International Journal of Engineering Research and Development
ISSN: 2278-067X, Volume 1, Issue 11 (July 2012), PP. 52-57
www.ijerd.com

Object Oriented Approach to Continuous Neighbor Discovery in Asynchronous Sensor Networks

T.V.Satya Sheela[1], N.T. Radha[2], P Suresh Babu[3]
[1]*Kaushik College of Engineering, Gambhiram, Visakhapatnam, AP, INDIA,*
[2,3]*Associate Professor, Kaushik College of Engineering, Gambhiram, Visakhapatnam, AP, INDIA,*

Abstract—*Sensor nodes can be used for a wide variety of applications such as social networks and location tracking. An important requirement for all such applications is that the nodes need to actively discover their neighbors with minimal energy and latency. Nodes in networks are not necessarily synchronized with each other, making the neighbor discovery problem all the more challenging. In this paper, we propose a neighbor discovery protocol in asynchronous sensor networks, which achieves neighbor discovery at minimal and predictable energy costs while allowing nodes. We provide a theoretical formulation of this asynchronous neighbor discovery problem, and evaluate it.*

Keywords— *Sensor nodes, social networks, asynchronous sensor network;*

I. INTRODUCTION

A sensor network may contain a huge number of simple sensor nodes that are deployed at some inspected site. In large areas, the sensor network usually has a mesh structure. In this case, some of the sensor nodes act as routers, forwarding messages from one of their neighbors to another. To minimize idle listening, which is the main source for energy waste [1], the sensors turn their communication hardware on and off. Energy expenditure is commonly assumed to be governed not by the amount of transmitted data but by the time the sensors spend in active mode [2-4].

This is true, for example, in alarm sensor networks, where the traffic is very low while the latency requirements are high. Thus, there is a clear tradeoff between energy expenditure and the end-to end delay. The energy expenditure and the end-to-end delay are both governed by the duty cycle of the nodes, which is defined by their wake-up frequency.

In order for two neighboring sensors to communicate, both must be active at the same time. There are two main approaches to scheduling simultaneous wake-up of neighboring sensors: global or local synchronization [5-8]. Since global synchronization is inefficient and very difficult to achieve in big networks [6-8], we assume that sensor wake-ups are only synchronized locally. That is, each node selects its wake-up schedule and informs neighboring nodes about its selection. A node that needs to send a packet through a neighbor must wake up and transmit it during the neighbor's duty cycle. This communication model imposes a clear tradeoff between the delay encountered by a packet routed along the sensor network and the time during which the sensors along the route are in active mode. This tradeoff was studied in several works[9-11]. Solutions for addressing this trade-off depend, to a large extent, on the specific sensor network model [12], and in particular on the following aspects:

- The data delivery model, i.e., whether data is delivered continuously by the sensor, or delivered only after an event of interest.
- The expected amount of data to be delivered.
- The routing scheme: whether, for example, a single route is used between each source and the gateway or multiple routes are concurrently employed, or whether routes are selected by the traversed nodes or by the sources ("source routing").
- Whether the intermediate nodes process the packets they receive in order to merge similar observations from different sources or forward them as is.

II. BACKGROUND

While many papers have been written on how to minimize energy consumption in sensor networks, very few have explicitly addressed the tradeoff between delay and energy. To the best of our knowledge, our work is the first to propose the assignment of different wake-up frequencies to nodes according to their role in the packet forwarding process. However, the energy-latency tradeoff has been thoroughly studied in sensor networks, as well as in other wireless networks. In this section we present related works, and compare their models and results with ours.

In the S-MAC protocol [9], packet latency caused by periodic sleeping of intermediate nodes is minimized by synchronizing the wake-up schedules of neighboring nodes. The duty cycles of all nodes are equal and predefined. The protocol is not intended to guarantee an upper bound on the end-to-end delay, but to minimize the energy consumption of the nodes.

