RESEARCH ARTICLE OPEN ACCESS

Packet Scheduling Mechanism Scheme for Wireless Sensor Networks

L Jagadeesh Naik [#1] C Y Bhaskar Reddy[#2] Dr Venkata Ramanaiah [*3]
[123]Department of Electronics and Communication Engineering
[12]BIT Institute of Technology, Ananthapur Dist. AP, India
[*3] YSR Engineering College, Yogivemana University, AP, India
jagadeeshnaik@gmail.com cybr415@gmail.com

Abstract—
In Wireless Sensor Networks scheduling different types of packets, such as real time and non-real-time data packets, WSN have limited energy resource constraints to reduce sensors energy consumptions. The offered packet-scheduling mechanisms of WSN use First Come First Served (FCFS), These algorithm obtain a high processing overhead and high end-to-end data transmission delay due to the FCFS concept, the high priority real-time data packets due to the transmission of a large data packet in non preemptive priority scheduling, the non-real-time data packets due to continuous arrival of real-time data in preemptive priority scheduling and improper allocation of data packets to queues in multilevel queue scheduling algorithms. In this paper Dynamic Multilevel Priority (DMP) packet scheduling evaluates the performance of the DMP packet scheduling. Simulations results show that the DMP packet scheduling scheme give better perform than others conventional schemes in terms of average data waiting time and end-to-end delay.

Keywords—Wireless sensor network, packet scheduling, FCFS and DMP.

I. INTRODUCTION

In the present existing packet scheduling schemes by classifying them based on several factors as is described in Figure 1.1

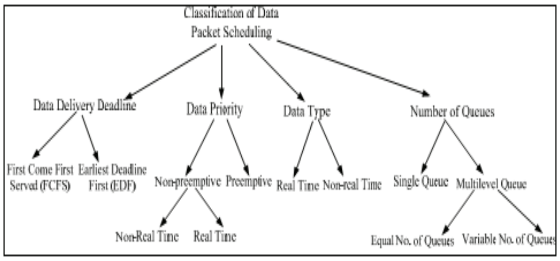

Figure 1: Classification of packet scheduling schemes

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.

The real-time communication architecture for large-scale sensor networks, whereby they use a priority-based scheduler. Data, that have travelled the longest distance from the source node to BS and have the shortest deadline, are prioritized. If the deadline of a particular task expires, the relevant data packets are dropped at an intermediate node. Though this approach reduces network traffic and data processing overhead, it is not efficient since it consumes resources such as memory and computation power and increases processing delay. The performance of the scheme can be improved by incorporating.

RACE a packet scheduling policy and routing algorithm for real-time large scale sensor networks that uses a loop-free Bellman-Ford algorithm to find paths with the minimum traffic load and delay between source and destination. RACE uses the Earliest.

Deadline First (EDF) scheduling concept to send packets with earliest deadline. It also uses a prioritized MAC protocol that modifies the initial wait time after the channel becomes idle and the back off window increases the function of the IEEE

802.11 standard. Priority queues actively drop packets whose deadlines have expired to avoid wasting network resources. However, local prioritization at each individual node in RACE is not sufficient because packets from different senders can compete against each other for a shared radio communication channel.

II. RELATED WORK

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 implementation, task t1carries on even if a higher priority packet t2 than the at present running packet t1arrives at the ready queue. Thus t2 has to wait in the ready queue until the carrying out of t1is complete.

Preemptive: In preemptive priority packet scheduling, higher priority packets are process first and can preempt lower priority packets by saving the framework of lower priority packets if they are previously running.

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.

In the present packet scheduling mechanisms that the system of WSN and classify them as either cooperative or preemptive. In the cooperative scheduling scheme can be based on a dynamic priority scheduling mechanism, such as EDF and Adaptive Double Ring Scheduling (ADRS) [32], 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. Cooperative schedulers in Tiny OS are appropriate for applications with partial system resources and with no hard real-time requirements. On the other hand, 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, where by 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.

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 process with the highest priority and delivered to the BS with a minimum achievable end-to-end delay**.**

III. DMP PACKETSCHEDULINGSCHEME

In the 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. On the other hand, 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 [5], each node at the lowest level has a single task queue considering that it has only local data to process Scheduling data among multiple queues. However the local data can also be real-time or non-real time and should be thus processed according to their priorities. Otherwise, emergency real-time data traffic may experience long queuing delays till they could be processed. In the Dynamic Multilevel Priority (DMP) packet scheduling scheme that ensures a tradeoff between priority and fairness.

