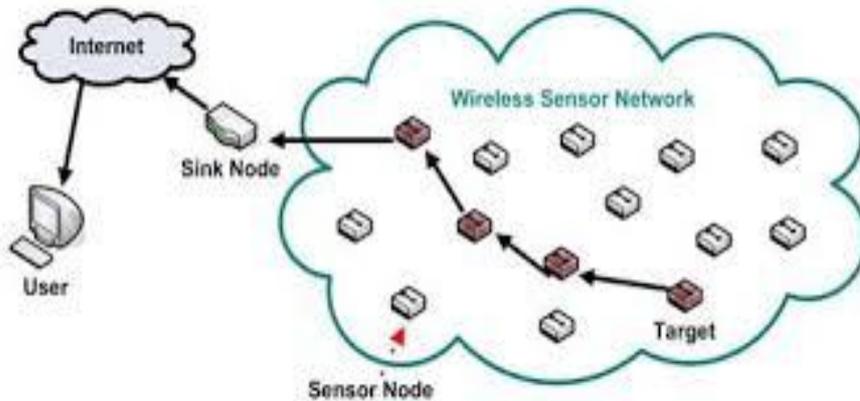


Figure 1.1 Wireless Sensor Network

Deadlock refers to the coordination and concurrency problem where two or more processes are waiting indefinitely for the release of a shared resource [2]. The deadlock problem involves a circular waiting where one or more transactions are waiting for resources to become available and those resources are held by some other transactions that are in turn blocked until resources held by the first transaction are released. [7]. Deadlock processes never terminate their executions and the resources held by them are not available to any other process.

Deadlock is an undesirable situation; some of the consequences of deadlock have been listed:

- Throughput of the system is affected.
- Performance of system suffers.
- Deadlock in real time applications is not appreciable at any cost.
- Entire or partial system is crippled.
- Utilization of the involved resources decreases to zero.
- Deadlock increases with deadlock persistence time.
- Deadlock cycles do not terminate by themselves until properly detected and resolved.
- No progress

These are the consequences of deadlock; let's have a look at some reasons why deadlock occurs in a resource allocation system:

- Deficiency of system resources.
- An inappropriate execution order of processes.
- Improper resource allocation logic [7]

Deadlock is a highly unfavorable situation, in deadlock situations the whole system or a part of it remains indefinitely blocked and cannot terminate its task. Therefore it is highly important to develop efficient control and scheduling algorithm to optimize the system performance while preventing deadlock situations. Basically the necessary four conditions for deadlock to occur which are known as Coffman conditions [7], are:

- Mutual exclusion condition:** A resource cannot be used by more than process at a time.
- Hold and wait condition:** Processes that already hold resource may request new resource.
- Non preemption condition:** No resource can be forcibly removed from a process that holds it, and resources can be released only by the explicit action of the process.
- Circular wait condition:** Two or more processes form circular chain where each process waits for a resource that the next process in the chain holds.

This paper presents a new threshold based resource allocation technique that will allocate the resources to the requesting process. The proposed technique reduces the overhead incurred by the deadlock avoidance and it guarantee



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

that a deadlock will never occur. However, as compared to the deadlock detection strategy, it reduces the frequency of deadlock occurrence in the system. The proposed threshold based technique increases the system performance by reducing the overhead incurred during both the deadlock avoidance and recovery.. The proposed technique also takes in account the arrival and burst time for the processes while granting the resources.

The rest of the paper is organized as follows: section II, illustrates a related works while section III describes the proposed approach. Section IV elaborates our simulation results . Finally, paper concludes with section V.

II.RELATED WORKS

In this section, we define the following terminologies and factors that are used in designing the DMP packet scheduling scheme.

A.Routing Protocol:

For the sake of energy efficiency and balance in energy consumption among sensor nodes, we envision using a zone-based routing protocol . In a zone based routing protocol, each zone is identified by a zone head (ZH) and nodes follow a hierarchical structure, based on the number of hops they are distant from the base station (BS).For instance, nodes in zones that are one hop and two hops away from the BS are considered to be at level 1 and level 2, respectively. Each zone is also divided into a number of small squares in such a way that if a sensor node exists in square S_1 , it covers all neighboring squares. Thus, this protocol reduces the probability of having any sensing hole [34] in the network even if the neighboring squares of a node do not have any sensor node.

B.TDMA Scheme:

Task or packet scheduling at each nodal level is performed using a TDMA scheme with variable-length timeslots. Data are transmitted from the lowest level nodes to BS through the nodes of intermediate levels. Thus, nodes at the intermediate and upper levels have more tasks and processing requirements compared to lower-level nodes. Considering this observation, the length of timeslots at the upper-level nodes is set to a higher value compared with the timeslot length of lower-level nodes. On the other hand, real-time and time critical emergency applications should stop intermediate nodes from aggregating data since they should be delivered to end users with a minimum possible delay.

