An Adaptive Self-Organization Protocol for Wireless Sensor Networks

Kil-Woong Jang[1] and Byung-Soon Kim[2]

[1] Dept. of Mathematical and Information Science, Korea Maritime University
1 YeongDo-Gu Dongsam-Dong, Busan, Korea
`jangkw@bada.hhu.ac.kr`
[2] Dept. of Computer Education, Andong National University
388 Songchon-Dong, Andong, Korea
`byungsoon_kim@hotmail.com`

Abstract. This paper proposes a new self-organization protocol for sensors with low-power battery in wireless sensor networks. In our protocol, sensor networks consist of a hierarchical architecture with a sink, which is a root node, using the spanning tree algorithm. Our protocol utilizes some control messages to construct a hierarchical architecture, and by exchanging the messages between nodes, maintains adaptively the network topology by reorganizing the tree architecture as the network evolves. We perform the simulation to evaluate the performance of our protocol over wireless sensor networks. We provide simulation results comparing our protocol with the conventional approaches. The results show that our protocol outperforms other protocols over a broad range of parameters.

1 Introduction

Sensor technology is one of the most challenging technical issues in ubiquitous networks. As sensors have currently the functions of sensing, data processing and communication, a wide range of monitoring applications, such as temperature, pressure, noise and so on, has been commonly studied in the literature [3-6]. Wireless sensor networks have characteristics as follows. A wireless sensor network consists of a large number of sensors, which may be very close to each other, and has a multi-hop wireless topology. Sensors are able to communicate directly in the transmission range. However, to send data to the destination beyond the transmission range, they have to communicate through some intermediate sensors. In addition, they have the constrained batteries, which cannot be recharged or replaced. That is, wireless sensor networks have the different environment than other wireless networks. Therefore, the conventional network protocols are not well applied to the application of wireless sensor networks. To accomplish the above functions, sensors must have capabilities to perform signal processing, computation, and network self-organizing capabilities to achieve scalable, robust and long-lived networks [1,2,7]. Specifically, the low power consumption of sensors is one of the most important requirements of network protocol in wireless sensor networks. The conventional wireless network protocols focus on high quality of services, but wireless sensor network protocols aim primarily to achieve power conservation owing to consisting of sensors with limited, irreplaceable batteries.

Several network protocols have been studied to extend network lifetime in wireless sensor networks. Direct communication protocol [5] is that each sensor transmits directly data to the sink. In this type of protocol, transmit power of each sensor is different due to the distances from the sensor to the sink. If the sensor is far away from the sink, it will quickly dissipate its power. On the other hand, it close to the sink can transmit data using a relative low transmit power. Minimum transmission energy (MTE) routing protocol [4] is designed each sensor to send data for next node by using minimum power. In this type of protocol, the sensors route data destined for the sink through intermediate sensors. Intermediate sensors act as routers for other sensors' data and are chosen such that the transmit power is minimized. The drawback of using this protocol is, due to relaying data, sensors closest to the sink die out first, whereas sensors furthest from the sink die out latest. Another power aware communication protocol is clustering protocol [5,6], where sensors are organized into clusters. In each cluster, a head exists in order to aggregate data from sensors and transfer them to the sink. The head can be decided by static or dynamic methods. For relaying other sensors' data, the head consumes its power more than others, so it would die quickly out.

In this paper, we propose a new self-organization protocol to extend the system life by solving the draws of the conventional protocols and reducing the power consumption of all sensors in the network system. Our protocol is designed by being based on the spanning tree algorithm and makes use of extra control messages to maintain the tree architecture. As all sensors send data destined for the sink by consuming minimum power and dissipating uniformly the power of all sensors, the lifetime of sensor networks is extended.