Another paper [10] proposes to minimize the delay using special scheduling of the nodes' wake-up periods. This paper extends another work [13] by the same authors, where the nodes are organized in a unidirectional tree. However, in [10] the authors assumption of arbitrary communication patterns renders the problem NP-Complete. The authors propose algorithms that find an optimal solution for specific topologies, such as trees and rings. They also show that their algorithms can be used as heuristics for general graphs.

In [14], the authors address the trade-off between delay and energy in sensor networks from a different viewpoint. They search for an optimal routing path from a source node to the gateway, such that latency is minimized and energy cost is not "too big." In their network model, sensors randomly switch between sleep and active states. Two alternatives are studied: a centralized global optimization approach and a distributed approach.

Energy efficiency can be achieved in different ways. For example, energy aware routing finds a routing path while taking into account energy cost and the sensor's available energy. In [15] the number of hops along the forwarding path is considered, while keeping in mind that transmission between close nodes is more energy efficient, even if the resulting route is longer. The authors use a random network model to show the energy-latency-throughput dependency and to find the optimal transmission power for nodes in an ad-hoc network. As already indicated, we do not address the routing issues in our work. The scheme proposed in [15], as well as many others, can be used for this purpose.

In [11], the tradeoff between energy and latency is investigated using probabilistic computation. The authors consider a network of nodes that switch from passive to active mode independently, but with a predefined frequency. The packets are not forwarded on predefined routes, as they are in our model, but are sent instead to all neighbors in active mode. Therefore, the network density and the duty cycle should be high enough to ensure that each packet will finally reach its destination. A probabilistic analysis finds the portion of time each node is required to be in an active state in order to ensure that the packet is delivered to the gateway on time. This model differs from ours in that our model does not use flooding and assumes that nodes are aware of their neighbor's duty cycles.

The tradeoff between energy and latency in general wireless networks was also studied in a different context. For example, [16] and [17] investigate this tradeoff when a technique called "modulation scaling" is used. The authors base their work on the observation that, in many coding schemes, the transmission of a packet requires a smaller amount of energy if it lasts longer. They solve the problem of finding an optimal transmission schedule for a node, given that it has to forward a random number of packets whose arrival times follow the Poisson distribution. The optimization criterion is to minimize the overall energy consumption and bound the maximum delay. Two algorithms are proposed: an off-line algorithm that finds an optimal solution, and an on-line algorithm that approximates the optimal solution. This approach is taken further in [18]. The authors deal with more general setting, assuming that each packet may have a different deadline and number of bits.

In [19], this problem is generalized by considering an aggregation tree with packets routed along the tree to the root. As in [16], the energy cost of a packet transmission is a decreasing convex function of its transmission time. The cost is different for each node because of the different amounts of data to be forwarded. The packet should be delivered to the sink within a limited time period. The authors propose an off-line algorithm for an optimal solution whose running time complexity is unknown, and an approximation algorithm with pseudo-polynomial running time that needs to know the network topology. Although our model is different, the considered problem is similar to ours. In our work, by making some assumption on the energy-latency dependency, we propose an optimal algorithm with linear complexity.

LEACH protocol[20] presumes that the sensors are able to change their transmitting power in order to build a better communication graph. The proposed solution combines the sensors into local clusters and allows only the cluster heads to contact the gateway. Since a cluster head expends much more energy than any other node, this role is periodically rotated between the cluster nodes.

III. PROPOSED SYSTEM ARCHITECTURE

In fig 1 node u is in the Init state, it performs initial neighbor discovery. After a certain time period, during which the node is expected, with high probability, to find most of its neighbors, the node moves to the Normal state, where continuous neighbor discovery is performed.

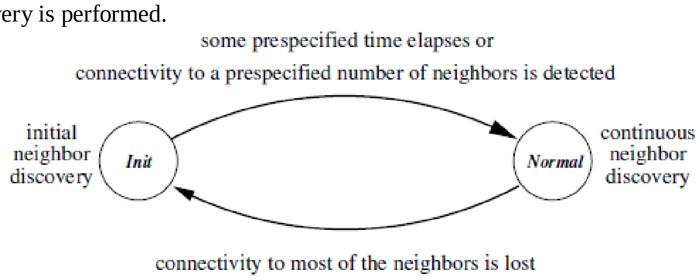

Fig. 1. Topology maintenance vs. neighbor discovery in sensor networks

A node in the Init state is also referred to in this paper as a hidden node and a node in the Normal state is referred to as a segment node. The main idea behind the continuous neighbor discovery scheme we propose is that the task of finding a new node u is divided among all the nodes that can help v to detect u.

