COMPARISON OF INITIAL AND CONTINUOUS NEIGHBOR DISCOVERY IN ASYNCHRONOUS SENSOR NETWORKS

[1]Laxmi Shabadi, [2]Praveen . D, [3]Ravi. K, [4]Sidram Raddy & [5]Vinod. P

Department of Computer Science and Engineering,
BLDEA's Dr. P. G. Halakatti College of Engineering and Technology,
(Affiliated to Visvesvaraya Technological University), Bijapur, India.
E-mail : laxmishabadi07@gmail.com

Abstract – **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. An efficient continuous neighbor discovery algorithm is used to discover the neighbor nodes. 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. Existing system is having some of the disadvantages like heavy traffic, long term process, and greater expense of energy. Hence we tried to overcome all these drawbacks and we compared it with the initial discovery.**

I. INTRODUCTION

The sensor network contains very large number of sensor nodes. These sensor nodes may be connected to each other inside a network by any mesh structure. These sensor nodes can sense various events very sensitively. Some of the sensor nodes act as routers and gateways to pass the message or the file from one particular sensor node to another sensor node. In order to pass the data there will be high consumption of bandwidth, energy and even power. Therefore we design this project in such a way that we can minimize these three critical issues. These issues can be overcome by alternatively putting the sensor nodes in active state and passive state. In this paper the sensor nodes are randomly distributed over a particular area and each sensor nodes have certain transmission area to cover. First we are detecting the immediate neighbours. That is called as initial discovery. The sensor nodes should have direct wireless communication between them. Then the sensor nodes should establish the particular shortest roots through which they can communicate with the other sensor nodes via any router or gateway in between. In order to communicate we need to first create communication between two sensor nodes. The sensor nodes will be awake for a very short period of time, that is in a sensor network the nodes are exist for a while and after that they can change their positions or move away from the network. Therefore there can be heavy traffic in the channel or in the particular transmission area. This paper presents a special neighbour discovery scheme that can be used to reduce the traffic that is being caused by the sensor nodes. Another important issue in the sensor network is that the sensor nodes despite of being static can change due to the following situation.

- Loss of local synchronization due to accumulated clock drifts.
- Disruption of wireless connectivity between adjacent nodes by a temporary event, such as a passing vehicle or animal or anything, 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.
- Sometimes the nodes may move away from the network and loses the connection with the other nodes present in a network.
- The increase in transmission power of some nodes, in response to certain events, such as detection of emergent situations.

After resolving the above issues the sensor nodes can be there in two states. One is the Init state and the second is the normal state. Now in this discussion a main idea is to discover the links during the normal operation, and this is referred to as Continuous neighbor Discovery. Now we have to discuss about how the nodes are being discovered by Continuous neighbor Discovery algorithm. At first when the sensor nodes are in Init state, we consider that all the nodes present in a network are active. Now this sensor node will search for any other sensor node which is active. If any sensor node is active at that period, the first sensor node repeatedly transmits data to the next active sensor node. The other sensor node replies back by sending the ACK packet to the previous sensor nodes and therefore the two way communication between the sensor nodes is being established.

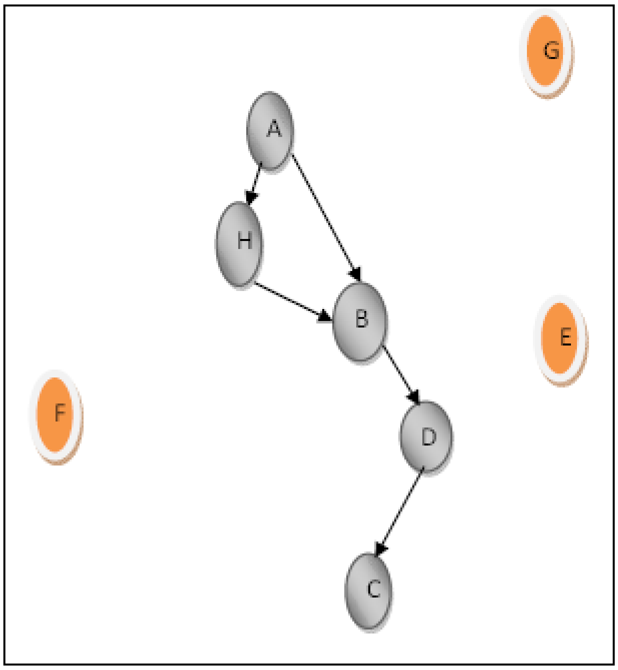

Fig1.1 : The transmission of data in initial and normal states.

The sensor nodes have to be detected by the other nodes by using the joint task of all the nodes. If there is a sensor node A and a sensor node C, then the neighbors of A can be detected by B with the help of H and D.

