CONTINUOUS NEIGHBOR DISCOVERY IN WIRELESS SENSOR NETWORK

R. AKILA1 &  S. G. SANTHI2
1M.E Student, Dept of CSE, Annamalai University, Chidambaram, (E-mail: aki.akila471@gmail.com)
2Assistant Professor, Dept of CSE, Annamalai University, Chidambaram, (E-mail: sgau2009@gmail.com)

Abstract: In sensor network the node connectivity is to be changes of disruption in wireless communication, transmission power changes or loss of synchronization between neighbour nodes. Hence a sensor is aware of its immediate neighbour it must continuously maintain of its view. It’s called continuous neighbour discovery. In this paper it differentiates between neighbour discovery during sensor network initialization and continuous neighbour discovery. It’s used to reduce the power consumption without increasing the time required to detect hidden sensors. In sensor network, the node is to detect their immediate neighbor which they direct communication that is used to establish routes to the gateway.

Keywords: Neighbor discovery, Sensor network.

1. INTRODUCTION

Networks are classified into two main types based on connectivity, wired and wireless networks. A wireless network provides flexibility over standard wired networks. Only the help of wireless networks, the users can retrieve information and get services even when they travel from place to place. Now a day Wireless Sensor Networks (WSN) has emerged as a promising tool for monitoring the physical world. Wireless Sensor Networks (WSN) can provide low cost solution to verify of real-world problems e.g. habitat monitoring, battle field surveillance, disaster management, health monitoring or industrial control etc. A Wireless Sensor Networks (WSN) is a system that consists of thousands of very small stations called sensor nodes. The sensor nodes are deployed in the target area. These nodes form a network by communicating with each other either directly or through the other nodes. The main function of sensor nodes is to monitor, record, and notify a specific condition at various locations to other stations and end users.

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices that use sensors to monitor physical or environmental conditions. These autonomous devices, or nodes, combine with routers and a gateway to create a typical WSN system. The distributed measurement nodes communicate wirelessly to a central gateway, which provides a connection to the wired world where it can be collect, process, analyze, and present the measurement of data. To extend distance and reliability in a wireless sensor network, it can use routers to gain an additional communication link between end nodes and the gateway.

Sensor network initially consists of small or large nodes called as sensor nodes because different sizes of sensor nodes work efficiently in different fields. Wireless sensor networks has such sensor nodes which are specially designed in such a typical way that they have a microcontroller which controls the monitoring, a radio transceiver for generating radio waves, different type of wireless communicating devices and also equipped with an energy source such as battery. The entire network worked simultaneously by using different dimensions of sensors and worked on the phenomenon of multi routing algorithm which is also termed as wireless ad hoc networking.

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. Some prespecified time elapses or Connectivity to a prespecified number of neighbors is detected.

Some prespecified time elapses or Connectivity to a prespecified number of neighbors is detected.

![Figure 1: Continuous Neighbor Discovery Vs Initial Neighbor Discovery in Sensor Networks](image)

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 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.

2. RELATED WORK

Reuven Cohen, and Boris Kapchits[1] described, distinguishes between neighbor discovery during sensor network initialization and continuous neighbor discovery. It focuses on the latter and views 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. J. Hill and D. Culler[2] presented wireless sensor node architecture to achieve high communication bandwidth with the flexibility to efficiently implement novel communication protocols.

A. Keshavarzian, E. Uysal-Biyikoglu and F. Herrmann [3], introduced and analyzes of two different approaches to link assessment: The first approach was a random nondeterministic scheme that allows for a probabilistic guarantee of collision-free packet exchange. An alternative method was described which employs 'constant-weight codes' and provides a deterministic guarantee of success.

E. B. Hamida, G. Chelius, and E. Fleury,[4], the neighbor discovery is studied for general adhoc wireless networks. The authors propose a random HELLO protocols, 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.

R. Madan and S. Lall [5] Neighbor discovery in wireless sensor networks is addressed in energy optimal algorithm. The authors purpose a policy for determining the transmission power of every node in order to guarantee that each node defects at least one of its neighbors using as little power as possible.

In [6] the IEEE 802.15.4 standard 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, this scheme does not supply any bound on the hidden neighbor discovery time.

S. Vasudevan, J. Kurose, and D. Towsley, [7] the author 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.

A. Keshavarzian and E. Uysal-Biyikoglu [9] The aim of this paper, continuous neighbor discovery algorithm is used to determine the frequency with which every node enters the HELLO period. Here it 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. This paper presents three methods to detect their neighbour nodes.

