COMPUSOFT, An international journal of advanced computer technology, 3 (4), April-2014 (Volume-III, Issue-IV)

ISSN:2320-0790

An International Journal of Advanced Computer Technology

ENERGY EFFICIENT TRACKING SYSTEM USING WIRELESS SENSORS

Thankaselvi Kumaresan[1], Sheryl Mathias[2], Digja Khanvilkar[3], Prof. Smita Dange[4]
Department of Computer Engineering
Fr. C. Rodrigues Institute of Technology
Sector 9-A, Vashi, Navi Mumbai

Abstract: One of the most important applications of wireless sensor networks (WSNs) is surveillance system, which is used to track moving targets. WSN is composed of a large number of low cost sensors which operate on the power derived from batteries. Energy efficiency is an important issue in WSN, which determines the network lifetime. Due to the need for continuous monitoring with 100% efficiency, keeping all the sensor nodes active permanently leads to fast draining of batteries. Hence there is a need to design a tracking system which conserves energy without affecting its ability to track accurately.

In our paper, we present energy efficient tracking system which keeps sensor nodes in sleep mode and awakens only those nodes which lie on the path of target. By allowing the nodes to be in sleep mode for maximum amount of time, this algorithm saves lot of energy. This is proved by simulating our system using OMNeT++ simulator and Castalia framework.

Keywords: WSN, energy, target tracking, sleep scheduling, duty cycling, Castalia, SPAN

I. INTRODUCTION

Recently, there has been a tremendous increase in the use of sensor networks for a variety of applications in different fields. This has been facilitated due to the diverse and distinguishing features of sensors. Sensors are tiny devices which are capable of sensing, processing and transmitting the required information. Some of the known applications of sensor networks include environmental monitoring, warfare, child education, surveillance, micro-surgery, industrial monitoring, agriculture[1].Though there are many advantages provided by wireless sensor networks, we also face many constraints and challenges during the use of sensors. The main difficulties that arise are related to hardware designing so as to achieve efficient communication. But the biggest and most important issue in wireless sensor networks is energy management. Energy is a non renewable resource. For the sensors to function properly, continuous power supply is a must. But the amount of communication and other processing required to be performed by the sensors drains out the power at a considerably faster rate. Also, it is difficult to replace the power supply of a sensor. Hence there is a need to devise methods for minimizing energy consumption in sensor networks.

In this paper, we have discussed energy efficient target tracking system using wireless sensors. The proposed system makes use of sensors which are put to sleep mode initially. Only those nodes which lie in the path of the target are awakened and others are allowed to sleep. In this way, all sensors are not active all the time, thereby leading to a considerable reduction in the amount of energy utilized. This system makes use of the concept of duty cycling and sleep scheduling wherein the awakening of nodes is controlled by the presence of target near them.

II. WIRELESS NETWORKS

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. WSNs are used to collect data from the environment. They consist of large number of sensor nodes and one or more Base Stations. The nodes in the network are connected via wireless communication channels. Each node has capability to sense data, process the data and send it to rest of the nodes or to Base Station. [6]

The WSN is built of "nodes" - from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts:

- A radio transceiver with an internal antenna or connection to an external antenna
- A microcontroller, an electronic circuit for interfacing with the sensors
- An energy source, usually a battery or an embedded form of energy harvesting.

A sensor converts the physical parameter into a signal which can be measured electrically.

III. ENERGY CONSUMPTION IN WSN

Energy is a factor of outmost importance in WSNs. The wireless sensor node, being a micro-electronic device, can only be equipped with a limited power source (<0.5 Ah, 1.2 V). Sensor node lifetime, therefore, shows a strong dependence on battery lifetime. Energy is a very scarce resource for sensor systems and has to be used wisely in order to extend the life of the sensor nodes for the duration of a particular mission. Energy consumption in a sensor node could be due to either "useful" or "wasteful" sources. Useful energy consumption can be due to transmitting or receiving data, processing query requests, and forwarding queries and data to neighbouring nodes. Wasteful energy consumption can be due to one or more of the following factors: idle listening to the channel, Retransmitting, overhearing, over-emitting. [13]

Considering the above-mentioned facts, a correctly designed protocol must be considered to prevent these energy wastes. Also it is important to design sensor networks aiming to maximize their life expectancy. Data communication is responsible for the greatest weight in the energy budget when compared with data sensing and processing. Therefore, it is desirable to use short range instead of long-range communication between sensor nodes because of the transmission power required.

