ViTAMin: A Virtual Backbone Tree Algorithm for Minimal Energy Consumption in Wireless Sensor Network Routing

Jaekwang Kim
Department of Electrical and Computer Engineering,
Sungkyunkwan University,
Suwon, Korea
linux@ece.skku.ac.kr

Jee-Hyong Lee
Department of Electrical and Computer Engineering,
Sungkyunkwan University,
Suwon, Korea
jhlee@ece.skku.ac.kr

Abstract—In wireless sensor networks (WSNs), routing algorithms are one of the important research topics because low energy consumption is strongly needed. In comparison with single hop and cluster routing scheme, virtual backbone tree schemes have recently attracted considerable attention in WSN routing algorithms as they offer energy-efficient transmission based on multi-hop routing approaches. The Energy-aware Virtual Backbone Tree (EVBT) is the pioneer of this concept which considers the energy-aware parameters in constructing virtual backbone tree. And the Multi-hop Cluster based stable Backbone Trees (MCBT) adds the clustering concept in the energy-aware virtual backbone tree. However there are still some energy wastes in their algorithms. In this paper, we propose ViTAMin, a Virtual backbone Tree Algorithm for Minimal energy consumption in WSN routing. ViTAMin provides more energy-efficient routing than do EVBT and MCBT, because of efficient selecting of upstream link. Though the simulation experiments, ViTAMin performed well with minimal energy consumption and a sufficient network lifetime in WSN routing.

Keywords-component: formatting; WSN, Virtual Backbone Tree, Energy Consumption, EVBT, MCBT.

I. INTRODUCTION

In wireless sensor network (WSN), low energy consumption is one of important topics because there are a huge amount of components which are distributed in the environment. Those components usually sense outside events, process those, and deliver the data to servers. Since the components, called sensor nodes, are independent and have limited power, how to efficiently operate sensor nodes is very important for the performance of WSNs.

Sensor nodes consist of four units: a sensing unit, a processing unit, a communication unit, and a power unit [1]. The main purpose of sensor nodes is to collect and send data to sink nodes which are connected to computing devices and external networks [2].

Although sensor networks are similar to ad-hoc networks, there are some differences [3]. First, sensor networks contain many sensors so they cannot be used with IP-based routing protocols. Second, sensor nodes usually use broadcasting communication instead of point-to-point communication. Third, sensor nodes have limited battery power, low computing power, and limited memory. Fourth, sensor networks need to allow for the cooperation of sensor nodes to prevent duplication of data transmission due to broadcasting.

There are many different routing algorithms in WSN. The main purpose of routing algorithms is energy-efficient transmission. Recently, two kinds of routing schemes have been proposed: the clustering scheme and the virtual backbone tree scheme. Generally, clustering schemes are based on single-hop routing approaches. These schemes lead to equal energy consumption among all nodes in a network. Each node directly identifies the sink nodes to which it will send sensed data, so the network model requires many sink nodes to gather the data sensed by the sensor nodes [4-5]. On the other hand, virtual backbone tree schemes are based on multi-hop routing approaches [6-9]. These schemes use a backbone that can efficiently transfer data from the sensor nodes to the sink node. Virtual backbone tree schemes are used in environments that differ from those of clustering schemes, i.e., the environment has fewer sink nodes and a relatively wide area, and thus requires multi-hop routing to gather the sensed data from the sensor nodes and send it to the sink nodes.

One of representative algorithms in virtual backbone tree schemes is EVBT [6-7]. This algorithm uses a tree structure to construct a virtual backbone so that the energy efficiency of data transfer is increased. It uses the characteristic distance, which guarantees the minimal energy consumption between two nodes in a virtual backbone [10]. That is, once a packet is relayed to one of tree nodes, the packet can be transferred efficiently to the sink node. The question is how to relay a packet from non-tree nodes to one of tree nodes. To address this issue, EVBT chooses the nearest tree nodes to the non-tree nodes to minimize the energy consumption from a non-tree node to the virtual tree, but this scheme does not minimize the energy consumption from non-tree nodes to the sink node [11].

Multi-hop Cluster Based stable Backbone Trees (MCBT) is one of the most up-to-date algorithm in WSN [9]. It is based
on the clustering scheme and backbone tree concept. MCBT assumes that the multiple sink nodes in the algorithm. It uses a distributed manner to create a stable backbone by selecting the nodes with higher energy as the cluster heads so that it can increase the network lifetime. This factor actually works well in consuming the battery of sensor nodes evenly. However, they still use the simple method to determine upstream links. This simple method cannot guarantee the minimized energy consumption in WSN.

