EMID: Maximizing Lifetime of Wireless Sensor Network by Using Energy Efficient Middleware Service

Chandrakant N[1], Tejas J[1], Harsha D[1]
Dept. Computer Science and Engineering,University Visvesvaraya College of Engineering,Bangalore, India
P Deepa Shenoy[1], Venugopal K R[1], L M Patnaik[2]
[2]Vice Chancellor,Defense Institute of Advanced Technology,Pune, India

Abstract:This paper introduces the processing of raw data from sensor nodes located at different places within the vicinity of header node. The middleware service of header node evaluates the assignment and requirement of each node that comes under its vicinity. Based on header node instructions each sensor node is in one of two modes: Active mode or Sleep mode. We have developed a software program to compute the essence of each node based on the raw information provided by each sensor node. If raw data of current sensor node is static at certain time interval or if the raw data of current sensor node is equal to the raw data of other sensor node, then the current node will be treated as qualified node to go to sleep for the time period of maxSleepTime. The proposed algorithm is well suited for military application or monitoring unmanned area.

Keywords-EMID, WSN, SleepWake Cycle

I. INTRODUCTION

Wireless Sensor Networks (WSN) is a set of sensor nodes that collects the information from environment and sends to base station (Header node or Central Node).Basically, WSN are application specific and all design considerations are different for each application. The requirements of WSNs are very specific, especially when it comes to military application.

Middleware is a software infrastructure that binds together the applications, network hardware, operating systems and network stacks. The main services of middleware are to provide standardized system services to diverse applications. It provides a runtime environment that can support and coordinate multiple applications. However the important mechanism of middleware is to achieve adaptive and efficient utilization of resources.

WSN is limited in energy and has individual resources (such as CPU and memory). These tiny devices could be deployed in hundreds or even thousands in harsh and hostile environments. In many cases physical contact for gathering data is impossible. In such cases wireless media is the only way for remote accessibility. Hence, middleware should provide mechanisms for efficient computation and memory use while enabling lower-power communication. A sensor node should accomplish its three basic operations: sensing, data processing, and communication without exhausting resources such as energy.

The development of middleware for sensor networks, however, places new challenges to middleware developers due to the low availability of resources like energy and processing capacity of the sensor nodes. A middleware layer should act as a broker between applications and the WSN, translating application requirements into WSN configuration parameters. Due to the dynamism of WSN environments, applications should have some degree of power awareness to best reach their network lifetime requirements. The middleware should supply mechanisms that allow the application to monitor the network state through a high level interface.

EMID (Energy efficient MIDdleware service) paper proposes a service-oriented middleware for WSNs. We address the problem of energy efficiency in wireless sensor applications. Considering raw data, the proposed algorithm is used to decide which node has to go for sleep and which node has to wake up.

II. RELATED WORK

The power related problem has been studied extensively in the context of power aware

communication mechanisms. The Aura project [4] and related work on SenSay: A Context-Aware Mobile Phone [9] investigates how a small set of sensors, may relieve the user from being constantly aware of and having to manage the telephones state.

The Solar system [1] is a prototype implementation of a graph-based abstraction for context collecting, aggregation, and dissemination of (sensor) generated events passing one or more operators and is finally delivered to a subscribing application.

The Context Toolkit [2] supports the development of context-aware applications using context widgets with different responsibilities that provide context information to applications. The lowest level interfaces to a physical sensor. The middle layer is concerned with abstracting and combining data. The highest level coordinates the underlying components and provides the callback interface to applications.

The Web Architectures for Service Platforms (WASP) [5]was designed to support context-aware applications specifically in the 3G environment using Web Services technologies and WASP Subscription Language (WSL) to communicate with the platform that connects context-aware applications with context providers (sensors) and third party service providers. The project "Context Recognition by User Situation Data Analysis (Context)" [7] studies characterization and analysis of information about users' context and use it in adaptation.

Mires [10] propose an adaptation of a message oriented middleware for traditional fixed distributed systems. Mires provide an asynchronous communication model that is suitable for WSN applications, which are event driven in most cases, and has more advantages over the traditional request-reply model. It adopts a component-based programming model using active messages to implement its publish-subscribe-based communication infrastructure.

This paper is organized as follows; Section III briefs the problem definition. Section IV gives details about middleware architecture for wireless sensor network. In Section V, we have proposed an algorithm to achieve maximum network life time. Finally Section VI gives our conclusion.

