

Energy-Latency Improved Sensor Networks Using Mobile Agents in Textile Industry

G. Sundari and P. E. Sankaranarayanan

Sathyabama University, Indi

sundari.sundarc@rediffmail.com, drpesanky37@hotmail.com

Abstract

The key performance metrics in wireless sensor networks are both network lifetime and an average time required to report an event reliably. The optimal solution must be taken into account to meet these metrics. Considering energy-latency constraints in sensor networks, we have developed a cluster architecture using mobile agents to detect the yarn break in spinning machine in textile industry. We have used the latency improved MAC layer protocol to access the channel in the cluster and also used energy efficient cluster formation algorithm based on residual energy of the sensor node. This protocol can handle multiple event messages efficiently. And, we mainly focused on simulating the behavior of mobile agents for information retrieval from a wireless sensor network in a event driven and poll driven wireless sensor network model.

Keywords: *Latency, MAC protocol, Energy efficiency, Sensor node, Mobile agent*

1. Introduction

Wireless sensor networks (WSN) are gaining popularity for industrial sensing applications due to their relatively low cost and simplicity for retrofitting into existing infrastructure. WSN is a network consisting of distributed self-organized autonomous devices built with sensors to monitor physical and environmental conditions such as vibration, motion, temperature and sound in a coordinated fashion. It is a group of specialized transducers with a communication infrastructure intended to monitor and record conditions at diverse locations[1]. They are particularly suited to power engineering applications as they do not require cabling, which will lead to shorter outages during installation and a lower capital outlay than their wired equivalent. In emerging Wireless Sensor Networks, including Sensor Webs, a 'deploy and ignore' approach is no longer possible[2]. Indeed, the conventional 'built-in' station is replaced by a large number of lower-power motes which often conserve their resources by having sleep/wake cycles. In addition, in mobile data collection, communication pathways and network topologies regularly change. Consequently, monitoring must be actively and continually updated. Especially in remote areas where manual maintenance is nearly impossible, monitoring must be carried out by the network in a way that it can detect events of interest and selfconfigure quickly and efficiently in order to collect and forward data to sinks.

Generally, there are three models of wireless sensor network: continuous, on-demand and event-driven. In continuous WSN, sensors send data periodically to the sink. There are always sensors in the network, which initiate the communication. In the on-demand WSN model, sensors send data only when they receive a request from a access point. Without request, sensors sense the information and store it in their local memory. In the event driven

WSN, in which we are interested, the sensors send data only when an event occurs[3]. We made an attempt to use an mobile agent-based system that will provide the abstraction for the different technologies used in spinning machine. Here the agent has the ability to communicate with the sensors in the environment, learn about the status of all sensors, analyze the decisions, and acts autonomously. The agent acts as a mediator between the administrator and the sensor network environment. This agent provides real-time solutions for energy-efficient query processing and data dissemination. The agent-based architecture will use a database approach where the users will be unaware of the implementation details. In other words, users would be able to query the network without worrying about the energy efficient query processing and routing techniques. The query and routing agents would be responsible for query processing and data dissemination respectively. Moreover, the application developers could easily build a standard application for the given sensors, without going into the hardware, routing, or sensor details[4]. Figure 1 shows the client-server based and mobile agent based architecture. Shaded circles represents the nodes.

In this paper, section 2 states the problem description. Section 3 describes some related MAC protocols and the energy efficient clustering formation algorithm, Section 4 details the background of mobile agents and the proposed model. In section 5, we showed the simulation results and section 6 concludes the paper.