IV. SOLUTIONS FOR MINIMIZING ENERGY CONSUMPTION

A. Duty cycling approach

The main energy conserving method is to put the radio transceiver in low power mode when communication is not needed. Ideally, the radio should be switched off as soon as there is no more data to send/receive and should be resumed as soon as a new data packet becomes ready. In this way, nodes alternate between active and sleep periods depending on network activity. So there is a constant switching between sleep and active states. This is duty cycling. Duty cycle represents the fraction of time the nodes are active. A sleep/wake up scheduling algorithm should accompany any duty cycling scheme which is a distributed algorithm based on which sensor nodes decide when a transit from active to sleep state is needed. [5]

B. Protocols used for efficient energy consumption

The clustering algorithm is considered as a kind of key technique used to reduce energy consumption. It can help in increasing the stability period and network life time. Clustering can be done in two types of networks, homogeneous and heterogeneous networks. In homogeneous networks, initial energy of all nodes is same while in heterogeneous network, all nodes have different initial energy. It is difficult to implement an energy aware heterogeneous clustering algorithm due to the complex energy configuration of the network. Some of the previous clustering algorithm for homogenous networks are LEACH (Low Energy Adaptive Clustering Hierarchy), PEGASIS and HEED and non-homogenous one's are SEP and DEEC. [9]

V. TARGET TRACKING SYSTEMS

Target tracking in WSNs is an important problem with a large spectrum of applications, such as surveillance, natural disaster relief, traffic monitoring, border security etc. However, the use of sensor networks for target tracking presents a number of new challenges. These challenges include limited energy supply and communication bandwidth, distributed algorithms and control, and handling the fundamental performance limits of sensor nodes, especially as the size of the network becomes large. The environment plays a key role in determining the size of the network, the deployment scheme, and the network topology. Power consumption is the most important design factor for WSNs.

A. Algorithms for Vehicle Detection and Tracking

Target tracking algorithms usually focus on the aspect of the sensor nodes' interaction with a target after the target has already been detected within the area the sensor nodes cover. Once the object has been detected, the nodes collect information and then use one of many different types of algorithms to calculate the current location of the object relative to the sensor nodes' locations. From here, it is the goal of the sensor network to track the object as it moves through the network. This may or may not involve predicting the next location of the object when it moves. In order to forewarn those nodes it will be heading towards to prepare to capture data. Some of the tracking algorithms are– Simple triangulation, Clusters (Hierarchical super nodes, Dynamic clustering, Probabilistic Localization, Distributive predictive tracking), Particle filtering, Rooted tree. [1]

B. Existing systems

The role of WSN in border surveillance, as in most WSN applications, focuses on information gathering from various types of sensors, such as seismic, camera, thermal camera, and motion detectors. Some advanced WSN process these raw data and send an abstracted alarm or aggregated data to the command centre, which, in turn, takes the appropriate defence action. Examples of this kind of usage include intrusion detection, geo-fencing (along with background monitoring) and area monitoring.

Some of the existing intrusion detection systems used by US military and Government agencies are:

1) ***Self-Powered Ad-hoc Network (SPAN)[12]:*** Aerospace and defense company Lockheed Martin has launched a new wireless-sensor mesh-network system designed for detecting intrusion, monitoring secured areas or tracking structure stress, using small devices that contain active RFID tags and can incorporate energy-harvesting technology. For the detection of movement at a border crossing, Self-Powered Ad-hoc Network (SPAN) nodes may be equipped with ground-vibration or acoustic sensors, while for structural-integrity applications, stress sensors would be employed. SPAN system is a network of “Field-and Forget” ground sensors that provides unobtrusive, continuous surveillance which can support a variety of missions and applications, such as border protection, area surveillance and even bridge, pipeline, aircraft and other structural monitoring requirements. SPAN is a mesh network of self-organizing, self-healing sensors. Information from this mesh is processed using proprietary algorithms that reduce false alarms, providing intelligent situational awareness for military, border patrol and structural monitoring applications. This smart sensor network can cue a camera or unmanned aerial vehicle to further study an area or call an engineer when a pipeline or bridge structure is in danger of fracture. The SPAN system also operates with extremely low power, thereby making it possible to power it via energy-harvesting technology, such as solar panels. Its ultra low sensor cost is predicated on the fact that each node within the SPAN network incorporates an energy harvesting subsystem that re-charges itself using simple energy sources in its surrounding environment.