2.1. Method 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.

2.2. Method 2 (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 neighbouring nodes will be active at the same time.

2.3. Method 3 (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.

2.4. An Efficient Continuous Neighbor Discovery Algorithm
In this section it presents 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 rate, where it wakes up every T1 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.

\[ 1 - (1 - P)^D \geq P \]

This can also be stated as

\[ P \geq 1 - \sqrt[1 - P]{D} \]

Since \( P = \frac{H / T_1}{(1-2H (1-\delta)/T_N (u))^{n}} \), we get

\[ \frac{H}{T_1} \left( 1 - (1 - 2(1-\delta)) \right)^n \geq 1 - \sqrt[1 - P]{D}, \]

and therefore

\[ T_N (u) \leq \frac{2H(1-\delta)}{1-n \sqrt[1 - P]{T_1 / H(1-\sqrt[1 - P]{D})}}. \]

Since node v does not know the exact value of \( n \), it can estimate it using the methods of estimating the in segment degree of a hidden neighbor.

3. PROPOSED METHOD
The proposed system is divided into four modules.

3.1. Design the Sensor Network
3.2. Initialization of Sensor Nodes
3.3. Implementation of Continues Neighbor Discovery Algorithm

3.4. Evaluating the Time Performance

3.1. Designing the Sensor Network

A sensor network with a large number of low cost nodes distributed over a wide area. Here 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. 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.

3.2. Initialization of Sensor Nodes

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.

3.3. Implementation of Continues Neighbor Discovery Algorithm

In this paper it presents an algorithm for assigning HELLO message frequency to the nodes of the same segment. This algorithm is used to determine the frequency with which every node enters the HELLO period. If a hidden node is discovered by one of its segment neighbors, its discovery 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.

3.4. Evaluating the Time Performance

It simulates a large sensor network, with that most of the nodes discover each other and enter nodes distributed randomly and uniformly over the area of interest. Assume that the nodes have an equal and constant transmission range. Communication is always bi-directional. It also assumes the continuous neighbor discovery state before the simulation begins. And also 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, are shown to be the best.
feedback when designing real world systems. This allows the designer to determine the correctness and efficiency of a design before the system is actually built and the user may explore the advantages and disadvantages of the new designs by altering the system parameters without actually physically building the systems. Another benefit simulator is that they permit system designers to study a problem at several different levels of abstraction.

NS-2 (Network Simulator version 2) is chosen as the simulation tool because NS-2 is an object-oriented, discrete-event-driven network simulator developed at UC Berkeley written in C++ and OTcl, targeted at networking research, which has been extensively used by the networking research community. Versions are available for Linux, Solaris, Windows, and MAC OS etc. NS-2 also provides substantial support for simulation of TCP, UDP, routing and multicast protocols over wired and wireless networks.

The purpose of our simulation is to observe the total energy consumption and the coverage distance achieved for RD and QRD of the sensor nodes. The number of nodes for the protocol input is varied from 50 to 150 and accordingly, the protocol performance is looked at. The constant bit rate source at the rate of 100kb and packet size of 512 is considered for the protocol input.

NS2 is considered as the appropriate platform to execute the project since it has all the in-built functionality which is necessary for the execution of the network projects. Front end is to be implemented using tcl and back end to back to be in C++ language which is object oriented to implement the project and the number of nodes and new protocol selection is done through the front end tcl.

We consider the network to be consisting of several nodes ranging from 50, 75, 100, 125 and 150. For each of this selection, simulation is run and the results are obtained. We set the network channel to wireless channel and the propagation model to Two Ray Ground. Network interface is set to wireless physical. Since we need to choose the proper MAC layer protocol, Mac 802.11 is chosen in our case. Interface queue is considered to be that of Drop tail’s priority queue. Considering the link layer protocol, Omni directional antenna is taken into consideration with X dimension of topology as 1000 and Y dimension of topology as 1000. Base routing protocol is considered to be that of AODV. Simulation time is taken as 50 and flow varied from one value to another value with specific range.

![Figure 4: Number of Nodes v/s Throughput](image1)

![Figure 5: Number of Nodes v/s Delay](image2)

![Figure 6: Number of Nodes v/s pwb](image3)
4. CONCLUSION

In this paper we identified and analyzed a new problem that arises in wireless sensor networks. It analyzed that continuous neighborhood discovery is crucial if the sensor nodes go to hidden. The neighborhood discovery process should be a continuous one to make involve all the nodes in the communication. So that the nodes can actively participate in the communication which reduces the communication cost. The nodes are connected as segments. This segmentation ensures the maximum probability of neighbor discovery. In this proposed algorithm detects the hidden nodes with the estimated probability.

Further work employs Multi-hop routing for data aggregation at the base station, where several nodes may forward data packets to the base station. This will lead to the reduced lifetime of the network and reduced battery life for sensor nodes near the base station, as they have to relay data from all parts of the network to the base station.

References