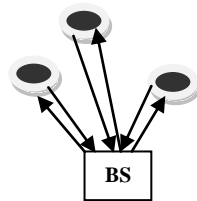


Fig 1a. Client-server based

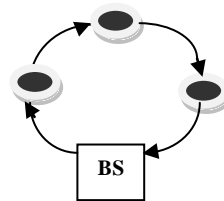


Fig 1b. Mobile Agent based

2. Problem Description

Detecting the yarn break in a spinning machine in a textile industry is the important task as it directly affects the production efficiency. The end breakage in ring spinning not only reduces the running efficiency of the process, but also deteriorates the quality of the yarn in terms of presence of piecing slubs. With the increase in spindle speed, the yarn breakage rate increases. A higher speed causes powerful centrifugal forces on the fibres resulting in poor spinning stability and an increase in yarn tension, which in turn causes high yarn breakage. An appropriate sensor network is always required to monitor the system behavior in such a way that suitable signal processing and damage-sensitive feature extraction on the measured data can be performed efficiently. The number of sensing systems available vary significantly depending upon the activity. Current sensing network in spinning machine uses wired concept. Wireless communication can remedy the cabling issues with the traditional monitoring system and significantly reduce the maintenance costs. A new wireless sensor network (WSN) paradigm, which has been investigated by the authors, is then presented.

3. MAC Protocols and Clustering Algorithm

Research on the medium access control of wireless sensor networks is very common. There is a need to improve MAC protocol management of communication time between sensors which consumes the most energy. MAC protocols are classified into two different types: Contention-Based and Contention-Free. Contention-free MAC is based on reservation

and scheduling[5]. Here, each node announces a time slot that they want to use to the coordinator of the network. This coordinator schedules the request and allocates to other nodes their respective time slots. In this way, a node can access the channel without colliding with others because it is the only node which can transmit during its time slot. This technique guarantees low energy consumption because each node in the network works only in its time slot without collisions. However, the main drawback of this technique is, the nodes must be well synchronized among them (about several μs), which is not easy to achieve in the widely distributed and scalable environment of a sensor network. Unlike this technique, contention-based MAC is a protocol where every node competes to access the channel. Before transmitting a message, a node listens to the channel to see whether there is already a transmission in the medium. If the channel is busy, it will wait for a random time and retry to detect it later. If the channel is free, it will transmit the message. Collision occurs when two or more interfering nodes observe that the channel is free at the same time and they transmit their message simultaneously[6]. The most well-known example of this technique is protocol IEEE 802.11 for wireless LAN network. Many research projects have been carried out to optimize the existing MAC methods. S-MAC[7] is considered to be the first MAC protocol proposal which tries to reduce idle time for sensors, it is not suitable for our event driven application. TMAC tries to improve SMAC by using a time out scheme to further reduce the idle listening wastage. DMAC[8] uses an improved staggered sleeping cycle to reduce the data latency of SMAC. Latency MAC protocol[9] sacrifices some packet latency to reduce the energy wastage of idle listening and to increase the accuracy of the prediction for packets incoming. Sift[10] is the MAC protocol specific to event-driven WSN and guarantees successful transmission of messages in the shortest delay but the distribution function is complicated and requires a lot of processing to compute the non-uniform probability for each transmission. So we have adapted Sift to improve the latency and to meet our event driven application.

Our work is motivated by the following observations: In most sensor networks, multiple sensors are deployed in the same geographic area, usually for fault-detection and reliability. In addition to sensing periodic observations, when an event of interest happens, the sensing node that senses the event send messages reporting the event. The result is spatially-correlated contention. Multiples sensors sharing the wireless medium all have messages to send at almost the same time because they all generate messages in response to external events. In our work, each node is deployed across each spindle to monitor the rotation of the traveler. So, only a single node is responsible for sending the event of interest.

In all contention window based protocols, each node picks a random contention slot in $[1, CW]$ to transmit. If two nodes pick the same slot, they will both transmit and cause a collision. When this happens, most protocols specify that the colliding nodes multiplicatively increase their value of CW. In contract to this, sift protocol uses a small and fixed CW of 32 slots, where each slot lasts on the order of tens of microseconds. In a shared medium, where R nodes sense an event and contend to transmit on the channel at the same time, we design a MAC protocol that minimizes the time taken to send R messages without any collision. The key idea is to use an increasing non-uniform probability distribution with in a fixed contention window, rather than varying the contention window size as in many traditional MAC protocols. If exactly one node happens to pick some contention slot, it will start to transmit in that slot. When its transmission finishes, all other competing nodes select new random contention slots and repeat the process of backing off over the fixed contention window. The same process happens if two or more stations happen to pick the same contention slot resulting in collision. So in our method, a node with larger report information is delayed by the other node.