C.Fairness:

This metric ensures that tasks of different priorities get carried out with a minimum waiting time at the ready queue based on the priority of tasks. For instance, if any lower priority task waits for a long period of time for the continuous arrival of higher-priority tasks, fairness defines a constraint that allows the lower-priority tasks to get processed after a certain waiting time.

D.Priority:

The real-time and emergency data should have the highest priority. The priority of non-real-time data packets is assigned based on the sensed location and the size of the data. The data packets that are received by node x from the lower level nodes are given higher priority than the data packets sensed at the node x itself. However, if it is observed that the lower priority non-real-time local data cannot be transmitted due to the continuous arrival of higher priority non-real-time remote data, they are preempted to allow low-priority data packets to be processed after a certain waiting period. Nevertheless, these tasks can be preempted by real-time emergency tasks. In case of two same priority data packets the smaller sized data packets are given the higher priority.

E.Motivational example:

Scheduling data packets among several queues of a sensor node is presented in Figure 2.1 Data packets that are sensed at a node are scheduled among a number of levels in the ready queue.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

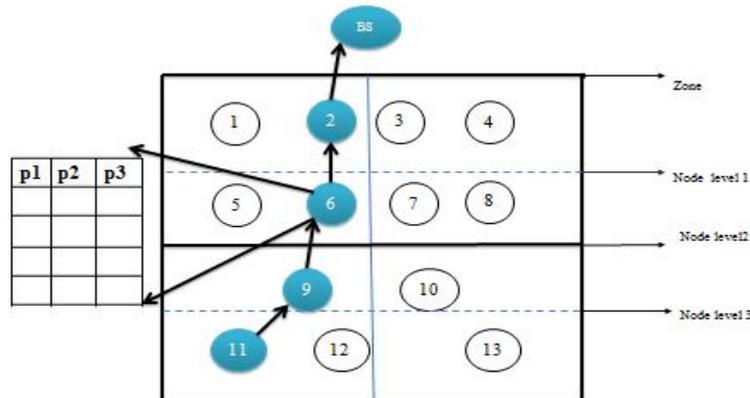


Figure 2.1 Zone based routing in DMP

In the dynamic multilevel priority scheme, each node, except those at the last level of the virtual hierarchy in the zone based topology of 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.

III. PROPOSED SYSTEM MODEL

This paper deals with resources allocation technique that allocates the resources to the requesting processes. The system is assumed to have m resource types, i.e., $R_1, R_2, R_3 \dots R_m$, with $a_1, a_2, a_3 \dots a_m$ instances of each type. Further, it consists of n independent processes $P_1, P_2, P_3 \dots P_n$ where, each process P_i has the attributes (a_i, e_i) that is the arrival and worst-case execution time respectively. The processes are assigned priority based on smallest execution time (Shortest job first, SJF). Several data structures are needed for maintaining the state of the resource allocation in the system; they can be defined as follows:

- i. **Available:** An array of m elements, indicating the number of instances available for each resource type. Thus, $Available(R_j)$, is the number of resource type R_j available in the system.
- ii. **Maximum:** A two dimensional array $n \times m$, defining the maximum resource demand of each process. If $Max[i][j]$ equal k , then process P_i may request at most k instances of resources type R_j in its life time.
- iii. **Allocation:** A two dimensional array $n \times m$, defines the number of resources of each type currently allocated to each process. If $Allocation[i][j]$ equals k , then process P_i is currently allocated k instances of resources type R_j .
- iv. **Need:** A two dimensional array $n \times m$, indicates the remaining resource need of each process. If $Need[i][j]$ equals k , then process P_i may need k more instances of resources type R_j to complete its execution. It can be estimated as $Need[i][j] = Maximum[i][j] - Allocation[i][j]$
- v. **Threshold:** An array of m elements, indicating the minimum number of remaining resource at least one process will require to complete. If $threshold[j]$ equals k , then process at least one process P_i will require at most k instances of resource type R_j to complete its execution. It can be estimated as $Threshold[j] = \lceil \text{need}[i][j] \forall i=1, 2 \dots n \rceil, 0$
- vi. **Request:** A two dimensional array $n \times m$, indicating the number of resource requested by process P_i during its execution. If $Request[i][j]$ equals k , then process P_i request k instances of resource type R_j for current execution
- vii. **Precheck :** when a transaction is at its pre check phase only the resource availability is checked at queue and the transaction is initiated, it is not checked whether the required resources are free or not.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