Here A is a source from where we are trying to send the data to the destination node C. To do this there exist two paths, that is AHBDC and the path ABDC. This is the initial discovery, it will show only the route from source to destination. Here orange colour nodes represent the nodes that are away from the network or the dead nodes. And gray colour nodes are the active nodes.

II. METHODOLOGY

In this section, we introduce the asynchronous neighbor discovery problem, provide its corresponding theoretical formulation, and develop relevant evaluation metrics for mobile sensor networks. For the benefit of the reader, we first introduce the simple neighbor discovery problem, and later generalize the formulation to describe the continuous neighbor discovery problem.

Mobile nodes in sensor networks continually move in space. Due to such motion, the network topology changes over time, nodes within communication range may move farther apart, and nodes outside the communication range move closer to each other. The neighbor discovery problem is one in which each mobile node keeps track of all other nodes within its communication range. This information changes over time and a neighbor discovery protocol is used to continuously update it. The asynchronous neighbor discovery problem is often encountered in mobile networks, where the mobile nodes discovering each other are not necessarily synchronized with each other.

Here we are using two main discovery algorithms namely first one is initial discovery and second is continuous discovery. Those are explained as below.

Initial neighbor discovery:

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.

Continuous neighbor discovery:

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. In a continuous neighbor discovery we are detecting all of it’s neighboring nodes as well as selecting the shortest path for transferring the file or the data.

III. IMPLEMENTATION

Modules

A. Client-Servers

B. Detecting all hidden links Inside and outside the segment

C. Neighbor Discovery Model

A. 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.

B. Hidden link participate Inside and outside the 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 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.

A random wake-up approach is used to minimize the possibility of repeating collisions between the 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 message transmission time is even shorter, the probability that two neighboring nodes will be active at the same time.

C. Neighbor Discovery Model

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

Initial Discovery Algorithm

INPUT: Number of nodes, source IP address, destination IP address.

OUTPUT: Discovers the path from source to destination.

- Enter the number of nodes and give the destination IP address.
- Insert the source node into the discovery table database.
- Discover the nearest node to the source node and enter it into the discovery table, for next operations consider it as the source node.
- Find the nearest node to previously found neighbour.
- Repeat the above steps until all nodes are found.
- Discovery table is now having the entry of nearest node to source and destination. Using these entries the data is transferred from source to destination.

Continuous Neighbor Discovery Algorithm

INPUT: Number of nodes, source IP address, destination IP address.

OUTPUT: Discovers the efficient path from source to destination.

- Enter the number of nodes and give the destination IP address.
- Insert the source node into the discovery table database.
- Discover the nearest node to the source node and enter it into the discovery table, for next operations consider it as the source node.
- If node exists go to step 3.
- Discovery table is now having the entry of nearest node to source and destination. Using these entries the data is transferred from source to destination.
- Discovery table is the database maintained to keep track of the neighboring nodes.

IV. COMPARISON

The already existing system which we have referred, in that they have considered the sensor network as a static network. The number of nodes in that are fixed and the positions are also fixed. They have given the energy for each node, after some time when the energy completes then that node is considered as a dead node and data is transferred through remaining active nodes. After some amount of time all the nodes will die, and there is no network is present to transfer the data. But in real world application the sensor network keeps changing, means nodes may added or deleted from the network.

Hence we are considering the network as a dynamic sensor network. Where we are considering the number of nodes and positions of the nodes are keeps on changing regularly. We consider the criteria

- Number of nodes may increases in sensor network.
- Number of nodes may decreases in sensor network.
- Energy level is fixed.

The following are the disadvantages of existing system

- No clues about structure of immediate surroundings.
- Heavy traffic.
- Long term process.
- Greater expense of energy

To overcome of this problems we are introduce the following methods

- Removing dead nodes is avoided
- Calculated service time and band width.
- Traffic is controlled and the process is smoother.
- Efficient use of energy and time.
- The number of nodes in a network may increase or decrease.

V. RESULT

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

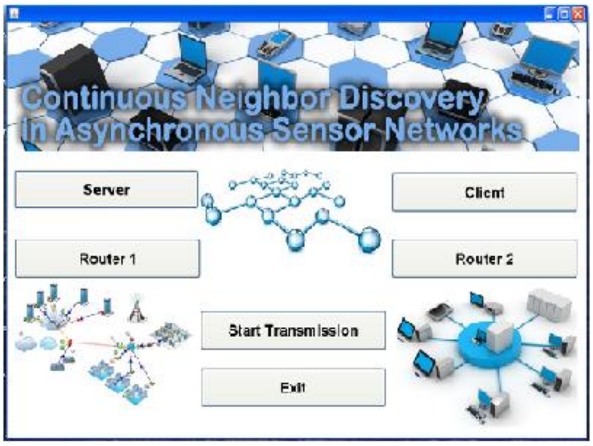
Continuous Neighbor Discovery
in Asynchronous Sensor Networks
Server
Client
Router 1
Router 2
Start Transmission
Exit

Fig 5.1: Modules of the project

This is home page of our project, it consist of six buttons such as server, client,router1,router2,strat transmission and exit button. On clicking each button it performs different action as shown in next pages.

