Uninterrupted Detection of Segment Nodes in Wireless Sensor Networks

[1]K. Hari Prasad, [2]D. Srinivas
[1,2]Dept. of CSE, Kakinada Institute of Engineering & Technology, Korangi, AP, India

Abstract
Neighbor discovery is an important task in wireless networks, and especially in sensor networks. Neighbor information can be used to improve routing, clustering and scheduling algorithms. A sensor network may contain a huge number of simple sensor nodes that are deployed at some inspected t site. In large areas, such a 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. The nodes are configured to turn their communication hardware on and off to minimize energy consumption. Therefore, in order for two neighboring sensors to communicate, both must be in active mode. In the sensor network model considered in this paper, the nodes are placed randomly over the area of interest and their first step is to detect their immediate neighbors - the nodes with which they have a direct wireless communication - and to establish routes to the gateway. In networks with incessantly heavy traffic, the sensors need not invoke any special neighbor detection protocol during normal operation. This is because any new node, or a node that has lost connectivity to its neighbors, can hear its neighbors simply by listening to the channel for a short time. However, for sensor networks with low and irregular traffic, a special neighbor detection scheme should be used.

Keywords
Sensor, Synchronization, Neighbor

I. Introduction

A. Sensor Networks

Sensor is a cheap tiny device that is able to detect local events and report them to a centralized gateway using wireless communication. A sensor network consists of many sensors and a gateway. The sensors perform some common task, like smoke detection or temperature measurement, and report to the gateway. Since not all the sensors are in the transmission range of the gateway, their messages should be forwarded by other sensors. Most often the network structure cannot be pre-engineered, since the sensors are placed randomly in the covered area.

A Wireless Sensor Network (WSN) consists of spatially distributed autonomous sensors monitor physical or environmental conditions, such as temperature, sound, vibration pressure, humidity, motion or pollutants and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

Area monitoring is a common application of WSNs. In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. A military example is the use of sensors to detect enemy intrusion; a civilian example is the geo-fencing of gas or oil pipelines.

B. Neighbor Discovery in Sensor Networks

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.

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 crucial first 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], medium access protocols [3, 1] and routing algorithms [4].

For these reasons, detecting new links and nodes in sensor networks must be considered as an ongoing process. In the following discussion we distinguish between the detection of new links and nodes during initialization, i.e., when the node is in Init state, and their detection during normal operation, when the node is in Normal state. The former will be referred to as initial neighbor discovery whereas the latter will be referred to as continuous neighbor discovery. While previous works [1-3], address initial neighbor discovery and continuous neighbor discovery as similar tasks, to be performed by the same scheme, we claim that different schemes The emerging wireless ad hoc network paradigm enables a new type of network in which collaborating devices relay packets from one device to another across multiple wireless links in a self-organizing manner. A number of applications based on this type of network have been established or are expected in the near future, such as environmental and building monitoring, disaster relief and military battlefield communication. Due to the self-organizing nature of ad hoc networks, every node in the network can be alternately functioning as a transmitter or a receiver. Oftentimes, a node can communicate directly with only several other nodes around itself, which are called its “neighbors”. In absence of a

central controller, every node has to discover its neighbors before efficient routing is possible. The process for a node to identify all its neighbors is called neighbor discovery, which is a crucial first step of constructing reliable wireless ad hoc networks.

The differences between neighbor discovery and topology management.

- For neighbor discovery, an aggressive protocol, one which requires the sensor to stay in active mode and expend a lot of energy until detection, is usually acceptable.
- Neighbor discovery is performed when the sensor has no clue about the structure of its immediate surroundings. In particular, the sensor is unable to perform any useful task. Hence, energy consumption in this state is less of an issue.
- When the sensor performs topology maintenance, it can perform topology maintenance together with these neighbors in order to consume less energy.

Despite the static nature of the sensors in many sensor networks, connectivity is still subject to changes even after the network has been established. The sensors must continuously look for new neighbors in order to accommodate the following situations:

- Loss of local synchronization due to accumulated clock drifts
- Disruption of wireless connectivity between adjacent nodes by a temporary event, such as a passing car or animal, a dust storm, rain or fog. When these events are over, the hidden nodes must be rediscovered.
- The ongoing addition of new nodes, in some networks to compensate for nodes which have ceased to function because their energy has been exhausted.
- The increase in transmission power of some nodes, in response to certain events, such as detection of emergent situations.