III. PROBLEM DEFINITION

One of the key tasks of WSN is their ability to bridge the gap between the physical and logical worlds, by gathering certain useful information from the physical world and communicating that information to more powerful logical devices that can process it. If the ability of the WSN is suitably harnessed, it is visualized that WSNs can reduce or eliminate the need for human involvement in information gathering in certain civilian and military applications. The lifetime of wireless sensor network is limited due to lack of battery power. In some cases physical contact for replacement and maintenance is impossible. In such situations we require a efficient service to overcome this problem. The objective of EMID algorithm is to increase the network lifetime by applying Optimal Sleep-Wake Policies for Wireless Sensor Network. The sleep mode is a power saving mode in which the sensor only harvests energy and performs no other functions so that the energy consumption is negligible [6].

IV. MIDDLEWARE ARCHITECTURE

WSN middleware is a software infrastructure that glues together the network hardware, operating systems, network stacks, and applications as given in the Fig 2. A complete middleware solution should include a runtime environment that supports and manages multiple applications, and standardized system services such as data aggregation, control and management policies adapting to target applications, and mechanisms to achieve adaptive and efficient system resources used to prolong the sensor networks life. The EMID will be introduced at the resource management layer to manage the resources based on the raw data provided by each node. The resource management layer also coordinates the resource sharing based on application needs, passed through the upper layers. Services provided by upper layers may need some resource sharing support, which is encapsulated in the communication layer.

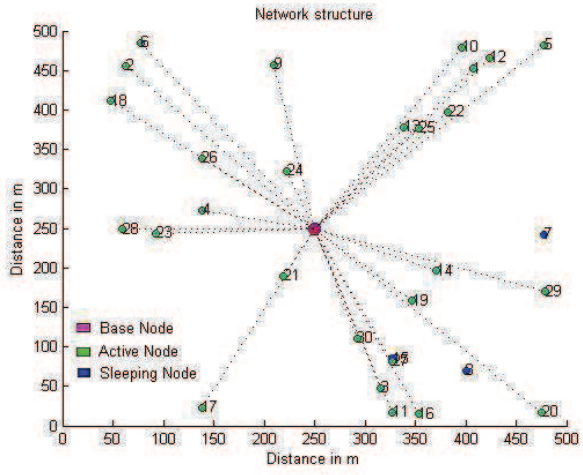


Figure 1. Structure of Wireless Sensor Network.

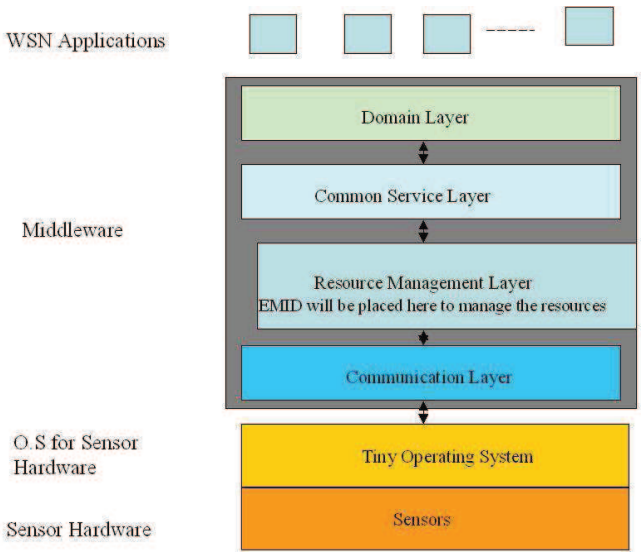


Figure 2. EMID overall architecture.

As an application uses such a service, the corresponding layer asks for the communication layer to manage the access control of the required resources. Indeed, the resource management layer commands the allocation and adaptation of resources, such that the QoS requirements specified by the applications can be met.

