에이전트 시스템 개발도구에 관한 연구

Dung T. Nguyen, 염상길, 추현승
성균관대학교 소프트웨어대학
e-mail : {ntdung, sanggil12, choo}@skku.edu

A new approach for minimum aggregation time scheduling in wireless sensor networks

Dung T. Nguyen*, Sanggil Yeoum*, Hyunseung Choo*
* College of Software, Sungkyunkwan University

요 약

Collecting data in wireless sensor networks in minimum time is a traditional problem which is known NP-hard. Previous studies built the schedule using the node-based or link-based metrics to prioritize the transmissions. In this work, we combine the effect of both metrics to obtain a smaller aggregation time. We compare our work with state of the art schemes and report the improvement.

1. Introduction

In this work we are addressing the Minimum Latency Aggregation Scheduling (MLAS) problem in Wireless Sensor Networks (WSNs) [2]. The goal of the MLAS problem is to find a transmission schedule for all nodes, in which the time required to collect data from the whole network are minimized. To date, the research on MLAS has gone through decades, however, improvements seems still limited.

Existing solutions follow two mainstream approaches: (1) building the aggregation tree first, and then scheduling the transmissions along the tree structure (sequential approach); (2) structure-free scheduling in which the schedule is built without a prebuilt tree (simultaneous approach) [2]. In the former approach, the scheduling order mimics the flow of data collection: leaf nodes are the first nodes to transmit their data, so they will be determined first. Then the algorithm moves toward internal nodes until it reaches the sink.

All the existing solutions are greedy algorithms. The prioritization metrics are either node degree, hop-distance to the sink, number of children, the number of link conflicts or some combination of them. Some prior work in the sequential approach are [4][5]. Regarding the simultaneous approach, Tian et al. proposed a top-down tree construction algorithm. They also argue that scheduling based on conventional shortest path tree or connected dominating set cannot guarantee a close to optimal solution.

In our work, we propose a scheduling algorithm that takes into account link conflict, and the node metric introduced in our previous work [1] to assign transmission schedule. We will compare our results with to some state-of-the-art aggregation scheduling solutions in both two approaches: sequential and simultaneous. The organization of the paper is as follows. Section 2 recaps preliminaries related to the problem. Section III briefly presents our proposed idea. Section IV illustrate our experiment settings and interim results.

2. Preliminaries

A. The Network Model

We consider a WSN modeled as a graph G(V; E) where V is the set of nodes including the sink, and E is the set of edges. Two nodes in V has an edge in E if their Euclidian distance is smaller than the communication range. We assume that all the nodes have data to send to the sink, and because of the multi-hop nature of WSNs, the nodes that are far away from the sink must transmit their data through a multi-hop path. The intermediate nodes will merge all the received data with its own to produce a single out going packet. We call this aggregation.

We assume time is slotted and each node will get a time slot assignment to transmit its data. In a data aggregation round, each node transmits only once. We consider the collision model as indicated in [3]. If a node $w \in V$ is in communication range of two other nodes $u \in V$ and $v \in V$, and u and v transmit their data (not necessarily intend to transmit to w) at the same time slot, then w cannot receive either of the packets.

B. Problem Statement

We refer to an aggregation schedule as the assignment of transmission time slots to all the nodes in the network. A valid schedule must avoid any collisions between the transmissions. The total number of time slots needed is called aggregation time. The aim of this work is to construct a schedule that minimize the aggregation time.

3. Proposed scheme

A. The Minimum Aggregation Time

We use the metric calculated on each node called Minimum Aggregation Time (MAT). Given a tree, MAT of a node is the minimum time it should wait to collect data from its subtree considering primary collisions only (primary collision happens between transmissions between the children of a common parent).

Detailed computation of the MAT can be found in our previous work []. In brief, the MAT metric must satisfy the following conditions:

- Leaf nodes have MAT = 0 since they don't need to collect data from any children
- MAT of a parent always greater than MAT of all its children. Let p be a parent and $c_1, c_2, \dots, c_k$ be the children of node p, then MAT of node p, denoted as $mat(p)$, is calculated as follows:

$$mat(p) = \max\{mat(c_i) + 1 + k - i \mid 1 \le i \le k\}$$

Figure 1 shows an example of MAT calculation for a tree rooted at the sink s. Initially, all the nodes in the tree is set MAT = 0. Starting from the leaf nodes (Figure 1a), the three leaves' MATs are 0. In the next step, the parents of the MAT-calculated nodes in the previous steps will be examined. In Figure 1b, nodes a and d gets its MATs: $mat(a) = 1$ and $mat(d) = 2$, because node a only has one leaf child, while node d has two. Similarly, the algorithm goes up toward the sink, and we can get $mat(b) = 3$ and $mat(s) = 4$.