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. The immediate surroundings should be detected as soon as possible in order to establish a path to the gateway and contribute to the operation of the network. Hence, in this state, more extensive energy use is justified. In contrast, continuous neighbor discovery is performed when the sensor is already operational. This is a long-term process, whose optimization is crucial for increasing network lifetime.

When the sensor performs continuous neighbor discovery, it is already aware of most of its immediate neighbors and can therefore perform it together with these neighbors in order to consume less energy. In contrast, initial neighbor discovery must be executed by each sensor separately.

A typical neighbor discovery protocol. In this protocol, a node becomes active according to its duty cycle. Let this duty cycle be in Init state and in Normal state. We want to have . When a node becomes active, it transmits can invoke another procedure to finalize the setup of their joint wireless link.

To summarize, in the Init state, a node has no information about its surroundings and therefore must remain active for a relatively long time in order to detect new neighbors. In contrast, in the Normal state the node must use a more efficient scheme. Such a scheme is the subject of our study. Figure 1 summarizes this idea. When 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 nd most of its neighbours, the node moves to the Normal state, where continuous neighbour discovery is performed. 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.

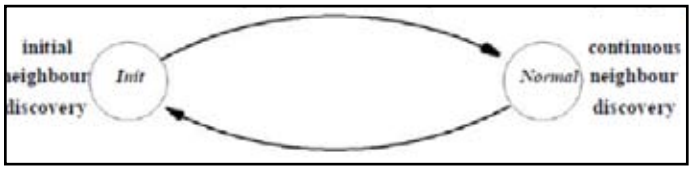

Fig. 1: Continuous Neighbor Discovery vs. Initial Neighbor Discovery in Sensor Networks

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. These nodes are characterized as follows: (a) they are also neighbors of u; (b) they belong to a connected segment of nodes that have already detected each other; (c) node v also belongs to this segment. Let degS (u) be the number of these nodes. This variable indicates the in-segment degree of a hidden neighbor u. In order to take advantage of the proposed discovery scheme, node v must estimate the value of degS (u).

II. Related Work

We consider the problem of neighbor discovery at the physical and medium access layers. In many networks (especially static wired networks) lists of neighbors may be entered manually by a network administrator. We are concerned with situations where this is not easy to do, or when self-configuration is desired.

Neighbor discovery can be done in a centralized or a distributed way. In centralized methods, there is a central controller. All nodes report to the controller, which determines the positions of the nodes, computes their neighbors, and informs each node. Central control of neighbor discovery is expected to cost a lot of energy, particularly when the number of nodes is large. Distributed algorithms have no central controller.

A distributed algorithm of Zheng et al. is described in [7]. The algorithm is analyzed under the assumption that radios can simultaneously send and receive on the same channel. This is difficult to achieve in practice, and is unlikely to be the case for sensor networks. Additionally, because their algorithm is deterministic, there is a non-zero probability that a cluster of neighboring nodes, all within transmission range of each other, will systematically interfere with each other, making neighbor discovery impossible. A large part of the analysis in this paper is devoted to determining when nodes should transmit or receive, and to addressing interference.

Nakano and Olariu describe algorithms to initially assign numbers to completely identical nodes, and to elect leaders in wireless ad hoc networks in [8,9]. These problems bear some similarity to neighbor discovery, as they would also take place early in the life of a sensor network. Their algorithms assume the use of the Global Positioning System (GPS) to achieve synchronous operation In the distributed algorithm of Baker and Ephremides [10], all nodes participate in a two-round round-robin schedule. In each round each node is assigned a single slot to announce its identity and the identities of neighbors discovered so far. Nodes listen in the other slots, and can determine all their neighbors, and all their neighbors' neighbors, within two rounds, under the assumption that nodes receive messages from neighboring nodes without errors.

Asynchronous, multi-channel neighbor discovery algorithms have been developed by the Bluetooth research community [6-7]. But they address a case where channel set is fixed, while in CRN channel set can vary among nodes. Also in Bluetooth, neighbor

discovery is asymmetric (master/slave configuration), but in CRN we consider a symmetric scenario. In fact, we have borrowed some ideas from Bluetooth neighbor discovery process when developing our algorithm.