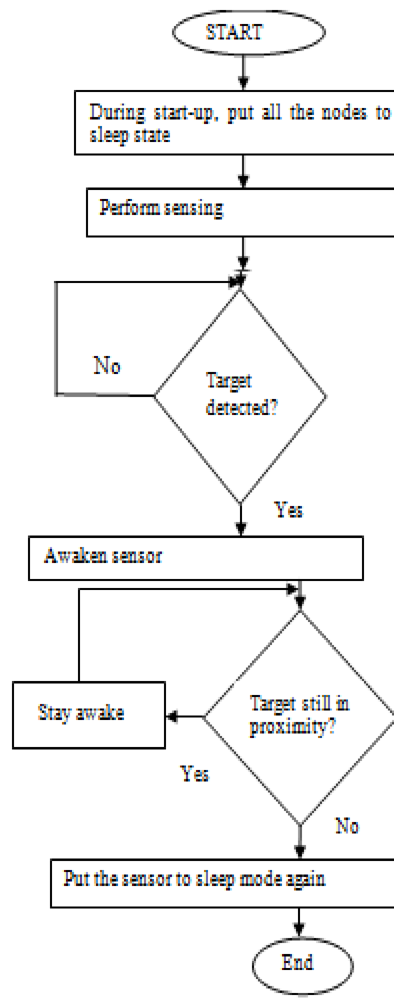
2) *MicroObserver System* [13]: MicroObserver system is a next-generation Unattended Ground Sensor (UGS) solution that provides situational awareness for applications ranging from perimeter defense and force protection to border security and law enforcement. With its advanced tracking and seismic detection technology, the MicroObserver system provides detection, classification and tracking of people and vehicles. Identification of targets is possible with sensor-cued infrared imaging sensors that utilize proven target detection algorithms to capture a series of mini-video images. MicroObserver sensor nodes are buried in the ground with no part—including the antenna, visible above the surface. The nodes utilize battery power to operate a vibration sensor and transmit data to other sensor nodes. The MicroObserver solution has been commercially available since 2008.Leveraging a radio optimized for low-power operation, the MicroObserver system is capable of both short- and long-term emplacements, with sensor node battery life options of 30-days for tactical operations and greater than two-years for permanent, long-term perimeter security deployment scenarios.

VI. PROPOSED SYSTEM

There are many techniques to track a mobile target such as the proactive wake up which awakens all the neighbouring nodes in the area where the target is expected to arrive without any differentiation. However, this consumes a lot of energy. In order to reduce the energy consumption we propose a system based on sleep scheduling. We designed a homogeneous static sensor network wherein the nodes are put to sleep initially. We assume that every node is aware of its own location and all sensor nodes are locally time synchronized. In sleep mode, the sensor nodes continue to sense, while the radio module is incapable of performing any operations. Thus, idle listening is avoided. As soon as the target arrives, only the nodes closer to the target awaken. These are the nodes whose sensor readings are important for tracking of target. The nodes remain active till the target is nearby them. As soon as the target moves away from the sensors, they again go to sleep mode. Thus in this way, nodes do not remain active unnecessarily thus saving energy. Here the radios of sensors do not indulge in unnecessary transmission of packets. Thus radio module is optimized in order to avoid wastage of energy. When the nodes are active, they send their readings to the controller/sink node. In this manner, transmitting redundant values which are not needed is prevented. Hence power consumption is reduced.

VII. WORKING OF PROPOSED SYSTEM

The following flowchart depicts the working of our proposed system.



