ISSN:0975-9646

Ramineni Saichand et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (3) , 2012,4089-4092

Continuous Energy-Efficient Link Assessment in Sensor Networks

Ramineni Saichand, B.Suvarna
Department Of Computer Science and Engineering,
Vignan University,vadlamudi-522213

Abstract- in wireless communication network nodes are static . the connectivity between the node are changed because of disruptions in wireless communication , transmission power changes or loss of synchronization between neighboring nodes . we use a process called continous neighbor discovery in that we continuously maintain immediate neighbors using sensors.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

keyword:- Neighbor discovery, sensor networks.

INTRODUCTION

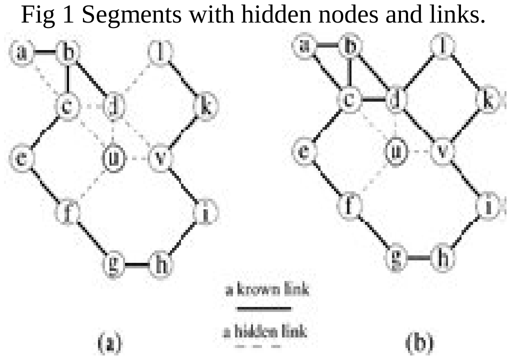
A sensor network may contain a huge number of simple sensor nodes that are deployed at some inspected site. In large areas, such a network usually has a mesh structure. In this case, some of the sensor nodes act as routers, forwarding messages from one of their neighbors to another. The nodes are configured 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. 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.

RELATED WORK

In a WiFi 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 message.1 A regular node that hears this message can initiate a registration process. The regular node can switch frequencies/channels in order to find 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]–[8]. The main differences between neighbor discovery in WiFi and in mesh sensor networks are that neighbor discovery in the former is 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 message 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 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 node detects at least one of its neighbors using as little power as possible.

As shown in the figure These two nodes can learn about their hidden wireless link using the following simple scheme, which uses two message types: 1) SYNC messages for synchronization between all segment nodes, transmitted over known wireless links; 2) HELLO messages for detecting new neighbors.

Fig 1 Segments with hidden nodes and links.



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 , denoted by $d_{egs}^{(u)}$ In this section, we present methods that can be used by node in the Normal (continuous neighbor discovery) state to estimate this value. Node is assumed to not yet be connected to the segment, and it is in the Init (initial neighbor discovery) state. Three methods are presented.

1)Node measures the average in-segment degree of the segment's nodes and uses this number as an estimate of the in-segment degree of . 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 Hence every node.

2)Node v discovers, using Scheme 1, the number of its in-segment neighbors, $d_{egs}^{(v)}$, and views this number as an estimate of $d_{egs}^{(u)}$. This approach is expected to yield better results than the previous one when the degrees of neighboring nodes are strongly correlated.

3)Node uses the average in-segment degree $d_{egs}^{(u)}$ of its segment's nodes and its own in-segment degree to estimate the number of node '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.

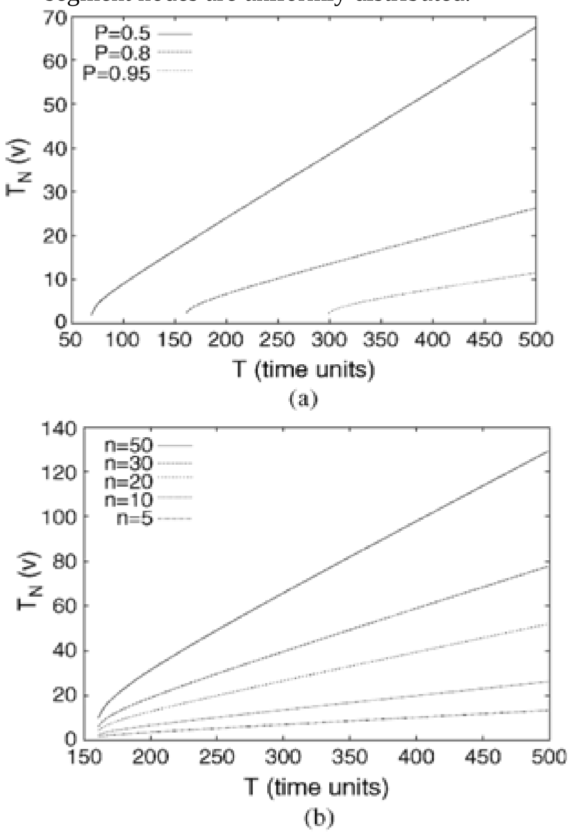


Fig 1

As show in figure 1 we present two graphs that show the dependency between T and $T_N(v)$. We assume that a hidden node wakes up once every 100 H time units on the average, and that $T_I = 100$, $H = 1$, and $\delta = 0.5$. In Fig. 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 three different values of P: 0.5, 0.8, and 0.95. In Fig. 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 a greater $T_N(v)$ to be chosen. In both graphs, the dependency between $T_N(v)$ and T is almost linear and, as we can see in Fig. 6(b), 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 $T_N(v)$

