ISSN 2348 – 7968

Comparison of LEACH (Low Energy Adaptive Clustering Hierarchy) Protocols for Wireless Sensor Networks

Manohar R[1], Byregowda B K[2]

[1] Department of Information Science and Engineering, MVIT, Bangalore-562157, Karnataka, India
[2] Department of Information Science and Engineering, MVIT, Bangalore-562157, Karnataka, India
mailmanu.r@gmail.com byregowda008@gmail.com

ABSTRACT

Wireless sensor networks are formed by a large number of sensor nodes which are commonly known as motes. These motes are small in size and have limited processing power, memory and battery life. Motes typically have sensors such as thermometers attached to them in order to gather data about the physical environment the device is a part of. Collectively they are capable of forming wireless ad-hoc networks in order to relay this sensor data throughout the network. In this paper a general overview of wireless sensor network technology is presented as well as an comparison of different versions of LEACH protocols for sensor networks. In this paper comparison of protocols defines, which can have significant impact on the overall power consumption, delay and delivery ratio of these networks. In comparison with Flat routing protocols there is low power usage and least delay with LEACH protocol can be observed in simulation.

Keywords **-Wireless sensor networks, Hierarchical routing protocols, Tiny OS, NesC, LEACH**

I INTRODUCTION TO WIRELESS SENSOR NETWORK

This paper is concerned with wireless sensor networks which are a type of adhoc network. Wireless sensor networks consist of a large number of sensor devices which are commonly known as motes. Motes are small devices with sizes typically ranging from matchbox size to the size of a pen tip. Motes usually consist of a battery, low clock rate processor, a small amount of memory and a component to allow wireless communication[1]. Motes have been built which use active communication (e.g. radio frequency or a laser module) and passive communication (e.g. a corner cube reflector to modulate and reflect laser beams aimed at the mote). Motes may also have sensors attached to them to monitor the physical environment in some way.

These sensors can be built directly into the motes main board or can come as add on boards which can be connected in some way to the mote. Wireless sensor networks are very versatile and can be used in many different application areas. Recent examples of applications developed for these networks include tracking military vehicles and monitoring forest fires. To support these applications the TinyOS operating system has been developed to control the operation of the mote devices. Among other things TinyOS provides a customizable networking stack which provides message passing functions. This allows motes to form an ad hoc network in order to communicate with other motes and possibly other types of devices. This paper is concerned with routing in wireless sensor networks. The scalability, limited processing power, memory and battery life of the motes present many challenges when it comes to routing in these networks. This paper will look at LEACH routing protocols to assess their suitability for use in wireless sensor networks. Simulation is done in TinyOS using modules written in NesC are explained in further sections of the paper.

TINYOS

TinyOS is an operating system that is designed to manage the operation of a variety of mote devices & attached sensors. Its libraries and applications are all written in the NesC language. In order to customize motes for different purposes a single application can be integrated with the TinyOS structure. When TinyOS is compiled all selected components, custom components and the application is integrated into a single multi-threaded program.

TinyOS defines components, implementation and concurrency model. Components in TinyOS: *modules* and *configurations*. Modules provide application code, implementing interfaces. Configurations are used to assemble other components together. This is called *wiring*. TinyOS executes only one program consisting of selected system components and custom components needed for a single application. There are two threads of execution: *tasks* and *hardware event handlers[2]*. Tasks are functions do not preempt one another. Hardware event handlers are executed in response to a hardware interrupt and also run to completion, but may preempt.

TinyOS provides the following: TOSSIM a discrete event simulator which has been designed to simulate results of sensor network applications. In real world simulates on motes. TinyDB a small database to store data or reading during data aggregation and simulation. TinyViz is graphical visualize tool in TinyOS used in connection with TOSSIM to view the results of any NesC application. As simulator gives a readable output, TinyViz provides graphical presentation of nodes and its connectivity. Provides an extensible graphical user interface for debugging, visualizing and interacting with TOSSIM.