VII. TARGET TRACKING IMPLEMENTATION

Target tracking will be implemented using a popular discrete event simulator OMNeT++ with Castalia framework. OMNeT++ is a discrete event simulation environment. Its primary application area is the simulation of communication networks. OMNeT++ provides component architecture for models. Components (modules) are programmed in C++, and then assembled into larger components and models using a high-level language (NED).An OMNeT++ based simulator is built from elements called modules. Simple module is a basic unit of execution and is written in C++. Compound module consists of other modules (simple or compound) that are linked by connections. Top-level compound module is called network module. Modules communicate via messages that are sent via connections or directly from module to module. [15]

Example of a WSN simulator built on top of OMNeT++ is Castalia. It is a generic simulator intended for the first order validation of high-level algorithms before moving to a specific sensor platform [16]. In Castalia, sensor nodes are implemented as compound modules, consisting of sub-modules that represent, for instance, network stack layers, application, and sensor. Node modules are connected to wireless channel and physical process modules.

To plot the positions of all the nodes and the path of the target, we used software called Gnuplot.

The main coding work for our proposed system spanned across Sensor manger, Application, Radio, Routing and MAC. The following types of files are used:

1. Omnetpp.ini: It specifies the structure information, specific parameters and their values. The language used here is NED.
2. .h files: Every module has a .h file which is a typical C header file. It has the names of all the variables and function declarations.
3. .cc files: This is the C/C++ file where the actual code is written to implement a given functionality.
4. .msg files: Contain the structure of messages to be used for communication.

Table I: Simulation parameters

Sr.no	Parameter	Value
1	Simulation time	50 secs
2	Physical process	Car
3	Inter arrival time between cars	1.5 secs
4	Maximum number of cars present at a time	1
5	Filed size	100x10
6	Number of sensors	61
7	Deployment	1.Uniform 2. Random
8	Distance between sensors in uniform deployment	5cm

When we kept all the sensors active without putting them to sleep at any point of time, the energy consumption was in the range of 3-4 mW. Whereas when we simulated our proposed system, the energy consumption was just 0.7-0.9 mW.

The screenshots of output are as shown below:

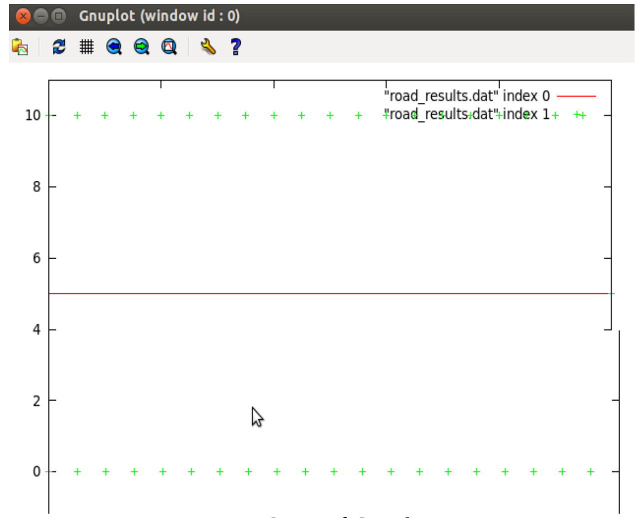
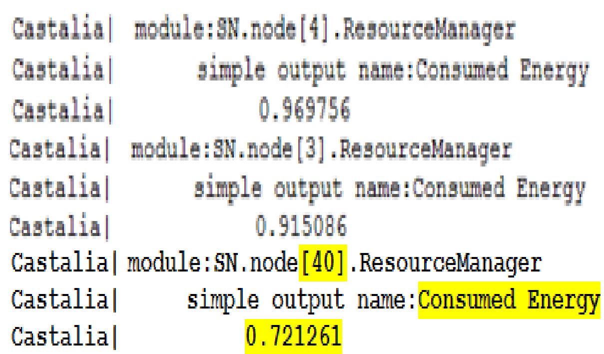
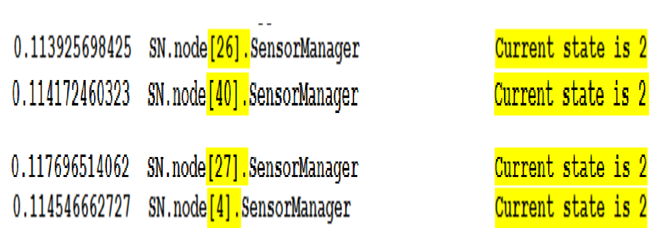


