IPHV7I10002X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

PROBABILISTIC PROTOCOLS FOR NODE DISCOVERY IN ADHOC NETWORKS

[1]PURNIMA TANUGULA, [2] K.SRINIVAS
1. M.Tech- Pursuing, 2. Asst.Professor
Dept of CSE, SREE CHAITHANYA COLLEGE OF ENGINEERING, KARIMNAGAR, AP.

ABSTRACT:

Network uses mesh structure where communication area are large especially in A sensor network may contain a huge number of simple sensor nodes that are deployed at some inspected site. In most sensor networks the nodes are static because of disruptions in wireless communication, transmission power changes, or loss of synchronization between neighbor nodes. The process of Neighbor Discovery is identifying the nearest node. So the identification can done through Neighbor Discovery Protocol (NDP) is a protocol . It is responsible for address auto configuration of nodes, discovery of other nodes on the link, and maintaining reachability information about the paths to other active neighbor 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.

Keywords: Networks , wireless sensors, neighbor discovery, hidden link.

I. 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 wireless sensor network is to have only one or two objectives, there is little reason to maintain the generality of the OSI model ("layered networking" [1]) within the network stack of the sensor. We should optimize lifetime or performance by making connections across layers of the network stack. In a general purpose computer, which must handle a large set of applications and conform to many networking standards, layering is a reasonable response to the great complexity of software. In a sensor which has a single application and no requirement to conform to a standard (at this time), layering serves as an obstacle to performance, lifetime, or both. For example, some routing algorithms decide which link to forward a packet onto by choosing the path with smallest end-to-end expected delay. A better fit for an energy-constrained node **[5]** might be to choose the path consuming least total energy. However, energy is a physical layer parameter, which would not be available in the routing calculation if we maintained a strict boundary between the routing, link and physical layers. It is quite unclear how the optimal communication system should work, absent layering. We largely keep the existing layered model, but allow information to be shared between the layers where there are clear advantages to doing so.

Contention-based medium access control (MAC), such as that which is in 802.11 wireless LANs, is often preferred to scheduled access to the wireless medium. However, contention-based MACs have well-known disadvantages, including wide variability in delay of transmissions, poor performance in heavily loaded networks, and wasted energy when multiple users attempt to transmit simultaneously. For energy-related metrics typical of sensor networks, scheduling would clearly be superior for mediating access[5][8].

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.

The main idea behind the continuous neighbour 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 neighbours 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 neighbour u. In order to take advantage of the proposed discovery scheme, node v must estimate the value of degS (u).

II. Existing Work

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 802.15.4 standard [12] proposes a rather simple scheme for neighbour 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

Disadvantages

1. In networks with continuously heavy traffic.

2. Long-term process.

3. Greater expense of energy than required in our scheme.

Proposed 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.

Advantages of Proposed System

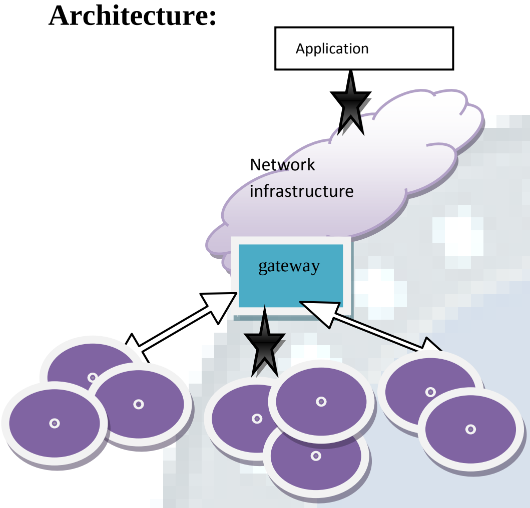
1. Detect their immediate neighbors.

2. Message does not collide with another.

ISSN Online: 2319 – 9253
Print: 2319 – 9245

3. Every node discovers its hidden neighbors independently.

Architecture:



Algorithm:

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 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. One of the three methods presented in Section is used to estimate the number of nodes participating in this effort.

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.

Asynchronous analysis of A

The primary goal of the asynchronous analysis of A is to determine the optimal values of W and pT . The larger W is, the longer the slots are. Large W gives a better chance for the message m to be successfully received during a slot, but for a given period t over which A is to be run, large W reduces the number S of slots We have assumed that if two neighbors simultaneously transmit their messages, X hears garbage during the period of their overlap; X must hear exactly one of his neighbors transmitting a complete message. If multi-user detection were possible, then more transmissions would be successfully received and the performance of this algorithm would be improved. The algorithm produces diminishing returns as it runs. At first, every reception of the message m from a neighbor is new; as the algorithm runs and more messages are received, more of them duplicate messages already received; finally, a node may continue to run A well after it has (unknowingly) discovered all its neighbors. If A runs long enough, a majority of the energy used by A is wasted, since only the first time one transmits successfully to a neighbor is one getting useful information across. Later successful transmissions to that neighbor are redundant. In this was assumed that the running time t is fixed.

If instead we were to choose a stopping point, we could use Equation (5.5) as a basis for determining performance as a function of time. The nodes might adapt local parameter settings based on observations. During A, they listen during every slot they do not transmit. If a nodes hears few transmissions, from the fact that pT is common across the network, it could conclude that it has few neighbors. If this were the case, a higher pT might be called for. Performance might be improved by simple adaptive behavior such as this[4][3].

Proposed Approaches

1. Client – Server

2. Detecting all hidden links Inside a segment

3. Detecting all hidden links Outside a segment

4. Neighbor Discovery Model

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.

Hidden link participate 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.

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 neighboring nodes will be active at the same time.

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.

CONCLUSION:

Neighbor discovery is an important task in sensor networks. This paper presented the asynchronous, distributed neighbor discovery algorithm. The algorithm A impose a state machine structure on the participating nodes and promise simple operation. Furthermore, the probabilistic nature of A makes it robust in the event that the actual number of neighbors is different from what was expected. The algorithm's performance was analyzed in slotted and unslotted time. A major difference between the algorithms is that A is probabilistic, so not all neighbor relations are necessarily discovered. However, simulations indicate that A can discover nearly all neighbors with proper parameter settings.

REFERENCES:

[1] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbour 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 neighbour 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 en-ergy deployment and exible neighbour discovery in ad hoc wireless networks,"

[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-ef_cient link assessment 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] S. A. Borbash, .Design considerations in wireless sensor networks,. Ph.D. dissertation, ISR, August 2004.

[8] G. Alonso, E. Kranakis, R. Wattenhofer, and P. Widmayer, .Probabilistic protocols for node discovery in ad-hoc, single broadcast channel networks,. in IPDPS, 2003, p. 218.

[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 Speci_cation v. 1.0.