PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks

Fan Ye, Gary Zhong, Songwu Lu, Lixia Zhang
{yefan, gzhong, slu, lixia}@cs.ucla.edu
UCLA Computer Science Department, Los Angeles, CA 90095-1596

1 Introduction

Small, inexpensive sensors with limited memory, computing power and short battery lifetimes are turning into reality. Due to adverse conditions such as high noise levels, extreme humidity or temperatures, or even destructions from unfriendly entities, sensor node failures may become norms rather than exceptions in real environments. To be practical, sensor networks must last for much longer times than that of individual nodes, and have yet to be robust against potentially frequent node failures.

This paper presents the design of PEAS[1], a simple protocol that can build a long-lived sensor network and maintain robust operations using large quantities of economical, short-lived sensor nodes. PEAS extends system functioning time by keeping only a necessary set of sensors working and putting the rest into sleep mode. Sleeping ones wake up now and then, probing the local environment and replacing failed ones. The sleeping periods are self-adjusted dynamically, so as to keep the sensors' wakeup rate roughly constant, thus adapting to high node densities.

PEAS shares the basic technique of turning off nodes with other energy-conserving protocols in wireless ad hoc networks (GAF[4], SPAN[2], AFECA[3]) and sensor networks (ASCENT[1]). However, existing work is intended for much more powerful mobile computers and/or a relatively stable network environment, whereas PEAS design targets at a much harsher or even hostile working environment, where (1) node failures should be considered norms rather than exceptions, (2) due to ad hoc deployment and the need for long-last operations, the nodes' deployment density can be several times or even a magnitude higher than the minimum required for normal functioning, and (3) nodes are too constrained in memory and computing resources to afford relatively complex protocols.

None of the related work addresses the issue of robust operations in the harsh environmental setting that PEAS design has assumed. PEAS achieves highly robust operation by randomizing the sleeping times of sleeping nodes to detect and replace failed nodes actively. It eliminates per-neighbor states, thus removing the complexity to tracking each neighbor in a dense deployment. These are in contrast with related work which either require maintenance of per-neighbor states or estimation of the active times of working nodes, which are difficult when nodes are densely deployed and subject to frequent failures.

[1]PEAS stands for Probing Environment and Adaptive Sleeping.

2 PEAS Design

PEAS consists of two simple algorithms: Probing Environment and Adaptive Sleeping, which determine (1) which sensors should work and how a wake-up sensor makes the decision of whether going back to sleep, and (2) how the average sleep times of sensors are dynamically adjusted to keep a relatively constant wake-up rate, respectively.

2.1 Probing Environment

The goal of Probing Environment is to maintain a required number of nodes in the presence of node failures. PEAS uses a simple probing mechanism to solve the problem. Initially all nodes are sleeping and they sleep for an exponentially distributed random time. When a node wakes up, it sends a PROBE message within a certain probing range R_p. Any working nodes within R_p should send back a REPLY message. A sleeping node starts working continuously only if it does not hear any REPLY message. Otherwise, it goes back to sleep again for another random time.

The exponentially distributed sleeping time allows us to measure the nodes' wakeup rate and adjust sleeping periods easily in Adaptive Sleeping (Section 2.2). The probing range R_p is given by the application depending on the degree of robustness it needs. An application requiring highly robust functioning may choose a small R_p to achieve a greater density, thus higher redundancy of working nodes. To avoid disconnecting the network, R_p is generally much less than the maximum transmission range R_t of a sensor. We assume the nodes can adjust their transmitting power to broadcast PROBE messages given the probing range R_p.