In this paper, we propose a new algorithm for WSN routing, ViTAMin, the Virtual backbone Tree Algorithm for Minimal energy consumption in wireless sensor network routing. ViTAMin constructs a virtual backbone tree using the characteristic distance and connects non-tree nodes to tree nodes so that guarantees the minimal energy consumption to transmit data to the sink node.

We evaluated ViTAMin through simulations. We experimentally confirmed that the total energy consumption of our algorithm is less than those of the existing ones, even including the additional cost of constructing a backbone tree. In this paper, we also show that our algorithm increases the network lifetime. We also compared our algorithm with MCBT algorithm which is the one of the up-to-date algorithm in this domain. The result shows that ViTAMin is superior to MCBT algorithm in energy consumption.

The paper is organized as follows: In Section 2, we briefly summarize the existing routing protocols for wireless sensor networks. In Section 3, we present ViTAMin, in Section 4, we evaluate the system performance, and Section 5 concludes the paper.

II. BACKGROUND

We will briefly explain the basic concept of constructing a virtual backbone tree in this section. We followed the network model in [12-14] and energy model in [9][15].

Our approach is basically based on EVBT [6]. So, we summarize the virtual backbone tree construction method of EVBT. When the virtual backbone tree is built, the sink node is used as the root. All data sensed by the sensor nodes is transmitted to the root along the virtual backbone tree, which is constructed according to the following procedure:

1) The sink node broadcasts the Backbone tree Construction Request (BCR) packet to the nodes within its broadcast range.

2) The nodes that received the BCR packet calculate the Fitness Indicator, \( f_i \), and waits for another BCR packet during \( t_d \). The length of \( t_d \) is inversely proportion to the value of \( f_i \).

3) If no further BCR packet is received during \( t_d \), the nodes are incorporated into the virtual backbone tree. The parent of the new tree node is the node that sent the BCR packet. The new tree nodes then broadcast BCR packets to continue tree construction.

4) If another BCR packet is received within \( t_d \), those nodes are fixed as sensor nodes, non-tree nodes, and they will choose one of the virtual backbone tree nodes as their upstream link.

At the fourth step, the nodes fixed as non-tree sensor nodes may receive more than two packets. Those sensor nodes will choose one of the BCR-sending nodes as its upstream link to the backbone tree. In EVBT, the sensor nodes choose the nearest node as the upstream link.

The fitness indicator, \( f_i \), represents the suitability of each node as a virtual backbone tree node [6]. Three factors are considered to evaluate: the distance, the remaining energies of the nodes, and the straightness of the virtual backbone tree [6]. Since every node is assumed to be aware of its location, the senders of BCRs can include this location information in the packets; BCR packets therefore include the header, the node ID, and the location of the sender.

III. ViTAMIN

We propose ViTAMin, a routing algorithm, for WSN based on virtual backbone trees. It constructs a virtual backbone using the characteristic distance, which guarantees the minimal energy consumption between two nodes in the virtual backbone, as described in Section 2. After creating a virtual backbone tree, EVBT connects non-tree nodes to the virtual backbone by identifying the tree node that is the closest to the non-tree node to minimize energy consumption. However, this approach cannot minimize the total energy required to send data from a sensor node to the sink node.

![Figure 1. Selecting the energy-efficient upstream link.](image)

For example, Figure 1 shows the sink node \( S \), two tree nodes \( A \) and \( B \), and one non-tree node \( C \). Tree node \( B \) is connected to the sink node \( S \), via \( A \) and the energy consumptions from \( A \) to \( S \) is 1,000, from \( B \) to \( A \) is 100, from \( C \) to \( A \) is 80, and from \( C \) to \( B \) is 40. Around \( C \), there are two tree nodes \( A \) and \( B \), so \( C \) needs to be connected to either \( A \) or \( B \) in order to transmit data to \( S \). In this case, EVBT chooses \( B \) because the energy consumption to \( B \) is smaller than that to \( A \), so the total energy consumption to \( S \) will be 1,140. However, if \( C \) is directly connected to \( A \), it will be more energy efficient, with a total energy consumption of 1,080.

