Power Aware, Probabilistic and Adaptable Routing Algorithm for Wireless Sensor Networks

M.B. Srinivas, G. Ramamurthy, S. Arora, S. Chaudhary
{srinivas,rammurthy}@iiit.net, {arora,shashank}@gdit.iiit.net

Abstract

Wireless sensor networks are the ‘hot-spots’ of research today. Due to a low per node cost, they are increasingly being deployed for various purposes. This paper introduces a novel power-aware routing algorithm which ensures a very high degree of reliability. This algorithm is ad-hoc, hierarchical, probabilistic and leverages the inherent multiplicity of routes in sensor nets to provide high reliability even under adverse conditions.

1. Introduction

With sensor networks falling in cost as well as in size, they are finding increasing utility in the defense as well as Industrial, Scientific and Medical (ISM) domain. They can be deployed in a factory to check that pressure levels of a boiler, for example, remain constant. If the pressure goes beyond acceptable limit, a sensor at that location would generate an ‘event’ and forward this event to a central ‘sink’ which would then process this event and take appropriate action. Currently, such sensor networks are applying traditional techniques for their routing, medium access, security and other aspects [9][16][17][18]. However, with increasing node densities, the inadequacies of these techniques are becoming increasingly apparent. Primary among these inadequacies is power. Nodes
have limited real-estate and irreplaceable batteries (it may be cheaper to discard a node than to replace batteries, for example, or it may not be possible, as in a battlefield). Traditional setups concentrated on increasing end to end reliability and throughput. This is achieved usually at the cost of battery and computing power. In sensor networks, preserving power is our primary concern. Thus, researchers are interested in devising techniques to ensure maximum utilization of available power. Various techniques at different levels (radio, MAC, network) have been proposed [3][6].

In this paper we present a routing protocol that inherently conserves power by not transmitting any control packets, or maintaining history and state tables and provides high reliability by transmitting multiple packets. Our protocol is highly scalable, reliable and adaptable. It can scale from high to very high node density since it does not utilize control packets (except for deciding the hierarchical position of a node) and hence will not fail in the presence of a very large number of nodes. We do not broadcast packets; instead, each packet takes a random directed walk. It is also reliable under most conditions. This is because a single packet takes multiple routes towards the ‘sink’. In unreliable environments, the transmission probability can be increased to ensure a higher probability of the packet reaching the sink, at the cost of power. The algorithm is adaptable, in that it can adapt to dynamic topologies by reconfiguring the hierarchy (or level) of the nodes. Many ad-hoc routing protocols depend upon a hierarchical partitioning of the network to achieve scalability and adaptability [14][15][13][12][11][10][4][7]. We also use hierarchical partitioning to ensure this. We are also working on a method for efficiently and dynamically configuring the hierarchy of sensors. This is the subject of a related research.
Our routing algorithm utilizes the fact that routing in sensor networks is many-to-one, i.e. every packet has a single final destination, called the ‘sink’. The sink is usually not constrained in power. The sensors are called the sources and generate packets to be routed to the sink. This is in contrast to traditional networks, where a message can have any node as its final destination.

Several algorithms for routing in wireless environments have been proposed in the literature [5][8][22]. They can broadly be classified as a priori and on demand. A priori algorithms set up routes between nodes before a message needs to be transmitted. When a message is received, a node looks up the routing table and decides where to forward the message. These types of algorithms suffer from having to maintain state information and other overheads which drastically reduce their utility in sensor networks. On demand attempt to set up routes only when a message is received. They work well under conditions of low traffic. Some examples of the current algorithms are Dynamic Source Routing (DSR) [17], Temporally Ordered Routing Algorithm (TORA) [16] and Ad-hoc On Demand Distance Vector Routing (AODV) [18]. Although they are ad-hoc, these algorithms suffer from the drawback of maintaining state information and utilizing control packets. DSR, for example, uses ROUTE REQUEST and ROUTE REPLY packets for route discovery and to build its tables. These add to the overhead in the sensors and contribute to unnecessary power usage, especially when the node density is high.

Our routing algorithm works on the technique of passive participation, first introduced by Aron and Gupta [23] in Witness Aided Routing (WAR), then by Marti et al [2] and then extended by Karlof et al [20].
Our algorithm differs from most other algorithms in that messages are not routed by specifically addressing them to the next hop. Instead, we make use of the broadcast property of wireless media as well as the high node density of sensor networks. Since nodes can listen in to messages even if they are not addressed to it, every node is a potential next hop. A node will decide, based on a probability inbuilt into the message header and the position of the node, whether it should transmit the packet it has received, or discard it. If the node is ‘closer’ to the destination than the previous ‘hop’, the node will look at the probability in the message otherwise it will discard the packet irrespective of the probability contained in the packet. If it decides to transmit, it will transmit the message to the level closer than its own level. The high node density ensures that there are enough nodes willing and able to further propel the packet. Thus, our algorithm does away with the notion of source and destination addressing and instead relies on the relative position of the nodes. We feel this is suitable for sensor networks because of the many to one routing prevalent there.

![Fig. 1. Example of Routing and Levels](image)
Fig. 1 shows how a typical sensor network. The solid circle is the sink while empty circles are sources. The possible paths followed by a message from an outermost node to the sink if it uses our protocol are also shown. Note that each packet takes a constant number of hops to reach the destination and the number of hops is equal to the level of the source.

2. The Algorithm

The following formula is used to compute the power required to transmit a message to a node d distance away.

\[ e = kd^c \]

where k and c are constants for a specific wireless system. c is usually between 2 and 4. Hence, direct transmission from source to sink is not a viable option, and the use of intermediate nodes will shorten the distance required to transmit and the power needed by the system.

