IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Nodes Self-Scheduling Approach for Maximising Wireless Sensor Network Lifetime

Sivasankari Selvam[*1], Surumbar Khuzhali [2]

Department of Electronics and Communication Engineering, Bharath University, Selaiyur, Chennai, India

sivasankariselvam@gmail.com

Abstract

Coverage and energy conservation are two major issues in wireless sensor networks (WSNs), especially when sensors are randomly deployed in large areas. In such WSNs, sensors are equipped with limited lifetime batteries and redundantly cover the target area. To face the short lifetime of the WSN, the objective is to optimize energy consumption while maintaining the full sensing coverage. A major technique to save the energy is to use a wake-up scheduling protocol through which some nodes stay active whereas the others enter sleep state so as to conserve their energy. This study presents an original algorithm for node

Self scheduling to decide which ones have to switch to the sleep state. The novelty is to take into account the remaining energy at every node in the decision of turning off redundant nodes. Hence, the node with a low remaining energy has priority over its neighbors to enter sleep state. The decision is based on a local neighborhood knowledge that minimizes the algorithm overhead. To verify and evaluate the proposed algorithm, simulations have been conducted and have shown that it can contribute to extend the network lifetime. A comparison with existing works is also presented and the performance gains are highlighted.

Existing Method

- The wireless sensor networks are mainly used in critical environments such as industries, military applications.
- The sensor nodes are equipped with limited energy resources such as battery.
- The sensor nodes need to consume the energy for sending / receiving/ sensing activities.
- Higher energy consumption leads the nodes into sleep state. (the state with low residual energy).
- At sleep state, the nodes cannot send / receive/ sense the data and cannot perform in routing.
- The wake-up scheduling protocols are used to preserve the nodes from entering into sleep state earlier.
- These protocols used ON / OFF technique to put the node in one of these modes respective of their residual energy.
- The nodes with low residual energy are put into OFF mode for a while to conserve its energy.
- But this technique leads to severe coverage issues where the nodes were put into OFF state.

And this technique also leads to connectivity issues in the network.

Proposed Solution

- Here, we propose an algorithm called Energy remaining greedy scheduling (ERGS) algorithm.
- This algorithm uses wake-up protocol with self-scheduling algorithm.
- The self-scheduling phase considered of the following two rounds.
  - Advertisement step
  - Eligibility step
- In advertisement step, each node transmits to its neighbour nodes an advertisement message (ADV), including its ID and its current remaining energy.
- When receiving an ADV message, the receiver node will compare the current residual energy to the transmitter node.
- If the current energy is higher than the transmitter node, the node is specified as HCS list. Otherwise specified into LCS list.
- The nodes belonging to the HCS list have less priority than the receiver node to be deactivated.
- The nodes belonging to the LCS list have more priority to be deactivated.
- In eligibility step, based on the results obtained in advertisement step the nodes are put into sleep state with Timestamp (Ts).

After the Ts expired, the nodes are awakening into active state.

Hardware Specification

PROCESSOR	:	PENTIUM IV
CLOCK SPEED	:	1.7 GHZ
RAM CAPACITY	:	1 GB
HARD DISK DRIVE	:	80 GB
CACHE MEMORY	:	1 MB
VIRTUAL MEMORY	:	512 MB
CD ROM	:	52X LG
POINTING DEVICE	:	LOGITECH MOUSE
KEYBOARD	:	104 KEYS
MONITOR	:	15" VGA COLOR
CABINET	:	ATX

Software Specification

CROSS PLATFORM	:	VMWARE
PROTOCOL DESIGN	:	AODV
LANGUAGE	:	TCL/TK,OTCL
SIMULATOR VERSION	:	NS-2.28
OPERATING SYSTEM	:	REDHAT LINUX 9.0
GRAPH TOOL	:	GNUPLOT
CODE EDITOR	:	GEDIT
TRACE ANALYZER	:	AWK,PERL

Software Implementation

Network Simulation

In communication and computer network research, network simulation is a technique where a program models the behavior of a network either by calculating the interaction between the different network entities(host/routers, data links, packets, etc) using mathematical formulas, or actually capturing and playing back observations from a production network. The behavior of the network and the various applications and services it supports can then be observed in a test lab.

