Energy Awareness Tree-Based Routing Protocol for Wireless Sensor Networks

IMAN ALMOMANI, Member IEEE, MAHA SAADEH,
HAMZEH AL JAWAWDEH, MOUSA AL-AKHRAS, Member IEEE
Computer Science Department, King Abdullah II School for Information Technology (KASIT)
The University of Jordan, Amman 11942, JORDAN

i.momani@ju.edu.jo, saadeh.maha@yahoo.com, hamzeh@jawawdeh.com, mousa.akhras@ju.edu.jo

Abstract: - Wireless Sensor Networks (WSNs) are resource-constrained networks that have been applied in several military and civilian applications. Many protocols have been proposed for the simple, low power, low cost WSNs. In this paper we propose a new routing protocol based on Tree Routing (TR). The proposed protocol exploits neighbors' links in addition to parent-child links to transmit messages while considering the residual energy at each node. Nodes should maintain neighbors' information in their neighbors table. The goal is to try to route the data over the shortest path and at the same time to prolong the network's lifetime by considering network power consumption and avoiding excessive messages between nodes. The proposed protocol is analyzed and compared with other tree-based routing protocols such as plus tree protocol.

Key-Words: - Wireless Sensor Network, WSN, Tree-Based Routing, Shortest Path, Power Consumption.

1 Introduction

Wireless Sensor Networks (WSNs) are infrastructure-based networks that consist of small sensors scattered in the sensing environment and one or more sink node(s). Sensors are used to sense environmental data and send it to the sink [1]. The sink node is used to process sensed data and connect the sensor network to the Internet. Sink node is usually a powerful device that is connected to a power supply.

WSNs are increasingly used in many applications such as environment monitoring, disaster relief, emergency rescue operation, military applications, biomedical and healthcare applications and others [1][2]. Some of these applications are considered sensitive in which the data should be private and confidential. Other applications need high reliability and bounded delivery time. To support these applications it is important to design and implement resource-efficient routing protocols that will transmit the data while considering nodes resource limitations and try to give best power saving, thus, it will prolong network life time and increase its availability.

Unlike traditional nodes, sensor devices have limited capabilities. As a result, sensor networks are subject to a set of resource constraints such as finite battery power, small amount of memory and limited processing capabilities. The design and implementation of WSNs protocols and operations should consider theses resource and energy limitations.

There are many factors affect network energy consumption such as nodes distance, number of sent and received messages between nodes, message size, number of intermediate nodes between source and destination, and the required level of local data processing. For routing protocols the challenge is to route the message using the suitable path and at the same time try to prolong network life time by avoiding excessive message exchange between nodes, thus, reducing the overall consumed power.

In this paper a new tree-based routing protocol is proposed. The goal of this protocol is to prolong the network lifetime by considering sensors power limitation and avoid excessive messages between nodes. A tree will be constructed between networks nodes and a new addressing scheme will be used during the tree construction to assign logical addresses to the network nodes. Each node should maintain neighbors' information such as neighbor's logical address, MAC address and power in its neighbors table.

The proposed protocol consists of many stages. First, construct a logical tree between network nodes, assign addresses to each one and build the neighbor tables. Second, exploit neighbor links to transmit the message considering intermediate nodes energy during transmission. Finally, solve the consequences of node failure and new nodes entrance.

The rest of the paper is organized as follows: section 2 summarizes some of the related works. Section 3 discusses the proposed routing protocol. The analysis and comparison are discussed in section 4. Finally, section 5 concludes the paper and presents possible future work.

2 Related Works

Many tree-based routing protocols have been proposed for WSNs. In this section we will review some of them.

One protocol proposed for WSNs is the Tree Routing (TR) which is supported by IEEE802.15.4 [3]. It is suited for small memory, low power and low complexity networks with lightweight nodes. This protocol aims to eliminate the overhead of path searching and updating, therefore, reduce extensive messages that are exchanged between network nodes. Two parameters are used by the TR protocol to control the tree construction. These parameters are the maximum number of children a node can have, and the maximum depth of the tree. As the number of children increases/decreases the depth will decreases/increases. An address scheme is used to assign logical network addresses to the network nodes.

