A New Data Aggregation Algorithm for Clustering Distributed Nodes in Sensor Networks

Suk-Jin Lee¹, Chun-Jai Lee¹, You-Ze Cho², and Sung-Un Kim³,*

¹ Pukyong National University, 599-1 Daeyeon 3-Dong Nam-Gu, Busan, 608-737, Korea
{stone, leecjc@pmail1.pknu.ac.kr}
² Kyungpook National University, 1370 San Kyok Dong, pook Gu, Daegu, 702-701, Korea
yzcho@ee.knu.ac.kr
³ Pukyong National University, 599-1 Daeyeon 3-Dong Nam-Gu, Busan, 608-737, Korea
kimsu@pknu.ac.kr

Abstract. The sensor nodes in sensor networks are limited in power, computational capacities, and memory. In order to fulfill these limitations an appropriate strategy is needed. Data aggregation is one of the power saving strategies in sensor networks, combining the data that comes from many sensor nodes into a set of the meaningful information. This paper proposes a new data aggregation algorithm named DAUCH (Data Aggregation algorithm Using DAG rooted at the Cluster Head) for clustering distributed nodes in sensor networks, combining the random cluster head election technique in LEACH with DAG in TORA. The proposed algorithm outperforms LEACH due to the less transmission power. Our simulation reveals that approximately a 4% improvement is accomplished comparing to the number of nodes alive with LEACH.

1 Introduction

Recent advances in MEMS-based sensor technologies, low-power analogy and digital electronics, and low-power RF designs have enabled the development of relatively inexpensive and low-power wireless micro sensors. These tiny sensor nodes can be used to a wide range of application areas such as health, military, home, and so on.

Realization of these sensor network applications requires wireless ad hoc networking techniques. But the sensor nodes in sensor networks are densely deployed, prone to failures, and are limited in power, computational capacities, and memory than the nodes in ad hoc networks. Moreover the topology of a sensor network changes very frequently, and sensor nodes mainly use a broadcast communication paradigm, whereas most of the ad hoc networks are based

* Corresponding Author
on point-to-point communications. And Sensor nodes may not have global identification (ID) as well, because of the large amount of overhead and the large number of sensors. Although many protocols and algorithms have been proposed for traditional wireless ad hoc networks, they are not well suited to the unique features and application requirements of sensor networks.

Many researchers are currently engaged in developing schemes that fulfill these requirements. Especially in a view of the network, the power consumption has been incurred to a critical issue, because the power efficiency is an important performance metric, directly influencing the network lifetime in sensor networks through.

Data aggregation is one of the power saving strategies in sensor networks, combining the data that comes from many sensor nodes into a set of meaningful information. In LEACH, the useful data is aggregated to the cluster heads that are randomly selected and allocate the time scheduling to their cluster members[1]. But LEACH needs clustering formation overheads before performing the task and the nodes which are away from a cluster head consume much more transmission batteries comparing to the nodes close to the cluster head. In order to eliminate redundancy power consume in LEACH, TORA (Temporally-Ordered Routing Algorithm) technique[2] which builds a DAG (Directed Acyclic Graph) rooted at the destinations in ad hoc networks can be used. With DAG each cluster head can construct an efficient cluster rooted at itself in sensor networks.

This paper proposes DAUCH data aggregation algorithm for clustering distributed nodes in sensor networks, combining the random cluster head election technique in LEACH with DAG in TORA. Using the efficient DAG, the proposed algorithm saves the radio energy dissipations of the transmitter and the receiver due to the short propagation distance. Our simulation reveals that approximately a 4 percentages improvement is accomplished comparing to the number of nodes alive with LEACH.

The rest of the paper is organized as follows: in section 2 we define the merits and demerits of data aggregation methods in sensor networks and in ad hoc networks, and show the properties of a new data aggregation method. In section 3 we define the new data aggregation algorithm, and Section 4 presents a model of the radio energy dissipation for the new data aggregation method and an experiment result showing an effect of the power consumption comparing with LEACH. We present our conclusion in section 5.

