무선센서네트워크에서 데이터 병합 트리를 위한 자기치유 방법

Duc Tai Le, Thang Le Duc, 염상길, Vyacheslav V. Zalyubovskiy, 추현승
성균관대학교 정보통신대학
{ldtai, ldthang, sanggil12, slava, choo}@skku.edu

Self-healing Method for Data Aggregation Tree in Wireless Sensor Networks

Duc Tai Le, Thang Le Duc, Sanggil Yeom, Vyacheslav V. Zalyubovskiy, and Hyunseung Choo
College of Information and Communication Engineering, Sungkyunkwan University

Abstract

Data aggregation is a fundamental problem in wireless sensor networks that has attracted great attention in recent years. On constructing a robust algorithm for minimizing data aggregation delay in wireless sensor networks, we consider limited transmission range sensors and approximate the minimum-delay data aggregation tree which can only be built in networks of unlimited transmission range sensors. The paper proposes an adaptive method that can be applied to maintain the network structure in case of a sensor node fails. The data aggregation tree built by the proposed scheme is therefore self-healing and robust. Intensive simulations are carried out and the results show that the scheme could adapt well to network topology changes compared with other approaches.

1. Introduction

An important application of Wireless Sensor Networks (WSNs) is data gathering, i.e., sensor nodes transmit data, possibly in a multi-hop fashion, to a Base Station (BS). Data aggregation is defined as the process of aggregating data from multiple sensors using data fusion and transmitting the aggregated data to the BS. Much existing research on data aggregation has focused on saving sensor energy by dividing the network into clusters. Only the Cluster Head (CH) of each cluster transmits data to the BS. Network energy is saved by reducing the number of nodes involved in long distance transmission, but data aggregation delay is often sacrificed in return. Cheng et al. [1] introduced a delay-efficient data aggregation tree and prove that the structure provides the minimum data aggregation delay when only primary interference is considered and sensors are required to be able to adjust the transmission range freely. This assumption may not be applied in many scenarios, especially in a widely deployed network and in sensors that have limited transmission range. Le et al. [2] considered a more practical scenario where sensors have only limited transmission range. They approximated the data aggregation delay and improved the network lifetime in compared with [1].

After an aggregation tree is built, sensors then periodically sense their nearby environment and send the information to the BS following the data gathering paths on the tree. After a period of time, some sensor nodes may deplete their batteries and removed from the network. The network topology is changed and the existing aggregation tree does not work properly any more. In case the change is not significant, it is not efficient to rebuild a whole new aggregation tree. We propose a method to dynamically modify the aggregation tree to adapt to those changes, which makes the aggregation scheme self-healing.

The remainder of this paper is organized as follows. In Section 2 includes the network model. The proposed method is presented in Section 3. Section 4 discusses the performance evaluation. Section 5 conclude the paper.

2. Preliminaries

2.1 System Model and Assumptions

The same system model in [2] is used in this paper. The network consists of N wireless sensors which are randomly deployed in the network area and one BS. Locations of sensor nodes and the BS are not changed after deployment. The transmission area of each sensor is roughly a disk centered at the sensor. We assume that all nodes have the same transmission range. The network topology is presented as a graph $G=(V,E)$, where V is the set of nodes and E is the set of edges in the network. An edge$(u,v) \in E$ if and only if u and v are in the transmission area of each other. Energy consumption of sensors is assumed to be proportional to the square of its transmission range.

Each sensor is equipped with only one transceiver, and multiple receptions by a node in the same time slot is not allowed. A node receives a data packet correctly when it hears only this packet at that moment. If a node hears two or more packets at the same time, it cannot receive any of them due to the collision. Data are generated by sensor nodes and passed through aggregation paths to be gathered at the BS. Perfect data fusion is assumed to be used, meaning that all sending data packets are of the same size. Data transmission from a node to its parent is assumed to be completed in one time slot.

2.2 Delay-Efficient Network Structure for Data Aggregation

The delay-efficient structure in [1] consists of one or multiple sub-trees, each of them has a different number of nodes which is a power of two (2^p). The sub-tree root is the CH and other nodes are Cluster Members (CM). The CH forms a data link with the BS directly. The rank of a node is the number of data links attached to it. A node of rank k has $k-1$ child nodes, while these child nodes have different ranks varied from 1 to $k-1$, as shown in Fig. 1.

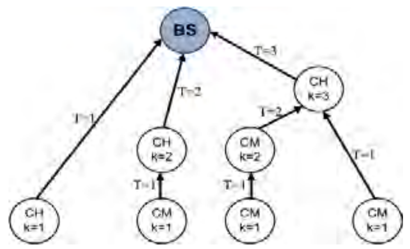

Fig. 1. An example of a delay-efficient data aggregation tree consisting of three sub-trees.