To solve this problem, we propose ViTAMin, which uses energy consumption information when constructing a virtual backbone tree and when choosing upstream links for non-tree nodes. There are two important characteristics in ViTAMin. First, this algorithm uses Extended BCR packets (EBCRs). A BCR packet includes the header and the location of the broadcasting node, but an EBCR packet also includes the energy consumption information from the broadcasting node to the sink node, along with its path in the backbone tree. For example, an EBCR broadcasted by node \( N \) includes the energy consumption, \( E_{N} \), to transmit a bit from node \( N \) to the sink node, that is, \( E_{BCR} = (BCR, E_{N}) \).

The second characteristic is the way in which non-tree
nodes choose their upstream links. In ViTAMin, the sensor nodes choose the upstream link that minimizes the total transmission energy to the sink node based on the information in the EBCRs. ViTAMin constructs a virtual backbone tree as follows:

1) The sink node, S, broadcasts EBCR packets, (BCR,0), to nodes within its broadcast range.

2) If a node receives an EBCR packet for the first time: Let us assume that node N receives an EBCR packet, (BCR, \( E_N^p \)), from node P. Node N evaluates the energy, \( E_N^p \), for transmitting a bit to node P from itself and the fitness indicator, and waits for another EBCR for time \( t_d \) which is inverse proportional to the fitness indicator.

3) If no more EBCRs are received during \( t_d \): Node N becomes a part of the virtual backbone tree and node P becomes the parent of node N. Node N broadcasts EBCR=(BCR, \( E_N \)) where \( E_N = E_p + E_N^p \).

4) If another EBCR is received within \( t_d \): Let us assume that node N received another EBCR=(BCR, \( E_Q \)), from node Q. Node N is fixed as a sensor node. Then, it evaluates \( E_Q \), the energy for transmitting a bit to node P from itself, and compares \( E_Q + E_N^p \) to \( E_Q + E_Q^p \). If \( E_Q + E_N^p \) is smaller, then node N chooses node P as the upstream link. Otherwise, node N chooses node Q.

Even though a node is fixed as a tree node or a non-tree node, it may receive more EBCR packets. If a tree node receives, it ignores EBCRs. If a non-tree node receives, however, it updates its upstream link. For example, let us assume that node N is a non-tree node and its current upstream link is node P. If node N receives one more EBCR, (BCR, \( E_R \)), from node R, it evaluates \( E_R \) and compares \( E_R + E_N^p \) with \( E_R + E_R^p \). If node R provides a more efficient path to the sink node, that is \( E_R + E_R^p \) is smaller than \( E_p + E_N^p \), node N will choose node R as the upstream link. With this step, any non-tree node can always have an energy efficient path to the sink node.

The details of ViTAMin are described with a simple example. We introduce four node states: white, grey, black, and blue. White nodes represent those that have not yet received any EBCR packets. Grey nodes are those that have received one EBCR packet, but the time delay has not yet expired. Black nodes are those that belong to the virtual backbone tree, that is, they are tree nodes. Blue nodes indicate those that do not belong to the tree and need to choose an upstream node to send their data to the sink node. Figure 2 shows the state diagram of a node.

![Figure 2. State diagram of a node.](image)

![Figure 3. An example of the construction of a virtual backbone tree using ViTAMin.](image)
ViTAMin constructs a backbone tree by first broadcasting an EBCR from the sink node. The energy consumption information in the EBCR packet sent by the sink node is 0, because no energy is necessary for transmitting one bit from the sink node to itself.

As shown in Figure 3(a), the sink node, S, broadcasts an EBCR packet, and the nodes that have received the EBCR packet (A, B and C) transition to the grey state. The three grey nodes are candidates for new virtual backbone tree nodes. The three grey nodes then calculate their fitness indicator values, \( \hat{f}_i \). The timer, \( t_{dt} \), in the node with the maximum indicator value will fire first, so the node will transition to the black state. Let us assume that B becomes black, the new black node then broadcasts another EBCR packet, as shown in Figure 3(b). Before broadcasting, the new black node calculates the energy required for transmitting data from itself to the sink node. Let us assume that the energy for transmitting one bit from B to S is 100. Node B broadcasts an EBCR (BCR, 100). Nodes A, C, and E are in the transmission range of B, so they receive the EBCR. Since A and C are grey and are waiting for another EBCR, they become blue. Node E is receiving an EBCR for the first time, so it becomes grey, as shown in Figure 3(c). Since there are no more grey nodes, it becomes black. It then calculates the energy for transmitting one bit from itself to the sink node. The energy information from B to S and the position of B are contained in the EBCR packet from B. Thus, the energy for transmitting one bit from E to B can be obtained. Let us say that this energy is 110, and the energy from B to S is 100. Thus, the energy from E to S is 210 = 110 + 100. That is, E will broadcast (BCR, 210) to the nodes in its transmission range (B, D, F and G), as shown in Figure 3(d). Node B is black, so it will ignore the EBCR. Nodes D, F, and G follow the same procedure described above. Finally, we assume that the backbone tree is constructed as shown in Figure 3(e).