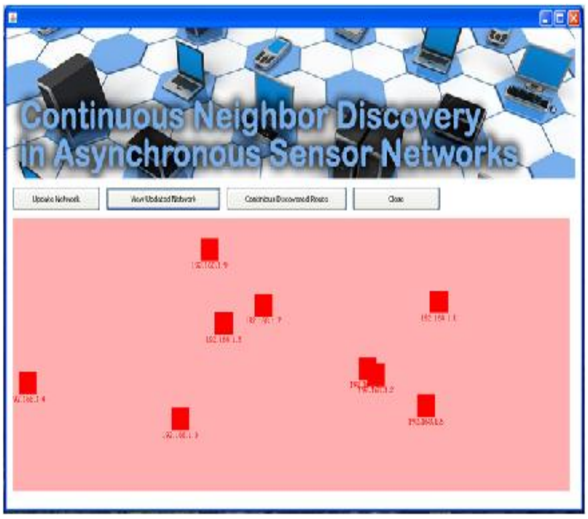
Continuous Neighbor Discovery
in Asynchronous Sensor Networks
Update Network
View Updated Network
Continuous Discovered Route
Close

Fig 5.2: Updating the network by adding random number of nodes in router2

On clicking Router2 continuous neighbor discovery takes place. Here we enter the start(source) IP address and we use the button update network to add the random no of nodes into network.

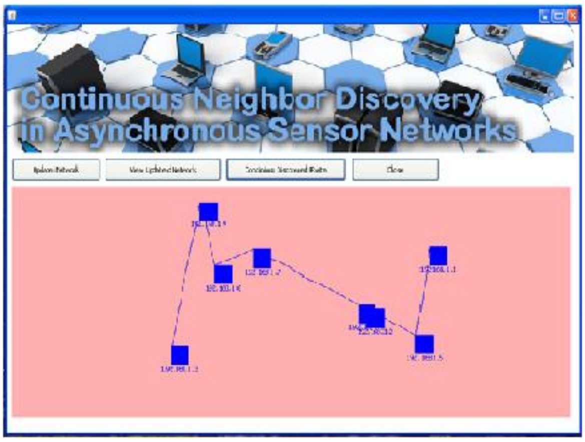
Continuous Neighbor Discovery
in Asynchronous Sensor Networks
Update Network
View Updated Network
Continuous Discovered Route
Close

Fig 5.3: Continuous discovery route in updated network

When we click on show transfer route it will search the minimum path from source node to destination node and show the minimum route.

VI. CONCLUSION

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 dynamic. 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. We then presented a continuous neighbor discovery algorithm that determines the frequency with which every node enters the message period. We simulated a sensor network to analyze our algorithms and showed that when the hidden nodes are uniformly distributed in the area.

VII. REFERENCE

[1] Book : Java 2: The Complete Reference, Fifth Edition –herbert sehildi

[2] S. Vasudevan, J. Kurose, and D. Towsley, .On neighbor discovery in wireless networks with directional antennas,. In INFOCOM, vol. 4, 2005, pp. 2502.2512.

[3] R. Madan and S. Lall, .An energy-optimal algorithm for neighbor discovery in wireless sensor networks,. Mob. Network. Appl., vol. 11, no. 3, pp. 317.326, 2006

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

[5] M. Haenggi, Opportunities and challenges in wireless sensor networks, in: M. Ilyas, I. Mahgoub (Eds.), Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, CRC Press, Florida, 2004, pp. 1.1–1.14.

[6] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Elsevier Comput. Net-work 38 (4) (2002) 393–422.

[7] M. J. McGlynn and S. A. Borbash, .Birthday protocols for low energy deployment and flexible neighbor discovery in

[8] 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.

[9] A. Keshavarzian and E. Uysal-Biyikoglu, .Energy-effcient link assessment in wireless sensor networks,. in INFOCOM, 2004.

[10] E. B. Hamida, G. Chelius, and E. Fleury, .Revisiting neighbor discovery with interferences consideration,. in PE-WASUN, 2006, pp. 74.81.

[11] A Survey on Neighbor Discovery in Asynchronous Wireless Sensor Network K.Sreekanth Asst Professor & S.Sarfara

[12] Amazon Web Services (AWS), Online at http://aws. amazon.com.9.

◈◈◈