3.1. Energy Efficient Cluster Formation Algorithm

Energy consumption of the cluster heads is relatively expensive, so the residual energy of sensor node is the main criteria for the election of cluster head[11][12]. The closer sensor nodes within a cluster, the lower energy they need to consume to send the data. Considering a sensor network which is randomly distributed over a region with a sink node. The sink node is located outside of the monitoring area with infinite energy. All these sensor nodes are grouped into clusters. All sensor nodes are assigned with an ID(identification), computing power and processing power. Initially some nodes are randomly chosen as cluster heads and a level value 0 is assigned to them along with their ID[13].

$$CH=(ID, \text{level } 0)$$

During each round, each sensor node calculates its own “concentration degree” defined to be the concentration degree of node i, namely the number of sensor nodes that it can sense during the rth round.

The residual energy $E_{rd(i)}$ of a node i is calculated as $E_{rd(i)} = \text{initial power} - (\text{total power consumed for transmission and reception})$. Sink node marks itself as level 1, and broadcast a message to neighbor nodes to construct a routing among clusters, the message is sent with “appointment of cluster heads” messages. The cluster heads one hop away from the sink will receive the message, and then marks itself as level 2, mark the ID as its father cluster head ID. Cluster head receiving the message will mark the node ID, from which the message is sent as its father cluster ID and use the level value in the message plus 1 as its new level value if its original level value is 0. The routing construction process will end when there is no level 0 cluster head in the sensor network. In the initialization phase of network, the sink broadcasts $E_{rd(r)}$, the average energy of sensor nodes in ‘cluster head election’ messages. When a node i, receives the broadcast message, it will first compare its own residual energy $E_{rd(i)}$ with $E_{rd(r)}$.

$$\begin{aligned} & \text{if} \\ & E_{rd(r)} < E_{rd(i)} \\ & \{ \\ & W_{(j,r)} = \alpha [E_{rd(i)} / E_{rd(r)}] + (1-\alpha) [D_{(j)} / ((N-1)/K)] \\ & \beta = E_{rd(r)} / E_{rd(i)} \\ & \alpha = 1 / (1 + \beta) \\ & \} \end{aligned}$$

α is adaptive factor

β is the residual energy of node j in round r

$W_{(j,r)}$ = weight factor

$E_{rd(i)}$ = residual energy of node i

$E_{rd(j)}$ = average residual energy

N = number of nodes

K = number of clusters

else

Node i gives up cluster head selection and chooses to join a cluster.

The sink chooses sensor nodes with maximum election weight as cluster heads. After that, cluster heads will broadcast to neighbor nodes to notify them that it has been elected as a

new cluster head. After receiving the broadcast message transmitted over a single-hop or multi-hop path, the non-cluster sensor node need to determine whether it is in the sensing range of the cluster head or not, if not they will join the CH with sensing range. Right before the beginning of a new round, source nodes should send the information of their residual energy to their cluster heads along with the sensing data for the purpose of cluster reconstruction[14][15].