3. Adaptive Method for Data Aggregation Tree

A node failure will cause a critical routing problem in which the descendants of the failed node cannot send their data until the data aggregation tree is reconstructed. In order to solve this problem, failures must be handled as soon as possible. We propose an adaptive method (ADA) for data aggregation tree in [2] to handle the problem. The approach only update the nodes in the local area around the failed nodes, thus only a part of the aggregation tree is affected by the event. While modifying the tree, our goal is to keep good properties of the delay-efficient network structure and limit the number of updates as small as possible.

3.1 Node Available Time Slots

Considering limited transmission range sensors, the aggregation tree T built in [2] can only approximate the structure in [1], thus some of the reserved time slots of nodes in T are probably not used. The set of unused time slots of node u is defined as $\mathrm{AvlTS}(u)$, which is also called the set of available time slots of u. Node u can accept a new child node v where $\mathrm{rank}(v) \in \mathrm{AvlTS}(u)$. Node v then can transmit data to u in the time slot equal to $\mathrm{rank}(v)$ without increasing the aggregation delay. Fig. 2 is an example to illustrate available time slots of sensor nodes.

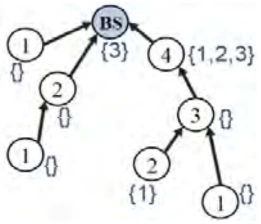

Fig. 2. An example of available time slots in a tree.

3.2 Aggregation Tree Maintenance

When node x is detected as failed, all children of x become orphans and look for a new parent. On another hand, the parent of x removes x from children list and updates its available time slots. In order to find a new parent, an orphan node u first determines all available time slots of its neighbors. There are three cases need to be considered as follows:

- Case 1: There exits one available time slot of u's neighbor greater than or equal to $\mathrm{rank}(u)$. Node u can use that time slot for transmitting data without increasing the data aggregation delay. In case node u can find more than one available time slots, it selects the smallest one because the big one should be reserved for later use.
- Case 2: Neighbors of u have available time slots, but none of them is greater than or equal to $\mathrm{rank}(u)$. In this case, u selects the maximum one for transmitting data. It detach all children having ranks greater than or equal to the selected available time slot. The detached children of u become orphans and have to find their new parents.
- Case 3: No neighbors of u have available time slots. This is the worst case because u has almost no choices but to select one of its neighbors v as its parent. To reduce the number of updates, u selects the neighbor closest to the BS as its parent and set $\mathrm{rank}(u) = \min\{z | z \neq \mathrm{rank}(w), \forall w \in \mathrm{children}(v)\}$. Node u also needs to detach all children having ranks greater than or equal to $\mathrm{rank}(u)$.

4. Performance Evaluation

The network parameters used in our simulations are the same as [2]. Each plotted data point is the average result of 100 runs. We vary the percentage of nodes failure from the initial network, and compare the performance of ADA with that of DEDA [2]. As DEDA is considered as the ideal values for data aggregation delay, the more ADA results get close to that of DEDA, the more it is effective. In Figs. 3, the aggregation delay of the adaptive method ADA is at most 6% higher than the ideal values produced by DEDA.

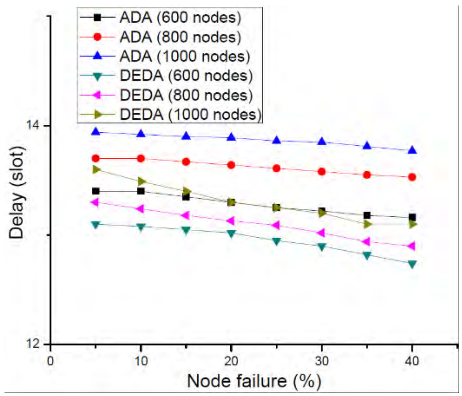

Fig. 3. ADA vs DEDA (re-construct a new tree).

5. Conclusion

This paper presents an adaptive method for a delay-efficient data aggregation tree to handle network topology changes due to node failure. Intensive simulations are carried out to evaluate the performance of our proposed scheme in comparison with recent related works. Simulation results show that the adaptive method is well adapted to network topology changes. The network built by the scheme is therefore robust and self-healing.

ACKNOWLEDGMENT

This research was supported in part by PRCP (NRF-2010-0020210), ICT R&D program (14-911-05-006), ICT R&D program (10041244), BSRP (NRF-2013R1A1A2064302), respectively.

References

[1] C. Cheng, C.K. Tse, and M. Lau, "A Delay-Aware Data Collection Network Structure for Wireless Sensor Networks," IEEE Sensors J., vol. 11, no. 3, pp. 699-710, Mar. 2011.
[2] H. N. Le, V. Zalyubovskiy, and H. Choo, "Delay-minimized Energy-efficient Data Aggregation in Wireless Sensor Networks," Proc. IEEE CyberC, pp. 401-407, Dec. 2012.