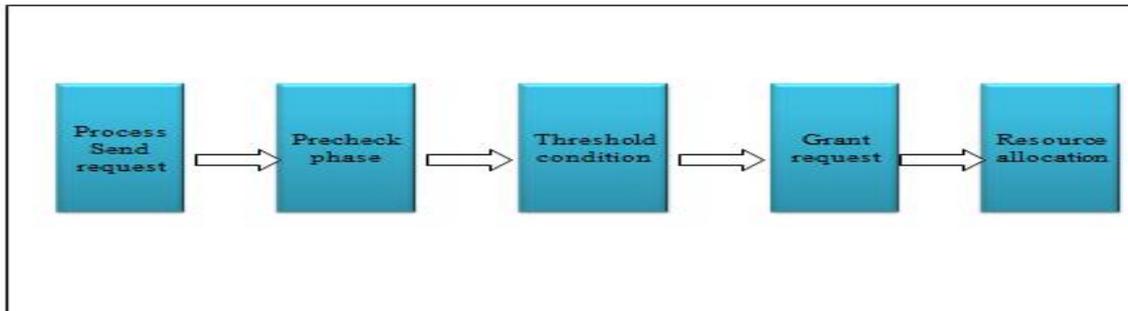


Figure 3.1 Threshold based Resource Allocation technique architecture

A. Working principle

The proposed a threshold based technique in which the system reserves a pool of threshold number of resources to ensure that at least one process will always complete and relieves all the resources it is holding. The proposed technique only considers the requesting process detail along with the system data structure to take the decision of granting or not granting of the resources incurring an overhead of waiting time. Figure 3.1 the proposed Threshold based Resource Allocation (TRA) technique architecture can be illustrate as follows: The proposed method deals with resources allocation technique that allocates the resources to the requesting processes. The system is assumed to have m resource types, i.e., R1, R2, R3... Rm, with $\alpha_1, \alpha_2, \alpha_3 \dots \alpha_m$ instances of each type. Further, it consists of n independent processes P1, P2, P3... Pn where, each process Pi has the attributes (a_i, e_i) that is the arrival and worst-case execution time respectively. The processes are assigned priority based on smallest execution time (Shortest job first, SJF). Several data structures are needed for maintaining the state of the resource allocation in the system.