4. Mobile Agents in Sensor Networks

4.1. Working of Mobile Agent

There are various model of mobile computing : client/server, peer based and agent based. In agent based modeling, a mobile agent consists of the program code and the program execution state (the current values of variables, next instruction to be executed, etc.)[16][17]. Initially a mobile agent resides on a computer called the home machine. The agent is then dispatched to execute on a remote computer called a mobile agent host (a mobile agent host is also called mobile agent platform or mobile agent server). Figure 2. describes the sequence of processes carried out during agent's lifetime. When a mobile agent is dispatched the entire code of the mobile agent(MA) and the execution state of the mobile agent is transferred to the host. The host provides a suitable execution environment for the mobile agent to execute. The mobile agent uses resources like CPU, memory, etc. of the host to perform its task[18]. After completing its task on the host, the mobile agent migrates to another computer. Since the state information is also transferred to the host, mobile agents can resume the execution of the code from where they left off in the previous host instead of having to restart execution from the beginning. This continues until the mobile agent returns to its home machine after completing execution on the last machine in its itinerary[19][20].

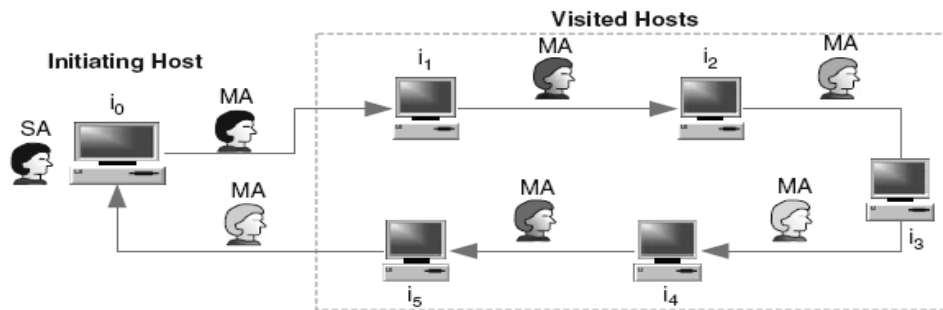


Fig 2. Agent's Lifecycle

Although, in the last several years, there has been extensive research going on in the development of mobile agent systems, there are few efforts in the study of their performance in real world applications of wireless networks. As a result, the spread of mobile agent technology in real world wireless applications cannot as yet be seen[21]. We describe the implementation of a tool to simulate the behavior of mobile agents and its utilization in industrial applications. The implementation consists of extending the existing network simulator NS2 to support mobile agent simulation.

4.2. Proposed Work

We mainly focus on the use of mobile agents in sensor networks for information retrieval and to improve the latency in the data delivery. We use the models of event driven (source initiated) and poll driven WSN model. In event driven model, the source sensor monitors for a given event (sensors are generally designed for specific applications) and reports it to base station. In a poll driven (sink initiated) model, the base station polls the source station whenever the information about a particular event is required. The architecture basically uses mobile agents ability to carry processing codes that allow the computation and communication resources at the sensor nodes to be efficiently harnessed in an undefined area. The MAs should adjust their behavior depending on quality of service needs (e.g. data delivery latency) and the network characteristics to increase network lifetime while still meeting those quality of service needs. The aim of the work would be to dedicate MAs to introduce retrieval of data from a sensor network. It also reduces the information redundancy and communication overhead at all levels so as to prolong network lifetime.

Consider a yarn break detection in a spinning machine application of a sensor network. The sensor nodes are deployed to monitor the rotation of a ring traveler which carries yarn in it. Any break in the yarn stops the rotation of the traveler. Now, the deployed sensor nodes across each spindle monitors the rotation of the traveler. If there is any break (event), it gives the report information to the cluster head. Sink sends the mobile agent to the cluster heads to gather the information.

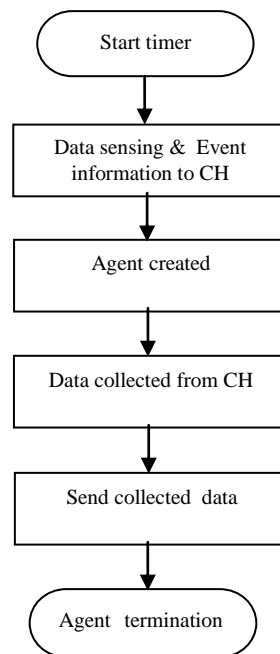


Fig 3. Flow Chart Representation of Our Model

The agent reaches the target region through multiple hops. Here it performs a three step process to retrieve and process the information: Registration- The agent checks the target node for integrity and authenticity using information being provided by the sink node. Once the authenticity check is performed, the information is collected for further processing. Association- The process of registration is followed by processing of the information gathered

