International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 4 Issue 6, June-2015, pp: (102-105), Impact Factor: 1.252, Available online at: www.erpublications.com

A Survey: Cluster Head Election for Minimum Average Energy Consumption in WSN

Shalini[1], Yudhvir Singh[2]
[1,2]University Institute of Engineering and Technology, Maharishi Dayanand University, Rohtak, India

ABSTRACT: Wireless sensor networks are increasingly prepared to handle more difficult functions, in-network processing may require these Threshold powered sensors to carefully use their constrained energy to extend the effective network life time exclusively in a heterogeneous settings. Clustered techniques have since been active to optimize energy consumption in this energy constrained wireless sensor networks. We introduce improved LEACH clustering algorithm in a three-tier node state to prolong the effective network life-time and average energy consumption with saturated round of nodes. This review would allow developers to directly observe the entire environment at once, while also providing access to all the debugging tools of a modern compiler environment. Our program attempts to mix the review of communication and routing protocols among sensors with virtual representations of a real world object, such as the sensors and other entities in the environment.

1. INTRODUCTION

In each sensory device is modeled as a separate object running in its own thread. Elements of the environment are modeled as additional objects, and the sensors may query aspects of the environment based upon their physical position and sensor type to determine information about environmental objects. LEACH (Low Energy Adaptive Clustering Hierarchy) is one of the most popular clustering algorithms for WSN. It forms clusters based on the received signal strength and uses the CH (Clustered Head) nodes as routers to the base-station. All the data processing such as data fusion and aggregation are local to the cluster. LEACH forms clusters by using a distributed algorithm, where nodes make autonomous decisions without any centralized control. Initially a node decides to be a CH with a probability p and broadcasts its decision. Each non-CH node determines its cluster by choosing the CH that can be reached using the least communication energy. The role of being a CH is rotated periodically among the nodes of the cluster in order to balance the load. The rotation is performed by getting each node to choose a random number T(n) between 0 and 1. A node becomes a CH for the current rotation round if the number is less than the following threshold

T (n) = p/1-p*(rmod1/p): if n E G
T (n) = 0 : otherwise

Where p is the desired percentage of CH nodes in the sensor node, r is the current round number, and G is the set of nodes that have not been CHs in the last 1/p rounds. Since the decision to change the CH is probabilistic, there is a good chance that a node with very low energy gets selected as a CH. When this node dies, the whole cell becomes dysfunctional. Also, the CH is assumed to have a long communication range so that the data can reach the base station from the CH directly.

This protocol is divided into rounds; each round consists of two phases:

Set-up Phase

- Advertisement Phase
- Cluster Set-up Phase

Steady Phase

- Schedule Creation
- Data Transmission

Setup Phase

Each node decides independent of other nodes if it will become a CH or not. This decision takes into account when the node served as a CH for the last time (the node that hasn't been a CH for long time is more likely to elect itself than nodes that have been a CH recently). In the following advertisement phase, the CHs inform their neighborhoods with an advertisement packet that they become CHs. Non-CH nodes pick the advertisement packet with the strongest received signal strength.

Steady-state Phase

Data transmission begins; Nodes send their data during their allocated TDMA slot to the CH. This transmission uses a minimal amount of energy (chosen based on the received strength of the CH advertisement).

2. MOTIVATION

Our goal is to simulate not only a sensor network, but also the physical environment in which it operates. We reviewed several samples of existing literature to determine whether anyone had previously attempted this kind of paper. For details, see the "Related work" section. We found that there were several packages for simulating wireless networks, but in general they seem to rely on external input for the signals that would occur in a real world environment.

In a time when physics students are hired to make increasingly realistic games, we see no reason why the physical events in a typical environment should not also be simulated. One advantage of this approach is that a developer could essentially create any event that he or she wishes to test, without going through the trouble of purchasing (for example) real tanks and airplanes to detect. A robust system would, of course, be capable of switching between input from a simulated object and real data retrieved from sensors. Providing a state of the art physics review is obviously beyond the scope of this paper; we are only concerned here with modeling and displaying a simple world review.