2 Preliminaries

2.1 Data Aggregation in Sensor Networks

Before starting the data aggregation techniques, we should investigate the routing models[3] that are assumed to consist of a single data sink attempting to gather information from a number of data sources. Figure 1 is a simple illustration of the difference between simple models of routing schemes that use data
Fig. 1. Illustration of AC routing vs. DC routing

aggregation (which we term Data-Centric (DC)), and schemes that do not (which we term Address-Centric (AC)). They differ in the manner that the data is sent from a source to a sink. In the AC routing, each source independently sends data along the shortest path to the sink based on the route that the queries took (end-to-end routing), whereas in the DC routing the sources send data to the sink, but routing nodes on the way look at the content of the data and perform some form of aggregation and consolidation functions on the data originating at multiple sources.

In ad hoc networks, a routing model follows the AC routing, so each source sends its information separately to the sink like the figure 1(a). In sensor networks, a routing model follows the DC routing, so the data from the two sources are aggregated at node A, and the combined data is sent from node A to the sink like the figure 1(b). Therefore in sensor networks, the data aggregation technique is a critical factor different from ad hoc networks to save the power consumptions of the nodes in order to extend the sensor network lifetime.

In sensor networks, the data aggregation tree can be thought of as the reverse of a multicast tree. So optimal data aggregation is a minimum Steiner tree on the network graph. Instead of an optimal data aggregation, sub-optimal data aggregations are proposed to generate data aggregation trees that are aimed to diminish the transmission power. The figure 2 summarizes the properties and disadvantages of sub-optimal data aggregation methods.

The prevenient data aggregation methods $^3$ are efficient to the model where a single point in the unit square is defined as the location of an ”event”, and all nodes within a distance $S$ (called the sensing range) of this event that are not sinks are considered to be data sources (which we term Event-Radius Model). In the model where some nodes that are not sinks are randomly selected to be sources, e.g. a temperature measurement and environment pollution detection (which we term Random-Source Model), it needs appropriate strategies for an efficient data aggregation.

In LEACH, all of the nodes in the field can be the source nodes in sensor networks, so this model can be considered Random-Source Model. The nodes in
LEACH organize themselves into local clusters, with one node acting as the cluster head, which allocates the time slot to its cluster members. All non-cluster head nodes directly transmit their data to the cluster head, while the cluster head node receives data from all the cluster members, performs signal processing functions on the data (e.g., data aggregation), and transmits data to the remote BS (Base Station). If the cluster heads were chosen a priori and fixed throughout the system lifetime, these nodes would quickly use up their limited energy because being a cluster head node is much more energy intensive than being a non-cluster head node. Thus LEACH incorporates randomized rotation of the high-energy cluster head position among the sensors to avoid draining the battery of any one sensor in the network. In this way, the energy load of being a cluster head is evenly distributed among the nodes. But LEACH needs clustering formation overheads before performing the task, and the nodes which are away from the cluster head consume much more transmission batteries comparing to the nodes close to the cluster head. So it needs a strategy to eliminate the redundancy power consume in LEACH.

2.2 Data Aggregation in Ad Hoc Networks

Most of the ad hoc networks are based on point-to-point communications, so the data aggregation in ad hoc networks is not considered a critical issue except the multipath routing. In some routing protocols such as DSR [4], AODV [5], LMR [6], TORA, and so on, multi-paths can be established from the sources to the destination. In that case the data aggregation can be performed through the overlapped paths en route. But it depends on each routing technique, which is implemented in ad hoc networks. Amongst the multipath routing techniques, TORA builds a directed acyclic graph rooted at the destination in ad hoc networks. So using DAG all data in the field can be assembled at the destination node.
3 DAUCH Data Aggregation Algorithm

3.1 Overview

The new data aggregation algorithm is illustrated by the figure 3. Each cluster head that is elected randomly creates the DAG rooted at the cluster head. The nodes that have more than one uplink node aggregate the data arrived from the uplink nodes then transmit them to the downlink node. This manner is continued until all data arrive at the cluster head. The cluster heads receive and aggregate the data from the adjacent neighboring node, and then transmit them to BS.