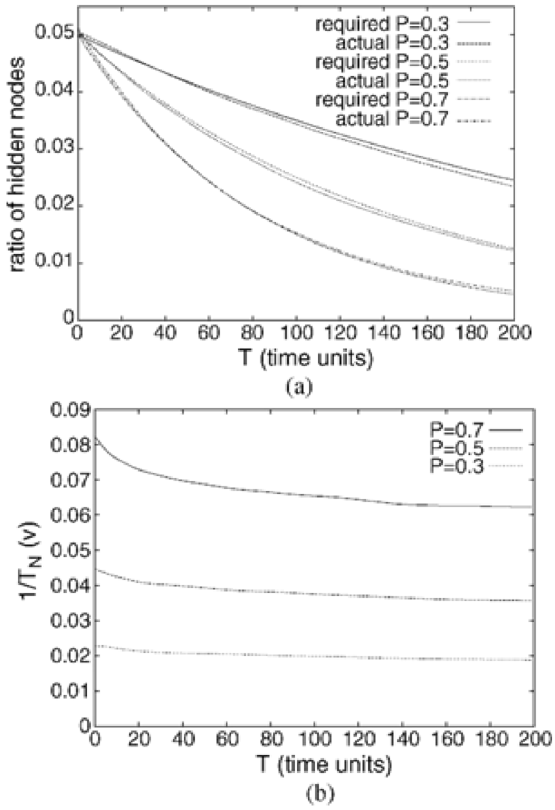


Fig 2 -Hidden neighbor detection for the case of uniform distribution. (a) Decrease in the ratio of hidden nodes. (b) Averag $_N$ as a function of time.

Fig. 2(a) shows the ratios of hidden nodes to the total number of nodes as a function of time. The initial ratio is 0.05. We can see that after 100 time units, this ratio decreases to 0.035 for $P=0.3$, to 0.025 for $P=0.5$, and to 0.015 for $P=0.7$. After 200 time units, the ratios of the hidden nodes are 0.025, 0.012, and 0.005, respectively. It is evident that these results are very close to the required ratios.

In the next simulation, we start with 50% hidden nodes. Fig. 2(b) shows the change in the average frequency of HELLO intervals of the segment nodes, as a function of time, for the same three values of P. We can see that for the smaller value of P (the lower curve), the frequency is almost 75% lower than the frequency for the larger value of P. We can also see that for a given value of P, the average frequency of HELLO intervals decreases with time. This is because as the segment grows, more nodes participate in the discovery process. Similar results are obtained for the case where the initial hidden node ratio was 0.05, but they can hardly be observed due to the small changes in the segment size during the simulation.

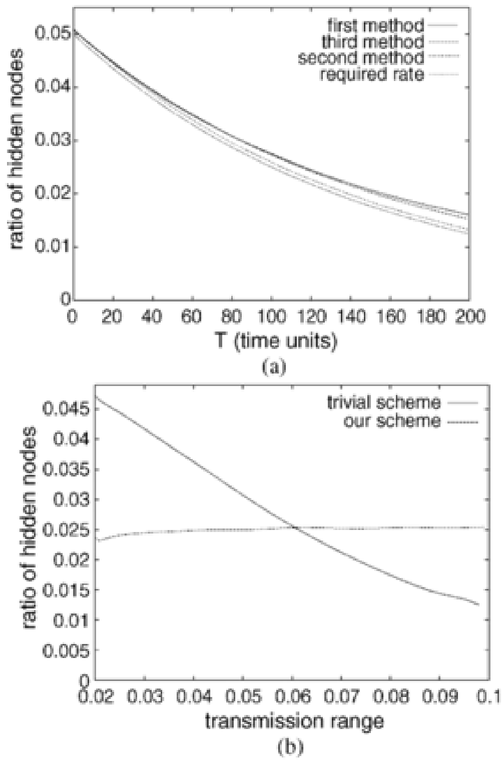


Fig. 3. Hidden neighbors detection with extreme point. (a) Decrease in hidden node ratio. (b) Our scheme compared to a trivial scheme that does not adjust wake-up frequency to the network density.

Fig. 3(a) shows the percent of hidden nodes as a function of time for the three estimation algorithms and P=0.5 Unlike in the uniform distribution case, here we can see some differences between the three algorithms: The second algorithm is the closest to the required rate (shown by a separate curve), where the first algorithm discovers the hidden nodes at a rate slower than the required one.Fig. 3(b) shows the ratio of hidden nodes after T for net-works with different transmission ranges, and hence with dif-ferent node average degrees. This graph reveals the flexibility of our scheme and its ability to adjust the wake-up frequency to the network density. We show this by comparing our scheme to a trivial scheme that does not take the network density into account. For the trivial scheme, all the nodes have the same wake-up frequency. The actual values, which depend on the wake-up frequency of the nodes, are not important. The com-parison shows that the trivial scheme is too aggressive in dense networks and not aggressive enough in sparse ones.

