Finding Neighbor Node in Asynchronous Sensor Network

Arvind Kishanrao Rathod
Lecturer in Computer Engineering, Department of Computer Engineering, Government Polytechnic Jintur, Maharashtra, India.

Abstract – In sensor network most of the time networks are static in nature. When there is change in the connectivity due to the disruptions in wireless communicating such as power changes, loss of communication between nodes. In this work it describes the continuous neighbor discovery or finding even if the network is asynchronous type. We describe the simple protocol by which nodes are coordinate with each other to get discover in the network. It reduces the power consumptions by the sensor node coordinating with neighbor node along with reduces the discovery time for the node to discover in the network.

1. INTRODUCTION

A sensor network may contain a huge number of simple sensor nodes that are deployed at some inspected 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 continuously heavy traffic, the sensors need not invoke any special neighbor discovery 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 discovery scheme should be used. This paper presents and analyzes such a scheme. 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:

1) Loss of local synchronization due to accumulated clock drifts.

2) 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.

3) The ongoing addition of new nodes, in some networks to compensate for nodes which have ceased to function because their energy has been exhausted.

4) The increase in transmission power of some nodes, in response to certain events, such as detection of emergent situations. 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], [2], [3] address initial neighbor discovery and continuous neighbor discovery as similar tasks, to be performed by the same scheme, we claim that different schemes are required, for the following reasons: 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.

Figure 1 show a typical neighbor discovery protocol. In this protocol, a node becomes active according to its duty cycle. Let this duty cycle be α in Init state and β in Normal state.

We want to have $\alpha << \beta$. When a node becomes active, it transmits periodical HELLO messages and listens for similar messages from possible neighbors. A node that receives aHELLO message immediately responds and the two nodes 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 2 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 find most of its neighbors, the node moves to the Normal state, where continuous neighbor 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. 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). The rest of the paper is organized as follows. In Section II we present related work. Section III presents a basic scheme and problem definition. The core of the paper is Section IV, which presents three methods for estimating the in-segment degree of a hidden neighbor and analyzes their accuracy. Section V concentrates on a special case where the network nodes are uniformly distributed. For this case, we are able to find a numeric value for the accuracy of the three methods presented in IV. Section VI presents our continuous neighbor discovery scheme, which is based on our findings in Section IV. Section VII presents simulation results that demonstrate the scheme's efficiency. It also includes a discussion of problems that arise when two small segments have to detect one another. Finally, Section VIII concludes this work.

2. RELATED WORK

In a Wi Fi network operating in centralized mode, a special node, called an access point, coordinates access to the shared medium. Messages are transmitted only to or from the access point. Therefore, neighbor discovery is the process of having a new node detected by the base station. Since energy consumption is not a concern for the base station, discovering new nodes is rather easy. The base station periodically broadcasts a special HELLO message1. A regular node that hears this message can initiate a registration process. The regular node can switch frequencies/channels in order to _nd the best HELLO message for its needs. Which message is the best might depend on the identity of the broadcasting base station, on security considerations, or on PHY layer quality (signal-to-noise ratio). Problems related to possible collisions of registration messages in such a network are addressed in [4]. Other works try to minimize neighbor discovery time by optimizing the broadcast rate of the HELLO messages [1], [5], [6], [7], [8]. The main differences between neighbor discovery in WiFi and in mesh sensor networks are that neighbor discovery in the former are performed only by the central node, for which energy consumption is not a concern. In addition, the hidden nodes are assumed to be able to hear the HELLO messages broadcast by the central node. In contrast, neighbor discovery in sensor networks is performed by every node, and hidden nodes cannot hear the HELLO messages when they sleep. In mobile ad-hoc networks (MANETs), nodes usually do not switch to a special sleep state. Therefore, two neighboring nodes can send messages to each other whenever their physical distance allows communication. AODV [9] is a typical routing protocol for MANETs. In AODV, when a node wishes to send a message to another node, it broadcasts a special RREQ (route request) message. This message is then broadcast by every node that hears it for the first time. The same message is used for connectivity management, as part of an established route maintenance procedure, aside from which there is no special neighbor discovery protocol.