The next step is for the blue nodes to choose the upstream nodes among the black nodes. This is done based on the EBCR packets received by blue nodes. For example, A receives two EBCR packets: one from S and the other from B. These two nodes are the candidates for the upstream link of A. The EBCR from S is (BCR, 0), and the EBCR from B is (BCR, 100). Node A evaluates the energies to B and S, respectively. Let us assume that the energy for transmitting from A to B is 20 and that from A to S is 95. Node A evaluates the total energy consumption for A to send a bit to S through B, it is 20 + 100: 20 from A to B and 100 from B to S. If B directly sends a bit to S, it consumes 95: 95 from B to S and 0 from S to S. Thus, B chooses S as its upstream link. Similarly, H chooses F because the energy from H to S via I is 90 + 410, and the energy from H to S directly via F is 100 + 305. The dotted lines in Figure 4(a) show the upstream links of the blue nodes.

In the case of EVBT, a blue node simply chooses the nearest one from the nodes having sent BCRs as its upstream link. Thus, the upstream link of A is B. Figure 4(b) shows the upstream links as determined by EVBT. EVBT minimizes the energy from a blue node to one of the backbone tree nodes, but fails to minimize the energy from a blue node to the sink node. For example, the energy from H to S is 500 by EVBT, but it is 405 by ViTAMin. The ViTAMin algorithm allows a reduction in transmission energy by determining the most efficient upstream links from the blue nodes.

IV. EVAUATION

In this section, we evaluate the performance of ViTAMin by comparing to EVBT and MCBT. EVBT is one of important algorithm among the virtual backbone tree algorithms and is reported to be more efficient than other schemes [6][7]. MCBT has been announced most recently [9].

A. Simulation methodology

We compare these three algorithms in three aspects: the tree construction efficiency, the energy efficiency and the network efficiency. To compare the tree construction efficiency, the virtual backbone tree construction costs of each algorithm are measured. The energy efficiency is measured in two aspects: the energy consumption at one event and the energy consumption until the first reconstruction. The energy consumption at one event is the energy consumed by all nodes until the sink node receives the packets as many as the number of the nodes. We compare the network efficiency within the transmitted data. We compare the network efficiency with two criteria. One is the transmitted data until the first tree reconstruction occurs. The other is the average transmitted data until the network is dead.

We use a simple reconstruction strategy for all the methods.
A backbone tree is reconstructed if the amount of data received by the sink node in a unit time interval is less than a threshold. We define that the network dies if no data packet from sensor nodes is received by the sink node after two successive backbone tree reconstructions.

We use the NS-2 simulator to evaluate ViTAMin with the following parameters by comparing in [6]: \( n=2, \alpha_1=\alpha_2=80nJ/\text{bit} \) and \( \alpha_2=100pJ/\text{bit}/m^2 \), \( d_{\text{thr}}=40m \), and \( c_1=c_2=c_3=1/3 \). That is, that the energy consumptions of \( tx \) and \( rx \) for one sensor are 240nJ and 80nJ, respectively, and every relay node consumes about 320nJ for the transmission of one bit. We also assume that the specifications of all nodes are the same. Each node has the initial energy 0.72J by the assumption of common sensor specification (0.2V \( \times \) 1mA h \( \times \) 1h = 0.72J). The data generation of each node is 50bytes/10sec. For tree construction, the header size of each algorithm is: MCBT Control packet = 15bytes, EBCR packet = 2bytes and BCR packet = 1byte. The sink node is placed at the center of the area with infinite energy, and other nodes are randomly deployed. Each sensor node sends 50bytes of data to the sink node at the sensing of every event, with an average duration between events of 10 sec. We also define the network as dead if the sink node does not receive data after two successive tree reconstructions. We fix the field size to 600m \( \times \) 600m, and perform the experiment by varying the number of nodes 1800, 3600, 7200, and 10800. We also fix the transmission range of each node to the characteristic distance, 40m. This is done because a virtual backbone tree performs with the highest energy efficiency if the transmission range is the same as the characteristic distance. We performed each experiment 100 times and averaged the results.