and removing the redundant /unwanted information. This process generally starts from second node onwards. Fusion- The process information is fused into the sink node to take appropriate action. The developed example application is structured and is shown in Figure 3. Once the timer starts, data sensing starts and the event is detected, which is then sent to cluster head, agent is created, data is collected by the agents, information is given to base station and finally the agent is terminated.

Table 1 shows the parameters of a mobile agent. The entire agent code is transmitted along with its data and execution stack. The agent's code is transmitted only if it is not available at the receiving node. The agent consists of B_{code} bytes of code, B_{data} bytes of data and B_{state} bytes of execution state and is described by the triple $B_A = (B_{code}, B_{data}, B_{state})$. The selectivity of the agent is given by σ . Thus the actual size of the results that agent is required to carry after remote processing is $(1-\sigma) B_{rep}$. The total network load for the migration of an agent A from node n1 to node n2 and then back from node n2 to node n1 is given as:

$$B_{MR} = B_{code} + B_{data} + B_{state} + (1-\sigma) B_{rep}$$

The execution time for a single agent migration from node n1 to node n2 will be

$$T_{Mig} = \delta + [(1/\tau) + 2\mu] (B_{code} + B_{data} + B_{state})$$

The total execution time for the entire agent task, neglecting any other processing time, can be given as:

$$T_{MR} = T_{Mig} + \delta + [(1/\tau) + 2\mu] B_{MR}$$

τ denotes the throughput of the link between nodes n1 and n2 and also μ represents the overhead. The above equation represents the mathematical model used in the simulation[22].

Table 1. Parameters of a Mobile Agent

ID	Unique ID or name given to a mobile agent.
code_size_data_size (bytes)	Represents the size of the code and data of a mobile agent respectively.
Node	Name of the current node of execution
State	Current state of the agent i.e. whether it is idle, active, mobile or terminating.
Size (bytes)	Total size of the agent migrating from one context to another.

5. Simulated Output

We have tried to build a model for simulating the general behavior of mobile agents by extending the existing network simulator NS2. Within the clusters, the average one hop delay is calculated using the event driven MAC protocol and is plotted in Figure 4 and the number of nodes reporting an event is also represented in Figure 5. Then, the average energy consumed is calculated using the energy efficient cluster formation algorithm and is shown in

Figure 6.. Using this model, the performance of the mobile agents are studied and compared with more traditional client-server approach. Figure 7 compares the time required for client-server approach(upper curve) and mobile agent based approach(lower curve).