Various attributes of the environment can also be modified in a controlled manner to assess how the network would behave under different conditions. When a simulation program is used in conjunction with live applications and services in order to observe end-to-end performance to the user desktop, this technique is also referred to as network emulation.

Network Simulator

A network simulator is a piece of software or hardware that predicts the behavior of a network, without an actual network being present.The network simulator is the program in charge of calculating how the network would behave. Such software may be distributed in source form(software) or packaged in the form of a dedicated hardware appliance. Users can then customize the simulator to fulfill their specific analysis needs. Simulators typically come with support for the most popular protocols in use today, such as IPv4,IPv6,UDP, and TCP.

Uses OF Network Simulators

Network Simulators serve a variety of needs. Compared to the cost and time involved in setting up an entire test bed containing multiple networked computers, routers and data links, network simulators are relatively fats and inexpensive. They allow engineers to test scenarios that might be particularly difficult or expensive to emulate using real hardware – for instance, simulating the effects of a sudden burst in traffic or a DoS attack on a network service. Networking simulators are particularly useful in allowing designers to test new networking protocols or changes to existing protocols in a controlled and reproducible environment.

Network simulators, as the name suggests are used by researchers, developers and QA to design various kinds of networks, simulate and then analyze the effect of various kinds of networks, simulate and then analyze the effect of various parameters on the network performance. A typical network simulator encompasses a wide range of networking technologies and help the users to build complex networks from basic building blocks like variety of nodes and links.With the help of simultors one can design hierarchial networks using various types of nodes like computers, hubs, bridges, routers, optical cross-connects, multicast router, mobile units, MSAUs etc.

Types of Network Simuators

Various types of Wide Area Network(WAN) technologies like TCP, ATM, IP etc and Local Area Network(LAN) technologies like Ethernet, token rings etc. can all be simulated with a typical simulator and the user can test, analyze various standard results apart from devising some nova or routing etc.There are a wide variety of network simulators, ranging from the very simple to the very simple to the very complex. Minimally, a network simulator must enable a user to represent a network topology ,specifying the nodes on the network , the links between those nodes and the traffic between the nodes.More complicated systems may allow the user to specify everything about the protocols used to handle network traffic . Graphical applications allow users to easily visualize the workings of their simulated environment. Text-based applications may provide a less intuitive interface,but may permit more advanced forms of customization.Others, such as GTNets,are programming-oriented,providing a programming framework that the user then customizes to create an application that simulates the networking environment to be tested.

Simulations

Most of the commercial simulators are GUI driven, while some network simulators require input scripts or commands (network parameters). The network parameters describe the state of the network(node placement, existing links) and the events (data transmission, link failures, etc).An important output of simulations are the trace files.Tracefiles can document every event that occurred in the simulation and are used for analysis.Certain simulators have added functionality of capturing this type of data directly from a functioning production environment, at various times of the day, week, or month,in order to reflect average,worst-case,and best-case conditions.Network simulators can also provide others tools to facilitate visual analysis of trends and potential trouble spots.

Simulation Techniques

Most network simulators use discrete event simulation, in which a list of pending "events" is stored , and those events are processed in order, with some events triggering future events—such as the event of the arrival of apacket at one node triggering the event of the arrival of that packet at a downstream node. Some network simulation problems, notably those relying on queueing theory, are well suited to Markov chain simulation in which no list of furure events is maintained and the simulation consists of transiting between different system "states" in a memoryless fashion . Markov chain simulation is typically faster but less accurate and flexible then detailed discrete event simulation. Some simulationa are cyclic based simulations and these are faster as compared to event based simulations.Simulation of networks can be a difficult task.