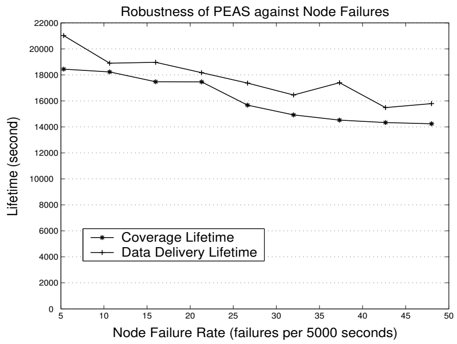

Figure 1. Robustness against node failures

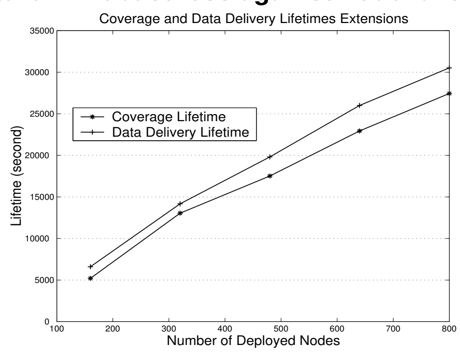

Figure 2. Extension of lifetimes

2.2 Adaptive Sleeping

The goal of Adaptive Sleeping is to keep the number of wakeups, thus the overhead, constant in unit time and independent from the local node densities, which vary in different deployments, at different locations or times. The desired frequency of wakeups is decided by the application. Adaptive Sleeping adjusts the wakeups of sleeping neighbors for each working node at appropriate levels, so that transient interruptions in sensing and communicating caused by node failures are within what is tolerated by the application.

The basic idea is to let each working node measure the aggregate probing rate $\bar{\lambda}$ it perceives from all its sleeping neighbors. The working node then includes the measured rate $\bar{\lambda}$ when sending a REPLY message to a probing neighbor. Each probing node then adjusts its sleeping times accordingly. The details are in a technical report [5].

3 Summary of Performance Evaluation

We implemented PEAS and evaluated its performance by two main metrics: coverage lifetime and data delivery lifetime. Coverage lifetime denotes the period during which the network has enough working nodes to ensure every place is being monitored by working nodes. Data delivery lifetime represents the time during which the network can deliver data reports successfully.

The results show that PEAS maintains robust operation against up to 38% node failures (besides energy exhaustions) at less than 1% energy overhead, and extends the system functioning time in proportion to the number of deployed nodes.

We first vary the node failure percentage from 7% to 38%. The coverage lifetime and data delivery lifetime decrease gracefully as more nodes fail (Figure 1). Even for the 38% node failures, the deductions are about 20%. Then we vary the node number in a field from 1 to 5 times the number needed for basic sensing coverage with a fixed 13% node failures. Both the coverage and data delivery lifetimes increase linearly to the number of deployed nodes (Figure 2). Finally we measure the overhead of PEAS in the energy consumed by probing and replying operations and find it is less than 1% of the total energy consumption.

4 Conclusions and Future Work

Existing energy-saving protocols have not paid adequate attention to the unique issues of unexpected node failures and extremely dense deployment in sensor networks. In this paper we presented the design of PEAS which achieves robustness and adaptivity from minimized protocol operations, reduced message exchanges, and elimination of per-neighbor states. We conducted simulations and analysis which confirmed the robustness and efficiency of PEAS.

We are continuing our efforts to further enhance PEAS. While PEAS has demonstrated its effectiveness in controlling working node density, we believe its performance can be further improved through more examinations of its interaction with the sensing data forwarding protocol. In addition to comparing PEAS' performance with related work, we also plan to develop formal analysis to estimate its performance bounds.

References

[1] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. *INFOCOM 2002, New York, NY, USA*.

[2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. SPAN: An Energy Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. *MOBICOM 2001*.

[3] Y. Xu, J. Heidemann, and D. Destrin. Adaptive Energy-Conserving Routing for Multihop Ad Hoc Networks. *USC/ISI Research Report 527*, October, 2000.

[4] Y. Xu, J. Heidemann, and D. Estrin. Geography Informed Energy Conmservation for Ad Hoc Routing. *MOBICOM 2001*.

[5] F. Ye, S. Lu, and L. Zhang. GRAdient Broadcast: A Robust, Long-lived Large Sensor Network. http://irl.cs.ucla.edu/papers/grab-tech-report.ps, 2001.