Minimizing energy consumption is an important target design in Bluetooth [10]. As in WiFi, the process of neighbor discovery in Bluetooth is also asymmetric. A node that wants to be discovered switches to an inquiry scan mode, whereas a node that wants to discover its neighbors enters the inquiry mode. In the inquiry scan mode, the node listens for a certain period on each of the 32 frequencies dedicated to neighbor discovery, while the discovering node passes through these frequencies one by one and broadcasts HELLO in each of them. This process is considered to be energy consuming and slow. A symmetric neighbor discovery scheme for Bluetooth is proposed in [11]. The idea is to allow each node to switch between the inquiry scan mode and the inquiry mode. The 802.15.4 standard [12] proposes a rather simple scheme for neighbor discovery. It assumes that every coordinator node issues one special “beacon” message per frame and a newly deployed node has only to scan the available frequencies for such a message. However, the standard also supports a beaconless mode of operation. Under this mode, a newly deployed node should transmit a beacon request on each available channel. A network coordinator that hears such a request should immediately answer with a beacon of its own. However, this scheme does not supply any bound on the hidden neighbor discovery time. Neighbor discovery in wireless sensor networks is addressed in [2]. The authors propose a policy for determining the transmission power of every node, in order to guarantee that each node detects at least one of its neighbors using as little power as possible. In [1], the authors study the problem of neighbor discovery in static wireless ad hoc networks with directional antennas. At each time slot, a sensor either transmits HELLO messages in a random direction, or listens for HELLO messages from other nodes. The goal is to determine the optimal rate of transmission and reception slots, and the pattern of transmission directions.

In [6], neighbor discovery is studied for general ad-hoc wireless networks. The authors propose a random HELLO protocol, inspired by ALOHA. Each node can be in one of two states: listening or talking. 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. In [5], the sensor nodes are supposed to determine, for every time slot, whether to transmit HELLO, to listen, or to sleep. The optimal transition rate between the three states is determined using a priori knowledge of the maximum possible number of neighbors. In [13], the Disco algorithm is proposed for scheduling the wake-up times of two nodes that wish to find each other. For this algorithm, each node chooses a prime number; the choice depends on the required discovery time. Using the Chinese Remainders theorem, it is proved that the wake-up periods of the nodes will overlap within the required time. However, [13] does not discuss the problem of many sensors in the same segment collaborating to reduce the energy they expend for discovering hidden nodes. As discussed in Section I, the sensor network nodes spend most of their time in sleep/idle mode, where they cannot receive or transmit messages. Therefore, the node's ability to discover a new neighbor is limited to periods when both are active. In [3], this neighbor discovery model is shown to be similar to the well-known .birthday paradox. In our work we use a similar analysis, in order to find the probability that a node will be discovered by one of its neighbors. A novel low-power listening (LPL) technique, proposed in [14] to overcome sensor synchronization problems, is implemented by the B-MAC protocol [15]. The transmission of a packet is preceded by a special preamble. This preamble is long enough to be discovered if each node performs periodic channel sampling. However, this technique can usually not be used for initial neighbor discovery, and cannot be used at all for continuous neighbor discovery, because it actually requires the node to stay awake during the entire time it is searching for a new neighbor.

3. CONCLUSION

This paper is only showing the methodology used to find the neighbor in a synchronous sensor network.

REFERENCES

[1] S. Vasudevan, J. Kurose, and D. Towsley, “On neighbor discovery in wireless networks with directional antennas,” in INFOCOM, vol. 4, 2005, pp. 2502–2512.

[2] R. Madan and 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 and S. A. Borbash, “Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks,” in Mobi Hoc: 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 and 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 and E. Uysal-Biyikoglu, “Energy-efficient link assess-ment in wireless sensor networks,” in INFOCOM, 2004.

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

[7] Author's names. "Title of Article” Name of Publication volume. Issue (year of publication): page numbers. G. Alonso, E. Kranakis, R. Watten hofer, and Widmayer, “Probabilistic protocols for node discovery in ad-hoc, single broadcast channel networks,” in IPDPS, 2003, p. 218.

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

[9] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” RFC 3561, July 2003.

[10] J. Haartsen, Bluetooth Baseband Specification v. 1.0.

[11] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. O. LaMaire, “Distributed topology construction of bluetooth personal area networks,” in INFO-COM, 2001, pp. 1577–1586.

[12] IEEE 802.15.4: Wireless Medium Access Control (MAC) and Physica Layer (PHY) Specifications for Low-Rate Wireless Persona (WPANs), IEEE 802.15 WPAN Task Group 4 (TG4), 2006.

[13] P. Dutta and D. Culler, “Practica asynchronous neighbor discovery and rendezvous for mobile sensing applications,” in SenSys: Proceedings of the 6th ACM conference on Embedded network sensor systems. New York, NY: ACM Press, 2008, pp. 71–84.

[14] J.Hill and D. Culler, “A wireless embedded sensor architecture for system-level optimization,” Technical report, U.C. Berkeley, 2001.

[15] R. Jurdak, P. Baldi, and C. V. Lopes, “Adaptive low power listening for wireless sensor networks,” IEEE Transactions on Mobile Computing,

Author

Arvind Kishanrao Rathod completed B.E in Computer Science and Engineering at MIT College of Engineering, Dr. B.A.M. University, Aurangabad. Presently Working as a Lecturer in Computer Engineering at Government Polytechnic, Jintur District Parbhani, Maharashtra, India, Has 15 Years of Teaching Experience.