NESC

The NesC language [3] is an extension of the C programming language which was designed to facilitate the implementation of the structuring concepts and execution model of TinyOS. NesC was primarily designed for use with embedded systems such as sensor networks. In NesC there is a separation of construction and composition. Programs are built out of components which are 'wired' together to form whole programs. Concurrency is supported in NesC in a way which supports the concurrency model of TinyOS. Concurrency in NesC is based on run-to-completion tasks and interrupts handlers. Interrupt handlers may interrupt tasks as well as other interrupt handlers. NesC uses the filename extension ".nc" for all source files -- interfaces, modules, and configurations.

II CLASSIFICATION OF ROUTING PROTOCOLS

Classification of protocols in Wireless Sensor Networks is done in different levels based on either application or network structure [4]. The features and functioning of these protocols will differ. Fig 1 displays classification of routing protocols.

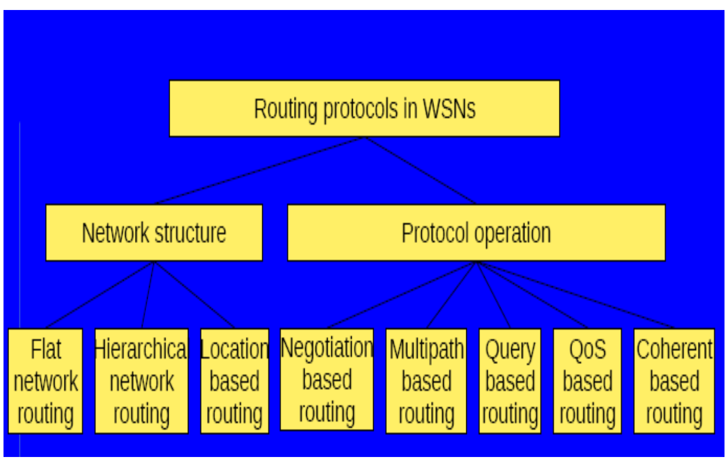

Fig 1. Classification of routing protocols.

III INTRODUCTION TO HIERARCHICAL ROUTING PROTOCOL

Hierarchical or cluster-based routing, originally proposed in wireline networks, are well-known techniques with special advantages related to scalability and e±cient communication. As such, the concept of hierar-chical routing is also utilized to perform energy-effcient routing in WSNs. In a hierarchical architecture, higher energy nodes can be used to process and send the information while low energy nodes can be used to perform the sensing in the proximity of the target. This means that creation of clusters and assigning special tasks to cluster heads can greatly contribute to overall system scalability, lifetime, and energy effciency. Hierarchical routing is an effcient way to lower energy consumption within a cluster and by performing data aggregation and fusion in order to decrease the number of transmitted messages to the BS. Hierarchical routing is mainly two-layer routing where one layer is used to select clusterheads and the other layer is used for routing. However, most techniques in this category are not about routing, rather on "who and when to send or process/aggregate" the information, channel allocation etc., which can be orthogonal to the multihop routing function.

a. Low-Energy Adaptive Clustering Hierarchy (LEACH)

LEACH is an adaptive clustering routing protocol proposed by Wendi B. Heinzelman, et al. The implementation process of LEACH includes many rounds. Each round consists of the setup phase and the steady data transmission phase. In the set-up phase, the cluster head nodes are randomly elected from all the sensor nodes and several clusters are constructed dynamically. In the steady data transmission phase, member nodes in every cluster send data to their own cluster head, the cluster head compresses the data that received from member nodes and sends the compressed data to the sink node. LEACH protocol periodically[5] elects the cluster head nodes and re-establishes the clusters according to a round time, which ensures energy dissipation of each node in the network is relatively evenly.

b. M-LEACH

The leach protocol [6] is a dynamic cluster based protocol. The leach protocol assumes that the network consists of a remote base station and a set of homogeneous sensor nodes that are energy constrained. All sensor nodes are capable of direct communication with the base station but this is expensive in terms of energy usage. Leach reduces the number of nodes that communicate directly with the base station by forming clusters. Leaf nodes connect to the cluster head which requires the least amount of power to communicate with. The cluster head nodes connect directly to the base station. Cluster head nodes allocate each leaf node that connects to it a time slot to communicate in. This allows the leaf nodes to sleep between its allocated communication slots.