Fig 1 Output of Gnuplot



Castalia| module:SN.node[4].ResourceManager
Castalia| simple output name:Consumed Energy
Castalia| 0.969756
Castalia| module:SN.node[3].ResourceManager
Castalia| simple output name:Consumed Energy
Castalia| 0.915086
Castalia| module:SN.node[40].ResourceManager
Castalia| simple output name:Consumed Energy
Castalia| 0.721261

Fig 2 Screenshot of result file



0.113925698425 SN.node[26].SensorManager Current state is 2
0.114172460323 SN.node[40].SensorManager Current state is 2
0.117696514062 SN.node[27].SensorManager Current state is 2
0.114546662727 SN.node[4].SensorManager Current state is 2

Fig 5 Nodes in sleep mode

0	SN.physicalProcess[0]	First car arrival at 1.90698
0	SN.node[0].MobilityManager	initial location(x:y:z) is 100:5:0
0	SN.node[1].MobilityManager	initial location(x:y:z) is 0:0:0
0	SN.node[2].MobilityManager	initial location(x:y:z) is 5:0:0
0	SN.node[39].MobilityManager	initial location(x:y:z) is 90:10:0
0	SN.node[40].MobilityManager	initial location(x:y:z) is 95:10:0

Fig 3 Location of nodes mentioned in trace file

0.000937400513	SN.node[11].SensorManager	Received STARTUP MESSAGE
0.000937400513	SN.node[11].SensorManager	Current state is 0
0.000937400513	SN.node[11].SensorManager	Node put to sleep
0.001199094159	SN.node[32].SensorManager	Received STARTUP MESSAGE
0.001199094159	SN.node[32].SensorManager	Current state is 0
0.001199094159	SN.node[32].SensorManager	Node put to sleep
0.014176254312	SN.node[40].SensorManager	Received STARTUP MESSAGE
0.014176254312	SN.node[40].SensorManager	Current state is 0
0.014176254312	SN.node[40].SensorManager	Node put to sleep

Fig 4 Putting nodes to sleep (Initial state is Rx)

1.906980964704	SN.physicalProcess[0]	New car arrives on the bridge, index 0
1.907322853432	SN.node[21].SensorManager	Current state is 2
1.907333853431	SN.node[21].Application	Sensed value 1.684 is less than the treshold (10), discarding
1.914104168521	SN.node[40].SensorManager	Current state is 2
1.91411516852	SN.node[40].Application	Accumulated 12/120 bytes of samples. Sensed value is 14.527
2.114096580543	SN.node[40].SensorManager	Current state is 0
2.114107580542	SN.node[40].Application	Accumulated 12/120 bytes of samples. Sensed value is 17.075
2.314088992565	SN.node[40].SensorManager	Current state is 0
2.314099992564	SN.node[40].Application	Accumulated 12/120 bytes of samples. Sensed value is 17.045
2.514081404587	SN.node[40].SensorManager	Current state is 0
2.514092404586	SN.node[40].Application	Accumulated 12/120 bytes of samples. Sensed value is 14.558
2.914066228631	SN.node[40].SensorManager	Current state is 0
2.91407722863	SN.node[40].Application	Sensed value 9.931 is less than the treshold (10), discarding
3.114058640653	SN.node[40].SensorManager	Current state is 2
3.114069640652	SN.node[40].Application	Sensed value 8.502 is less than the treshold (10), discarding

Fig 6 Transitions in state of node 40

VIII. TESTING

We first simulated an instance of existing system wherein all the nodes were active. They were never put in sleep mode. We tested the proposed system by simulating it for different kinds of scenarios. The parameters were assigned varied values and the consistency of results was checked. Basically there were two types of application environments: One with uniform placement of nodes and the other with random placement of nodes. Parameters specifying the size of the field, inter arrival time of cars, simulation time and the number of nodes were modified and the simulation was executed. The system worked well in each of these scenarios. The only case in which the system consumed the same amount of energy as that of the existing system was when the number of cars present simultaneously in the road was high(indicating high traffic). Otherwise the performance of system was as desired. This is an indication to the fact that our proposed system is ideal for use in border surveillance, intrusion detection or roads with light traffic.