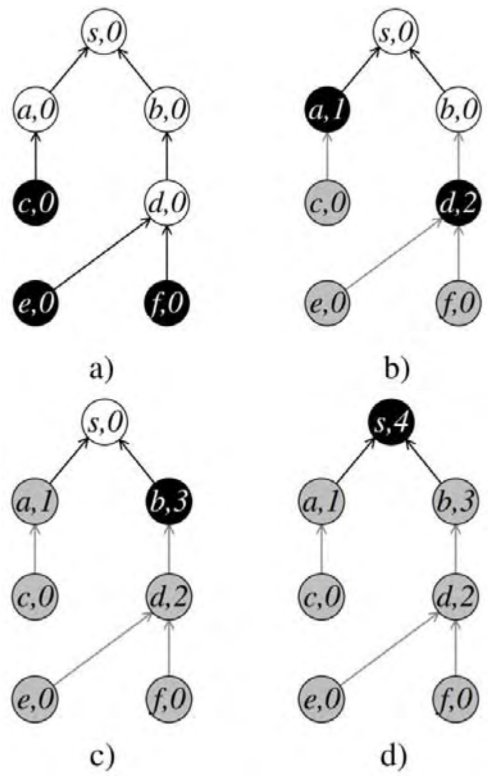

Figure 1. Step by step calculation of node metric

We can see that MAT somehow reflects the aggregation time toward the node. Therefore, a node with higher MAT value should be treated with higher priority during a scheduling process. We first construct a simple shortest path tree (SPT) given the network, and calculate MAT for all nodes based on the SPT. Those MATs will be used in the scheduling process. Note that constructing an SPT is just for the MAT calculation. In the scheduling process, the SPT is completely removed, and the scheduling process starts with the original network topology.

B. Scheduling algorithm

We propose a top down aggregation tree construction as in Algorithm 1. Initially, the scheduled set consists of the sink node only (line 1), and the non-scheduled set consists of all the remaining nodes. The algorithm goes iteratively by time slot, starting from time slot $t = 0$ (line 3), and t increases until all the nodes are scheduled. In each iteration, corresponding to a specific time slot, the algorithm identifies the set C of links between S and NS. The C is basically the link candidates that can be scheduled at time slot t. Then among C, we find the maximum number of links that can be scheduled collision-freely (line 8).

Algorithm 1 Scheduling algorithm

Input: $G = (V, E)$ and MAT of all nodes in V

Output: A time slot assignment for all nodes

1: $S \leftarrow \{sink\}$
2: $NS \leftarrow V \backslash \{sink\}$
3: $t \leftarrow 0$
4: **while** $S \neq V$ **do**
5: $\quad t \leftarrow t + 1$
6: $\quad C \leftarrow \{(u, v) | u \in S, v \in NS\}$
7: $\quad CP \leftarrow \{u \in V | \exists v \in V : (u, v) \in C\}$
8: $\quad$ MAT-based matching$(CP, C \backslash CF)$
9: $\quad$ **for** $v \in C$ **do**
10: $\qquad v.timeslot = t$
11: $\quad$ **end for**
12: **end while**

The function named MAT-based matching (Algorithm 2) greedily selects the link one by one based on the degree of conflict it causes to other links in the set C. If there are several nodes with the same degree, we break the tie by their MAT values.

Algorithm 2 MAT-based matching

Input: $G = (V, E)$ and MAT of all nodes in V

Output: A time slot assignment for all nodes

1: **while** $C \neq \emptyset$ **do**
2: $\quad$ compute conflict degree for the links in C
3: $\quad d \leftarrow \min\{f(u, v) | (u, v) \in C\}$
4: $\quad D \leftarrow \{(u, v) \in C | f(u, v) = d\}$
5: $\quad (u_0, v_0) \leftarrow \arg\max_{MAT(v)}\{(u, v) \in D\}$
6: $\quad$ Remove links that conflicts with (u_0, v_0) from C
7: **end while**

4. Experimental results

We select well known aggregation scheduling schemes: Minimum Lower bound Spanning Tree (MLST) for tree construction and Neighbor Degree Ranking (NDR) for link scheduling following the tree structure. To build an aggregation tree, the MLST algorithm sum up the number of children and hop distance to the sink and use it as a metric to minimize during the tree construction process. NDR basically relies on the number of neighbors in the two hops distance to determine the transmission priority. For the simultaneous scheduling approach, we select the Greedy Growing Tree (GGT). The simulation settings are presented in Table 1.