Once the cycle of slots has completed the data from that cycle is aggregated by the cluster head to save power and then it is sent back to the base station. When the cluster heads are fixed the cluster heads are required to do a lot of work which requires a lot of energy. Eventually this would lead to node failure. In order to avoid this leach uses rounds. At the beginning of each round each node in the network decides if it should become a cluster head based on a probability factor and whether it has been a cluster head in a previous round. This results in dynamic clusters with each node taking a turn in forwarding packets to the base station. This reduces the energy drain on particular nodes caused by static clusters and spreads energy usage more evenly across the network.

c. E-LEACH

Fan el. Al. [7] proposes a new protocol Energy-Leach which improves the CH selection procedure. Like LEACH protocol, E-LEACH protocol also devided into rounds, in the first round, every node has the same probability to turn into CH, that mean nodes are randomly selected as CHs, in the next rounds, the residual energy of each node is different after one round communication and taken into account for the selection of the CHs. That mean nodes have more energy will become a CHs rather than nodes with less energy. E-LEACH tries to optimize the energy consumption of the network by ensuring that nodes belonging to hot regions have a high probability of becoming a cluster heads. Thus nodes belonging to hot regions, which are expected to transmit data more frequently, now do it over shorter distances, thereby leading to balanced energy consumption over the network. E-LEACH selects a node to be a cluster head depending upon its hotness value and residual energy. This is an improvement over stochastic approach used in LEACH in terms of energy efficiency. The E-LEACH approach considers two additional parameters for cluster-head selection. These are the residual energy of a node and the hotness of the region sensed by the node. These two factors are used in a fashion which leads to Spatio-temporal adaptation for optimum energy usage.

d. C-LEACH

Wendi *et al.* [8] proposed LEACH-C protocol which uses a centralized algorithm. LEACH-C protocol can produce better performance by dispersing the cluster heads throughout the network. During the set-up phase of LEACH-C, each node sends information about its current location (possibly determined using GPS) and residual energy level to the sink. In addition to determining good clusters, the sink needs to ensure that the energy load is evenly distributed
among all the nodes. To do this, sink computes the average node energy, and determines which nodes have energy below this average. The steady-state phase of LEACH-C is identical to that of the LEACH protocol.

e. V-LEACH

In our new version of LEACH[9] protocol, the cluster contains; CH (responsible only for sending data that is received from the cluster members to the BS), vice-CH (the node that will become a CH of the cluster in case of CH dies), cluster nodes (gathering data from environment and send it to the CH). In the original leach, the CH is always on receiving data from cluster members, aggregate these data and then send it to the BS that might be located far away from it. The CH will die earlier than the other nodes in the cluster because of its operation of receiving, sending and overhearing. When the CH die, the cluster will become useless because the data gathered by cluster nodes will never reach the base station. In our V-LEACH protocol, besides having a CH in the cluster, there is a vice-CH that takes the role of the CH when the CH dies because the reasons we mentioned above. By doing this, cluster nodes data will always reach the BS; no need to elect a new CH each time the CH dies. This will extend the overall network life time.

f. TL-LEACH

Two-Level Hierarchy LEACH[10] is a proposed extension to the LEACH algorithm. It utilizes two levels of cluster heads (primary and secondary) in addition to the other simple sensing nodes. In this algorithm, the primary cluster head in each cluster communicates with the secondaries, and the corresponding secondaries communicate with the nodes in their sub-cluster. Data-fusion can also be performed as in LEACH. In addition, communication within a cluster is still scheduled using TDMA time-slots. The organization of a round will consist of first selecting the primary and secondary cluster heads using the same mechanism as LEACH, with the a priori probability of being elevated to a primary cluster head less than that of a secondary node.