![Fig. 3. Illustration of the new data aggregation algorithm](image)

3.2 The Properties of DAUCH

In DAUCH, each cluster head that is elected randomly in the field creates a DAG centered at itself. Using DAG DAUCH creates the more effective cluster comparing to LEACH, and the nodes that are away from their cluster head save the data transmission power consumption using multi hop transmissions because of the shorter radio propagation distance. Moreover before the whole data get to the cluster head, some data aggregations are performed in the overlapped routes, so the task effort of data aggregations of the cluster head is distributed to other nodes that are not a cluster head.

But it is only effective supposed that all of the nodes are evenly distributed in the sensor field. Moreover there is a time delay in the case that the data of the nodes that are away from the cluster head arrive at the cluster head, and some overhead in the case that the links that connect the uplink with the downlink are unstable because of the dynamic movement of some nodes.

For the development of DAUCH, we made some assumptions about the sensor nodes and the underlying network model. For the sensor nodes, we assume
that all nodes can transmit with enough power to reach the BS if need, that each
node has the computational power to perform signal processing functions, and
that all nodes are synchronized by each other in a sensor field. These assumptions
are reasonable due to technological advances in radio hardware, low-power com-
puting, and time synchronization techniques. For the network, we use a model
where nodes always have data to send to the end user, nodes located close to
each other have correlated data, all sensor nodes are evenly distributed and quasi
static, and the message from the sender is correctly accepted by the receiver.
Although DAUCH is optimized for this situation, it will continue to work if it
were not true.

3.3 Requirements

The new algorithm uses five types of messages to communication.

DAG construction packet - When the cluster head creates DAG rooted at
itself, the cluster head generates the DAG construction packet and broadcast the
packet to the adjacent neighboring nodes. It contains the information of Clus-
terHeadID and DownlinkNodeID, where ClusterHeadID is the current cluster
head’s ID and DownlinkNodeID is the node that transmits the current DAG
construction packet, and each DAG construction packet contains a cluster ra-
dius and a height. The former indicates the node hop number that is the farthest
node of a cluster, and the latter express the hop number how far the current
node is apart from the cluster head, respectively.

DAG deconstruction packet - When each round is ended, the cluster head
generates a DAG deconstruction packet and sends it to all the cluster members
in order to inform them of the end of a current round and the beginning of a
new round. It contains ClusterHeadID and a cluster radius, and they inform the
current cluster head’s ID and the node hop number that is the farthest node of
a cluster, respectively.

ACK packet - When a link breakage occurs, an uplink node sets a downlink
flag and sends ACK packet to a downlink node whose height is the same as or
less than itself in order to reconstruct the broken link. The ACK packet contains
ClusterHeadID and TransmitterID. Herein TransmitterID presents the node ID
that sends the ACK packet.

NonACK packet - When a node receives more than one message from its
neighboring nodes and then collision occurs, it sends the packet to its neigh-
boring node to retransmit. The NonACK packet contains ClusterHeadID and
TransmitterID.
DATA packet - The nodes that have the data to send it to the cluster head use DATA packet to transmit the data. DATA packet is sent to the downlink node and on the way the packet is aggregated by the nodes that have more than one neighboring node. Finally all the data of a cluster are aggregated by the cluster head and sent to BS. The data packet contains SourceID, ClusterHeadID, TransmitterID, and Data. Herein SourceID presents the node that senses the event in a sensor field.

Each node caches the adjacent neighboring node IDs and stores their heights as well.

3.4 The Operation of DAUCH

The operation of DAUCH is divided into rounds and each round consists of five phases logically.

Cluster head selection phase - DAUCH’s cluster heads are stochastically selected like LEACH. In order to select cluster heads, each node $n$ determines a random number between 0 and 1. If the number is less than a threshold $T(n)$, the node becomes a cluster head for the current round. The threshold is set as follows:

$$
\begin{align*}
T(n) &= \frac{P}{1-P(r \mod \frac{1}{P})} \\
T(n) &= 0 \quad \text{[otherwise]} \\
\forall n \in G
\end{align*}
$$

with $P$ as the cluster head probability, $r$ as the number of the current round, and $G$ as the set of nodes that have not been cluster head in the last $1/P$ round. This algorithm ensures that every node becomes a cluster head exactly once within $1/P$ rounds.