2 Our Protocol

For analyzing and evaluating the fundamental performance of our protocol, we first describe the power conserving behavior of the protocol. In wireless sensor networks, all sensing nodes have the maximum transmission range, and they send data to the sink directly or through intermediate nodes in this range. In our protocol, if the sink exists in the maximum transmission range of nodes, the nodes directly send data to the sink. Otherwise, they send data to the sink through intermediate nodes closest to the sink.

We first define notations used in our protocol before describing the detail operation of our protocol. A sink and all nodes have a level. The level of the sink is initially set to zero and that of all nodes is set by an infinite value. To distinguish a node with others, each node has a unique identification (ID),

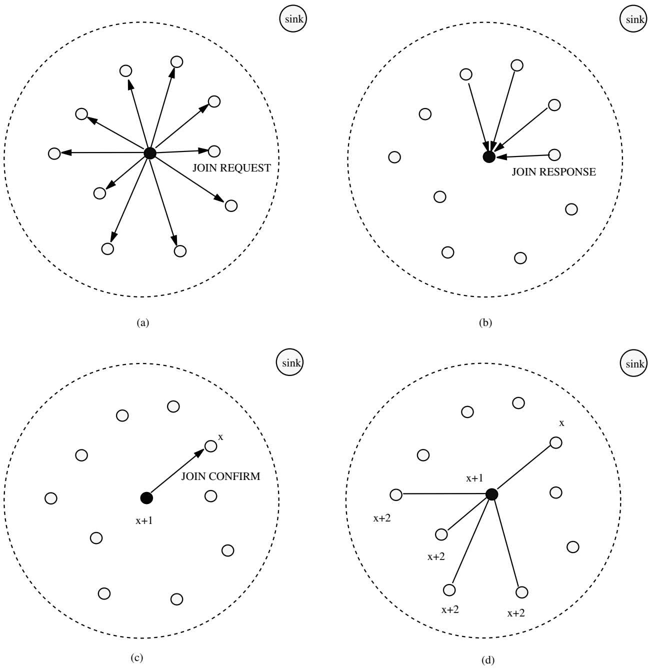

Fig. 1. An example of the join phase in our protocol. Node A is marked with •, and a dashed line is the maximum transmission range of node A. Labels (x, x+1 and x+2) on nodes are their level. (a) node A sends a *JOIN REQUEST* message to neighbors. (b) neighbors reply node A with a *JOIN RESPONSE* message. (c) node A sends a *JOIN CONFIRM* message to the parent. (d) network links after the join phase for node A.

which is generally its MAC address. The operation of our protocol consists of the join and rejoin phases. The join phase is a process that each node joins the network when it powers on. In our protocol, each node is initially set by a constant threshold value, and if its energy is less than this value, it is unable to act as intermediate node for other nodes. Suppose a node sends data to the sink through an intermediate node. If energy of the intermediate node is less than the threshold value, the intermediate node cannot relay data to the sink. To maintain the connection with the sink, the node must join the network again by choosing new intermediate nodes, which is called the rejoin phase.

Initially, when each node powers on, it operates the following procedure to join a wireless sensor network, as shown in Fig. 1. To simply describe the operation, suppose a node A powers on right now. Node A first sends a *JOIN REQUEST* message to the neighbors, which are sensors or the sink. This message includes node A's ID and level. Upon receiving the message, the neighbors compare their level with node A's level. If their level is lower than node A's level, they send a *JOIN RESPONSE* message to node A. This message includes the sender's ID, node A's ID, the sender's level and the number of children of sender. If node A receives the *JOIN RESPONSE* messages, node A chooses the node sending the message with the lowest level as its parent, and replaces its level by parent's level plus one. For example, node A receives the message from the sink, the sink is parent of node A and node A's level becomes to one. If more than nodes with the lowest level can be chosen, node A chooses a node, whose number of children is the lowest, as its parent. In addition, if there are chosen some parent, then node A randomly chooses one of the nodes as its parent. The main reason using this approach is because each node can send data to the sink using minimum hop count and we can balance the number of child of intermediate nodes. Therefore, the advantages of our protocol is that it minimizes the energy consumption in each node, so prolong network lifetime. On selection of the parent, node A sends a *JOIN CONFIRM* message to the parent. This message includes node A's and parent's IDs. Finally, the parent records node A's ID and increments the number of children in its memory. However, if node A does not receive any response messages from neighbors, node A fails to join the network because no neighbor exists in the transmission range of node A.