Although this protocol works well, it suffers from two drawbacks. First, message transmission depends on source depth; the deeper the node the longer the path. Second, it suffers from node/link failure that causes nodes isolation.

To overcome the drawbacks of TR many protocols have been proposed to enhance TR performance. The authors in [4] proposed a Plus Tree (PT) routing protocol that utilizes the neighbors' links in order to find the shortest path to sink. To transmit the message, PT first constructs the parent-child links and then each node broadcast its ID to construct the neighbors' tables. Although plus tree finds the shortest path and solves link failure problem, it does not consider energy consumption in its solutions. In [5], the authors propose Enhance Tree Routing (ETR) protocol for zigbee networks. In this protocol each node should maintain information about its neighbors in a neighbors table. A structured address assignment scheme is used to assign addresses to tree nodes. Then the relations between nodes' addresses are exploited to find a shorter path to sink. Another protocol for zigbee networks is ImpTR that is proposed in [6]. However, both ETR and ImpTR do not consider network energy.

In this paper we propose a new tree-based routing protocol that will consider both number of intermediate nodes to the destination and residual energy at each node.

3 Proposed Routing Protocol

We propose a tree based routing protocol that aims to enhance the tree routing and tries to prolong network life time. The proposed protocol consists of different stages. First, construct a logical tree between network nodes. During the construction each node will get an address (ID) and construct the neighbors table. Second, exploit neighbor links to transmit the message and consider intermediate nodes energy and depth while transmitting. Finally, reconstruct the tree due to node failure and new nodes entrance. In the next subsections, the protocol stages will be discussed in details. Different control messages are defined for this protocol. These messages are listed in Table 1 with their descriptions.

3.1 Network Model

The network model is described as follows:

- The sensors are scattered in the network field without isolation.
- All sensor nodes have same capabilities, same transmission ranges and limited power resources.
- Symmetric model is assumed. That means of sensor A is located within the transmission range of sensor B, then B is also located within A's transmission range.
- All sensors sense data and transmit it to the sink for processing.
- The sink node assumed to have unconstrained resources.
- All sensor nodes are located in fixed places without mobility.

ISBN: 978-960-474-281-3

Table 1: Proposed Protocol Control Messages.

Message Type	When To Be Sent	Actions By Receivers	Message Structure
Ready Message	Is sent when: 1-The node gets an ID, so it's used to broadcast the ID and tell other nodes that it is ready to accept children. 2-The node receives a *NewNode* or *RequestParent* messages.	1- Store the information in the neighbors table. 2- If receiver node does not have ID, it will send *Engagement* message to the node with the maximum power.	1- Node ID 2- Node power
UnReady Message	It is used only to broadcast the ID.	Store ID in the neighbor table.	1- Node ID 2- Node power
Engagement Message	Is sent when receiving a *Ready* message and used to request for ID and request for Parent.	May send *EngagementAcceptance* message.	
EngagementAccep-tance Message	Is sent as a reply to an *Engagement* message.	1- Refresh neighbors table. 2- Calculate the ID. 3- Send *Ready* message	1- Node ID 2- Node power 3- Offered ID
NewNode Message	Is sent when new node want to join the network.	Either send *Ready* message or *UnReady* message.	
RequestParent Message	Is sent when a node can not reach its parent.	Send *Ready* message or *UnReady* message.	
Inform Message	Is sent to tell the neighbors that the node will go down.	Reconstruct the tree according to their relation with the dead node.	Node ID
ChangeID Message	Is sent when any node change its ID due to some failure to tell other nodes to modify the ID in their tables.	1- Modify the ID in neighbors table. 2- If the receiver is one of sender's children, it will update its own ID and send *ChangeID* message	1- Node ID 2- Node Power

3.2 Proposed Protocol Stages

The proposed protocol consists of four stages; logical tree construction, message transmission, new node entrance and node failure. This subsection will discuss these stages in details.

Tree Construction: A sink rooted tree should be constructed between network nodes before sending a message to the sink. A new addressing scheme is used in this stage to assign a logical ID for each node. Each node uses the ID to calculate its depth and its neighbors' depth. When any node receives *Engagement-Acceptance* message it will calculate its own ID = *parentID* || *offeredID*. Each *offredID* is represented by m digits where m is the digits required to represent C_{max} nodes. Where C_{max} is the maximum number of children. For example if $C_{max} < 10$ then we need only one digit to represent the offered ID 0…9, but for large networks if $C_{max} < 100$ then we need 2 digits 00…99, and so on. Using this addressing scheme each node will be able to know the depth for a particular ID.