DAG construction phase - After the cluster head selection phase, the cluster heads of each cluster broadcast the DAG construction packet to the neighboring nodes, as shown in figure 4(a). The nodes that receive the DAG construction packet set their heights that present the hop number how far the current node is apart from the cluster head, and then rebroadcast the packet to the neighboring nodes (figure 4(b)). If a node receives more than one DAG construction packet, it takes the message whose hop count is smaller than the others, and the messages that do not be taken are discarded. This manner is continued until the hop counter of the DAG construction packet reaches a cluster radius. After all, each cluster constructs DAG rooted at its cluster head.

DATA transfer phase - After a DAG construction is completed, the nodes that are farthest from the cluster head send the data to the neighboring downlink node, as shown in figure 5(a). After receiving the data, the node aggregates the data with own data, and then sends the aggregated data to the neighboring downlink node (figure 5(b), (c)). After all, every data within a cluster is assembled and aggregated at the cluster head, and then it is sent to BS, as shown in figure 5(d).
A New Data Aggregation Algorithm for Clustering Distributed Nodes

Fig. 4. The procedure of DAG construction

(a) The cluster head broadcasts the DAG construction packet to the neighboring nodes.
(b) The nodes that receive DAG construction packet set their height and then rebroadcast the packet.
(c) Using DAG construction packet, each node establishes the downlink rooted at the cluster head.
(d) Each cluster constructs DAG rooted at the cluster head.

Fig. 5. The procedure of DATA transfer

(a) Node A, B send DATA packet to the neighboring downlink.
(b) Node C aggregates DATA packet arrived from node A and B with its own data.
(c) Every node within a cluster is assembled at the cluster head.
(d) After aggregating the data, the cluster head sends the aggregated data to BS.
Downlink failure phase - When a node’s link that connects the neighboring two nodes is broken due to the node’s movement or complete battery consumption, the uplink node sends ACK packet to reconnect a downlink path with the node which height is the same as or less than a previous downlink node with setting downlink flag. If there doesn’t exist the node which height is the same as or less than a previous downlink node, or there isn’t any respond to the ACK packet within TTL, the node directly sends its data message to the cluster head.

DAG deconstruction phase - Before the end of each round, the current cluster heads generate DAG deconstruction packets and send them to all the cluster members in order to inform the cluster members of the end of a current round and the beginning of a new round. And the packet is delivered until the nodes located in the end of the cluster, i.e. cluster radius.

4 Analysis and Simulation for DAUCH

4.1 Radio Energy Dissipation Model for the New Algorithm

For our experiment, we used a 100-node network where nodes were randomly distributed between (x=0m, y=0m) and (x=200m, y=200m) with BS at location (x=200m, y=200m). We assume a simple model for the radio hardware energy dissipation where the transmitter dissipates energy to run the radio electronics ($E_{Tx-elec}$) and the power amplifier ($E_{Tx-amp}$), and the receiver dissipates energy to run the radio electronics ($E_{Rx-elec}$), as shown in figure 6. Therefore the radio energy dissipation of the transmitter is set as follows:

$$
\begin{align*}
E_{Tx}(k, d) &= E_{Tx-elec}(k) + E_{Tx-amp}(k, d) \\
E_{Tx}(k, d) &= E_{elec} \times k + \epsilon_{amp} \times k \times d^n
\end{align*}
$$

(2)

and the radio energy dissipation of the receiver is set as follows:

$$
\begin{align*}
E_{Rx}(k) &= E_{Rx-elec}(k) \\
E_{Rx}(k) &= E_{elec} \times k
\end{align*}
$$

(3)

Fig. 6. Radio energy dissipation model
where $d$ is a distance from a transmitter to a receiver, $k$ is k-bit messages, and $n$ is an exponential factor depending on the distance between the transmitter and the receiver. In the free space channel model, $n$ is set to 2, and in the multipath fading channel model, $n$ is set to 4 \[7\]. In our model, the distance between each cluster member and a cluster head is set to the free space channel model, and the distance between cluster heads and BS is set to the multipath fading channel model.