After completely joining the network, if node's energy is lower than threshold, the rejoin phase is operated, as shown in Fig. 2. Suppose the energy of node A's parent is below threshold. Node A's parent sends a *RELEASE REQUEST* message to node A, its child. Node A receiving the message, in order to find a new parent, sends a *PROBE REQUEST* message its own ID and level to neighbors. Neighbors, except parent and childen of node A as well as the nodes with energy lower than threshold, send a *PROBE RESPONSE* message to node A. The message includes the sender's ID, node A's ID, the sender's level and the number of child of sender. Upon receiving the message, node A chooses a node with lowest level as new parent, and replaces its level by the parent's level plus one. If more than nodes with the lowest level can be chosen, node A chooses a node having the lowest number of child as new parent. If more than nodes with the lowest number of child can be chosen, then node A randomly chooses one of the nodes as new parent. On selection of new parent, node A sends a *PROBE CONFIRM* message to the parent. This message includes node A's and parent's ID. Finally, the parent records node A's ID and increments the number of child in its memory, and finally the rejoin phase is accomplished. However, if node A does not receive any *PROBE RESPONSE* messages from neighbors, node A reset its own level as infinite and must carry out the join process again. In addition, node A sends a *RELEASE REQUEST* message to children.

During this phase, if node A's level is changed by other value, node A must inform children. Thus it sends a *CHANGE LEVEL* message with its own ID and level to children. Upon receiving the message, each child replaces its level by node A's level plus one, and if it has children, it also sends a *CHANGE LEVEL* message to children.

Intermediate nodes relaying data between nodes and the sink play a role in our protocol such like the head in the clustering protocol. In the clustering protocol, the head is generally chosen without respect to a distance from the

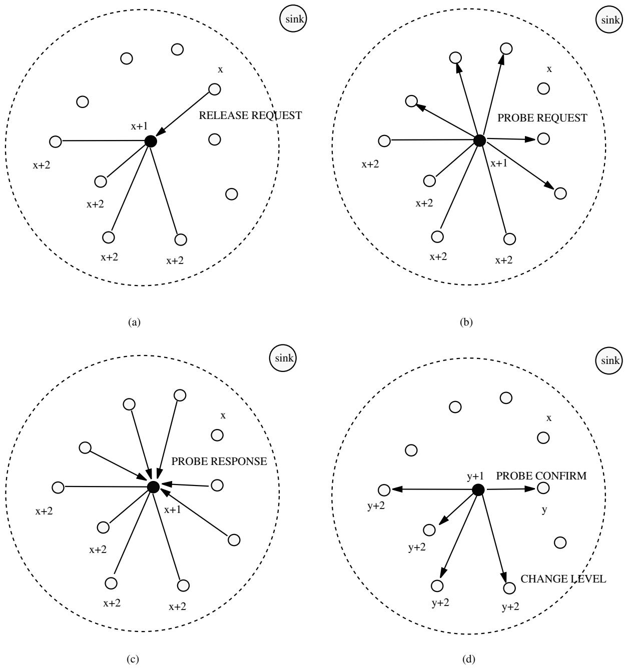

Fig. 2. An example of the update phase in our protocol. (a) node A's parent sends a *RELEASE REQUEST* message to node A. (b) node A sends a *PROBE REQUEST* message to neighbors. (c) neighbors reply a *PROBE RESPONSE* message to node A. (d) node A sends a *PROBE CONFIRM* message to new parent. If node A's level is changed, node A sends a *CHANGE LEVEL* message to children.

sink and aggregates data from nodes, thus the head dies out quicker than other nodes. However, in our protocol, each node sends data to the intermediate node closest to the sink, and changes the intermediate node having low energy with a new intermediate node having highier energy. Thus, our protocol can reduce energy dissipated from nodes and solve the problem, that specific nodes die out quickly. In addition, in the direct communication protocol, nodes furthest from the sink die out first, but this problem also can be solved by using our protocol.

