IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Multi-Level Priority Based Packet Scheduling Scheme For Wireless Sensor Network

P.Palanisamy[1], V.S.Sathish kumar[2]

Assistant Professor, Department of ECE, Nightingale Institute of Technology, Coimbatore, India[1]

Assistant Professor, Department of ECE, Nandha Engineering college, Erode, India[2]

Abstract: Scheduling different types of packets, such as real-time and non-real-time data packets, at sensor nodes with resource constraints in Wireless Sensor Networks (WSN) is of vital importance to reduce sensors energy consumptions and end-to-end data transmission delays. The main objective of packet scheduling scheme in wireless sensor network is to minimize the average end to end delay and waiting time of all real-time data packets and also non-real time data packets. Due to the minimization of these factors, we can get the minimum energy consumption. Then, the life time of battery will be increased. So here we were considering these factors for multiple levels and multiple zones. Scheduling different types of packets, such as real-time and non-real-time data packets, at sensor nodes with resource constraints in Wireless Sensor Networks is of vital importance to reduce sensors' energy consumptions and end-to-end data transmission delays.

Keywords: Wireless sensor network, packet scheduling, preemptive priority scheduling, non-preemptive priority scheduling, Dynamic Multilevel Priority scheduling, real-time, non-real-time, data waiting time, FCFS.

1. INTRODUCTION

1.1 OVERVIEW OF WIRELESS SENSOR NETWORKS

Wireless sensor networks are a trend of the past few years, and they involve deploying a large number of small nodes. The nodes then sense environmental changes and report them to other nodes over flexible network architecture. Sensor nodes are great for deployment in hostile environments or over large geographical areas.

Wireless sensor networks open up new application areas such as tactical surveillance, intelligent environmental and structural monitoring and target tracking . In a WSN, large numbers of tiny nodes (sensor motes) may be deployed in an ad hoc manner. These nodes automatically configure a topology by communicating and coordinating with each other. Nodes assume the roles of both sensing device and router. Messages are relayed to other nodes or to a hub in a multi-hop fashion.

Multi-hop routing in an energy-constrained WSN has been shown to give rise to significant gains in network performance. With more nodes, the area being monitored can be increased or with the same area, the increase in node density gives more precise and timely data and also provides a degree of operational reliability.

A wireless sensor network consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. They are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control. In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery.

A sensor node might vary in size from that of a shoebox down to the size of a grain of dust although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor node is similarly variable, ranging from hundreds of dollars to a few pennies, depending on the size of the sensor network and the complexity required of individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth. A sensor network normally constitutes a wireless ad-hoc network, meaning that each sensor supports a multi-hop routing algorithm (several nodes may forward data packets to the base station). In computer science and telecommunications, wireless sensor networks are an active research area with numerous workshops and conferences arranged each year.
Sensor networks have emerged as a promising tool for monitoring (and possibly actuating) the physical worlds, utilizing self-organizing networks of battery-powered wireless sensors that can sense, process and communicate. In sensor networks, energy is a critical resource, while applications exhibit a limited set of characteristics. The requirements and limitations of sensor networks make their architecture and protocols both challenging and divergent from the needs of traditional Internet architecture.

The basic goals of a WSN are to
(i) Determine the value of physical variables at a given location,

 DOI 10.17148/IJARCCE.2016.53267

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

(ii) Classify a detected object, and
(iii) Track an object.

The important requirements of a WSN are
(i) Use of a large number of sensors,
(ii) Attachment of stationary sensors,
(iii) Low energy consumption,
(iv) Self organization capability,
(v) Collaborative signal processing.

A sensor network is a network of many tiny disposable low power devices, called nodes, which are spatially distributed in order to perform an application-oriented global task. These nodes form network by communicating with each other either directly or through other nodes. One or more nodes among them will serve as sink(s) that are capable of communicating with the user either directly or through the existing wired networks.

The primary component of the network is the sensor, essential for monitoring real world physical conditions such as sound, temperature, humidity, intensity, vibration, pressure, motion, pollutants etc. at different locations. One or more nodes among them will serve as sink(s) that are capable of communicating with the user either directly or through the existing wired networks. The tiny sensor nodes, which consist of sensing, on board processor for data processing, and communicating components, leverage the idea of sensor networks based on collaborative effort of a large number of nodes. Figure 1.1 shows the structural view of a sensor network in which sensor nodes are shown as small circles.

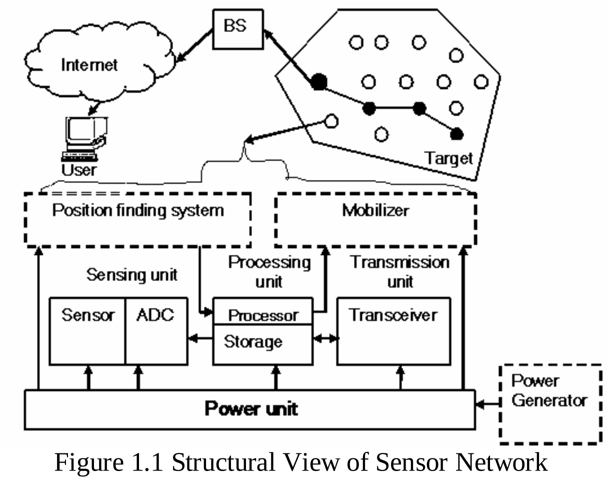

Figure 1.1 Structural View of Sensor Network

Each node typically consists of the four components: sensor unit, central processing unit , power unit, and Communication unit. They are assigned with different tasks. The sensor unit and Analog to Digital Converter. The sensor unit is responsible for collecting information as the ADC requests, and returning the analog data it sensed. ADC is a translator that tells the CPU what the sensor unit has sensed, and also informs the sensor unit what to do. Communication unit is tasked to receive command or query from and transmit the data from CPU to the outside world. CPU is the most complex unit. It interprets the command or query to ADC, monitors and controls power if necessary, processes received data, computes the next hop to the sink, etc.