Communication of data from source node to sink is achieved in two steps:

1. Secondary nodes collect data from nodes in their respective clusters. Data-fusion can be performed at this level.

2. Primary nodes collect data from their respective secondary clusters. Data-fusion can also be implemented at the primary clusterhead level.

The two-level structure of TL-LEACH reduces the amount of nodes that need to transmit to the base station, effectively reducing the total energy usage.

IV SIMULATION AND RESULTS

Metrics

The following metrics will be used when evaluating the protocols:

• Latency – the time taken to deliver a packet to the base station from the origin node will be looked at when evaluating the protocols. In addition the per hop time delay will also be looked at. Lower latency is preferable to higher latency.

• Battery usage – the amount of power used during the simulation will be monitored and used for evaluating the protocols. Batteries have a finite amount of power and nodes die once power runs out. For this reason lower power usage is preferable to higher power usage. In addition the distribution of power usage across the network will be looked at. Uniform Drain is preferable.

• Loss of data – the number of packets received from a node at the base station will be compared with the number of packets sent by a node in order to calculate the number of packets lost. A low packet loss rate is preferable to a high packet loss rate.

• Connectivity – the number of nodes that have a route to the base station will be used to assess the node connectivity provided by a particular routing protocol. More connected nodes in a network are preferable to fewer connected nodes.

CONCLUSION

During the design and implementation of the protocols it was clear that performance gains could have been made in places. Leach has been evaluated as better than flat and location based routing protocols. Comparatively other protocol uses a large amount of power and would drain the batteries of the nodes quickly. However LEACH has a high connectivity and high loss rate due to packets being dropped once the maximum hop count is reached. The leach protocol uses the least amount of power and also spreads the power drain out across all nodes in a network. Hierarchical routing protocols are non optimal, simple and no collision. As channel allocation is fair energy dissipation is uniform. Considerably lower latency and reduce d duty cycle & periodic.

FUTURE WORK

Future work could include improving the protocols developed during this report and the development of new networking components for the system. Data delivery ratio and connectivity should be improved in LEACH and optimal path module can be developed for efficient functioning at real time.

ACKNOWLEDGEMENT

We would like to thank to Prof. Rajashree V Biradar for anonymous review, helpful guidance and suggestions. This work is summary of academic project.

REFERENCES

[1] Crossbow - Wireless Sensor Networks Product Page, http://www. xbow.com/products/Wireless_Sensor_Networks.htm, April 2004

[2] TinyOS Programming, October 27 2006, Book written by Philip Levis and David Gay

[3] Jonathan Hui, "Deluge 2.0-TinyOS Network Programming", found in Feb. 2007 at http://www.cs.berkeley.edu/ whui/research/deluge/deluge-manual.pdf20

[5]W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000

[4] A Survey on Routing Protocols for Wireless Sensor Networks: http://www.cs.umbc.edu/~kemal1/mypapers/Akkaya_Younis_JoAdHocRevised.pdf

[6] C.Schurgers and M.B.Srivatava, Energy Efficient Routing in Wireless Sensor Networks, http://fleece.ucsd.edu/~curts/papers/MILCOM01.pdf, December 2002

[7]Fan Xiangning, Song Yulin (2007) *International Conference on Sensor Technologies and Applications.*

[8] Wendi B. Heinzelman, Anantha P. Chandrakasan and Hari Balakrishnan(2002) *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4.

[9] Bani Yassein M., Al-zou'bi A., Khamayseh Y., Mardini W. (2009) *JDCTA: International Journal of Digital Content Technology and its Applications*, Vol. 3, No. 2, pp. 132-136.

[10] Loscri V., Morabito G., Marano S. (2005) Vehicular Technology Conference, VTC-2005, Volume: 3,1809-1813.