As a media access control protocol to transmit data and control messages, we use the IEEE 802.11 power saving mechanism [8]. In IEEE 802.11 power saving mechanism, power management is done based on Ad hoc Traffic Indication (ATIM). Time is divided into beacon intervals, and every node in the network is synchronized by periodic beacon transmissions. However, since our protocol maintains the hierarchical tree structure, after synchronization between parent and child is carried out only once at first, then all nodes in this structure do not need to synchronize more. So every node will start and finish each beacon interval almost at the same time. At the start of each beacon interval, there exists an interval called ATIM window, where every node should be in awake state and be able to exchange messages. If a node A joins or rejoins neighbor, it sends a corresponding message to the node during this interval. On the other hand, if node A has data destined for sink, it sends an ATIM packet to intermediate node during this interval. If the intermediate node receives this message, it will reply back by sendig ATIM-ACK to node A, and both nodes will stay awake for that entire beacon interval. If the intermediate node has not sent or received any messages or ATIM packets during the ATIM window, it enters doze mode and stays until the next beacon time. This allows the radio component of each node to be turned off at all times except during its transmit times, thus minimizing the energy dissipated in the individual sensors.

3 Performance Evaluation

We carried out the computer simulation to evaluate the performance of our protocol. In this section,we describe performance metrics, simulation environment and simulation results.

3.1 Performance metrics

In order to compare the performance of our protocol and the conventional protocols, the performance metrics that we are interested in are

a) network lifetime (T),
b) number of nodes alive (N), and
c) total energy dissipated in the network (E).

3.2 Simulation Environment

The network model for simulation consists of randomly placed nodes in a constant size square area. Let s denote the network diameter and n denote the total number of nodes in the network. We assume that there are n nodes distributed randomly in a $s\times s$ region, and a sink is positioned at the center of the network. For example, if the network has a 100×100 meter area, the coordination of the sink is positioned at (x=50, y=50). Simulations are performed in wireless LAN environment. The data rate is 11Mbps, and the transmission range of each node is 15 m. In addition, beacon interval is set to 100 ms, and ATIM windows are 200 ms. We assume that each node has 2000 bits data packets and 160 bits control packets, and energy and threshold of each node were assigned 0.5 J and 0.5, respectively. We also assume that if a node dies out, it is not recharged or replaced by a new battery, and the event sensing by nodes is exponentially distributed with rate λ.

In a wireless sensor network, the energy of nodes is mainly dissipated in transmit and receive modes. In order to measure the energy dissipation of nodes, we use a radio model developed in [5]. In this model, nodes have the transmitter and receiver circuitries, which operate independently. The transmitter circuitry

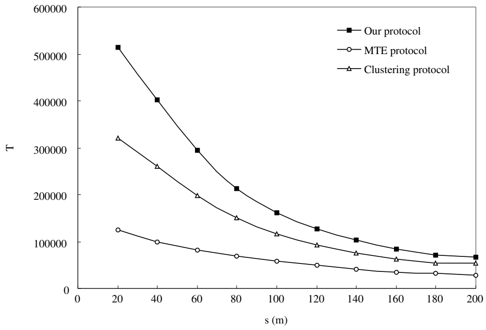

Fig. 3. T for various values of s

consists of a transmit electronics and a transmit amplifier, and the receiver circuitry consists of a receive electronics. Let E_e be the energy dissipated in transmit and receive electronics and E_a be the energy dissipated in transmit amplifier. We assume that $E_e = 50\ nJ/\text{bit}$ and $E_a = 100\ pJ/\text{bit}/m^2$. We also assume that the energy loss happens according to a distance between source and destination. Therefore, transmit energy, $E_t(k,d)$, and receive energy, $E_r(k)$, dissipated to send a k bit data packet to a destination apart a distance d are as follows:

$$E_t(k,d) = E_e * k + E_a * k * d^2. \tag{1}$$

$$E_r(k) = E_e * k. \tag{2}$$

3.3 Simulation Results

Using some results obtained by simulation, we compare our protocol with the MTE and clustering protocols. In the simulation, according to s, n and λ over a network maintained by a sink, we obtained some performance results, which are T, N and E.

We first experiment on different diameters of the network. We measure the network lifetime for our protocol, the MTE and clustering protocols. The result is in Fig. 3. In this simulation, there are a sink and 200 nodes in the network. In this figure, we can see that our protocol performs better than other protocols over all ranges of parameters. If s<60 m, the lifetime of our protocol is on average 1.5 times longer than that of the clustering protocol and average 2 times longer than that of the MTE protocol. If s>60 m, the lifetime of our protocol is also on 20 $\sim$ 50 % longer than that of both. The reason is because all nodes constantly dissipate their energy and they use the optimized transmission path in order to send data to the sink.

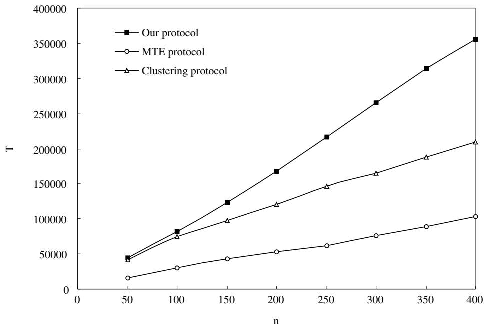

Fig. 4. T for various values of n

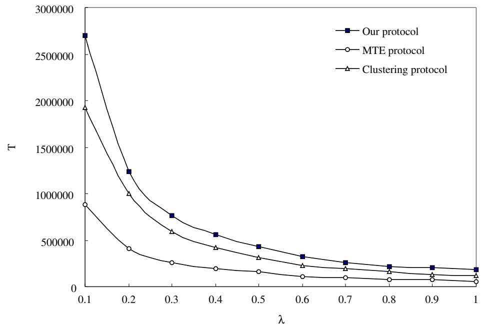

Fig. 5. T for various values of λ

Fig. 4 shows the network lifetime according to number of nodes in the network. In this simulation, we used the network with a 200 m diameter. For all the cases we clearly see that as n increases, T for all protocols increases accordingly. Specially, our protocol achieves 2 or 3 times extension in network lifetime compared with other protocols.

Fig. 5 shows network lifetime as we increase λ of nodes for s=200 m and n=200. As like the above results, our protocol outperforms more than other protocols. Note that increase of λ causes to increase an amount of data destined for the sink. As shown in Fig. 5, as λ increases, T for all protocols exponentially decreases but our protocol outperforms other protocols by a slight difference.

Fig. 6 shows the number of nodes alive for various simulation times, t. In this simulation, 200 nodes are deployed in 200×200 meter environment with rectangular topology. In this figure, we see nodes using the MTE protocol die out earlier than other protocols. To send data to the sink, the number of nodes in the MTE protocol need more than in other protocols. Therefore, since the