A. Existing System

Initial neighbor discovery is usually performed when the sensor has no clue about the structure of its immediate surroundings. In such a case, the sensor cannot communicate with the gateway and is therefore very limited in performing its tasks.

Disadvantages:

- In networks with continuously heavy traffic.
- Long-term process.
- Greater expense of energy than required in our scheme.

B. Proposed System:

We distinguish between neighbor discovery during sensor network initialization and continuous neighbor discovery. We focus on the latter and view it as a joint task of all the nodes in every connected segment. Each sensor employs a simple protocol in a coordinate effort to reduce power consumption without increasing the time required to detect hidden sensors

Advantages:

- Detect their immediate neighbors.
- Message does not collide with another.
- Every node discovers its hidden neighbors independently.
-

C. Algorithm:

Algorithm 1 For a given value of maximum delay, this algorithm determines the wake-up frequency for every node such that the overall energy is minimized.

1) Calculate the optimal wake-up frequency assignment by executing function Calculate-Frequency Division(root).
2) Find E(root)
3) Calculate the precise wake-up frequencies of all the nodes by calling Assignrequencies(root,E(root)).

Algorithm 2 an efficient continuous neighbor discovery algorithm***:***

In this section we present an algorithm for assigning HELLO message frequency to the nodes of the same segment. This algorithm is based on detecting all hidden links inside a segment. Namely, if a hidden node is discovered by one of its segment neighbors, it is discovered by all its other segment neighbors after a very short time. Hence, the discovery of a new neighbor is viewed as a joint effort of the whole segment. Suppose that node u is in initial neighbor discovery state, where it wakes up every T_I seconds for a period of time equal to H, and broadcasts HELLO messages.

Suppose that node u is in initial neighbor discovery state, where it wakes up every TI seconds for a period of time equal to H, and broadcasts HELLO messages. Suppose that the nodes of segment S should discover u within a time period T with probability P. Each node v in the segment S is in continuous neighbor discovery state, where it wakes up every $T_N(v)$ seconds for a period of time equal to H and broadcasts HELLO messages.

We assume that, in order to discover each other, nodes u and v should have an active period that overlaps by at least a portion δ, $0 < \delta < 1$, of their size H. Thus, if node u wakes up at time t for a period of H, node v should wake up between $t-H(1-\delta)$ and $t+H(1-\delta)$. The length of this valid time interval is $2H(1-\delta)$. Since the average time interval between two wake-up periods of v is $T_N(v)$, the probability that u and v discover each other during a specific HELLO interval of u
is $2H(1-\delta)/T_N(v)$.

IV. IMPLEMENTATION

In this paper our proposed method consists following modules.

1. Client – Server
2. Detecting all hidden links Inside a segment
3. Detecting all hidden links Outside a segment
4. Neighbor Discovery Model

Client – Server: Client – Server computing is distributed access. Server accepts requests for data from client and returns the result to the client. By separating data from the computation processing, the compute server's processing capabilities can be optimized. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system.

Hidden link participate Inside a segment: This scheme is invoked when a new node is discovered by one of the segment nodes. The discovering node issues a special SYNC message to all segment members, asking them to wake up and periodically broadcast a bunch of HELLO messages. This SYNC message is distributed over the already known wireless links of the segment. Thus, it is guaranteed to be received by every segment node. By having all the nodes wake up .almost at the same time. for a short period, we can ensure that every wireless link between the segment's members will be detected.