NS (Simulator)

Ns or the Network simulator(also popularly called ns-2) is a discrete event network simulator. It is popular in academia for its extensibility(due to its open source model) and plentiful online documentation. Ns is popularly used in the simulation of routing and multicast protocols, among others, and is heavily used in ad-hoc networking research. Ns supports an array of popular network protocols, offering simulation results for wired and wireless networks alike. It can be also used as limited –functionality network emulator. Ns is licensed for use under version 2 of the GNU General Public License.

About TCL

Tool Command Language (TCL) is an interpreted script language developed by Dr.John Ousterhout at the University of California, Berkely, and now developed and maintained by Scriptics. Tcl is comparable to : Netscape JavaScript Microsoft's Visual Basic .The UNIX-derived Practical Extraction and Reporting Language IBM's Restructured Extended ExecutorIn general,script languages are easier and faster to code in than the more structured,complied languages such as C and C++. Script languages are sometimes considered good "glue"languages for tying several compiled programs together.Or,as stand-alone programs,they can allow you to create simple but powerful effects on their own.TclBlend is a version of Tcl that can access certain Java languages facilities. Tcl has a companion program,Tool kit (Tk), to help create a Graphical User Interface with Tcl.

About OTCL

OTCL is an object orinted extension of Tcl and created by David Wetherall.It is used in network simulator(NS-2) and usually run under Unix environment.

About GEDIT

GEDIT is a UTF-8 compatible text editor for the GNOME computer desktop environment. Designed as a general purpose text editor,gedit emphasizes simplicity and ease of use.It includes tools for editing source code and structured text such as markup languages.It is designed to have a clean,simple graphical user interface according to the philosophy of the GNOME project ,and it is the default text editor for GNOME.

Gedit includes syntax highlighting for various program code and text markup formats. Gedit also has GUI tabs for editing multiple files.Tabs can be moved between various windows by the user.It can edit remote files using GVFS(Gnome VFS is now deprecated) libraries.It supports a full undo and redo system aS well as search and replace.Other typical code orinted features include line numbering,bracket matching,text wrapping, current line highlighting, automatic indentation and automatic file backup.Some advanced features of gedit include Multilanguage spellchecking and a flexible plugin system allowing to dynamically add new features.

Modules

- Network creation and routing
- Analysing power consumption
- Result analzation
- Ergs algorithm implementation
- Result analysis and comparison

Network Creation and Routing

In this module, a sample network is to be created. A network with 'n' number of nodes is to be created. All the nodes are deployed randomly across the network. All the nodes can communicate each other. The wireless properties are given to the network. Since our network is Sensor Network, a BASE STATION should be created. To configure the base station a patch file “sensorsim-2.27” is to be added The normal sensor nodes are to be configured in the network. A protocol called AODV is to be implemented to route the packets across the network. UDP, NULL agents are used to configure the sender and receiver nodes. CBR (constant bit rate) is provided with the sender and receiver that provide the packet flow between the nodes.

Analysing Power Consumption

To face the short lifetime of the WSN, the objective is to optimise energy consumption while maintaining the full sensing coverage. that significant energy savings can be achieved by scheduling node’s activities in high-density WSNs. Specifically, some nodes are scheduled to sleep whereas the remaining ones provide continuous monitoring to minimise the number of active nodes in order to maximise the network lifetime and at the same time to ensure the required quality of service (QoS) for applications. Particularly, coverage may be considered as the measure of the QoS of the sensing function for a WSN.

Result Analzation

At optimising the functioning of the WSN, while conserving, as long as possible, the full coverage of the target area, by preserving the redundant nodes. Thus, a self-scheduling algorithm is introduced. The remainder of this paper is organised as follows: The coverage problem and related work addressing the scheduling of the node’s activities.