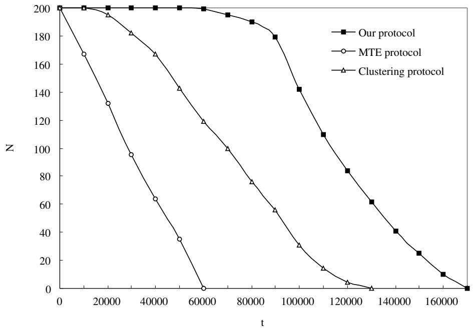

Fig. 6. N for various values of t

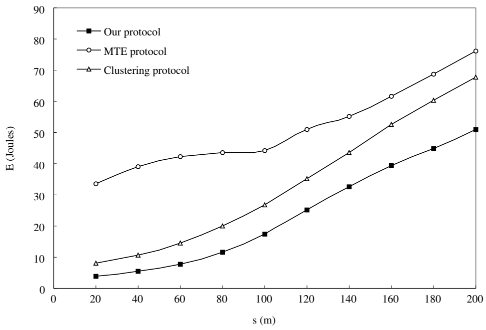

Fig. 7. E for various values of s

intermediate nodes dissipate their energy more, the nodes become to die out quickly. Similar to the above reason, in the clustering protocol, the head in each cluster dies out quicklier than other nodes. In this simulation, we do not consider the battery recharge or replacement of nodes. Note that as the number of die-out nodes increase, remained active nodes will dissipate their battery more than previous. Therefore, in Fig. 6, the MTE and clustering protocols decrease faster than our protocol.

Fig. 7 shows the total energy dissipated in the network for different values of s and n=200. In this figure, we see all the protocols increase accordingly the dissipated energy. If s<100 m, plot of the MTE protocol increases the different pattern than other protocols. Although a distance between nodes and the sink is close, they use an optimal path to send data to the sink. Thus the hop count is increased and the dissipated energy of the nodes on the path will be increased. On the contrast, if the network size is small, our protocol sends data to the sink directly. In addition, if the network size increases, our protocol establishes the optimal path for all nodes and thus can increase the energy efficiency.

4 Conclusions

In this paper, we have presented a new self-organization protocol to maintain efficiently the energy of sensors in wireless sensor networks. The basic idea of our protocol is to establish an optimal path of each node destined for a sink, and balance the energy of all nodes in wireless sensor networks. To achieve our protocol, each node has its level related to a distance between the node and the sink, and chooses a node with the lowest level and the lowest number of child as its parent. Each node also adaptively changes the parent with another node using the threshold and node's level, If intermediate node, which relays data destined for the sink, has its energy less than the threshold, another node, which has energy more than the threshold, is chosen as new parent. Therefore, we are able to maintain the optimal path of nodes continuously and reduce the number of die-out nodes. Moreover, we can extend network lifetime to balance the energy of all nodes without respect to a distance between the node and the sink. Using the simulation, we evaluated the performance of our protocol in terms of the network lifetime, the number of node alive and the energy dissipated in the network. The simualtion results illustrated that our protocol outperformed the conventional protocols over various range of parameters.

References

1. I. Akyildiz, W. Su, Y. Sankarasubramanian and E. Cayiraci.: Wireless sensor networks: a survey. Computer Networks, No. 38, (2002) 393–422
2. K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie.: Protocols for Self-Organization of a Wireless Sensor Network. IEEE Personal Communications, Oct. (2000) 16–27
3. A. Cerpa and D. Estrin.: ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies. Proceedings of IEEE Infocom, (2000)
4. X. Lin and I. Stojmenovic.: Power-Aware Routing in Ad Hoc Wireless Network. In SITE, University of Ottawa, TR-98-11, Dec. (1998)
5. W. R. Heinzelman, A. Chandrakasna and H. Balakrishnam.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks, IEEE Transaction on Wireless Communications, Vol. 1, No. 4, Oct. (2002) 660–669
6. M, Younis, M. Youssef and K, Arisha.: Energy-aware management for cluster-based sensor networks. Computer Networks, No. 43, (2003) 649–668
7. E. H. Callaway.: Wireless Sensor Networks - Architectures and Protocols, Auerbach publications, (2004)
8. IEEE Standard 802.11 for Wireless Medium Access Control and Physical Layer Specifications, Aug. (1999)