To construct the tree we assume that all nodes do not have IDs and they have full energy. Parents can have at most C_{max} children, and at the end of this stage each node should have an ID from which it can calculate its depth.

The sink node will start the tree construction by broadcasting a *Ready* message to its neighbors, this message contains the sink ID (sink ID = initial ID), and sink energy. Each node receives this message will store the sink information in their neighbors table and after a short period send *Engagement* message to the sink. The *Engagement* message has two purposes: request for a parent, and request for an ID. For each *Engagement* message the sink node

ISBN: 978-960-474-281-3

will reply by sending *EngagementAcceptance* message only if its children are less than C_{max}.

When the node gets the *OfferedID* it will calculate its ID and broadcast a *Ready* message allowing its neighbors to send *Engagement* messages. Note that the *Engagement* messages are sent after a short period during which they can receive *Ready* messages from other nodes. This waiting will force the node to be associated with the best possible node among others (the node that has the maximum amount of power), thus, keep the balance between nodes. This process will continue until all nodes get IDs and no more *Ready* messages are sent.

New Node Engagement: When any node wants to join the network it should broadcast a *NewNode* message. Then all neighbors will reply by either *Ready* message if number of children < C_{max}, or *UnReady* message if number of children = C_{max}. The new node will store all information in its neighbors table and associate with the parent that has maximum power. Then it will broadcast *Ready* message.

Message Transmission: As we said the tree should be rooted at the sink node and all other nodes will send data to it. The message should be forwarded over the best path. To choose the next hop, the sender will consider both neighbors depth and power. The neighbor that has the minimum depth and a power larger than a specific threshold will be chosen. If all smaller depth neighbors have critical energy then the sender will send the data through the parent. In this way the load is balanced between nodes instead of overloading the parent node as in TR or the less depth neighbor node as in [4] [5] [6].

Tree Reconstruction: If node energy reaches specific threshold it should inform its parent, children, and neighbors that it will go down by broadcasting its ID in an *Inform* message, so that they can take an action and prepare themselves to reconstruct the tree.

Each node has a relation with the dead node should take an action. There are three different cases; the first one when the dead node is a parent. In this case the children have to find a new parent. Each child broadcast *RequestParent* message and only neighbors with children less than C_{max} will reply by a *Ready* message, other nodes send *UnReady* message. The child then chooses the node that has maximum power as a parent. Then it will broadcast *ChangeID* message to its neighbors to update the ID in their neighbors tables. If any neighbor is a child for this node it will change its ID and broadcast *ChangeID* message. This process continues until all IDs are modified.

The second case is when the dead node is a child. In this case parent should remove the node from its neighbors table and decrement the number of children. Finally, the last case is when the dead node is a neighbor then neighbors will remove it from their neighbors' tables.

In some cases the node went down before informing other nodes that it run out of power. In this case any neighbor node (could be child or parent) discovers this absence should broadcast the dead node ID in an *Inform* message and then each node will take an action as discussed above.

4 Analysis and Comparison

In this section we will compare our protocol with Plus-Tree (PT) Protocol [4] in terms of number of sent and received messages during tree construction and the consumed power due to messages exchange.

4.1 The Number of Sent and Received Control Messages

This subsection shows the maximum number of messages that is exchanged between network nodes during the tree construction in both PT and the proposed protocols.

Proposed protocol: Three messages are used during tree construction. The first one is *Ready* message where each node will send one *Ready* message to broadcast its ID. For N nodes the total is N *Ready* messages. Note that each node will receive at most Ne(ni) *Ready* messages from its neighbors, and that required $\sum^{N}_{i=1}Ne(n_i)$ receives. Where Ne(ni) is the neighbors for node i. The second one is *Engagement* message where in the worst case each node will send *Engagement* to every received *Ready* message. Each node expects to receive an *Engagement* from its neighbors. The maximum *Engagement* messages that will be received at each node = Ne(ni) (the total = $\sum^{N}_{i=1}Ne(n_i)$). Finally, the third one is the *EngagementAcceptance* where each node will send at most C_{max} *EngagementAcceptatnce* messages since it can have at most C_{max} children.