V. IMPLEMENTATION AND RESULTS

A key assumption in EMID is that, header node will act as central node for all sensor nodes that comes under his vicinity. A central node will have an additional or constant power supply for computation and management of nodes.
Middleware is placed in central node or header node as it has constant or additional power supply. Also, each sensor is preloaded with certain power to sense the environment. Since all sensor nodes are homogeneous in nature, their transmission ranges are assumed to be the same. Table 1 enlists all variables used to implement this paper. This section discusses the algorithm and variables used in the implementation. Central node manages the set of sensor nodes. Hence all children nodes should report to the header node. Each sensor node will transmit the raw data detected at the environment to the header node. Header node will extract the raw data. If the raw data of current node is equal to its previous owned raw data that has been taken at dataAtSampleTime or if the raw data of current node is equal to the raw data of other node, then SendSleepSignal function will be executed which sends current node to the sleep mode.

TABLE I.

TABLE I. PARAMETERS USED IN PERFORMANCE ANALYSIS

Symbol	Description	Value
Tmaxsleep	Maximum sleep Time	5 Seconds
Tsample	Sample Time	1 second
Psleep	Power dissipated during Sleep Mode	600μW
PCS	Power dissipated when Processing	15mW
Et1	Power dissipated for sending 1 bit of data	1mW
Ed1	Power for transmission over a distance d	1mW
N	Number of nodes considered	30 and 500
A	Area of network	500X500 m2
Pinit	Initial power of each node	10W
Ttone	Time required to send the wakeup signal	Variable
PRX	Power when receiving	45mW
PTX	Power when transmitting	60mW
Pwu	Power when in wakeup mode	177W
Fwu	Frequency of the wakeup signal	862Hz
Fmsg	Frequency of sending message	Variable
Tmsg	Time needed for sending a msg+ACK	21 ms
Thdr	Time needed for sending just a header	7 ms

```
Begin
  Signal ← Wakeup
     if (currentData= previousData)
         Signal ← Sleep
          return Signal
     end if
  count← N
  while(count != 0)
      if(buff_rawData[count--] =
previousData)
                  Signal ← Sleep
                  break
      end if
  end while
  return Signal
```

```
  t ← currentTime;
  if (s[i].t == maxSleepTime)
    sendWakeUpSignal (s[i] )
    break;
  end if
  currentData ←s[i]. curData
```

Each node can stay in sleep mode for maximum of maxSleepTime only. After completion of this period, the middleware service will send a message called sendWakeUpSignal to wake up the sensor node for regular service.
This work has been carried out in MATLAB, and file details are as follows.

node: We define a class which specifies the properties and initial conditions of a sensor node.
create: This program creates nodes and sets timer object. We can set sample time using the timer object and specify the functions (or file) to be executed. We have taken sample time of 1 second for testing purpose.
emid: Our algorithm (EMID) is implemented in this file. Here base node processes the raw data received from each node as per the algorithm and sets the flag to each node. Depending on the value of flag base node sends wakeup/sleep signal to the node. This function (file) executes for every sample time and calculations are done.
graph: This will plot the graph of energy v/s time for a random node and total energy of system v/s time. Energy level without our algorithm is also plotted for comparison purpose.
efficiency: Executing this file will calculate the increase in efficiency.

Algorithm 1 and 2 demonstrates the above said logic. The energy spent in transmission of a single bit [8] is given by

$$e_{tx}(d) = e_{t1} + e_{d1} \times d^{n} \quad (1)$$

where e_{t1} is the energy dissipated per bit in the transmitter circuitry and $e_{d1}xd^{n}$ is the energy dissipated for transmission of a single bit over a distance d, n being the path loss exponent (usually 2.0 < n < 4.0). The latency of a (one-hop) message transfer consists of the time needed to send the wakeup signal (Ttone), and the actual transfer time over the primary radio (Tmsg)[3]. This transfer time includes waking up the radio, receiving the message (including headers), and sending back an acknowledge frame. For simplicity, transmission errors and collisions are not considered. Hence retransmissions are not modeled.

$$T_{tone} = 10/F_{wu} \quad (2)$$

$$L_{tone} = T_{tone} + T_{msg} \quad (3)$$

$$P_{wu} = P_{wu} + F_{msg} \times ((T_{tone} + T_{msg}) \times P_{TX} + T_{msg} \times P \quad (4)$$

The average power consumed by a node depends on the frequency at which messages are sent through the network, denoted by Tmsg. Each message transfer adds energy to the basic costs of the wakeup circuitry (Pwu). Receiving the message also takes Tmsg time.
We have started experimenting by taking 30 and 500 nodes in each network. We have calculated the time and energy usage by each network as shown in Fig 3 and Fig 4 respectively. The graph shows the statistics of the network by using EMID technique and without using EMID technique. EMID certainly increases the network lifetime by applying sleep and wakeup strategies. In Fig 5, we have randomly selected a node in the network to check the energy level v/s time.
By analysing Fig 3, Fig 4 and Fig 5, we have increased the network life time about 5.63% in the network of 30 nodes and 7.19% in the network of 500 nodes , hence EMID is well suited for big size network and/or crowded network.