Recall that the goal of our scheme is not to discover nodes as quickly as possible, but to impose an upper bound on the discovery time while minimizing energy consumption. In light of this goal, we see that our scheme performs better because its discovery rate is fixed, and so is its overall expended energy. The simulation starts with 5% hidden nodes, and each node in *Init* is configured with $P = 0.5$. For all transmission ranges, our scheme indeed guarantees that after T time units, the percentage of hidden nodes will decrease by half, to 2.5%. Interestingly enough, the trivial scheme discovers half of the hidden nodes only when the transmission range is ≈0.06. When the transmis-sion range is shorter, the trivial scheme discovers a smaller frac-tion of the hidden nodes. For instance, for a range of 0.03, the ratio of hidden nodes is reduced from 0.05 to 0.04. When the transmission range is greater than 0.06, the trivial scheme dis-covers more nodes during a time period of T. However, this is, of course, with a much greater expense of energy than required in our scheme. We conclude that our algorithm can self-adjust to invest the minimum energy needed to guarantee the required discovery rate, whereas the trivial algorithm cannot. Fig. 4 shows simulation results for the discovery by a small detecting segment. The transmission range is set to 0.3 of the graph, but similar results have been obtained for other trans-mission ranges. It is evident that the desired discovery rate is achieved for a segment of three or more nodes. For segments of two nodes, the discovery rate is faster than the desired rate. In such a segment, the in-segment degree of every node v and the in-segment degree of v's neighbor are both 1. Thus, every in-segment node v estimates the degree of a hidden neighbor u to be 1, while the actual expected degree of u is 1.58 as follows from Theorem 1. Our simulations reveal that for a two-node segment, the in-segment degree of a hidden neighbor should be taken to be 1.4, in which case the target discovery rate is achieved, whereas Algorithm 3 should be used for a larger seg-ment. On the basis othese results, we claim that our algo-rithms can be used for every segment size, despite our assump-tion during the analysis that the segment is "big enough."

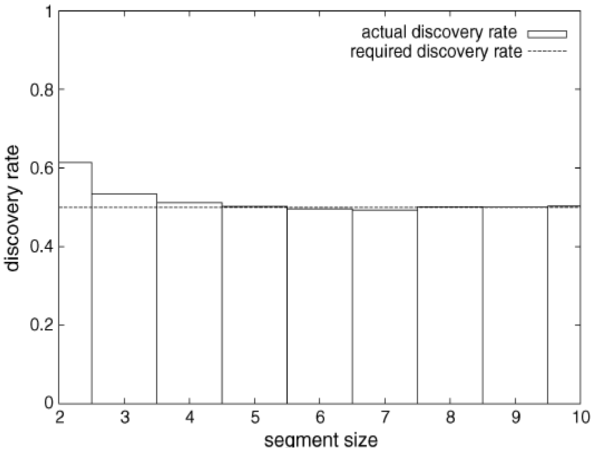


Fig 4.Hidden neighbor discovery rate for a small detecting segment.

FUTURE ENHANCEMENT

In terms of future work, we are currently exploring the problem of node discovery under different network models and with more complex protocols. We are also in the process of studying adaptive protocols that change the probability of transmission or the protocol used as a function of the perceived system size. Such protocols will then be the basic building block for scatter net formation protocols. With this, we expect to be able to define protocols that behave optimally in a wider range of system's sizes and that will shorten considerably the time to set up an ad-hoc network.

CONCLUSION

We exposed a new problem in wireless sensor networks, re-ferred 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 to-gether 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 con-nected to a segment estimates the in-segment degree of its pos-sible hidden neighbors. To this end, we proposed three estima-tion algorithms and analyzed their mean square errors. We then presented a continuous neighbor discovery algorithm that deter-mines the frequency with which every node enters the HELLO period. We simulated a sensor network to analyze our algo-rithms 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.

REFERENCE:

[1]Energy Efficient Neighbor Discovery for Multi-Channel Single-Radio Wireless Sensor Networks, Luca Stabellini Wireless@KTH, the Royal Institute of Technology, Electrum 418, SE-164 40 Kista, and Sweden Email: luca.stabellini@radio.kth.se

[2] Probabilistic Protocols for Node Discovery in Ad-hoc, Single Broadcast Channel Networks (Extended Abstract), G. Alonso, E. Kranakis, R.Wattenhofer, and P.Widmayer, Department of Computer Science Swiss Federal Institute of Technology, ETH Zurich Switzerland

[3] Revisiting Neighbor Discovery with Interferences Consideration, Elyes Ben Hamida, Guillaume Chelius and Eric Fleury, CITI/ARES -INRIA De France

[4] Adhoc on-Demand Distance Vector Routing, Author: Charles E_ Perkins, Sun Microsystems Laboratories, Advanced Development Group