Power unit supplies power to sensor unit, processing unit and communication unit. Each node may also consist of the two optional components namely Location finding system and mobilizer. If the user requires the knowledge of location with high accuracy then the node should pusses Location finding system and mobilizer may be needed to move sensor nodes when it is required to carry out the assigned tasks.

Instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data. The sensor nodes not only collect useful information such as sound, temperature, light etc., they also play a role of the router by communicating through wireless channels under battery-constraints.

Sensor network nodes are limited with respect to energy supply, restricted computational capacity and communication bandwidth. The ideal wireless sensor is networked and scalable , fault tolerance, consume very little power, smart and software programmable, efficient, capable of fast data acquisition, reliable and accurate over long term, cost little to purchase and required no real maintenance.

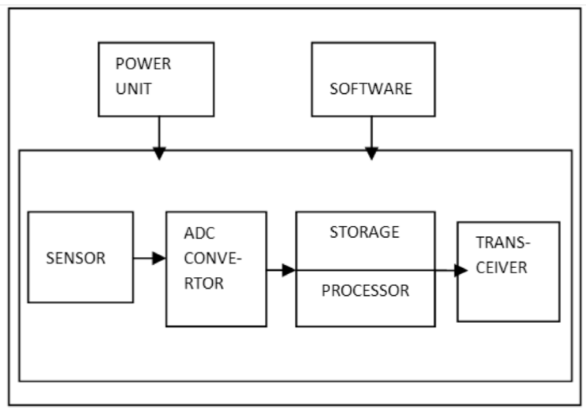

Figure 1.2 Sensor Node Architecture

2. RELATEDWORK

2.1 BALANCED ENERGY SLEEP SCHEDULING SCHEME FOR HIGH DENSITY CLUSTER-BASED SENSOR NETWORKS

Some sensor nodes may be put into the sleep state while other sensor nodes remain active for the sensing and communication tasks. However, determining which of the sensor nodes should be put into the sleep state is non-trivial. As the goal of allowing nodes to sleep is to extend network lifetime, we propose and analyze a Balanced-energy Scheduling scheme in the context of cluster based sensor networks. The BS scheme aims to evenly distribute the energy load of the sensing and communication tasks among all the nodes in the cluster, thereby extending the time until the cluster can no longer provide adequate sensing coverage. Two related sleep scheduling schemes, the Distance-based Scheduling scheme and the Randomized Scheduling scheme are also studied in terms of the coefficient of variation of their energy consumption.

DOI 10.17148/IJARCCE.2016.53267

Analytical and simulation results are presented to evaluate the proposed BS scheme. It is shown that the BS scheme extends the cluster's overall network lifetime significantly while maintaining a similar sensing coverage compared with the DS and the RS schemes for sensor clusters. The sleeping technique has been used to conserve energy of battery powered sensors. Rotating active and inactive sensors in the cluster, some of which provide redundant data, is one way that sensors can be intelligently managed to extend network lifetime. Some researchers even suggest putting redundant sensor nodes into the network and allowing the extra sensors to sleep to extend the network lifetime. This is made possible by the low cost of individual sensors. When a sensor node is put into the sleep state, it completely shuts itself down, leaving only one extremely low power timer on to wake itself up at a later time.1 This leads to the following Sleep Scheduling Problem: How does the cluster head select which sensor nodes to put to sleep, without compromising the sensing coverage capabilities of the cluster? we generalized and proposed two sleep scheduling schemes, termed the Randomized Scheduling scheme and the Distance-based Scheduling scheme. In the RS scheme, sensor nodes are randomly selected to go into the sleep state.

In the DS scheme, the probability that a sensor node is selected to sleep depends on the distance it is located from the cluster head. One possible drawback of the RS and the DS schemes is that the average energy consumptions of sensors with different distance to the cluster head might be different. In the RS scheme, all the sensor nodes in the cluster have the same sleep probability even though the sensor nodes on the border of the cluster may consume more energy than others.

The main objective of sleeping technique has been used to conserve energy of battery powered sensors. Rotating active and inactive sensors in the cluster, some of which provide redundant data, is one way that sensors can be intelligently managed to extend network lifetime. Some researchers even suggest putting redundant sensor nodes into the network and allowing the extra sensors to sleep to extend the network lifetime.

2.2 COVERAGE AND DETECTION OF A RANDOMIZED SCHEDULING ALGORITHM IN WIRELESS SENSOR NETWORKS

In wireless sensor networks, some sensor nodes are put in sleep mode while other sensor nodes are in active mode for sensing and communication tasks in order to reduce energy consumption and extend network lifetime. This approach is a special case ($k = 2$) of a randomized scheduling algorithm, in which k subsets of sensors work alternatively. In this paper, we first study the randomized scheduling algorithm via both analysis and simulations in terms of network coverage intensity, detection delay, and detection probability. We further study asymptotic coverage and other properties. Finally, we analyze a problem of maximizing network lifetime under Quality of Service constraints such as bounded detection delay, detection probability, and network coverage intensity.