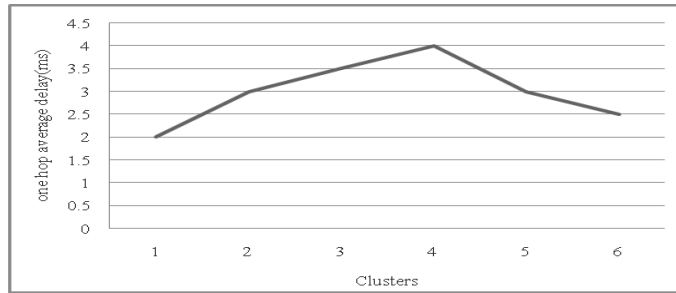


Fig 4. Number of clusters vs average one hop delay

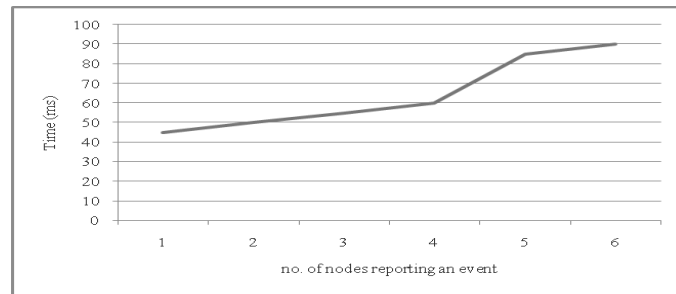


Fig 5. Number of nodes reporting an event vs time

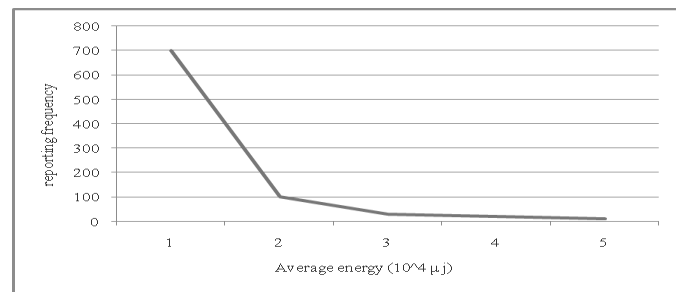


Fig 6. Energy consumption

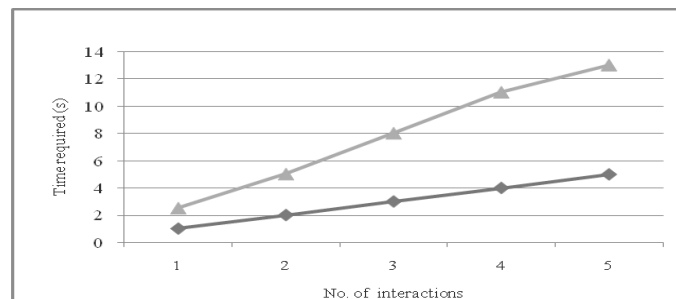


Fig 7. Effective number of Interactions

The simulation results suggest that a combination of energy efficient cluster formation, MAC layer protocol for event driven application and mobile agents to reduce the latency, may actually provide greater energy saving as well as much better latency.

6. Conclusion

The field of wireless sensor networks offers a rich, multi-disciplinary area of research, in which a variety of tools and concepts can be employed to address a diverse set of applications. As such, many potentials of this field have been under study both in academia and in the industry. To the best of our knowledge, we are the first to investigate the use of wireless sensor networks in spinning machine to detect the yarn break considering energy-latency problems using mobile agents. In this perspective, the results obtained from the simulation studies support the current trend of integrating the mobile agent technology into the mainstream of industrial software development rather than treating it as an exotic and specialized offshoot.

