Performance of Asynchronous Sensor Networks

[1]K. Parimala, [2]P. Radha Krishna, [3]K. Leela Lakshmi Devi

[1,2,3]Dept. of CSE, NOVA College of Engineering & Technology, Vijayawada, AP, India

Abstract

We exposed a new problem in wireless sensor networks, referred to as ongoing continuous neighbor discovery. We argue that continuous neighbor discovery is crucial even if the sensor nodes are static. If the nodes in a connected segment work together on this task, hidden nodes are guaranteed to be detected within a certain probability P and a certain time period T, with reduced expended on the detection. We showed that our scheme works well if every node connected to a segment estimates the in-segment degree of its possible hidden neighbors. To this end, we proposed three estimation algorithms and analyzed their mean square errors. We then presented a continuous neighbor discovery algorithm that determines the frequency with which every node enters the HELLO period. 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. When the hidden nodes are concentrated around some dead areas, the third algorithm, which requires every node to take into account not only its own degree, but also the average degree of all the nodes in the segment, was shown to be the best.

Keywords

Asynchronous, Sensor Networks, Analysis

I. Introduction

Wireless Ad-Hoc Networks, particularly, static Ad-Hoc networks such as sensor networks and community mesh networks, have generated tremendous amount of interest recently. Sensor networks have applications such as surveillance and tracking [16], environmental observation [2], habitat monitoring [7], and health monitoring [13], while mesh networks [8], enable nodes to connect home networks together forming a community Ad-Hoc network. A characteristic requirement of these Ad-Hoc networks is that they be .self-configuring., i.e., that a large number of wireless nodes organize themselves to efficiently perform the tasks required by the application after they have been deployed. Examples of self-configuration include construction of routing paths, clustering, and formation of minimum weight spanning trees. Self-configuring. Ad-hoc networks are very attractive since they reduce the cost of installation and allow for building large scale systems.

In this paper, we consider an aspect of self-configuration in wireless Ad-Hoc networks referred to as neighbor discovery. After nodes are deployed, they need to discover their one-hop neighbors. Knowledge of one-hop neighbors is essential for almost all routing protocols, medium-access control protocols and several other topology-control algorithms such as construction of minimum-energy spanning trees. Neighbor discovery is, therefore, a crucialrst step in the process of self-organization of a wireless ad-hoc network. Ideally, nodes should discover their neighbors as quickly as possible as rapid discovery of neighbors often translates into energy efficiency, since nodes have to spend less energy discovering neighbors. Also, rapid discovery allows for other protocols (such as topology control, medium access and routing protocols) to quickly start their execution. We emphasize that the focus of this paper is on neighbor discovery alone and not how the discovered neighbor information is used by topology control algorithms [6, 11, 15], medium access protocols [1, 3] and routing algorithms [4]. There has been earlier work on neighbor discovery in wireless networks [5, 9]. In these papers, the authors present algorithms for neighbor discovery in wireless networks where nodes have Omni-directional antennas and operate in a synchronous fashion. Our work differs from the existing work in two important ways. First, we address the problem of neighbor discovery when nodes have directional antennas. Second, we also consider the case in which the neighbor discovery algorithms operate asynchronously. Directional antennas offer many advantages over unidirectional antennas such as increased spatial reuse, increased transmission range and increased capacity. However, discovery of neighbors becomes harder since nodes must control the direction of their antennas in order to transmit or receive data packets from their neighbors. Thus the efficiency of neighbor discovery algorithms using directional antennas depends not only on how often nodes transmit and receive but also on antenna properties such as their direction and beam width. In this paper, we propose several probabilistic neighbor discovery algorithms in which nodes perform random independent transmissions in different directions to discover their one hop neighbors. The goal of these neighbor discovery algorithms is to maximize the probability of discovery of neighbors within a given amount of time. We consider both synchronous and asynchronous algorithms and their optimal design. While the algorithm in [9], can be made asynchronous in the manner described in this paper, synchronization is a requirement for the algorithm described in [5]. In this paper, we present several probabilistic neighbor discovery algorithms, both synchronous and asynchronous. Our neighbor discovery algorithms can be classified into two groups, viz. Direct-Discovery Algorithms in which a nodes discovers its neighbor only when it successfully hears a transmission from that neighbor and Gossip-Based Discovery Algorithms in which nodes gossip about each others' location information to speed up discovery. Some of the important contributions of our work are: 1. A simple mathematical model to derive the optimal parameter settings for synchronous direct-discovery and gossip-based algorithms. 2. A simulation-based performance comparison of the gossip-based and the direct-discovery algorithms, demonstrating that nodes discover their neighbors significantly faster using the gossip-based algorithm than using the direct-discovery algorithm. Interestingly, we also see that while the performance of direct-discovery algorithm degrades as node density increases, the gossip-based algorithm remains insensitive to an increase in node density.