We prove that the optimal solution exists, and provide conditions of the existence of the optimal solutions. This paper focuses on performance modelling and mathematical properties of a random coverage algorithm (also called k-set randomized scheduling algorithm) for WSNs. The algorithm is designed as follows: Let S denote the set including all the sensor nodes in a WSN. Each sensor node is randomly assigned to one of k disjoint subsets (S_j, $j=1; 2, . . . , k$), which work alternatively. In other words, at any time, only one set of sensor nodes are working, and the rest of sensor nodes sleep. Network lifetime is the elapsed time during which the network functions well, and the formal definition is given in a later section. In case that there is an intrusion such as an enemy tank invading a field covered with sensor nodes.

The main objective of this paper is to study network coverage intensity, asymptotic coverage intensity, detection probability, and detection delay. And analyze the problem of maximizing network lifetime under QoS constraints such as the bounded detection delay, detection probability, and coverage intensity. Then study the properties and asymptotic properties, disclose that the optimal solution exists, and present the conditions of the existence of the optimal solutions.

2.3 WAKEUP SCHEDULING IN WIRELESS SENSOR NETWORKS

A large number of practical sensing and actuating applications require immediate notification of rare but urgent events and also fast delivery of time sensitive actuation commands. In this paper, Authors consider the design of efficient wakeup scheduling schemes for energy constrained sensor nodes that adhere to the bidirectional end-to-end delay constraints posed by such applications. We evaluate several existing scheduling schemes and propose novel scheduling methods that outperform existing ones. We also present a new family of wakeup methods, called multi-parent schemes, which take a cross-layer approach where multiple routes for transfer of messages and wakeup schedules for various nodes are crafted in synergy to increase longevity while reducing message delivery latencies.

We analyze the power-delay and lifetime-latency tradeoff’s for several wakeup methods and show that our proposed techniques significantly improve the performance and allow for much longer network lifetime while satisfying the latency constraints Scheduled wakeups: In this class, the nodes follow deterministic (or possibly random) wakeup patterns. Time synchronization among the nodes in the network is generally assumed. However, asynchronous wakeup mechanisms which do not require synchronization among the different nodes are also categorized in this class. Although asynchronous methods are simpler to implement, they are not as efficient as synchronous schemes, and in the worst case their guaranteed delay can be very long. Wakeup on-demand (out-of-band wakeup): It is assumed that the nodes can be signalled and awakened at any point of time and then a message is sent to the node.

This is usually implemented by employing two wireless interfaces. The radio is used for data communication and is triggered by the second ultra low-power (or possibly passive) radio which is used only for paging and signalling. STEM and its variation , and passive radio-triggered solutions are examples of this class of wakeup methods. Although these methods can be optimal in terms of both delay and energy, they are not yet practical. The cost issues, currently limited available hardware options which results in limited range and poor reliability, and stringent system requirements prohibit the widespread use and design of such wakeup techniques. Consequently, there is a need for efficient scheduled wakeup schemes which are reliable and cost-effective and can also guarantee the delay and Lifetime constraints. In this paper, we focus on the synchronous scheduled wakeup methods which provide bidirectional delay guarantees. We analyze and compare the existing methods and introduce new efficient wakeup methods that outperform the existing ones. We present a novel class of wakeup methods called multi-parent schemes which assign multiple parents (forwarding nodes) with different wakeup schedules to each node in the network. This method takes a cross-layer approach and exploits the existence of multiple paths between the nodes in the network to significantly improve the energy efficiency of wakeup process and therefore increase the lifetime of the network while meeting the message delay constraints. We derive the best-case, worst-case, and generally the distribution of delay for many existing and our new wakeup schemes, and also characterize the trade-off between power consumption (or lifetime) and guaranteed delay for many different wakeup mechanisms.

The main objective is to analyze different wakeup schemes and obtained their delay distribution and delay-power trade-off curves. All existing wakeup patterns such as synchronized and staggered patterns are considered and we also introduced new efficient wakeup patterns such as crossed-ladders pattern which outperforms other methods. We also presented the new cross-layer idea, called multi-parent technique, where by assigning multiple parents with different wakeup schedules to each node in the network, significant performance improvement is achieved.

2.4 SLEEP SCHEDULING FOR CRITICAL EVENT MONITORING IN WIRELESS SENSOR NETWORKS

Critical Event Monitoring in wireless sensor networks, where only a small number of packets need to be transmitted most of the time. When a critical event occurs, an alarm message should be broadcast to the entire network as soon as possible. To prolong the network lifetime, some sleep scheduling methods are always employed in WSNs, resulting in significant broadcasting delay, especially in large scale WSNs. In this paper, we propose a novel sleep scheduling method to reduce the delay of alarm broadcasting from any sensor node in WSNs. Specifically, we design two determined traffic paths for the transmission of alarm message, and level-by-level offset based wake-up pattern according to the paths, respectively. When a critical event occurs, an alarm is quickly transmitted along one of the traffic paths to a center node, and then it is immediately broadcast by the center node along another path without collision.

Therefore, two of the big contributions are that the broadcasting delay is independent of the density of nodes and its energy consumption is ultra low. Exactly, the upper bound of the broadcasting delay is only 3D+2L, where D is the maximum hop of nodes to the center node, L is the length of sleeping duty cycle, and the unit is the size of time slot. Extensive simulations are conducted to evaluate these notable performances of the proposed method compared with existing works. As sensor nodes for event monitoring are expected to work for a long time without recharging their batteries, sleep scheduling method is always used during the monitoring process. Obviously, sleep scheduling could cause transmission delay because sender nodes should wait until receiver nodes are active and ready to receive the message.