Hidden link participate Outside a segment: A random wake-up approach is used to minimize the possibility of repeating collisions between the HELLO messages of nodes in the same segment. Theoretically, another scheme may be used, where segment nodes coordinate their wake-up periods to prevent collisions and speed up the discovery of hidden nodes. Since the time period during which every node wakes up is very short, and the HELLO transmission time is even shorter, the probability that two neighboring nodes will be active at the same time.

Neighbor Discovery Model: Neighbor Discovery is studied for general ad-hoc wireless networks. A node decides randomly when to initiate the transmission of a HELLO message. If its message does not collide with another HELLO, the node is considered to be discovered. The goal is to determine the HELLO transmission frequency, and the duration of the neighbor discovery process.

V. RESULT ANALYSIS

In this section we present a simulation study for the schemes presented in the paper. We simulate a large sensor network, with nodes distributed randomly and uniformly over the area of interest. We assume that the nodes have an equal and constant transmission range. Communication is always bi-directional. We also assume that most of the nodes discover each other and enter the continuous neighbor discovery state before the simulation begins. Our simulation model consists of 2,000 sensor nodes, randomly placed over a 10,000 x 10,000 grid. Figure 2(a) and (b) shows the ratios of hidden nodes to the total number of nodes as a function of time. The initial ratio is 0.05. We can see that after 100 time units, this ratio decreases to 0.035 for P = 0:3, to 0.025 for P = 0:5, and to 0:015 for P = 0:7. After 200 time units, the ratios of the hidden nodes are 0:025, 0:012 and 0:005 respectively. It is evident that these results are very close to the required ratios.

Fig 2 (a) Decrease in the ratio of hidden nodes

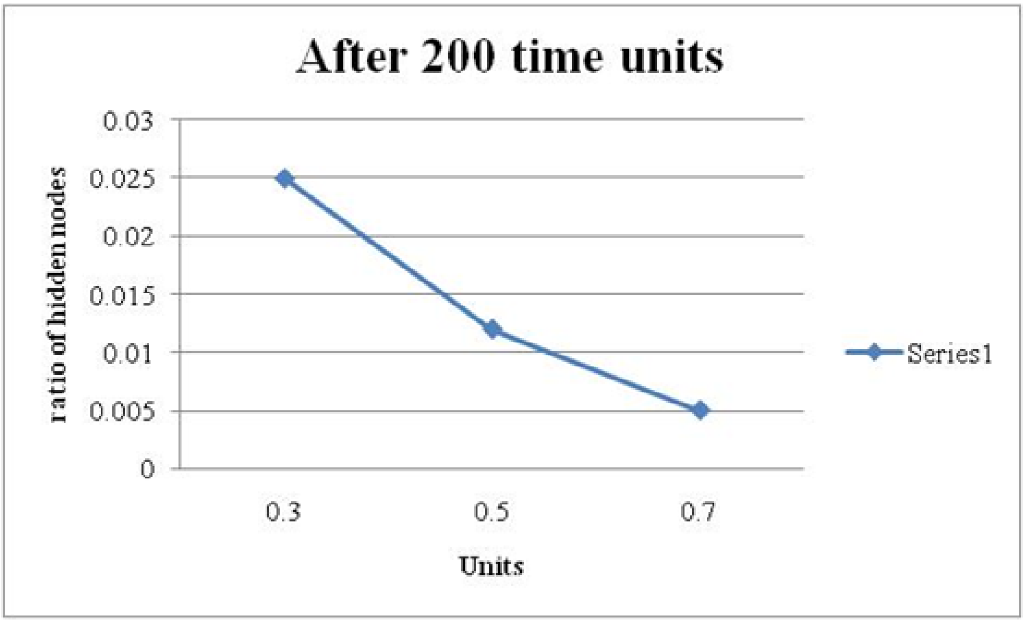

Fig 2 (b) Decrease in the ratio of hidden nodes

VI. CONCLUSIONS AND FUTURE WORK

This paper shows an approach for getting a high efficiency continuous neighbor discovery in asynchronous sensor networks with an even higher efficiency synchronous communication protocol. We showed that our scheme works well if every node connected to a segment estimates the in-segment degree of its possible hidden neighbors. We simulated a sensor network to analyze our algorithms and showed that when the hidden nodes are uniformly distributed in the area, the simplest estimation algorithm is good enough.