### 4.2 Analysis of the Radio Energy Dissipation Model

In order to analyze the different radio energy dissipation of LEACH and DAUCH, we apply the equation (2) and (3) to radio energy dissipation model for the data transmission and reception. Figure 7 presents the different data transfer model of LEACH and DAUCH. The inner and outer circles are implemented to easily compute the radio energy dissipation quantities based on the distance between each cluster member and a cluster head. We set the radius of the inner circle to $d$, and the radius of the outer circle to $2d$.

![Fig. 7. Comparison LEACH model with the new algorithm model](image)

In LEACH, the radio energy dissipations for the data transmission and reception can be computed using the figure 7(a). The energy dissipation of the inner nodes is set as follows:

$$8[E_{elec} \times k + \epsilon_{amp} \times k \times d^2 + E_{elec} \times k]$$

(4)

the energy dissipation of the outer nodes is set as follows:

$$12[E_{elec} \times k + \epsilon_{amp} \times k \times (2d)^2 + E_{elec} \times k]$$

(5)

and the total energy dissipation of the nodes is set as follows:

$$20[E_{elec} \times k] + 56[\epsilon_{amp} \times k \times d^2] + 20[E_{elec} \times k]$$

(6)

In the new algorithm, the radio energy dissipations for the data transmission and reception can be computed using the figure 7(b). The energy dissipation of
the inner nodes is set as follows:

\[ 8[E_{elec} \times k + \epsilon_{amp} \times k \times d^2 + E_{elec} \times k] \] ,

(7)

the energy dissipation of the outer nodes is set as follows:

\[ 12[E_{elec} \times k + \epsilon_{amp} \times k \times d^2 + E_{elec} \times k] \] ,

(8)

and the total energy dissipation of the nodes is set as follows:

\[ 20[E_{elec} \times k] + 20[\epsilon_{amp} \times k \times d^2] + 20[E_{elec} \times k] \] (9)

assuming that the total power consumption of the data aggregation is much smaller than the energy dissipation of the data transmission and reception, with comparing equation (6) with equation (9), we conclude that the new algorithm is superior to LEACH with regard to the radio energy dissipations for the data transmission and reception.

4.3 Simulation Results

For the presented simulations we use C++ programming. Each node is equipped with an energy source whose total amount of energy accounts for 2 J(Joule) at the beginning of the simulation. Every node transmits a 500 bytes message. The cluster head probability \( P \) is set to 0.05 [1].

Figure 8 and 9 illustrate simulation results of our sample network. According to our simulation results, the proposed method improves approximately 4% comparing to the number of nodes alive with LEACH. This is due that the new data aggregation uses smaller energy dissipation of the data transmission and reception than LEACH. Therefore the total amount of the new data aggregation algorithm’s data received at the BS over time is better than that of LEACH due to the extended sensor network life time.

5 Conclusions and Future Work

This paper proposes a new data aggregation algorithm for clustering distributed nodes in sensor networks, combining the random cluster head election technique in LEACH with DAG which constructs an efficient cluster in TORA. The proposed algorithm outperforms LEACH by diminishing the radio energy dissipations for the data transmission and reception, and extending sensor network lifetime in comparison with LEACH. Even though our simulation result shows a good performance, the scheme is based on the assumption that the cluster head probability should be optimized like as to the case of LEACH. If we find a suitable cluster head number for the proposed algorithm, we can expect a more efficient outperforming result. This is our future work to make efforts to find an optimal cluster head probability.

In conclusion, this algorithm can be used in periodic data gathering application such as a temperature measurement and environmental pollution detection due to an efficient data gathering and low power consumption.
A New Data Aggregation Algorithm for Clustering Distributed Nodes

Figure 8. Number of nodes alive over time

Figure 9. Number of data signals received at the base station over time
Acknowledgment. This work was supported by Korea Research Foundation Grant (KRF-2003-002-D00243).

References