B. Tree construction efficiency

Only the backbone tree construction cost is compared in this analysis. EVBT and ViTAMin have a similar tree construction algorithm. ViTAMin uses EBCR packet which is one-byte bigger than BCR packet of EVBT. MCBT and ViTAMin have a different tree construction algorithm and the header packet of MCBT is much bigger than that of ViTAMin. We change the number of nodes as shown in Figure 5.

Figure 5 shows that when the number of node is 1800, EVBT consumes 761.40mJ for constructing the virtual backbone tree, MCBT consumes 1,040.94mJ and ViTAMin consumes 931.14mJ, respectively. EVBT consumes the smallest amount of energy. MCBT consumes the largest amount of energy for tree construction. As the number of node increases the gap of each algorithm also increases. The tree construction cost depends on control packet size strongly so that the MCBT consumes the highest tree construction energy. EVBT consumes the lowest tree construction energy. (Control packet size: MCBT (15bytes), ViTAMin (2bytes), and EVBT (1byte), respectively).

C. Energy efficiency

To compare the energy efficiency of each algorithm, we measure the average energy which consumes the sink node to receive the same number of packets as the number of nodes. For example, if there 1800 nodes we compare the energy for the sink node to receive 1800 packets. In this comparison, we do not include the energy for the tree construction, so we can compare the pure energy consumption for transmission. Figure 6(a) shows the comparison results. As Figure 6(a) shows, ViTAMin represents the best performance in the average energy consumption. When the number of node is 10800, the energy saving ratio between EVBT and ViTAMin is about 19.8% and the ratio between MCBT and ViTAMin is about 32.3%. This result shows that ViTAMin is superior to other algorithms in energy efficiency itself.

We also compared the average energy consumption before tree reconstruction. There are three or four data generations of each node until tree reconstruction in each algorithm. Figure 6(b) shows the result. As Figure 6(b) shows, ViTAMin also has the best performance in the average energy consumption. The energy consumption of all three algorithms increases when the number of node increases. By comparing with Figure 6(a) which shows the energy consumption at one event, the energy consumption in Figure 6(b) is not much large. That is because that there are some node deads between one event and two events. When the number of node is 1800, the saving ratio of ViTAMin is 12.7% over EVBT and 30.1% over MCBT. When the number of node is 10800, the saving ratio is 20.1% over EVBT and 29.3% over MCBT.

D. Network efficiency

We compare the network efficiency with two criteria. One is the transmitted data until the first reconstruction. The other is the transmitted data until the network dies. That is, we try to measure how efficiently a network operates while network lives. If an algorithm efficiently manages the network, then the life time is longer. Thus, the amount of data transmitted to the sink node increases. In this comparison, we include the virtual tree construction cost. Figure 7(a) shows the average transmitted data to the sink node before tree reconstruction. Figure 7(a) shows that when the number of node is 1800, the transmitted data before the first tree reconstruction is 94,868.28bytes (EVBT), 87037.02bytes (MCBT) and 104,253.39bytes (ViTAMin), respectively. Our algorithm, ViTAMin, turns out the best performance in all cases. When the number of node is 10800, the increase ratios of ViTAMin are 11.7% over EVBT and 21.3% over MCBT, respectively.

Also Figure 7(b) presents the transmitted data to the sink node before the network dies. This means network lifetime. As Figure 7(b) shows, when the number of node increases, the average transmitted data also increases. ViTAMin shows the
highest increase rate, EVBT is the second and MCBT is the last. When the number of node is 10800, the increase ratios of ViTAMin are 21.2% over EVBT and 68.5% over MCBT, respectively.

![Figure 6. Energy consumption (J).](image)

(a) at one event

(b) until first reconstruction

Figure 6. Energy consumption (J).

![Figure 7. Transmitted data (bytes).](image)

(a) before the first reconstruction

(b) until the network dies

Figure 7. Transmitted data (bytes).

V. CONCLUSION

In this paper, we propose ViTAMin, which is an energy efficient routing protocol. This algorithm adds energy consumption information to the BCR packet so that nodes that are not included in the virtual backbone tree can select the energy efficient upstream node. Through experimental results, we showed that the energy efficiency of ViTAMin is superior to those of the EVBT and the MCBT, even considering the additional tree constructing costs. We also show that ViTAMin shows better in the network efficiency. Through these improvements, we expect that this algorithm will be a contribution to future green computing.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0075109).

REFERENCES