The network consisting of static sensor nodes deployed in a 2-dimensional square area with the size of L×L in which the sink node is at the center of the area. The communication range of every node is equal and normalized to 1. The network density, which is the average number of neighbors within a disk area of radius 1, can be computed as follows:

$$D = \frac{n\pi}{L^2}$$

We will vary the network density in range {5 – 95} and network side length in {2, 4, 7}.

Table 1. Simulation settings

Parameter	Value
Network density	15-95
Network side length	2, 4, 7
Sink position	Center
Number of experiments	30

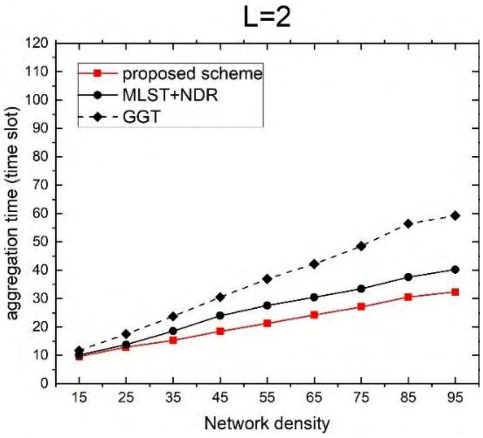

Figure 2. Aggregation time when L=2.

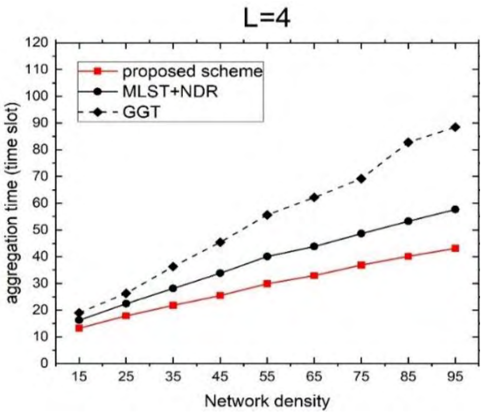

Figure 3. Aggregation time when L=4.

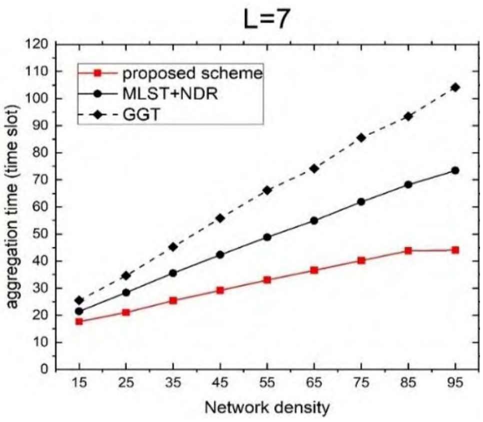

Figure 4. Aggregation time when L=7.

Figure 2, 3 and 4 show that our algorithm outperforms the MLST+NDR and the GGT schemes. When the network density increases, the aggregation time increases because the there are more sensor nodes, and the level of collision also increases. Proposed scheme provides about 26% smaller delay than the MLST+NDR, and about 50% better than the GGT.

5. Conclusion

In this paper we presented a simulation results of the proposed data aggregation scheduling algorithm. In the future we plan to investigate and implement more existing algorithms for comparison. We will also extend the simulation scenario to see the effect over different settings.

ACKNOWLEDGMENT

본 논문은 기초연구사업 (NRF-2010-0020210)과 과학기술정보통신부 및 정보통신기획평가원의 Grand ICT 연구센터지원사업 (IITP-2019-2015-0-00742)의 연구 결과로 수행되었음.

References

[1] Nguyen, Dung T., Sanggil Yeoum, and Hyunseung Choo. "A top-down aggregation scheduling approach for delay-constrained data aggregation in wireless sensor networks." 한국정보과학회 학술발표논문집 (2018): 1091-1093.

[2] Tian, Chen, et al. "Neither shortest path nor dominating set: aggregation scheduling by greedy growing tree in multihop wireless sensor networks." IEEE Transactions on Vehicular Technology 60.7 (2011): 3462-3472.

[3] Kang, Byungseok, et al. "A distributed delay-efficient data aggregation scheduling for duty-cycled WSNs." IEEE Sensors Journal 17.11 (2017): 3422-3437.

[4] Pan, Cheng, and Hesheng Zhang. "A time efficient aggregation convergecast scheduling algorithm for wireless sensor networks." Wireless Networks 22.7 (2016): 2469-2483.

[5] Malhotra, Baljeet, Ioanis Nikolaidis, and Mario A. Nascimento. "Aggregation convergecast scheduling in wireless sensor networks." Wireless Networks 17.2 (2011): 319-335.