III. Existing System

Initial neighbor detection 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.

A. Disadvantages

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

IV. Problem Statement

In the following discussion, 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 in Figure 4(a). 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.

Scheme 2 (detecting a hidden link outside a segment):

Node u wakes up randomly, every T (u) seconds on the average, for a fixed period of time H. During this time it broadcasts several HELLO messages, and listens for possible HELLO messages sent by new neighbors.

Node u wakes up randomly, every T (u) seconds on the average, for a fixed period of time H . During this time it broadcasts several HELLO messages, and listens for possible HELLO message 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 (see Figure 4(b)). 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.

A. Estimating the In-Segment Degree of a Hidden Neighbor

We consider the discovery of hidden neighbors as a joint task to be performed by all segment nodes. To determine the discovery load to be imposed on every segment node, namely, how often such a node should become active and send HELLO messages, we need to estimate the number of in segment neighbors of every hidden node u, denoted by degS (u). In this section we present methods that can be used by node v in the Normal (continuous neighbor discovery) state to estimate this value. Node u is assumed to not yet be connected to the segment, and it is in the Init (initial neighbor discovery) state. Three methods are presented

- Node v measures the average in-segment degree of the segment's nodes, and uses this number as an estimate of the in-segment degree of u. The average in-segment degree of the segment's nodes can be calculated by the Segment leader. To this end, it gets from every node in the segment a message indicating the in-segment degree of the sending node, which is known due to Scheme 1.We assume that the segment size is big enough for the received value to be considered equal to the expected number of neighbors of every node.
- Node v discovers, using Scheme 1, the number of its in-segment neighbors, degS (v), and views this number as an estimate of degS (u). This approach is expected to yield better results than the previous one when the degrees of neighboring nodes are strongly correlated.
- Node v uses the average in-segment degree of its segment's nodes and its own in-segment degree degS (v) to estimate the number of node u's neighbors. This approach is expected to yield the best results if the correlation between the in-segment degrees of neighboring nodes is known. An interesting special case is when the in-segment nodes are uniformly distributed.
- The in-segment degree of v and u depends on how the various nodes are distributed in the network. Let X be a random variable that indicates the degree degS (v) of v, a uniform randomly chosen node in the segment S. Let Y be a random variable that indicates the degree degS (u) of u, a uniform randomly chosen hidden neighbour of v, which we want to estimate. Note that u itself is not aware of the value of Y . Let Y 0 be the estimated value of Y . Clearly, we want Y 0 to be as close as possible to Y . We use the mean square error measure (MSE) to decide how good an estimate is.

The in-segment degree of v and u depends on how the various nodes are distributed in the network. Let X be a random variable that indicates the degree degS(v) of v, a uniform randomly chosen node in the segment S. Let Y be a random variable that indicates the degree degS(u) of u, a uniform randomly chosen hidden neighbor of v, which we want to estimate. Note that u itself is not aware of the value of Y. Let Y' be the estimated value of Y. Clearly, we want Y' to be as close as possible to Y. We use the mean square error measure (MSE) to decide how good an estimate is. The

MSE is defined as E ((Y – Y')2). Since v and u are two random neighbors in the same graph, X and Y have the same distribution. Let us denote the correlation between X and Y ,corr(X; Y), by C. Throughout the section we assume that degS(v) is small compared to the network size.

Denote the average graph degree by u, Clearly, E(X) =E(Y) = u, Thus, for the first method the following holds:

$$\begin{aligned} MSE_1 &= E((Y-Y')^2) = E((Y-\mu)^2) \\ &= \mathrm{Var}(Y). \end{aligned} \quad (1)$$

For the second method we have Y'=X, hence

$$\begin{aligned} MSE_2 &= E((Y-Y')^2) = E((Y-X)^2) \\ &= \sum_x \sum_y (y-x)^2 P(X=x, Y=y) \\ &= \sum_x \sum_y (y^2 - 2xy + x^2) P(X=x, Y=y) \\ &= E(X^2) + E(Y^2) - 2E(XY). \end{aligned} \quad (2)$$