REFERENCES

[1]. Cintia B. Margi and Katia Obraczka. Instrumenting network simulators for evaluating energy consumption in power-aware ad-hoc network protocols. In MASCOTS, pages 337–346, Washington, DC, 2004. IEEE Computer Society.
[2]. Mario Strasser, Andreas Meier, Koen Langendoen, and Philipp Blum. Dwarf: Delayaware robust forwarding for energy-constrained wireless sensor networks. In DCOSS, pages 64–81, 2007.
[3]. Abtin Keshavarzian, Huang Lee, and Lakshmi Venkatraman. Wakeup scheduling in wireless sensor networks. In MobiHoc: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, pages 322–333, New York, NY, 2006. ACM Press.
[4]. Reuven Cohen and Boris Kapchits. An optimal algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network. In INFOCOM, pages 258–266, 2007.
[5]. Qun Li and Daniela Rus. Global clock synchronization in sensor networks. IEEE Trans. Comput., 55(2):214–226, 2006.
[6]. Jeremy Elson and Kay R¨omer. Wireless sensor networks: a new regime for timesynchronization. SIGCOMM Comput. Commun. Rev., 33(1):149–154, 2003.
[7]. Yuan Li, Wei Ye, and John Heidemann. Energy and latency control in low duty cycle MAC protocols. In Proceedings of the IEEE Wireless Communications and Networking Conference, pages 676–682, New Orleans, LA, March 2005.
[8]. Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. SIGOPS Oper. Syst. Rev., 36(SI):147–163, 2002.
[9]. Wei Ye, John S. Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans. Netw., 12(3):493–506, 2004.
[10]. Gang Lu, Narayanan Sadagopan, Bhaskar Krishnamachari, and Ashish Goel. Delay efficient sleep scheduling in wireless sensor networks. In INFOCOM, pages 2470–2481, 2005.
[11]. Olivier Dousse, Petteri Mannersalo, and Patrick Thiran. Latency of wireless sensor networks with uncoordinated power saving mechanisms. In MobiHoc, pages 109–120, 2004.
[12]. Sameer Tilak, Nael B. Abu-Ghazaleh, andWendi Heinzelman. A taxonomy of wireless micro-sensor network models. SIGMOBILE Mob. Comput. Commun. Rev., 6(2):28– 36, 2002.
[13]. Gang Lu, Bhaskar Krishnamachari, and Cauligi S. Raghavendra. An adaptive energyefficient and low-latency mac for data gathering in wireless sensor networks. In IPDPS, 2004.
[14]. Wei Lai and I. C. Paschalidis. Routing through noise and sleeping nodes in sensor networks: latency vs. energy trade-offs. In Proceedings of the 45th IEEE Conference on Decision and Control, pages 2716–2721, December 2006.

[15]. Abbas El Gamal and James Mammen. Optimal hopping in ad hoc wireless networks. In INFOCOM, pages 1–10, 2006.
[16]. Balaji Prabhakar, Elif Uysal-Biyikoglu, and Abbas El Gamal. Energy-efficient transmission over a wireless link via lazy packet scheduling. In INFOCOM, pages 386–394, 2001.
[17]. Chandra Nair, Abbas El Gamal, Balaji Prabhakar, Elif Uysal-Biyikoglu, and Sina Zahedi. Energy-efficient scheduling of packet transmissions over wireless networks. In INFOCOM, pages 1773–1782, 2002.
[18]. Lei Miao and Christos G. Cassandras. Optimal transmission scheduling for energy efficient wireless networks. In INFOCOM, pages 1–11, 2006.
[19]. Yang Yu, Bhaskar Krishnamachari, and Viktor K. Prasanna. Energy-latency tradeoffs for data gathering in wireless sensor networks. In INFOCOM, 2004.
[20]. Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energyefficient communication protocol for wireless microsensor networks. In HICSS, 2000.