Delay-Efficient Data Collection with Dynamic Traffic Patterns in Wireless Sensor Networks

Phuc Nguyen Van, Vyacheslav Zalyubovskiy, and Hyunseung Choo
College of Information and Communication Engineering
Sungkyunkwan University
Suwon, South Korea
nvphuc@skku.edu, slava@ece.skku.ac.kr, choo@skku.edu

Abstract—Data collection is one of the most important applications in Wireless Sensor Networks (WSNs), where the data are gathered from sensor nodes to the base station. To reduce energy consumption, the sensor node may not report every sensed data sample to the base station. Thus, the network traffic of continuous data collection application often varies unpredictably over different sampling intervals.

In this paper, we propose an energy-efficient scheme, Delay-Efficient Traffic Adaptive (DETA), for collecting data from sensor nodes with minimum delay according to the traffic load. The DETA scheme minimizes data collection delay by constructing a delay-efficient, collision-free schedule, and by using an adaptive mechanism to enable every node to self-adapt to the change of traffic. The simulation results show that our proposed solution could significantly decrease data collection delay and obtain reasonable values of energy consumption compared with other schemes.

Keywords: scheduling; data collection; dynamic traffic, wireless sensor networks.

1. Introduction

Data collection from sensor nodes in a network over a tree based structure is a fundamental problem in WSNs. In many applications of WSNs, such as military surveillance [1], habitat monitoring [2], or structural maintenance [3], data collection is a key function in which the base station collects all data generated by sensor nodes in the network. Because sensor nodes are often powered by batteries that may not be recharged, reducing energy consumption has attracted great attention in recent years. Moreover, in many applications, it is crucial to guarantee the data collection time as quickly as possible. For instance, when the sensor nodes are used to detect gas, oil, or structural damage and so on, sensing data must be gathered as soon as possible. As a result, energy-efficiency and delay-efficiency are always important targets in WSNs.

The TDMA scheduling method is used quite commonly in WSNs. In a TDMA schedule, the collisions are eliminated by scheduling only non-interfering transmissions to proceed in the same transmission slot. There are many research [4]−[8] which provide the TDMA schedules for data collection in WSNs. In contrast to many realistic applications, most of these methods are designed for the static network traffic pattern where every sensor node has at least one data packet to send to the base station in a sampling interval. However, to reduce energy consumption, a sensor node may not report its data to the base station in every sampling interval. This causes the traffic network of data collection to vary unpredictably over different sampling intervals.

To the best of our knowledge, Wenbo Zhao et al. [9] are the first researchers to consider sensor data collection with dynamic traffic patterns. In this approach, the authors propose a TDMA scheduling algorithm which effectively deals with the change of network traffic. However, this scheme cannot guarantee good data collection delay because the parent node waits to receive data from all its children before sending its own data to its parent. In this paper, we propose a Delay-Efficient Traffic Adaptive (DETA) scheme to solve the data collection problem with dynamic traffic pattern in WSNs. The main contributions of this paper are summarized:

• Proposing an algorithm for scheduling sensor nodes to report data with minimum delay.
• Providing an adaptive mechanism to allow the sensor nodes to reduce their idle listening according to the change of network traffic.

The simulation results show that our proposed scheme can achieve up to 20% improvement in terms of data collection delay and keep energy consumption at reasonable values, compared with the existing scheme.

The remainder of this paper is organized as follows. In Section 2, we briefly review the related work. Section 3 defines our system model and assumptions. The proposed scheme is presented in Section 4. The performance evaluation is shown in Section 5. Finally, we conclude the paper in the last section.

2. Related Work and Motivation

Many research on sensor data collection in WSNs have been investigated in recent years. Data collection in WSNs
consists of two types: (1) *Aggregate data collection* is to enable each internal node in the tree to aggregate all the data received from its children before forwarding them toward the base station. A node compresses the data from all its children and its own data into one packet, and then sends that data packet to its parent. Thus, only one transmission slot is assigned to a sensor node to send all data in its sub-tree if any, fit into one packet and the data packets created by different sensor nodes are not aggregated on the way to the base station. We eliminate collisions by scheduling only non-interfering transmissions to proceed in the same transmission slot. Note that the proposed algorithm is a scheduling strategy, thus it is independent of the interference models. To simplify the presentation, we only consider pair-wise conflict relationships in which a transmission from a node to its parent conflicts with the transmission of its siblings, parent and grandparent over the tree $T$, same as in [9].

4. Proposed Scheme

4.1 The Overall Approach

Our proposed scheme consists of two phases: the scheduling and the data transmission phase. In the scheduling phase, we assign transmission slots to all sensor nodes under the assumption that each of them has data to send. To achieve a minimum delay, we use a parallel strategy which enables each sensor node has the chance to send their own data as early as possible. In the data transmission phase, after the schedule information is obtained, each sensor node applies an adaptive traffic mechanism to reduce idle listening according to the real data distribution of current network traffic. Using this mechanism, the base station can conclude data collection earlier instead of listening until the end of schedule.

4.2 Delay-Efficient Data Collection Scheduling

In this section, we introduce a TDMA collision-free schedule algorithm, called a DETA schedule. We assume that each sensor node has one data packet to send to the base station and use an example shown in Fig. 1(a) to explain the algorithm. We define three states for each sensor node: *Wait*, *Ready*, and *Scheduled*. Initially, all the sensor nodes are in the *Wait* state. A node is changed to *Ready* state if it is a leaf node or a node whose all children have been scheduled. The DETA schedule assigns transmission slots to the nodes only if they are in *Ready* state. After being assigned transmission slots, the state of that node changes to *Scheduled*.