Ergs Algorithm Implementation

The ERGS algorithm aims to provide full coverage over an area of interest whileminimising the number of active nodes. Thus, it maximises the duration of the coverage and, consequently the WSN lifetime. Such decisions must be done using minimumknowledge and messages exchange, and must ensure the robustness of the decisions despite the loss of messages

Result Analysis and Comparison

To compare the proposed and the considered algorithms, some simulations were made on static WSNs.The performance results are shown here and graphical representation shows the efficiency of proposed algorithm.

Literature Survey

Arms Chris Townsend, S.: ‘Wireless sensor networks: principles and applications’, in Wilson, J.S. (Ed.): ‘Sensor technology handbook’ (Jon S. Wilson 2004), pp. 439–450

Wireless sensor networks can be used by various applications such as surveillance, forest management, weather prediction, Avalanche land-slide prediction, road safety, marine movement control, etc. These applications pose a set of common difficulties. Specifically, in the remote large-scale networks, network topology, security, self-configuration, connectivity, maintenance, power management, time synchronization etc. are major challenges. In this paper we present an overview of issues related of wireless sensor networking. Different aspects of sensor networking are discussed and sensor network architecture is proposed that can satisfactorily overcome these problems.

Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.:‘Coverage problems in wireless ad-hoc sensor networks’. INFOCOM 2001: 20th Annual Joint Conf. IEEE Computer and Communications Societies, 2001, vol. 3, pp. 1380–1387

In this paper, we address one of the fundamental problems, namely coverage. Coverage in general, answers the questions about quality of service (surveillance) that can be provided by a particular sensor network. We first define the coverage problem from several points of view including deterministic, statistical, worst and best case, and present examples in each domain. By combining computational geometry and graph theoretic techniques, specifically the Voronoi diagram and graph search algorithms, we establish the main highlight of the paper - optimal polynomial time worst and average case algorithm for coverage calculation. We also present comprehensive experimental results and discuss future research directions related to coverage in sensor networks

Thai, M.T., Wang, F., Hongwei Du, D., Jia, X.: ‘Coverage problems in wireless sensor networks: designs and analysis’,Int. J. Sens. Netw.,2008,3, (3), pp. 191–200

Wireless sensor networks ensure a wide range of applications [2], starting from security surveillance in military and battlefields, monitoring previously unobserved environmental phenomena, smart homes and offices, improved healthcare, industrial diagnosis, and many more. For instance, a sensor network can be deployed in a remote island for monitoring wildlife habitat and animal behavior ornear the crater of a volcano to measure temperature, pressure, and seismic activities.

Ammari, H.M., Giudici, J.: 'On the connected k-coverage problem in heterogeneous sensor nets: the curse of randomness and heterogeneity'. ICDCS '09: Proc. 2009 29th IEEE Int. Conf. on Distributed Computing Systems, 2009, pp. 265–272

Coverage is an essential task in sensor deployment for the design of wireless sensor networks. While most existing studies on coverage consider homogeneous sensors, the deployment of heterogeneous sensors represents more accurately the network design for real-world applications. In this paper, we focus on the problem of connected k-coverage in heterogeneous wireless sensor networks. Precisely, we distinguish two deployment strategies, where heterogeneous sensors are either randomly or pseudo-randomly distributed in a field. While the first deployment approach considers a single layer of heterogeneous sensors, the second one proposes a multi-tier architecture of heterogeneous sensors to better address the problems introduced by pure randomness and heterogeneity.

Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., Gill, C.: 'Integrated coverage and connectivity configuration in wireless sensor networks', ACM Trans. Sens. Netw., 2005, 1, (1), pp. 28–39

Sensing coverage characterizes the monitoring quality provided by a sensor network in a designated region. Different applications require different degrees of sensing coverage. While some applications may only require that every location in a region be monitored by one node, other applications require significantly higher degrees of coverage. For example, distributed detection requires every location be monitored by multiple nodes, and distributed tracking and classification [9] requires even higher degrees of coverage. The coverage requirement also depends on the number of faults that must be tolerated. A network with a higher degree of coverage can maintain acceptable coverage in face of higher rates of node failures. The coverage requirement may also change after a network has been deployed due to changes in application modes or environmental conditions. For example, a surveillance sensor network may initially maintain a low degree of coverage required for distributed detection