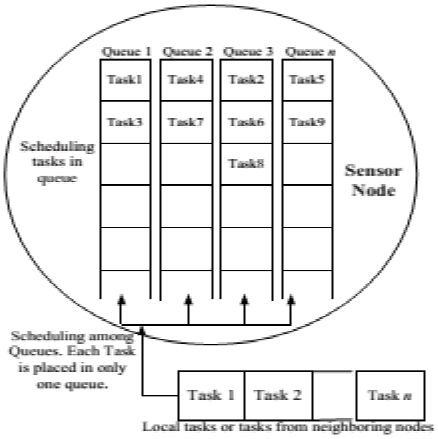
Queue 1
Queue 2
Queue 3
Queue n
Task1
Task4
Task2
Task5
Task3
Task7
Task6
Task9
Task8
Scheduling tasks in queue
Sensor Node
Scheduling among Queues. Each Task is placed in only one queue.
Task 1
Task 2
Task n
Local tasks or tasks from neighboring nodes

Figure 2: Scheduling data among multiple queues

IV. Working principle of DMP

The working principle of DMP is packet scheduling scheme with its pseudo-code. Scheduling data packets among several queues of a sensor node is presented in Figure 2. 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. For instance, Figure 2 demonstrates that the data packet, Data1 is scheduled to be placed in the first level, Queue1. Then, Data1 and Data3 of Queue1 are scheduled to be transmitted based of different criteria. The general working principle of the proposed DMP scheduling scheme is described in Figure 3. Shown below the 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 those are located at the lowest level and the second lowest level can be allocated timeslots 1 and 2, respectively. Here consider three-level of queues, that is, the maximum number of levels in the ready queue of a node is three they are priority 1, priority 2 and priority 3 queues. Real-time data packets go to pr1, the highest priority queue, and processed using FCFS. Non-real-time data packets that arrive from sensor nodes at lower levels go topr2, the second highest priority queue. Finally, non-real time data packets that are sensed at a local node go to pr3, the lowest priority queue.

The possible reasons for choosing maximum three queues are to process (i) real-time pr1 tasks with the highest priority to achieve the overall goal of WSNs, (ii) non real-time pr2 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 pr3 tasks with lower priority to achieve fairness by preemptingpr2 tasks if pr3 tasks wait a number of consecutive timeslots queue sizes differ based on the application requirements. Since preemptive priority scheduling incurs overhead due to the context storage and switching in resource constraint sensor networks, the size of the ready queue for preemptive priority schedulers is expected to be smaller than that of the preempt able priority schedulers. The idea behind this is that the highest-priority real-time tasks rarely occur. They are placed in the preemptive priority task queue (pr1queue) 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 lower level are placed in preempt able priority queue (pr2queue).

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 usually processed in FCFS fashion. Each packet has an ID, which consists of two parts, namely level ID and node ID. When two equal priority packets arrive at the ready queue at the same time, the data packet which is generated at the lower level will have higher priority.

This phenomenon reduces the end-to-end delay of the lower level tasks to reach the BS. For two tasks of the same level, the smaller task (i.e., in terms of data size) will have higher priority it is expected that when a node x senses and receives data from lower-level nodes, it is able to process and forward most data within its allocated timeslot; hence, the probability that the ready queue at a node becomes full and drops packets is low. However, if any data remains in the ready queue of node x during its allocated timeslot, that data will be transmitted in the next allocated timeslot Timeslots at each level are not fixed. They are rather calculated based on the data sensing period, data transmission rate and CPU speed. They are increased as the levels progress through BS. However, if there is any real-time or emergency response data at a particular level, the time required to transmit that data will be short and will not increase at the upper levels since there is no data aggregation. The remaining time of a timeslot of nodes at a particular level will be used to process data packets at other queues. Since the probability of having real-time emergency data is low, it is expected that this scenario would not degrade the system performance. Instead, it may improve the perceived Quality of Service (QoS) by delivering real-time data fast. Moreover, if any node x at a particular level completes its task before the expiration of its allocated timeslot, node x goes to sleep by turning its radio off for the sake of energy efficiency.

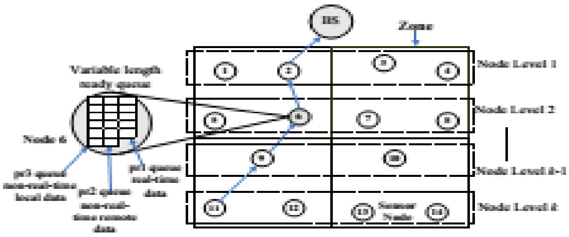

Figure 3: Proposed dynamic multi-level priority (DMP) packet scheduling scheme

V. RESULTS AND DISCUSSIONS

The performance evaluation of the DMP and FIFO scheduling scheme simulation results has been carried out using the NS 2 simulator. In this paper simulation results are shown by fixed number of nodes. The simulation results are shown below.