The delay could be significant as the network scale increases. Therefore, a delay-efficient sleep scheduling Method needs to be designed to ensure low broadcasting delay from any node in the WSN. To minimize the broadcasting delay, it is needed to minimize the time wasted for waiting during the broadcasting. The ideal scenario is the destination nodes wake up immediately when the source nodes obtain the broadcasting packets. Here, the broadcasting delay is definitely minimum. Based on this idea, a level-by-level offset schedule was proposed. The packet can be delivered from node a to node c via node b with minimum delay. Hence, it is possible to achieve low transmission delay with the level-by-level offset schedule in multi hop WSNs. For the critical event monitoring in a WSN, sensor nodes are usually equipped with passive event detection capabilities that allow a node to detect an event even when its wireless communication module is in sleep mode. Upon the detection of an event by the sensor, the radio module of the sensor node is immediately woken up and is ready to send an alarm message. Time of sensor nodes in the proposed scheme is assumed to be locally synchronous, which can be implemented and maintained with periodical beacon broadcasting from the center node. Hence, it is possible to achieve low transmission delay with the level-by-level offset schedule in multi hop WSNs.

3. INTRODUCTIONS TO PACKET SCHEDULING

Among many network design issues, such as routing protocols and data aggregation, that reduce sensor energy consumption and data transmission delay, packet scheduling (interchangeably use as task scheduling) at sensor nodes is highly important since it ensures delivery of different types of data packets based on their priority and fairness with a minimum latency. For instance, data sensed for real-time applications have higher priority than data sensed for non-realtime applications. Though extensive research for scheduling the sleep-wake times of sensor nodes has been conducted only a few studies exist in the literature on the packet scheduling of sensor nodes.

DOI 10.17148/IJARCCE.2016.53267

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

That schedule the processing of data packets available at a sensor node and also reduces energy consumptions. Indeed, most existing Wireless Sensor Network operating systems use First Come First Serve 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.

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.

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 predetermined and static, and cannot be changed in response to a change in the application requirements or environments. For example, in many real-time applications, a real-time priority scheduler is statically used and cannot be changed during the operation of WSN applications. Furthermore most existing packet scheduling algorithms of WSN are neither dynamic nor suitable for large scale applications since these schedulers.

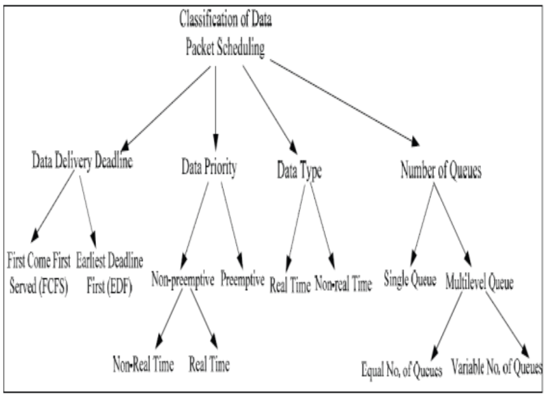

Figure 3.1 Classification of Packet Scheduling Schemes

3.1 PROPOSED SYSTEM

A proposed Circular Wait and Preemptive 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. This is because we 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 scheduler scheme since it is very efficient in terms of average task waiting time.

3.2 DEADLINE

Packet scheduling schemes can be classified based on the deadline of arrival of data packets to the base station, which are as follows.

First Come First Served: Most existing WSN applications use First Come First Served 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 but data from nearby neighbouring 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: 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 research work of a 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 FCFS. 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 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.

3.3 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 t carries on even if a higher priority packet than the currently running packet t. It arrives at the ready queue. Thus t has to wait in the ready queue until the execution of t 1.

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.

Packet scheduling mechanisms that are used in 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, 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 are suitable for applications with limited 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 scheme. EF-RM is an extension to Rate Monotonic, a static priority scheduling, whereby the shortest-deadline job has the highest priority. EF-RM divides WSN tasks into Period Tasks, 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.

3.4 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.

Though packet scheduling mechanisms are simple and are used extensively in sensor nodes, they cannot be applied to all applications: due to the long execution time of certain data packets, real-time packets might be placed into starvation. Moreover, the data queue can be filled up very quickly if local data packets are more frequent that causes the discard of real-time packets from other nodes.An improved priority-based soft real-time packet scheduling algorithm. Schedulers traverse the waiting queue for the data packets and choose the smallest packet ID as the highest priority to execute. Each packet is assigned an Execute Counter, EXECUTE MAX TIME, i.e., the largest initial task execution time. The management component compares the current packet ID with the previous packet ID. If it is the same, the system executes it and decrements the counting variable. Otherwise, if the counting variable is null, the management component terminates this packet and other packets get the opportunity to be executed. packet priorities are decided during the compilation phase, which cannot be changed during the execution time. If high priority packets are always in execution, the low priority packets cannot be implemented. If low-priority packets occupy the resources for a long time, the subsequent high-priority packets cannot get response in time.

3.5 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 . the main concept behind multi-level queue scheduling algorithms.