By the correlation of random variables and the fact that var(X)=var(Y), we get

$$\mathrm{corr}(X,Y) = \frac{\mathrm{cov}(X,Y)}{\mathrm{Var}(X)}.$$

Using the definition of covariance, we get

$$\mathrm{cov}(X,Y) = E((X-E(X))(Y-E(Y))$$

$$= (XY - XE(Y) - YE(Y) + E(X)E(Y))$$

$$= (XY) - E(X)E(Y) - E(Y)E(X) + E(X)E(Y)$$

$$= (XY) - E(X)E(Y).$$

Hence,

$$\begin{aligned} E(XY) &= \mathrm{cov}(X,Y) + E(X)E(Y) \\ &= \mathrm{corr}(X,Y)\,\mathrm{Var}(X) + E(X)E(Y) \\ &= \mathcal{C}\,\mathrm{Var}(X) + E(X)E(Y). \end{aligned} \quad (3)$$

Substituting into Eq. 2 and keeping in mind that X and Y have the same distribution, we get

$$\begin{aligned} MSE_2 &= E(X^2) + E(Y^2) - 2(\mathcal{C}\,\mathrm{Var}(X) + E(X)E(Y)) \\ &= E(X^2) + E(X^2) - 2\mathcal{C}\,\mathrm{Var}(X) - 2E(X)E(X) \\ &= 2E(X^2) - 2E(X)^2 - 2\mathcal{C}\,\mathrm{Var}(X) \\ &= 2\,\mathrm{Var}(X) - 2\mathcal{C}\,\mathrm{Var}(X) \\ &= (2-2\mathcal{C})\,\mathrm{Var}(X). \end{aligned}$$

For the third estimation approach, we define a linear prediction problem. We seek the values of β and v that minimize the MSE function E ((Y –'Y')2), where Y' = X + . By differentiating the MSE with respect to, we get

$$\begin{aligned} \frac{\delta MSE}{\delta(\gamma)} &= \frac{\delta(E((Y'-Y)^2) = E((\beta X + \gamma - Y)^2))}{\delta(\gamma)} \\ &\cdots \\ &= 2\gamma + 2\beta\mu - 2\mu. \end{aligned}$$

Equating the result to 0 yields

$$\hat{\gamma} = \mu - \beta\mu. \quad (4)$$

B. 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 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.

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 Continuous neighbor discovery state, where it wakes up every TN(v) seconds for a period of time equal to H and broadcasts HELLO messages.

V. Conclusion & Enhancement

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 propose a novel routing-driven key management scheme, which only establishes shared keys for neighbor sensors that communicate with each other. We utilize RSA Cryptography in the design of an efficient key management scheme for sensor nodes. The performance evaluation and security analysis show that our key management scheme can provide better security with significant reductions on communication overhead, storage space and energy consumption than other key management schemes.

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 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.

References

[1] S. Vasudevan, J. Kurose, D. Towsley,"On neighbor discovery in wireless networks with directional antennas".

[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-efficient 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.
[8] G. Alonso, E. Kranakis, R. Wattenhofer, P. Widmayer, Probabilistic protocols for node discovery in Ad-Hoc, single broadcast channel networks", in IPDPS, 2003, pp. 218.
[9] C. Perkins, E. Belding-Royer, S. Das,"Ad hoc on-demand distance vector (AODV) routing", RFC 3561, July 2003.
[10] T. Salonidis, P. Bhagwat, L. Tassiulas, R. O. LaMaire, "Distributed topology construction of bluetooth personal area networks", in INFOCOM, 2001, pp. 1577.158.

Mr. K Hari Prasad is a student of Kakinada Institute of Engineering & Technology (KIET) Korangi.currently he is pursuing his M.Tech (CSE) (10B21 D5807) from this college. College. He received his Graduation from V.S.MCollege Ramachandra puram in the Year 2007. His Areas of Interests are DBMS and NETWORKING

Mr. D.Srinivas is working as an Assistant Professor in KIET. He has Five years of Teaching Experience. He completed his B.Tech from KIET in 2007.He complete M.Tech from GIET Rajahmundry in 2010.His Areas of interests are DBMS & Networks He had Published his paper in International Journal of computer science & Technology.