II. Existing System

Two nodes are said to be neighboring nodes if they have direct wireless connectivity. We assume that all nodes have the same transmission range, which means that connectivity is always bidirectional. During some parts of our analysis, we also assume that the network is a unit disk graph; namely, any pair of nodes that are within transmission range are neighboring nodes. Two nodes are said to be directly connected if they have discovered each other and are aware of each other's wake-up times. Two nodes are said to be connected if there is a path of directly connected nodes between them. A set of connected nodes is referred to as a segment. Consider a pair of neighboring nodes that belong to

the same segment but are not aware that they have direct wireless connectivity. See, for example, nodes a and c These two nodes can learn about their hidden wireless link using the following simple scheme, which uses two message types: (a) SYNC messages for synchronization between all segment nodes, transmitted over known wireless links; (b) HELLO messages for detecting new neighbors. Scheme 1 (detecting all hidden links 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. To better understand the benefit of Scheme 1, we now compare its performance to the performance of a trivial algorithm where every node discovers its hidden neighbors independently. When Scheme 1 is used, a hidden node is discovered by all of its in-segment neighbors as soon as it is discovered by the first of them. In contrast, when Scheme 1 is not used, the hidden node is discovered by all of its in-segment neighbors only when it is discovered by the last of them. To analyze the time slots at which these nodes are discovered. Suppose that the time axis is divided into slots such that the probability that a node discovers a given hidden neighbor is p. Consider a node u with m in-segment hidden neighbors. The probability that u discovers its first in-segment hidden
Neighbor only at slot k + 1

$$p_m(k) = (1-p)^{m^k}(1-(1-p)^m).$$

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. However, finding an efficient time division is equivalent to the well-known node coloring problem, which is NP-hard and also cannot be well approximated. 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 is practically 0. In the rare case of collisions, CSMA/CD can be used to schedule retransmissions. By Scheme 1, the discovery of an individual node by any node in a segment leads to the discovery of this node by all of its neighbors that are part of this segment. Therefore, discovering a node that is not yet in the segment can be considered a joint task of all the neighbors of this node in the segment. As an example, consider Figure 4(a), which shows a segment S and a hidden node u. In this figure, a dashed line indicates a hidden wireless link, namely, a link between two nodes that have not yet discovered each other. A thick solid line indicates a known wireless link. After execution of Scheme 1, all hidden links in S are detected. The links connecting nodes in S to u are not detected because u does not belong to the segment. Node u has 4 hidden links to nodes in S. Hence, we say that the degree of u in S is degS(u) = 4. When u is discovered by one of its four neighbors in S, it will also be discovered by the rest of its neighbors in S as soon as Scheme 1 is re invoked. Consider one of the four segment members that are within range of u, node v say. Although it may know about the segment members within its own transmission range, it does not know how many in-segment neighbors participate in discovering u. In the next section we study three methods that allow v to estimate the value of degS(u) for a hidden node u, and compare their accuracy and applicability.

III. Proposed Work

In this section, we present an algorithm for assigning HELLO message frequency to the nodes of the same segment. This algorithm is based on Scheme 1. 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. One of the three methods presented in Section IV is used to estimate the number of nodes participating in this effort. 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 TN(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 _, 0 < _ < 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 $\frac{2H(1-\delta)}{T_N(v)}$.

Let n be the number of in-segment neighbors of u. When u wakes up and sends HELLO messages, the probability that at least one of its n neighbors is awake during a sufficiently
(8) long time interval is $1-(1-\frac{2H(1-\delta)}{T_N(v)})^n$.

For the sake of our analysis, consider a division of the time axis of u into time slots of length H. The probability that u is awake in a given time slot is $\frac{H}{T_I}$, and the probability that u is discovered during this time slot is $P_1 = \frac{H}{T_I}(1-(1-\frac{2H(1-\delta)}{T_N(v)})^n)$. Denote by D the value of $\frac{T}{H}$. Then, the probability that u is discovered within at most D slots is $P_2 = 1-(1-P_1)^D$. Therefore, we seek the value of $T_N(v)$ that satisfies the following equation:

$$1-(1-P_1)^D \geq P,$$

which can also be stated as

$$P_1 \geq 1 - \sqrt[D]{1-P}.$$

Since $P_1 = \frac{H}{T_I}(1-(1-\frac{2H(1-\delta)}{T_N(v)})^n)$, we get

$$\frac{H}{T_I}(1-(1-\frac{2H(1-\delta)}{T_N(v)})^n) \geq 1 - \sqrt[D]{1-P},$$

and therefore

$$T_N(v) \le \frac{2H(1-\delta)}{1-\sqrt[n]{1-\frac{T_I}{H}(1-\sqrt[D]{1-P})}}. \qquad (9)$$

Since node v does not know the exact value of n, it can estimate it using the methods presented in Section IV.