A multilevel queue scheduler scheme that uses a different number of queues according to the location of sensor nodes in the network. This approach uses two kinds of scheduling: simple priority-based and multi-FIFO queue-based. In the former, data enter the ready queue according to priority but this scheduling also has a high starvation rate. The multi-FIFO queue is divided into a maximum of three queues, depending on the location of the node in the network. If the lowest level is , nodes that are located at level have only one queue but there are two queues for nodes at level . Each queue has its priority set to high, mid, or low. When a node receives a packet, the node decides the packet’s priority according to the hop count of the packet and accordingly sends it to the relevant queue. A priority queue scheduling algorithm for WSN. In this scheduling scheme, buffer space of intermediate nodes is divided into four queues to hold three different types of video frames and one regular data frames. Data in the first three queues have the highest priority and are scheduled in round robin fashion. Data in the fourth queue is transmitted when the first three queues are empty. However, these scheduling schemes do not consider variable number of queues based on the position of sensor

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

nodes to reduce the overall end-to-end delay. If the lowest level is , nodes that are located at level have only one queue but there are two queues for nodes at level . Each queue has its priority set to high, mid, or low. When a node receives a packet, the node decides the packet's priority according to the hop count of the packet and accordingly sends it to the relevant queue.

4. WORKING PRINCIPLE

Scheduling data packets among several queues of a sensor node is presented in Figure 3.2. 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 3.2 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 scheduling scheme is illustrated in Figure 3.3. 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.

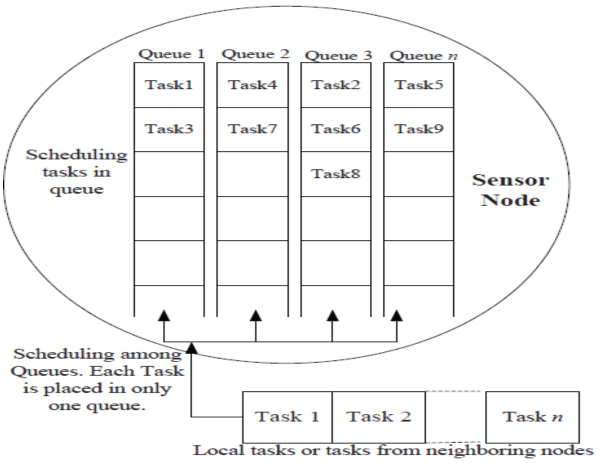

Figure 4.1 Scheduling data among multiple queues.

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. We consider three-level of queues, that is, the maximum number of levels in the ready queue of a node is three: priority 1 (pr1), priority 2 (pr2), and priority 3 (pr3) queues. Real-time data packets go to pr1, the highest priority queue, and are processed using FCFS. Non-real-time data packets that arrive from sensor nodes at lower levels go to pr2, 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 preempting pr2 tasks if pr3 tasks wait a number of consecutive timeslots. In the proposed scheme, 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 preemptable priority schedulers. The idea behind this is that the highest-priority real-time/emergency tasks rarely occur. They are thus placed in the preemptive priority task queue (pr1 queue) 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, nonreal- time packets that arrive from the sensor nodes at lower level are placed in the preemptable priority queue (pr2 queue). 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.

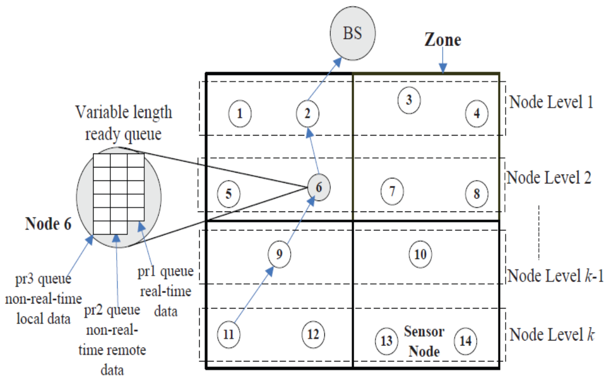

Figure 4.2 Proposed Packet Scheduling

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. Moreover, 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.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

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.

4.1 METHODOLOGY

- We present a new MAC protocol, which is referred to as hybrid MAC, which is suitable for WSNs in terms of energy efficiency, latency, and design complexity. HMAC combines channel-allocation schemes from existing contention-based and time-division multiple-access based MAC protocols to allow the realization of tradeoffs between different performance metrics.
- It uses a short slotted frame structure and a novel wakeup scheme to achieve high-energy performance, low delivery latency, and improved channel utilization.
- Our proposed protocol combines energy-efficient features of the existing contention-based and time-division multiple access based MAC protocols and adopts a short frame structure to expedite packet delivery.
- HMAC is simple and scalable since each node does not have to maintain neighbourhood information.
- HMAC provides routing layer coarse-grained quality-of-service support at the MAC layer. To the best of our knowledge, very few existing MAC layer works handle such QoS issues in WSNs.
- Quality of service-aware medium access control assigns each flow a channel-access priority to reduce the queuing delay for high-priority flows but it still suffers from a long end-to-end delay.
- The MAC protocols presented in reduce the end-to-end delivery latency while increasing control overhead without considering different performance demands between flows.