References

- [1] I.F. Akyldiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey, Computer Networks", Vol. 38, No. 4, pp. 393-422, March 2002.
- [2] Ulmer C, "Organization techniques for wireless in-situ, sensor networks", <http://www.craigulmer.com/research/>, accessed March 20, 2007.
- [3] Abdelkader Outtagarts, "Mobile Agent-based Applications : a Survey", IJCSNS International Journal of Computer Science and Network Security, Vol .9 No.11, November 2009.
- [4] Suriyakala C.D, Sankaranarayanan. P.E, " Smart Multiagent Architecture for Congestion Control to Access Remote Energy Meters", International Conference on Computational Intelligence and Multimedia Applications, 2007.
- [5] Ilker demirkol, Cem Ersoy and Fatih Alagoz, "MAC protocols for Wireless sensor networks: A Survey", IEEE Communications Magazine Topics in Ad hoc and Sensor networks, April 2006.
- [6] Chung-Jung Fu, et al., "A Latency MAC Protocol for Wireless Sensor Networks", International Journal of Future Generation Communication and Networking, Vol 2, No.1, March 2009.
- [7] I. Marin, et al., "LL-MAC: A low latency MAC protocol for wireless self-organised networks", Elsevier, Microprocessors and Microsystems 32 (2008) pp197-209.
- [8] Lu.G, Krishnamachari.B, Raghavendra C.S., "An adaptive energy efficient and low-latency MAC for data gathering in wireless sensor networks", Proceedings of 18th International parallel Distributed Processing symposium, Pages: 224, 26-30 April 2004.
- [9] Ye.W, Heiemann.J, Estrin. D, "An energy efficient MAC protocol for wireless sensor networks", IEEE Proc.Globecom 2001, pp 2944-2948.
- [10] Kyle Jamieson, et al., "Sift : A MAC protocol for Event-driven Wireless sensor networks", EWSN 2006, LNCS 3868, pp 260-275.
- [11] Fatma Bouabdallah, Nizar Bouabdallah, "The tradeoff between maximizing the sensor network lifetime and the fastest way to report reliably an event using reporting nodes selection", Elsevier, Computer communications 31, 2008, pp 1763-1776.
- [12] Yuzhong Chen and Yiping Chen, "An energy efficient clustering algorithm based on residual energy and concentration degree in wireless sensor networks", ISCSCT'09, Dec 26-28, 2009, pp306-309.
- [13] Bolian Yin, Hongchi Shi, and Yi Shang, "Analysis of Energy Consumption in Clustered Wireless Sensor Networks", 2nd International symposium on Wireless Pervasive Computing, ISWPC '07.
- [14] Ming Zhang and Chenglong Gong, "Energy-Efficient Dynamic Clustering Algorithm in Wireless Sensor Networks", International Symposium on Computer Science and Computational Technology, 2008, pp303-306.
- [15] Adeel Akhtar, et al., "Energy Aware Intra Cluster Routing for Wireless sensor networks", International Journal of Hybrid Information Technology, Vol.3, No.1, January, 2010.

- [16] Francesco Aiello et al., "A Java-Based Agent Platform for Programming Wireless Sensor Networks", The Computer Journal (2010) doi: 10.1093.
- [17] Abdelkader Outtagarts, "Mobile Agent based Applications -A Survey", International Journal of Computer Science and Network Security, Vol 9, No.11, November 2009.
- [18] Hongjoong Sin, et al., "Agent-based Framework for Energy Efficiency in Wireless Sensor Networks", World Academy of Science, Engineering and Technology 46, 2008.
- [19] Sajid Hussain et al., "Agent-based system architecture for wireless sensor networks", 20th International conference on Advanced Information Networking and Applications, April 18-20, 2006.
- [20] Osborne and K. Shah, "Performance Analysis of Mobile Agents in Wireless Internet Applications using simulation", A Scientific and Technical Publishing company, OACTA press.
- [21] Al-Jaljoui. R, Abawajy J, "Agents Based e-Commerce and Securing Exchanged Information", Pervasive computing, Springer London, 2009, pp 383-404.
- [22] Kunal Shah, "Performance Analysis of Mobile Agents in wireless Internet Applications using simulation", A thesis presented to the faculty of the College of Graduate Studies, Lamar University.

Authors



Sundari G. received the undergraduate degree in electronics and communication engineering from Institute of Engineers, Calcutta, in 1997 and the post graduate degree in applied electronics from Anna university, Chennai in 2001. She is currently doing her research in faculty of electronics engineering at Sathyabama University. Her areas of interest includes wireless sensor network, agent based technology, distributed systems and energy optimization.



Sankaranarayanan P. E. The significant engineering achievements of Prof. P. E. Sankaranarayanan are in the field of Instrumentation - sensor based electronic system development for process control applications His major contribution as a pioneer has been in the area of electronic instrumentation systems development pertaining to Aeronautics, Oceanography & Process Industries like - pulp & paper, leather and food processing Industries. He has more than 300 publications in international/national journals, proceedings, reports etc., to his credit. He also holds 6 patents and guided more than 75 students for the M.Tech. M.E., and M.Sc. and 25 towards Ph.D./M.S. (By Research) degrees in the field of Electronics.