The goal for the sensor network routing protocol modeled was to achieve low message overhead, scalability and a relatively simple implementation. These goals allow our review to accurately model a real world sensor network routing protocol. We considered three types of sensor network protocols, "Multi-hop routing protocols", "Zone Based routing" and Low Energy Adaptive Cluster Head routing protocol (LEACH) [1]. Of these, the LEACH protocol was the most suitable for our implementation. For more details, see the "Related work".

3. RELATED WORK

3.1. Target Tracking Algorithms

The "hierarchical multiple tracking Algorithm" [3] implemented randomization of targets and focused on tracking multiple targets. The algorithm took false alarms and noise into consideration for a more realistic application. The design was scalable, autonomous, and supported low communication. The algorithm used a weighted sum in conjunction with a shortest path calculation. All data was sent to "super-nodes" that were used to convey information. This implementation is similar to our approach, but does not implement clusters.

3.2. Network Simulators

We considered three existing examples of sensor network reviews: MATLAB, J-Sim, and Ptolemy II. While these papers contained important ideas, none of them were concerned with the goal that we were trying to accomplish. In their own words:

"MATLAB captures the behavior and interactions of networks of thousands of TinyOS motes at network bit granularity… The MATLAB architecture is composed of five parts: support for compiling TinyOS component graphs into the review infrastructure, a discrete event queue, a small number of re-implemented TinyOS hardware abstraction components, mechanisms for extensible radio and ADC models, and communication services for external programs to interact with a review."[4]

Ptolemy II -
The Ptolemy II paper studies heterogeneous modeling, review, and design of concurrent systems. The Ptolemy II software has the ability to model complex systems through a simple actor/component based user interface."[6]

These two applications were designed to model communication within a network, but do not model physical objects in the real world. MATLAB does allow external input to handle simulated physical objects, but does not provide its own method for modeling these objects. Ptolemy simulates sensor nodes at a very low level, dealing with ports and battery life. Our implementation is not concerned with the low level behavior of the sensor nodes. Instead, we are working at a level that might best be described as middleware. The paper on J-Sim[7] does mention several scenarios where target tracking is modeled, but in each case they rely on input captured from video or other representations of real events.

3.3. Sensor Networks

In constructing the review environment, we considered three sensor network routing protocols: "Multi-hop routing protocols", "Zone Based routing" and Low Energy Adaptive Cluster Head routing protocol (LEACH) [1].

3.4. Multi-hop routing protocols or static clustering

The "Multi-hop routing protocols or static clustering" [8, 9] use clusters to broadcast information about energy to static members of the cluster. We found that this implementation was not very scalable, so we could not model large sensor networks.

3.5. Zone Based routing

The "Zone Based routing" [8, 10] is a hierarchical approach to routing in sensor networks which groups nodes into geographic zones to control routing. It uses energy estimation based on data transmission direction. This implementation was not chosen because geographic networks were not being simulated.

3.6. LEACH

The LEACH routing protocol was designed to save power by randomizing the energy distribution in sensor networks. The sensor nodes form "clusters" and elect a "cluster-head" based on probability. We found this protocol most suitable to our implementation. It provided low message overhead, scalability, and was easy to simulate. Our implementation differs from LEACH because we do not simulate energy consumption or the randomization of cluster-heads. This applies most directly to sensor networks with wired sensors. Otherwise it adheres to LEACH, in that we implement similar cluster-head election, cluster formation, and cluster communication.

4. NETWORK ARCHITECTURE

We created the following class hierarchy:

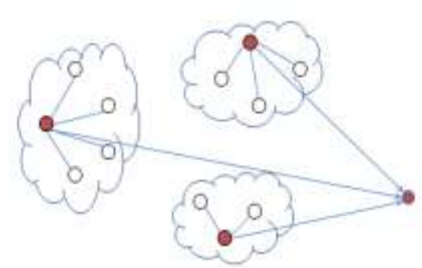

Figure 1: Node Cluster Hierarchy

Our physical world is represented by the Environment class, which serves as an all-seeing observer. The environment contains all instances of "Entities", which are our term for any object that has a spatial location. The car starts in a fixed location; the Monitor is an omniscient Entity not represented on the screen; all sensors are created dynamically by the user after the program starts.