A. Description

We assume that each node knows its location in the hierarchy (level). A technique for this is the subject of related research. The level of the node is determined by its distance from the sink. The sink is always located at level 0 and is the only node at that level. Growing radially outwards from the ‘sink’, nodes start falling in level 1,2,…, and so on. A node needs to know only its level. An addressing scheme, if required, can be utilized, but is not required for the purpose of this algorithm.
A message is introduced by a node into the network. The message contains a unique message ID (for example, the source address+event ID), the event ID, the level of the source and the transmission probability. The transmission probability is set by the source and depends on how reliable the network is. A greater probability will increase the probability of the message reaching the sink, but will also increase the power usage in the network. In a sparse network, or networks where nodes frequently fail, die out, or are prevented from transmitting due to obstructions, a high transmission probability (> 9.5) would be preferred. In very dense networks, with reliable transmission, a low probability of transmission is preferred.

When a node receives a message, it checks the level of the source. If the source level is greater than its own level, i.e. the node is closer to the sink than the source of the message, the node will retransmit the message with a probability equal to the transmission probability. Before transmitting,
the node updates the source level field in the message with its own level, and if necessary the
transmission probability. In networks with a high node density at the periphery (further away from
the sink) and a low node density near the sink, subsequent nodes in the route should increase the
transmitting probability before transmitting the packet. In networks with a high node density near
the sink, and low node density at the periphery, the nodes should decrease the probability of
transmission. If the density of nodes is uniform throughout, the transmission probability should
remain constant. If the node decides to transmit the message, it will broadcast the message with
sufficient power such that the packet can reach up till the end of the next level and not further.
Since the width of each level remains constant, this power value can also be kept constant. An
optimization is that a node transmits with a lower probability than that in the message header when
the node is low on power (when the node is about to die, we would like it conserve power). Further,
due to the high node density in sensor networks, there is a high probability of at least one node at a

Pseudo Code to process a received packet in a node using our routing algorithm. threshold (6) and factor (7) can be set as per requirements of individual networks. Transmit probability (4,7,10,11) is contained in message header. A node may or may not change the outgoing transmit probability (10) depending upon the network conditions and topology.

```
1. function Process Received Packet returns NULL;
2.  Check Packet Source Level;
3.  If Source Level > Own Level
4.     Get Packet Transmit Probability;
5.     Check Own Power Level;
6.     If Own Level < threshold
7.         Transmit Probability = Transmit Probability / factor;
8.     End If
9.  Change Packet Source Level to Own Level;
10. Change Packet Transmit Probability;
11. Transmit Packet with Transmit Probability;
12. End If
13. Return;
```


lower level listening to the packet. If the node decides not to transmit the packet, it quietly drops the packet. The flowchart of the algorithm is given in Fig. 2.

B. Performance Analysis
To analyze our algorithm, we simulated networks of 100 and 1000 nodes uniformly distributed over an area. Events are generated randomly. The simulator did not have any MAC or Radio layer, and statistics gathered were reliability, number of messages/events generated and number of messages delivered. We assume power consumption of 45 mW while transmitting and no power usage in receiving or idling. Graph 1 shows the reliability of the network (i.e. number of messages delivered to number of messages/events generated) in a 100 node network with a density of 0.83 nodes/unit area with different transmission probabilities. Graph 2 displays the same information with density equal to 0.5. We note that at higher densities, the performance of our algorithm is much better. Also note that at lower transmission probabilities ($P = 0.5$), reliability does not degrade even after a large number of packets have been transmitted. Graph 3 illustrates the importance of choosing the appropriate transmission probability. The time to partition (number of messages delivered until the first node dies) v/s the transmission probability is plotted for a 1000 node network. A slight change in the probability has a significant impact on time to partition. We also note that although the network partitions early, it remains reliable even after a considerable number of messages have been delivered (as shown in graphs 1 and 2). Node failures are frequent and expected in sensor networks [1]. In our simulation, nodes die only when they run out of power. Each node has enough power to transmit 10,000 messages in its lifetime i.e. maximum lifetime of a network of 100 nodes is 1,000,000 messages. In our simulation, we allow events to occur at any location. Events that occur at ‘dead’ nodes count towards the total number of messages but are counted as unsuccessfully delivered messages. Thus, once the network is ‘dead’ (that all nodes have no power left) events are still generated but they don’t propagate in the network. The graphs
given above relate the number or messages originated to the number of messages delivered successfully.

Further simulations (in GloMoSim [19]) and analysis are going on.

3. Conclusion

We have developed a power aware routing algorithm for wireless sensor networks. Our initial analysis shows that the protocol is reliable for a large lifetime of the network. Further analysis is ongoing. A related development for partitioning of wireless networks is an offshoot of this research.

4. References

5. R.D. Poor “Gradient Routing in Ad Hoc Networks”


10. A.B. McDonald, T. Znati ‘A Mobility Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks”

11. M. Joa-Ng, I. Lu ‘A Peer -to-Peer Zone Based Two-Level Link State Routing for Mobile Ad Hoc Networks”, IEEE Journal on Selected Areas in Communication vol.17 no. 8 , Aug 1999

12. G. Pei, M. Gerla, X. Hong ‘LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility”, 2000


15. N. Deijar ‘Zone Routing Protocol”


20. C. Karlof, Y. Li, J. Polastre “Algorithm for Robust Routing in Volatile Environments”