The total is N messages since each node can get only one *OfferedID*. On the other hand, each node will receive only one *EngagementAcceptance* message with the *OfferedID*.

PT Protocol: In this protocol the root (parent) sends an association message to its neighbors then each neighbor can attach by sending a reply. The parent checks if it can accept the child then respond by sending a message containing the logical ID. For N nodes the total is N association messages and $\sum_{i=1}^{N} Ne(n_i)$ association replies. Each node will have only one ID so there will be only N parents' ID responses. When the tree is constructed each node will broadcast its ID and collect its neighbors' ID to construct the neighbors table.

4.2 The Consumed Power

Sensor power is affected by local processing and communication operations. Since communication operations consumed more power than data processing, sensors will lose most of its power according to sending and receiving messages [7]. According to [8], the node requires ETx(k,d) to send k bits message to destination at distance d, and ERx(k) to receive k bits message.

$$ETx(k,d) = Eelec*k + Eamp*k*d^2 \quad (1)$$

$$ERx(k) = Eelec*k \quad (2)$$

Where Eelec = 50 nJ/bit, and Eamp = 100 pJ/bit/m2. By Eq.1 and Eq.2 we can calculate the maximum power that will be consumed during the tree construction. Table 2 illustrates the comparison between the proposed protocol and the PT protocol. As illustrated in the table, the proposed protocol requires less number of messages to construct the tree, consequently, less consumed power.

Table 2: Comparison Between The Proposed Protocol and PT Protocol.

	Plus Tree (PT)	Proposed Protocol
Sent Messages	$3N + \sum_{i=1}^{N} Ne(n_i)$	$2N + \sum_{i=1}^{N} Ne(n_i)$
Received Messages	$N + 3\sum_{i=1}^{N} Ne(n_i)$	$N + 2\sum_{i=1}^{N} Ne(n_i)$
Consumed Power	ETx(k,d) * sent + ERx(k) * received	ETx(k,d) * sent + ERx(k) * received

5 Conclusions and Future work

In this paper we propose a tree-based routing protocol for WSNs that considers both shortest path and energy balance between nodes. The proposed protocol consists of different stages; sink rooted tree construction, messages transmission, and node failure problem solving. The protocol is compared with other tree-based protocols such as Plus Tree routing protocol (PT). The results showed that the new protocol is more energy-efficient than the PT. As a future work we will implement this routing protocol to consider more sophisticated scenarios and compare it with other related protocols.

References:

[1] Al-Karaki and Kamal, "Routing Techniques In Wireless Sensor Networks: A Survey", Wireless Communications, vol.11, issue.6, pp.6 – 28, 2004.

[2] Biradar, Patil, Sawant, and Mudholkar, "Classification And Comparison Of Routing Protocols In Wireless Sensor Networks", Special Issue on Ubiquitous Computing Security Systems, UbiCC Journal, vol.4, pp.704 – 711, 2009.

[3] ZigBee Specification Version 1.0, ZigBee Alliance, 2005.

[4] Jung and Park, "Plus-Tree: A Routing Protocol For Wireless Sensor Networks", Lecture Notes In Artificial Intelligence, vol.4413 , pp.638 – 646, 2007.

[5] Qiu, Skafidas, and Hao, "Enhanced Tree Routing For Wireless Sensor Networks", Ad Hoc Networks, vol.7, issue 3, pp.638 – 650, 2009.

[6] Al-Harbawi, Rasid, and Noordin, "Improved Tree Routing (ImpTR) Protocol For ZigBee Network", International Journal of Computer Science and Network Security, vol.9, No.10 , pp.146 – 152, 2009.

[7] Zheng, and Jamalipour, "Wireless Sensor Network A Networking Perspective", A John Wiley & Sons, Inc. Publication, 2009.

[8] Heinzelman, Sinha, Wang, and Chandrakasan, "Energy-Scalable Algorithms And Protocols For Wireless Microsensor Networks". ICASSP, vol.6. Istanbul, Turkey, pp.3722-3725, 2000.