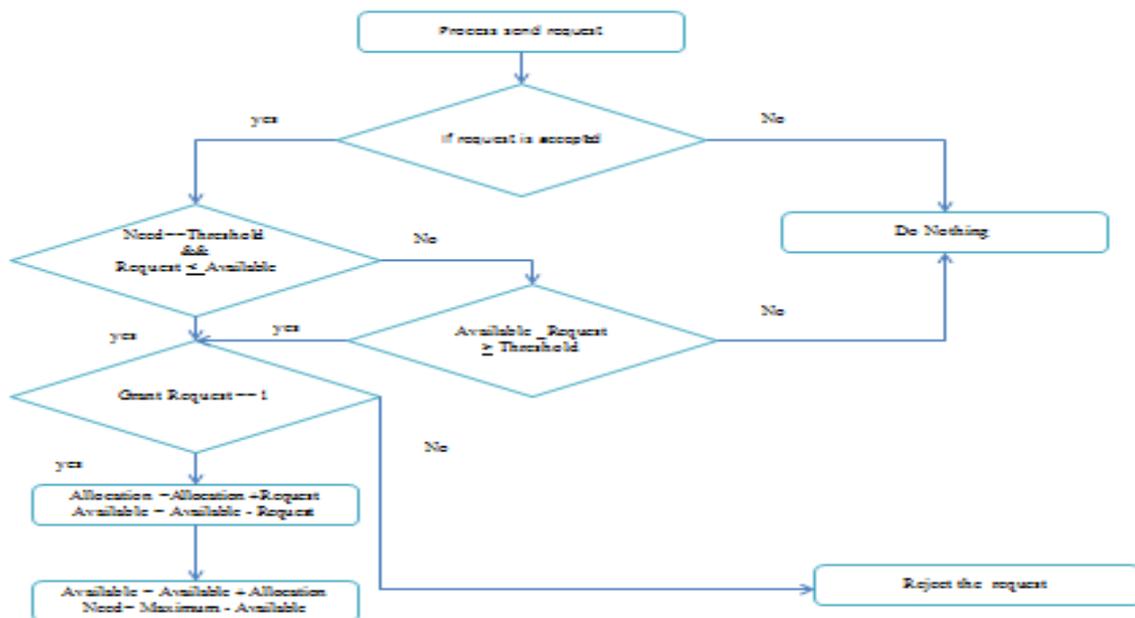


Figure 3.2 Working principle of TRA

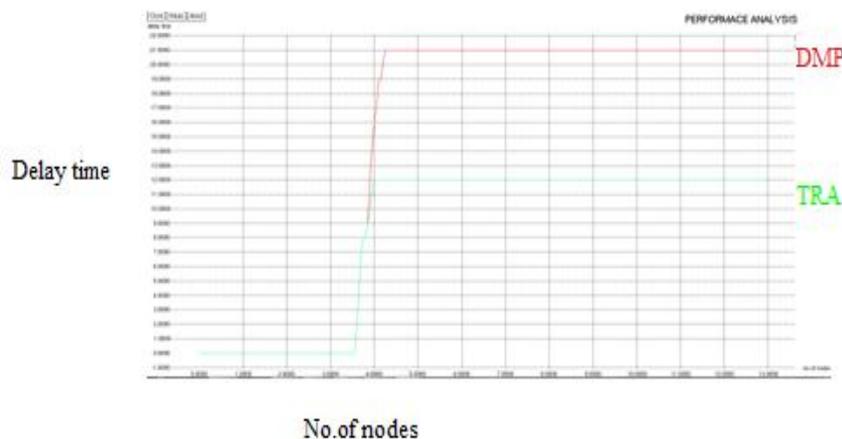
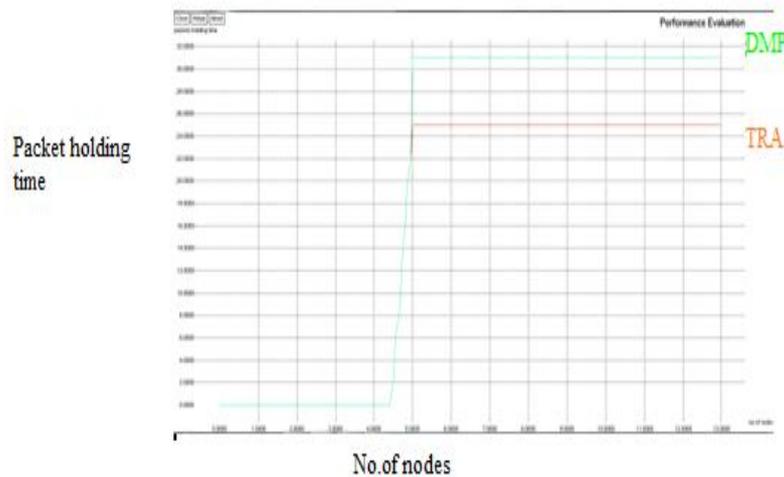
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

IV PERFORMANCE EVALUATION

The simulation model is implemented using the C programming language. It is used to evaluate the performance of the proposed threshold based resource allocation, comparing it against the DMP packet scheduling scheme. The comparison is made in terms of average packet waiting time, and end-to-end data transmission delay.



V.CONCLUSION

In this paper, we propose an efficient packet scheduling scheme in wireless sensor network to prevent the circular wait condition using the threshold based resource allocation technique. If a real-time task holds the resources for a longer period of time, other tasks need to wait for an undefined period time, causing the occurrence of a deadlock. 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, the threshold based resource allocation technique that focused on deadlock avoidance by ensuring that at least one process always has the requisite number of resources to complete. Simulation results illustrate that the threshold based resource allocation technique to outperforms conventional schemes in terms of average data waiting time and end-to-end delay.



ISSN(Online): 2320-9801
ISSN(Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

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. Goswami, Vaisla and Ajit Singh, "VGS Algorithm: An Efficient Deadlock Prevention Mechanism for Distributed Transactions using Pipeline Method" International Journal of Computer Applications (0975 – 8887) Volume 46– No.22, May 2012
3. M. Yu, S. J. Xiahou, and X. Y. Li, "A survey of studying on task scheduling mechanism for TinyOS," in *Proc. 2008 International Conf. Wireless Commun., Netw. Mobile Comput.*, pp. 1–4.
4. N. Habermann, "Prevention of System Deadlocks," Comm. ACM, vol. 12, no. 7, pp. 373-377, 385, July 1969.
5. N. Edalat, W. Xiao, C. Tham, E. Keikha, and L. Ong, "A price-based adaptive task allocation for wireless sensor network," in *Proc. 2009 IEEE International Conf. Mobile Adhoc Sensor Syst.*, pp. 888–893.
6. Nidal Nasser, Lutful Karim & Tarik Talib, "Dynamic Multilevel Priority Packet Scheduling Scheme for wireless sensor network", IEEE Trans on wireless communication, vol 12, NO. 4, April 2013
7. R.C. Holt, "Some Deadlock Properties of Computer Systems", ACM Computing Surveys, vol. 4, no. 3, pp. 179-196, Sept. 1972
8. S. Paul, S. Nandi, and I. Singh, "A dynamic balanced-energy sleep scheduling scheme in heterogeneous wireless sensor network," in *Proc.2008 IEEE International Conf. Netw.*, pp. 1–6, 2008.
9. TinyOS. Available: <http://webs.cs.berkeley.edu/tos>, accessed June 2010.