End to end delay: The average end to end delay of transmitting different priority data packets to the base station (BS) with n no of packets as shown in equation below

$$delay_{t_1} \geq \sum_{i=1}^{n_{pr1}} (delaypr_1)_i$$

Average Waiting Time: the average waiting time of tasks at different workloads. Let us assume that pri_{ji} represents the processing time of the j^{th} priority task at a node x, where, $1\leq i\leq 3$ *and* $1\leq ji \leq ni$

Total processing time: $pr_i(t) = \sum_{j_i=1}^{n_i} pr_{ij_i}(t)$

$$AvgWaitingTimePr_1(t) = \frac{\sum_{j_1=1}^{n_1-1}\sum_{m=1}^{j_1} pr_{1,m}(t)}{n_1}$$

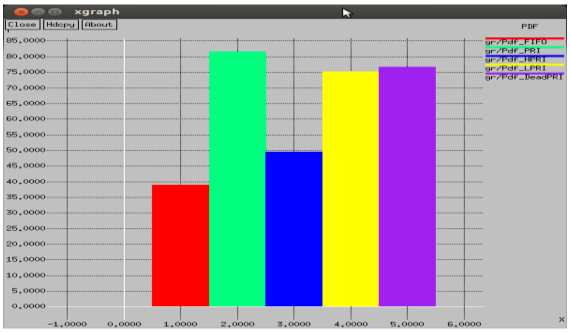

Figure 1: Comparison of packet delivery fraction

To obtained packet delivery fraction of DMP ,FIFO and others priority mechanism with respect to simulation time. Their comparison is given by the superimposed plot shown in Figure 1. We can observe from the graph that the packet delivery fraction of DMP is much higher than other packet scheduling.

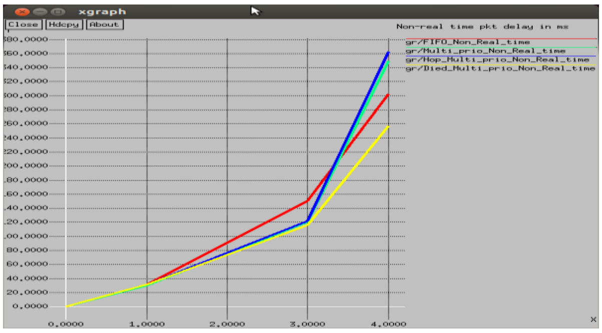

Figure 2: Comparison of non real time packet delay

We can observe from the figure 2 that the non real time packet delay DMP is much lesser than other packet scheduling. In FCFS delay is very high compared to DMP mechanism.

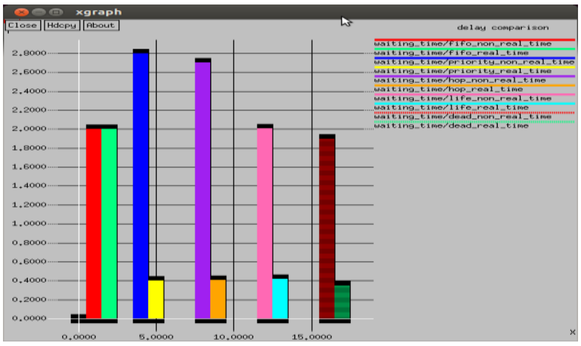

Figure 2: Comparison of waiting time

To obtained real and non real waiting time of DMP, FIFO and others priority mechanism with respect to simulation time. Their comparison is given by the superimposed plot shown in Figure 3. We can observe from the graph that the real time packet delay DMP is much lesser than other packet scheduling. In FCFS delay is very high compared to DMP mechanism.

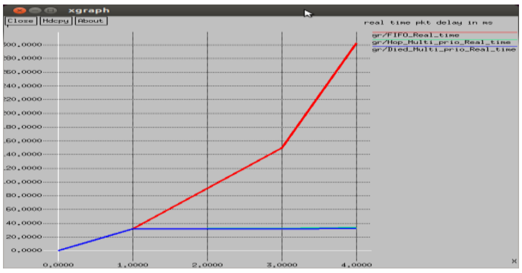

Figure 4: Comparison of real time packet delay

To obtained real time packet delay of DMP, FIFO and others priority mechanism with respect to simulation time. Their comparison is given by the superimposed plot shown in Figure 4. We can observe from the graph that the real time packet delay DMP is much lesser than other packet scheduling. In FCFS delay is very high compared to DMP mechanism.

CONCLUSION

In this paper evaluate the performance of packet scheduling using network simulator NS2. DMP, FIFO and other packet scheduling the simulation results are carried out its show that the average waiting time and delay is less compare to other scheduling mechanism. The DMP packet scheduling gives better performance in terms of the parameters average end to end delay, average waiting time.

REFERENCES

[1] 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, pp 1448-1459, April 2013

[2] 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

[3] G. Bergmann, M. Molnar, L. Gonczy, and B. Cousin, “Optimal period length for the CQS sensor network scheduling algorithm,” in Proc. 2010 International Conf. Netw. Services, pp. 192–199

[4] 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.