5. DISCUSSION

The discussion for the paper included the ability to have our review framework be applicable to disparate underlying sensor networks. Many measures were taken in order to ensure middleware portability of the code, and this pays dividends to the review technology as a whole. An elaborate and highly extensible hierarchy, as described above, sets a precedent by incorporating robust, extensible functionality at different levels of the API, allowing for intuitive and rapid development. Sensors natively support management of neighbors. This allows for specific Sensor subtypes to be implemented easily which have innate neighbor-relational functionality. One level above them, Nodes include abstractions for messaging – sending, handling, enqueuing and dequeuing. This layer allows for the abstraction of different networking protocols and services as well as the message types and formats that are sent between network nodes. A cluster is a collection of nodes organized by proximity to send messages to a central point (cluster-head). A cluster-head is a node that has been identified by other nodes to forward messages to a base station. After organization, the cluster-heads stay awake to receive messages from cluster members. The other cluster members wake periodically to check for a proximity violation.

Cluster-head election takes place during "Review Phase". Sensors elect a cluster-head by calculating the distance (cluster radius) between its position and other sensor nodes. If a node is within the cluster radius, it is considered as part of the same cluster. If the sensor is part of the same cluster, the current cluster-head identification is compared against the new sensor's identification. If the new sensor has a lower identification, it is considered to be the cluster-head.

6. CONCLUSION

Our review does indeed succeed at the challenges it set out to address. It is our belief that the architecture of this product is sufficiently in the realm of extensibility and adaptability that it forms a valuable framework for development of sensor network reviews for all types of environmental anomalies, network protocols, and node types. **For future work, better** modeling of how sound and vibrations work, etc. Our review of the LEACH protocol made some attempt to model the fact that sensors have limited communication range; however, they are allowed to freely communicate with the Monitor from any location.

REFERENCES

[1]. Heinzelman , Wendi; Chandrakasan, Anantha; Balakrishnan, Hari: Low Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Massachusetts Institute of Technology, Cambridge.
[2]. Cugola, G; Jacobsen, HA. "Using publish/subscribe middleware for mobile systems." ACM SIGMOBILE Mobile Computing and Communications Review, 2002.
[3]. Oh, Songhwai and Sastry, Shankar. A hierarchical multiple tracking Algorithm for Sensor networks. Berkeley CA, Jan 2004. http://webs.cs.berkeley.edu/retreat-1-04/slides/songhwai_oh-hmtt_nest-retreat04.ppt
[4]. Levis, Philip; Lee, Nelson; Welsh, Matt; and Culler, David. MATLAB: Accurate and Scalable Review of Entire TinyOS Applications. http://www.cs.berkeley.edu/~pal/pubs/MATLAB-sensys03.pdf
[5]. Levis, Philip and Lee, Nelson. MATLAB: A Simulator for TinyOS Networks. http://www.cs.berkeley.edu/~pal/pubs/nido.pdf
[6]. Baldwin, Philip J. Sensor network Modeling and Review in Ptolemy 2. Berkeley CA, Aug 2003. http://chess.eecs.berkeley.edu/papers/ITR/2003/BaldwinPaper.pdf
[7]. Ahmed Sobeih, Wei-Peng Chen, Jennifer C. Hou, Lu-Chuan Kung, Ning Li, Hyuk Lim, Hung-Ying Tyan, and Honghai Zhang. J-Sim: A Review and Emulation Environment for Wireless Sensor Networks http://www.j-sim.org/v1.3/sensor/JSim.pdf
[8]. Kyatham, Raghavendra. Algorithms in sensor networks. http://ranger.uta.edu/~ gdas/Courses/Fall2004/advAlgos /finalPresentations /Raghavendra_Algorithms%20in%20sensor%20networks.ppt#256
[9]. Woo, Ale; Tong, Terrence; Culler, David. Taming the underlying Challenges of Reliable Multihop Routing in Sensor networks. Berkeley CA, ACM 2003. http://www.cs.berkeley.edu/~awoo/sensys_awoo03.pdf
[10]. Li, Qun; Aslam, Javed; Rus, Daniella. Hierarchical Power-aware Routing in Sensor Networks Rutgers University, May 21, 2001.http://www.cs.wm.edu/~liqun/paper/dimacs.pdf