Figure 3. Energy v/s Time for 30 nodes.

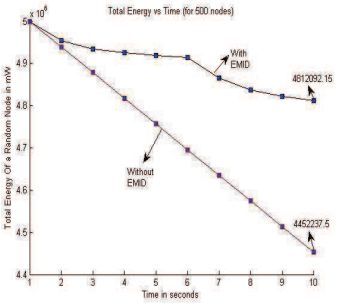


Figure 4. Energy v/s Time for 500 nodes

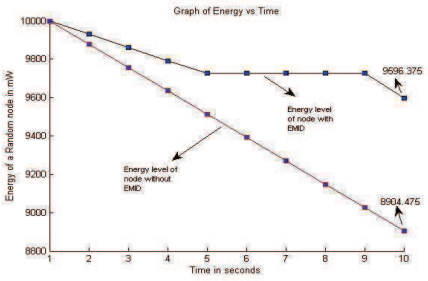
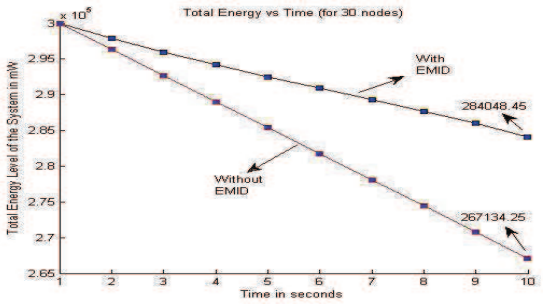


Figure 5. Energy v/s Time for randomly selected node



VI. CONCLUSIONS

In this paper, we propose EMID, an energy efficient middleware service for wireless_sensor network. The proposed algorithm increases the network lifetime by computing the essence of each node based on the raw information provided by each sensor node in the network. The average percentage energy saved per network is found to be around 5-7%. Thus there is an enhancement of energy management in EMID by applying wakeup and sleep strategies. As a future work, the application layer can also be included by means of the application metrics.

REFERENCES

[1] *G. Chen and D. Kotz. Solar: A pervasive-computing infrastructure for context-aware mobile applications. Technical report, Department of Computer Science, Dartmouth College, Hanover, NH, USA, 2002.*

[2] *A. K. Dey, D. Salber, and G. D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of contextaware applications. Human-Computer Interaction, 16:97–166, 2001.*

[3] *B. V. D. Doorn, W. Kavelaars, and K. Langendoen. A prototype low-cost wakeup radio for the 868mhz band. International Journal of Sensor Networks, 5:22– 32, 2009.*

[4] *D. Garlan. Aura: Distraction-free ubiquitous computing. In IFIP International Conference on Engineering for Human-Computer Interaction, pages 1–2, 2001.*

[5] *B. Han, W. Jia, J. Shen, and M.-C. Yuen. Contextawareness in mobile web services. Parallel and Distributed Processing and Applications, 3358:519–528, 2005.*

[6] *V. Joseph, V. Sharma, and U. Mukherji. Optimal sleepwake policies for an energy harvesting sensor node. In IEEE International Conference on Communications, pages 1–6, 2009.*

[7] *J. Mantyjarvi, J. Himberg, P. Kangas, U. Tuomela, and P. Huuskonen. Sensor signal data set for exploring context recognition of mobile devices. In International conference on Pervasive Computing, pages 1–6, 2004.*

[8] *A. Seetharam, A. Acharya, A. Bhattacharyya, and M. K. Naskar. An energy efficient data gathering protocol for wireless sensor networks. Journal of Applied Computer Science, 1:30–34, 2008*

[9] *E. Souto, G. Guimar˜aes, G. Vasconcelos, M. Vieira, N. Rosa, and C. Ferraz. A message-oriented middleware for sensor networks. In Workshop on Middleware for pervasive and ad-hoc computing, pages 127–134, 2004*