Zhang, H., Hou, J.: 'Maintaining sensing coverage and connectivity in large sensor networks', Ad Hoc Sens. Wirel. Netw., 2005, 1, (2) pp. 89–124

Based on the optimality conditions, we then devise a decentralized density control algorithm, Optimal Geographical Density Control(OGDC), for density control in large scale sensor networks. The OGDC algorithm is fully localized and can maintain coverage as well as connectivity, regardless of the relationship between the radio range and the sensing range. Ns-2 simulations show that OGDC outperforms existing density control algorithms with respect to the number of working nodes needed and network lifetime (with up to50% improvement), and achieves almost the same coverage as the algorithm with the best.

Conclusion

In this paper, the problems of energy conservation and full sensing coverage in large WSNs where nodes are randomly deployed have been addressed. Specifically, an original algorithm, the ERGS algorithm, has been introduced based on a wake-up scheduling concept allowing one to extend the lifetime of the WSN. The ERGS algorithm relies on the novel idea of exploiting the remaining energy in making decision on which node has to enter sleep state. The first main feature of the ERGS algorithm consists in applying an equity principle by balancing the remaining energy of nodes. This has contributed to extend the WSN lifetime. The second main feature consists in avoiding negotiation phases, as decision to enter sleep state uses a computed priority based on a one-hop neighbourhood knowledge.

Future Work

The whole coverage of the target area after the deactivation of redundant nodes is formally proven in this paper. This has shown that the local feature of the deactivation decisions has no impact on the general coverage. Obviously, the reliability of the ERGS algorithm relies on the hypothesis of an initial coverage of the WSN with a sufficient redundancy of nodes. It would be interesting, in a future work, to precisely determine how sufficient redundancy must be in a real implementation, and what impact the redundancy degree has on the performance of the algorithm.

References

[1] Arms Chris Townsend, S.: 'Wireless sensor networks: principles and applications', in Wilson, J.S. (Ed.): 'Sensor technology handbook' (Jon S. Wilson 2004), pp. 439–450

[2] Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.:'Coverage problems in wireless ad-hoc sensor networks'. INFOCOM 2001: 20th Annual Joint Conf. IEEE Computer and Communications Societies, 2001, vol. 3, pp. 1380–1387

[3] Thai, M.T., Wang, F., Hongwei Du, D., Jia, X.: 'Coverage problems in wireless sensor networks: designs and analysis', Int. J. Sens. Netw., 2008, 3, (3), pp. 191–200
[4] Ammari, H.M., Giudici, J.: 'On the connected k-coverage problem in heterogeneous sensor nets: the curse of randomness and heterogeneity'. ICDCS '09: Proc. 2009 29th IEEE Int. Conf. on Distributed Computing Systems, 2009, pp. 265–272
[5] Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., Gill, C.: 'Integrated coverage and connectivity configuration in wireless sensor networks', ACM Trans. Sens. Netw., 2005, 1, (1), pp. 28–39
[6] Zhang, H., Hou, J.: 'Maintaining sensing coverage and connectivity in large sensor networks', Ad Hoc Sens. Wirel. Netw., 2005, 1, (2) pp. 89–124
[7] Chang, J.-H., Tassiulas, L.: 'Energy conserving routing in wireless adhoc networks'. INFOCOM 2000: 19th Annual Joint Conf. IEEE Computer and Communications Societies, 2000, pp. 22–31.
[8] Zhang, H., Wang, H., Feng, H.: 'A distributed optimum algorithm for target coverage in wireless sensor networks'. Asia-Pacific Conf. on Information Processing, 2009, pp. 144–147
[9] Chen, A.: 'Designing localized algorithms for barrier coverage'. Proc. ACM MobiCom'07, 2007