4.2 PSEUDOCODE FOR SCHEDULING A SLOT

1. for all each link $(i, j) \in E[G]$ **do**
2. $F[i, j] \leftarrow 0$
3. $S[i, j] \leftarrow 0$
4. end for
5. $Q[G] \leftarrow$ NIL
6. while More than one link $(i, j) \in E[G]$ **do**
7. **while** More than one link $(i, j) \in E[G]$where $S[i, j] = 0$
8. do
9. Randomly select a link $(i, j) \in E[G]$ such that
10. $F[i, j] = 0$ and $S[i, j] = 0$
11. Add link (i, j) to $Q[G]$
12. UPDATE NETWORK CONFIGURATION(G,E,Q,S)
13. end while
14. Select a link $(i, j) \in Q[G]$ such that $D[i, j]/R[i, j]$ is minimal
15. **for** each required slot m in Mi do
16. Try assigning slot s = 1;
17. **while** any of the 3 interference criteria is not
18. satisfied **do**
19. Try assigning the next slot s[i,j] = s + 1;
20. end while
21. Assign slot s to required slot m of node i;
22. end for
23. $F[i, j] \leftarrow 0$ and $D[i, j] \leftarrow 0$
24. for all link $(m, n) \in Q[G]$ where $D[m, n]$ _= 0 **do**
25. $D[m, n] \leftarrow D[m, n] - D[i, j] \times R[m, n]/R[i, j]$
26. end for
27. UPDATE NETWORK CONFIGURATION(G,E,Q,S)
28. end while

- $S[i, j]$ is a link-blocking parameter that indicates whether link (i, j) is interfered by any other active links
- $F[i, j]$ indicates whether the demand of link (i, j) is already scheduled in the current schedule period link (i, j) is delivering datawith the traffic demand $D[i, j]$
- $U[G]$ is a subset of the nodes whose demands have not been satisfied
- $P[G]$ is a subset of the wireless links along the selected path
- A wireless link (i, j) is an element of $E[G]$ if and only if node i and node j are within the maximum transmission range of each other
- Graph –G
- E----element
- $Q[G]$ be the subset of the links whose demands are satisfied.

5. PERFORMANCE EVALUATION

5.1 SIMULATION RESULTS AND ANALYSIS

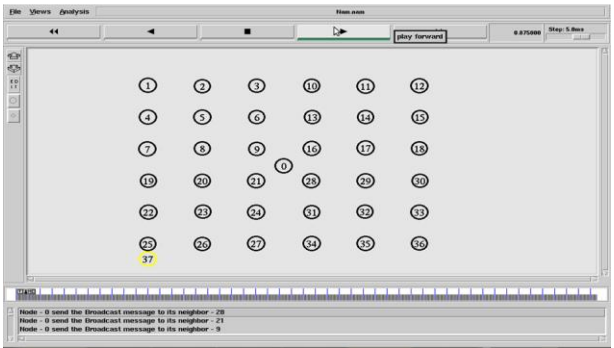

Figure 5.1 Node Creation

In node creation, we are considering a group of nodes. These group of nodes are arranged in level by level fashion in each zone. So here we are selecting one cluster head based on highest energy level. Each node in a zone will be having a communication with it's cluster heater. Node configuration essentially consists of defining the different node characteristics before creating them. They may consist of the type of addressing structure used in the simulation, defining the network components for mobile nodes, turning on or off the trace options at Agent/Router/MAC levels, selecting the type of adhoc routing protocol for wireless nodes or defining their energy model.

Figure 5.2 Shows the node communication. In which the chosen cluster header is node0. Which is indicated as Base

DOI 10.17148/IJARCCE.2016.53267

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Station (BS). So here the routing is takes place based on Hop counting. Each node in the cluster will sending Hello message to each other to conform the node status. If the node receives Hello message then the node is in ON state. If not so then node will be in OFF state.