We now give a simple example for the proposed algorithm. Suppose that nodes in the Init state remain active until they enter the Normal state. Suppose also that the requirement is to discover a hidden node within 10 time units with probability 0.5. Consider a segment node in the Normal state, where continuous neighbor discovery is performed, that estimates the degree of its hidden neighbor as 1. Following our definitions, $D = 10, T_I = H = 1$ and $n = 1$. Substituting these values into Eq. 9 yields $T_N \approx 15$. Note that the intuitive value of $T_N = 20$ is wrong because it would yield a detection probability of $1-(1-\frac{1}{20})^{10} \approx 0.4$ to discover a hidden node within 10 time units.

If one needs to enforce not only the expected hidden neighbor discovery delay but also an upper bound on it, each node can be assigned a wake-up period according to the rules described in [13].

In Figure 6 we present two graphs that show the dependency between T and $T_N(v)$. We assume that a hidden node wakes up once every $100H$ time units on the average, and that $T_I = 100$, $H = 1$, and $\delta = 0.5$. In Figure 6(a) the estimated value of n is 10. The curves present the value of $T_N(v)$ as a function of the desired discovery time T for 3 different values of P: 0.5, 0.8 and 0.95. In Figure 6(b) P is set to 0.8 and n varies between 5 and 50. Again, $T_N(v)$ is calculated as a function of the desired discovery time. As expected, the nodes have to work harder to achieve a greater discovery rate in less time, while the increase in the density of segment nodes allows to a greater TN(v) to be chosen. In both graphs the dependency between TN(v) and T is almost linear and, as, the slope of the curves is almost linear in the value of n as well. This means that a node v can use linear approximation to compute the value of TN(v).

IV. Results

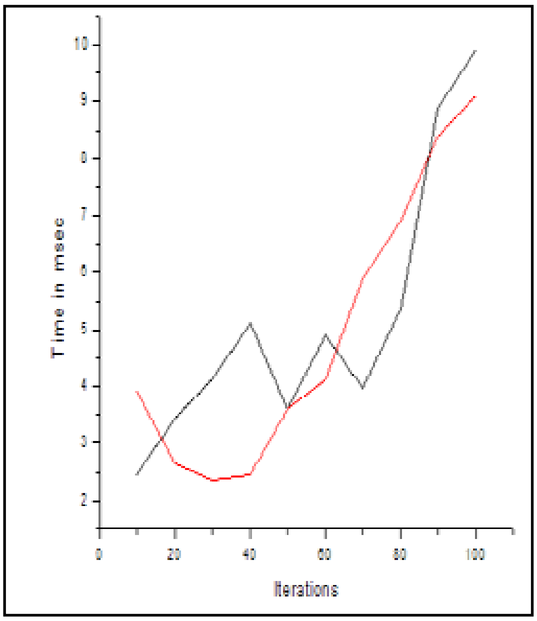

Fig. 1:

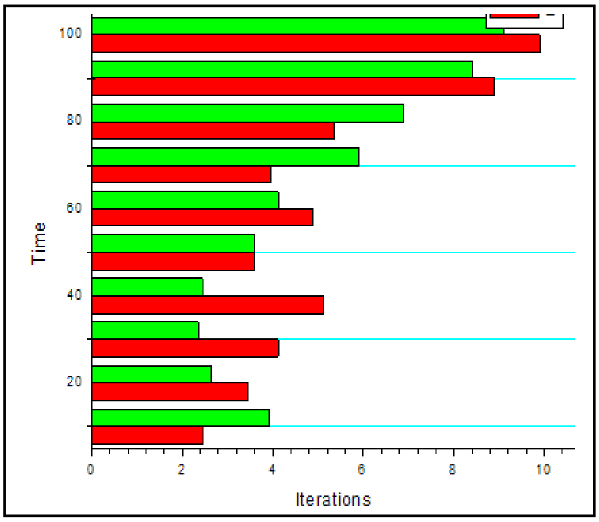

Fig. 2:

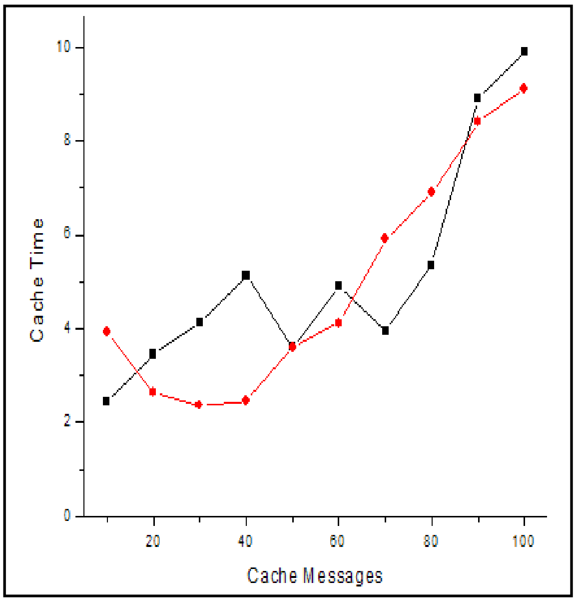

Fig. 3:

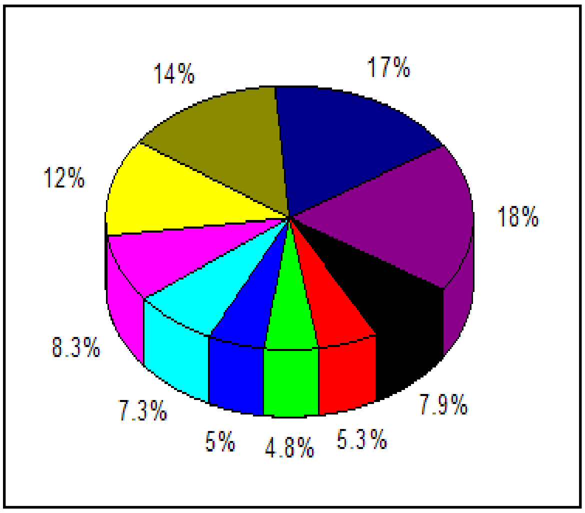

Fig. 4:

V. Conclusions

In most sensor networks the nodes are static. Nevertheless, node connectivity is subject to changes because of disruptions in wireless communication, transmission power changes, or loss of synchronization between neighboring nodes. Hence, even after a sensor is aware of its immediate neighbors, it must continuously maintain its view, a process we call continuous neighbor discovery. In this work 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.

References

[1] S. Vasudevan, J. Kurose, D. Towsley,"On neighbor discovery in wireless networks with directional antennas", in INFOCOM, Vol. 4, 2005, pp. 2502-2512.

[2] R. Madan, S. Lall, "An energy-optimal algorithm for neighbor discovery in wireless sensor networks", Mob. Netw. Appl., Vol. 11, No. 3, pp. 317-326, 2006.

[3] M. J. McGlynn, S. A. Borbash,"Birthday protocols for low energy deployment and _exible neighbor discovery in ad hoc wireless networks", in MobiHoc: Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing. New York, NY, USA: ACM Press, 2001, pp. 137-145.

[4] D. Baker, A. Ephremides,"The architectural organization of a mobile radio network via a distributed algorithm", in IEEE Transactions on Communications, Vol. 29, Nov. 1981, pp. 1694-1701.

[5] A. Keshavarzian, E. Uysal-Biyikoglu,"Energy-ef_cient link assessment in wireless sensor networks", in INFOCOM, 2004.

[6] E. B. Hamida, G. Chelius, E. Fleury,"Revisiting neighbor discovery with interferences consideration", in PE-WASUN, 2006, pp. 74-81.

[7] S. A. Borbash,"Design considerations in wireless sensor networks", Ph.D. dissertation, ISR, August 2004.

K.Parimala has completed B.Tech from Potluri VaraPrasad Siddhartha Institute & Technology. in 2008 and Pursuing final M.Tech in NOVA College of Engineering Technology for Women. Her research work includes concepts of Data Mining and Data Warehousing.

Mr. P.Radha Krishna did his Master in Computer Application at Lakkireddy Bali eddy college of engineering. He started his teaching career at Anurag Engineering College, Kodad. Where He had guided be so many batches in their projects. He guided students in doing IBM projects at Anurag. He played an active role in the NBA Accreditation process at Anurag Engineering College in to 2007, He had acted as In charge Hod of MCA at Anurag. Mr. P Radha Krishna is a Members of ISTE. He had participated many workshops and paper presentations, which include sunmicrosystem and open systems (LAMP) workshops. Mr. Radha Krishna is the secretary of Alumni Association of LBRCE. He is now working as the Head of the Department of CSE of Nova college of Engineering & Technology, Jupudi, Ibrahimpatnm.

K.Leela Lakshmi Devi has completed M.Sc in CSE from Annabathuni Satya Narayana P.G & Degree College in 2010 and Pursuing final M.Tech in NOVA College of Engineering Technology for Women. Her research work includes concepts of Data Mining and Data Warehousing.