IX. CONCLUSION

In this paper, we studied the problem of energy efficiency in wireless sensor networks especially target tracking systems. We studied the various reasons for need of energy conservation and commonly used solutions. We have proposed an energy efficient tracking systems based on sleep scheduling. This method is simpler than other solutions like energy harvesting. We have simulated this algorithm using OMNeT++ and Castalia framework. Simulation results indicate that this system consumes lesser amount of energy.

ACKNOWLEDGEMENT

First and foremost, we would like to thank Prof. H.K.Kaura, Head of Computer Engineering Department, for providing us with all the facilities required by us for the successful completion of this paper. We would also like to thank Prof.Kiruthika, coordinator of Computer Engineering

Department, for her encouragement and support. We express our sincere gratitude to Prof. Smita Dange, our internal guide for her constant guidance, advice and supervision. Without her expertise and experience, we would not have been able to prepare this paper. We also owe our profound gratitude to Prof. MritunjayOjha, who took keen interest in our project and guided us all along the way, by providing all the necessary information required for this project. We also extend our thankfulness to all the faculty members of Computer Engineering Department for their co-operation and assistance.

REFERENCES

[1] G.Padmavathi, D. Shanmugapriya, M.Kalaivani, "A study on vehicle detection and tracking using wireless sensor networks", Scientific Research-Wireless Sensor Network, 2010, pg 173

[2] Bo Jiang, Binoy Ravindran, Hyeonjoong Cho, "Probability-based prediction and sleep scheduling for energy-efficient tracking in sensor networks", IEEE Transactions in Mobile Computing, VOL 12, NO. 4, April 2013, pg 735

[3] F. Shebli, I. Dayoub and J.M. Rouvaen ,"Minimizing energy consumption within wireless sensor networks", University of Valenciennes (IEMN, UMR CNRS 8520) Le Mont Houy, BP311, 59313-Valenciennes Cedex, France

[4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: A survey" , Elsevier, 2001, pg 393

[5] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, Andrea Passarella, "Energy Conservation in Wireless Sensor Networks: a Survey", FP6-2005-NEST-PATH MEMORY PROJECT

[6] http:// en.wikipedia.org/wiki/Wireless_sensor_network

[7] http:// www.engineersgarage.com/articles/sensors

[8] http://www.intechopen.com/books/wireless-sensor-networks-application-centric-design/target-tracking-in-wireless-sensor-networks

[9] http://en.wikipedia.org/wiki/Low_Energy_Adaptive_Clustering_hierarchy

[10] http://www.cse.wustl..edu/~jain/cse567-11/ftp/sensor/index.html

[11] http://home.iitk.ac.in/~rajeevs/snw.html

[12] http://www.lockheedmartin.com/us/products/span.html

[13] http://www.textrondefense.com/products/brigade-combat-team-modernization

[14] http://www.intechopen.com/books/wireless-sensor-networks-technology-and-applications/power-optimization-for-wireless-sensor-networks

[15] OMNet++ Installation guide version 4.3

[16] Athanassios Boulis, "Castalia User manual"

[17] Emad Felemban, "Advanced Border Intrusion Detection and Surveillance Using Wireless Sensor Network Technology", Int. J. Communications, Network and System Sciences, 2013, pg 251

[18] Gnu plot user manual

[19] Prof. Sachin Deshpande, Prof. Umesh Kulkarni, Prof. Mrityunjaykumar Ojha, "Target Tracking in Wireless Sensor Network", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 9, September 2013, pg 177

[20] Wensheng Zhang, Guohong Cao, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks", IEEE Transactions on Wireless Communications, VOL 3, NO 5, September 2004, pg 1689