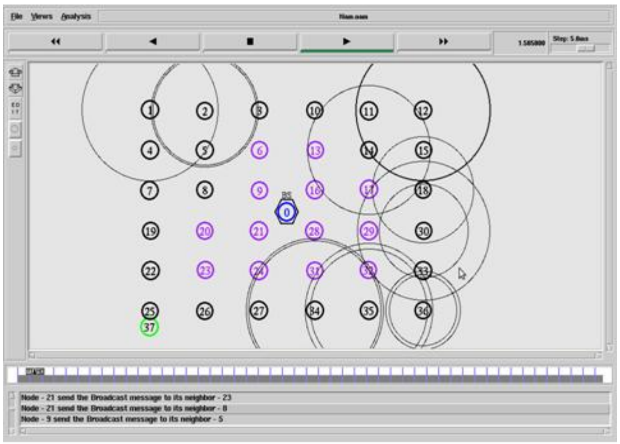

Figure 5.2 Node Communication

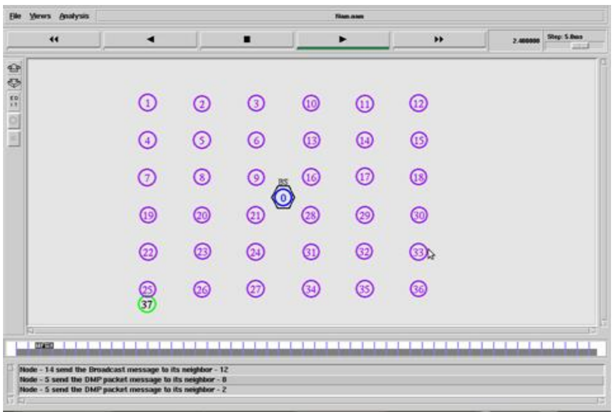

Figure 5.3 Active Nodes

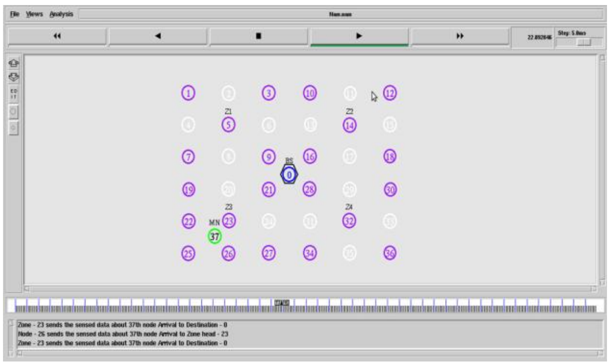

Figure 5.4 Zone Head Selection

Figure 5.3 shows the active nodes in the cluster. So totally we have taken 38 nodes. In which the cluster head is indicated as Base Station that is node 0. Base Station will known about all nodes information. Figure 4.5 shows the Zone Head Selection. We consider four zones. Each zone having nine nodes including zone head. Zone head directly contacted with the base station. When the node detects event then it will forwards to the zone head. So the zone head will forwards to the base station. When there is no event accrues for long time means the node will be put into sleep. Then it will wake up when event accrue.

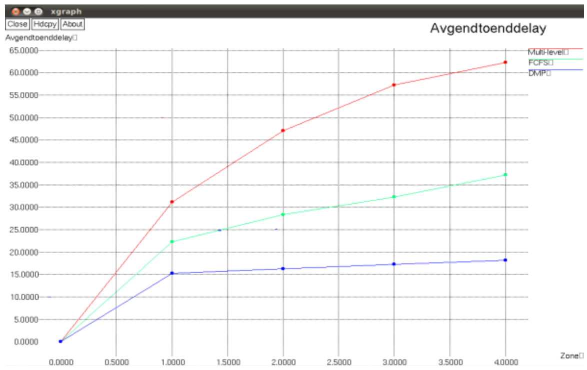

Figure 5.5 Average end to end delay

Figure 5.5 shows the average end to end delay. So here we are considered number of zones for calculating the average end to end delay. In this graph we compared our method DMP with the mechanisms FCFS and Multi-level. So our method produces the better result than the other two mechanisms. It illustrates the end-to-end data transmission delay of real-time tasks over a number of zones. We observe that the proposed DMP scheduling scheme outperforms the existing FCFS, and Multilevel Queue scheduler. This is because the proposed scheduling scheme gives the highest priority to real-time tasks and also allows real-time data packets to preempt the processing of non-real time data packets. Thus, real-time data packets have lower data transmission delays.

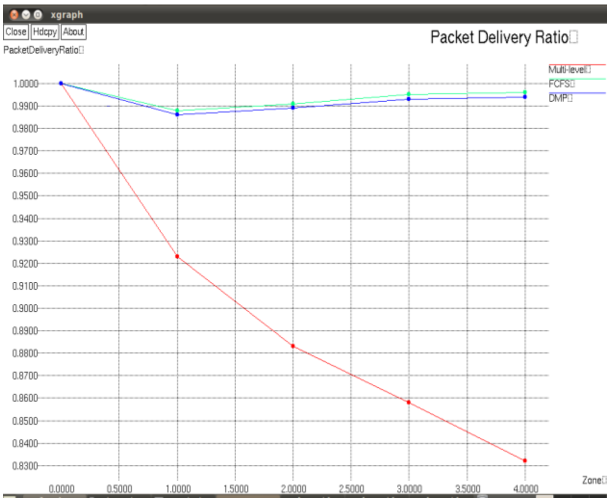

Figure5.6 Packet Delivery Ratio

Packet delivery ratio is shown in Figure 5.6 Packet delivery ratio measures the percentage of data packets generated by nodes that are successfully delivered, expressed as

$$\frac{\text{Total number of data packets succesfully delivered}}{\text{Total number of data packets sent}} \times 100\%$$

Figure 5.7 shows the Control overhead. We compared our proposed DMP packet Scheduling scheme with different methods like FCFS and Multi-level algorithms. When we compare we have better result. Control overhead is nothing but RTS and CTS and Acknowledgement like that. So which is some pre processing before starting to communicate.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

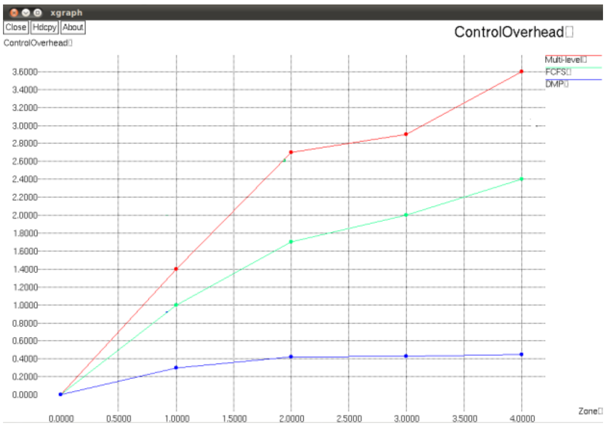

Figure 5.7 Control overhead

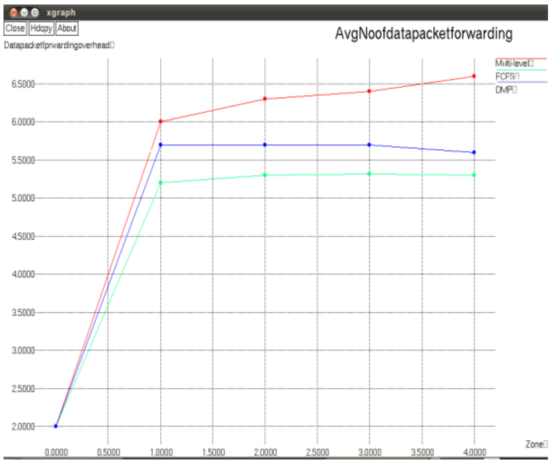

Figure 5.8 Average Number of data Packets Forwarding

Figure 5.8 shows the Average number of data packets forwarding from each node to the cluster head and not to be received successfully at the cluster head. We have calculated it for multiple zones and compared with previous algorithms.

6. CONCLUSION

In this paper, we propose a Multilevel Priority Based packet scheduling scheme for Wireless Sensor Networks. The scheme 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 the highest priority data while exhibiting acceptable fairness towards lowest-priority data. Experimental results show that the proposed 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. The proposed scheme, we envision assigning task priority based on task deadline instead of the shortest task processing time. To reduce processing overhead and save bandwidth, we could also consider removing tasks with expired deadlines from the medium. Furthermore, 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.

This project may have the way for the evolution of new methods for multi-target tracking based on other optimization techniques that may be more energy and power efficient than this method.

REFERENCES

[1] A. R. Swain, R. C. Hansdah, and V. K. Chouhan, "An energy aware routing protocol with sleep scheduling for wireless sensor networks," in Proc. 2010 IEEE International Conf. Adv. Inf. Netw. Appl., pp. 933–940.

[2] B. Zeng, Y. Dong, and D. Lu, "Cooperation-based scheduling algorithm in wireless multimedia sensor networks," in Proc. 2011 International Conf. Wireless Commun., Netw. Mobile Comput., pp. 1–4.

[3] 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.

[4] F. Liu, C. Tsui, and Y. J. Zhang, "Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks," IEEE Trans. Wireless Commun., vol. 9, no. 7, pp. 2258–2267, July 2010.

[5] J. Liu, N. Gu, and S. He, "An energy-aware coverage based node scheduling scheme for wireless sensor networks," in Proc. 2008 International Conf. Young Comput. Scientists, pp. 462–468.

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

[7] 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.

[8] O. Khader, A. Willig, and A. Wolisz, "Distributed wakeup scheduling scheme for supporting periodic traffic in wsns," in Proc. 2009 European Wireless Conf., pp. 287–292.

[9] 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.

[10] X. Yu, X. Xiaosong, and W. Wenyong, "Priority-based low-power task scheduling for wireless sensor network," in Proc. 2009 International Symp. Autonomous Decentralized Syst., pp. 1–5.

[11] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in Proc. 4th ACM Int. Symp. Mobihoc, Annapolis, MD,Jun. 2003, pp. 35–45.

[12] X. Yang and N. H. Vaidya, "A wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay," in Proc.IEEE 10th RTAS, Toronto, Canada, May 2004, pp. 19– 26.

[13] M. Peng and W. Wang, "An adaptive energy saving mechanism in the wireless packet access network," in Proc. IEEE WCNC, Las Vegas, NV, Apr. 2008, pp. 1536–1540.

[14] R. Yu, Y. Zhang, Z. Sun, and S. Mei, "Energy- and QoS-aware packet forwarding in wireless sensor networks," in Proc. IEEE ICC, Glasgow, U.K., Jun. 2007, pp. 3277–3282.

[15] A. O. Fapojuwo, "Energy consumption and message delay analysis of QoS-enhanced base-stationcontrolled dynamic clustering protocol for wireless sensor networks," IEEE Trans. Wireless Commun., vol. 8, no. 10, pp. 5366–5374, Oct. 2009.246810121400.20.40.60.811.21.41.61.82x 104No of zonesReal time Task End to End delay DMPDRR

[16] Z. Ye, A. A. Abouzeid, and J. Ai, "Optimal stochastic policies for distributed data aggregation in wireless sensor networks," IEEE/ACM Trans. Netw., vol. 17, no. 5, pp. 1494–1507, Oct. 2009.

[17] 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.

[18] 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.

[19] 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.

DOI 10.17148/IJARCCE.2016.53267

[20] F. Liu, C. Tsui, and Y. J. Zhang, "Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks," IEEE Trans.Wireless Commun., vol. 9, no. 7, pp. 2258–2267, July 2010.
[21] J. Liu, N. Gu, and S. He, "An energy-aware coverage based node scheduling scheme for wireless sensor networks," in Proc. 2008 International Conf. Young Comput. Scientists, pp. 462–468.
[22] O. Khader, A. Willig, and A. Wolisz, "Distributed wakeup scheduling scheme for supporting periodic traffic in wsns," in Proc. 2009 European Wireless Conf., pp. 287–292.
[23] B. Nazir and H. Hasbullah, "Dynamic sleep scheduling for minimizing delay in wireless sensor network," in Proc. 2011 Saudi International Electron., Communications Photon. Conf., pp. 1–5.
[24] 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.

BIOGRAPHIES

Mr. P. Palanisamy working as Assistant Professor in Nightingale Institute of Technology, Coimbatore, India. Completed his Master of Engineering under Anna University in Communication Systems (2014) in Sriguru Institute of Technology, Coimbatore, T.N, India. Received Bachelor of Engineering (2012) under Anna University in Electronics and Communication Engineering in V.S.B Engineering College, Karur, T.N, India.

Mr.V.S. Sathishkumar working as Assistant Professor in Nandha Engineering college, Erode, India. Completed his Master of Engineering under Anna University in Applied Electronics (2015) in Surya Engineering College, Erode, T.N, India. Received Bachelor of Engineering (2012) under Anna University in Electronics and Communication Engineering in V.S.B Engineering College, Karur, T.N, India.