Algorithm 1 presents the pseudo code of the DETA algorithm. There are some variables which will be frequently used in this presentation: $T$ is a given data collection tree, *ReadySet* is the set which contains all sensor nodes in *Ready* state. We consider that sensor node $v$: *Ch*(v) contains all descendants of $v$, $p(v)$ is a parent node of $v$, *CS*(v) is a set of nodes that conflict with $v$ when they send data in the same transmission slot with $v$, *ReqTS*(v) is the number of required transmission slots of node $v$ to send all data in the sub-tree rooted at node $v$, $TS(v)$

---

3. System Model and Assumptions

In our approach, we model the sensor network by a unit-disk graph $G = (V, E)$, where $V$ is the set of nodes including the sink $S$, and $E$ is the set of links. An edge $(u, v) \in E$ if and only if node $u$ is in the transmission range of node $v$. Due to limited transmission range, a routing infrastructure has to be constructed to transport data from sensor nodes to the base station. A common practice is to organize the sensor nodes into a tree structure rooted at the base station [4]–[8]. In this paper, we also assume that the routes from all sensor nodes to the base station have been formed by a given data collection tree $T$.

Using the same assumptions as in [4]–[10], we suppose that the data reported by each sensor in a sampling interval, if any, fit into one packet and the data packets created by different sensor nodes are not aggregated on the way to the base station. We divide the time into slots, and a node only can send or receive one data packet during one transmission slot. Our goal is to find a TDMA schedule strategy which can minimize data collection delay and energy consumption. As mentioned before, the network traffic frequently varies over different sampling intervals, thus the proposed schedule must effectively deal with these changes.
contains all transmission slots assigned to node $v$. For any set of nodes $V$, $TS(V) = \cup_{v \in V} TS(v)$, DETA scheduling algorithm starts from leaf nodes and works in rounds. At the beginning of each round, all leaf nodes are in Ready state; other nodes are in Wait state. We then apply an algorithm called SCHEDULE-ROUND to assign transmission slots to all sensor nodes in the ReadySet of $T$. Each schedule round ends when the SCHEDULE-ROUND algorithm is finished. If a sensor node is assigned the number of required transmission slots, it will be removed from the tree $T$.

The SCHEDULE-ROUND algorithm is presented by the pseudo code in Algorithm 2 where the goal is to schedule all sensor nodes in the ReadySet. First, it picks a node in the ReadySet, and assigns transmission slots for that node. The transmission slots obtained are the set of transmission slots when a node is in Scheduled state, which will be presented later. Then, the node changes its state to Scheduled, and informs its state and schedule information to its neighbors within two hops in the tree. Next, it is removed from the ReadySet. On receiving schedule information from its children, the parent node will check whether all its children are in Scheduled state. If so, its state changes to Ready and the node is put into ReadySet. The algorithm continues to assign the transmission slots to the Ready nodes until the ReadySet is empty. It means that all children of the sink are in Scheduled state. As mentioned before, the schedule round also ends at this moment, the algorithm removes the nodes which have assigned the number of required transmission slots from the tree and starts the new round. In the same way, the next round repeats all the procedures with new tree structure.

All the transmission slots are obtained by the SCHEDULE-NODE algorithm, where its pseudo code is presented in Algorithm 3. There are three types of transmission slots that can be assigned to the node. The first type is assigned to allow the leaf nodes to send their own data and the internal tree nodes to forward previously received data. There is at most one transmission slot assigned to a sensor node over a schedule round. The type two transmission slot enables the internal nodes to send their own data earlier; thus only one data packet of type two is assigned to one node in a whole algorithm. According to receive the data packets from some descendants early, the node is also responsible to forward those packets as soon as possible. The third type of transmission slot is used to do this duty. We define $T_1(v)$ as the type one set of transmission slots, $T_2(v)$ is the type two transmission slot, and $T_3(v)$ is the set of transmission slots of type three of node $v$. $T_4(v)$ is the set of transmission slots when a node $v$ receives early data packets from its children. Initially, $T_1(v)$, $T_3(v)$, and $T_4(v)$ are empty, $T_2(v)$ is equal to zero. The three types of transmission slots can be obtained by the following rules:

Rule #1: “Each schedule round is responsible to assign transmission slots to the sensor nodes to collect data from the leaf nodes.” A parent node receives data from all its children, and then forwards them one by one. Therefore, if the child has no data to send, it means that there is no remaining data in its sub-tree. In Fig. 1(a), since node $B, L, I, O, N$ and $V$ are the leaf nodes, they are assigned transmission slots 1, 1, 2, 1, and 1, respectively, to send their data. Then, the parent of these sensor node must be assigned transmission slots to forward data received from them, such as node $U$, it is assigned transmission slots 2 to forward data.
Algorithm 3 SCHEDULE-NODE(v)
1: // Assigning type one of transmission slot
2: \( t_1(v) \leftarrow \min\{r | r > 0, r > t_1(x), \forall x \in Ch(v), r \notin TS(CS(v))\} \)
3: \( T_1(v) \leftarrow T_1(v) \cup \{t_1(v)\} \)
4: \( TS(v) \leftarrow TS(v) \cup \{t_1(v)\} \)
5: // Assigning type two of transmission slot
6: if \((t_2(v) = 0)\) and \((|TS(v)| < ReqTS(v))\) then
7: \( Z \leftarrow [1, t_1(v) - 1] \)
8: if \( Z \neq \emptyset \) then
9: \( t_2(v) \leftarrow \min\{r \in Z, r \notin TS(CS(v))\} \)
10: \( TS(v) \leftarrow TS(v) \cup \{t_2(v)\} \)
11: end if
12: end if
13: // Assigning type three of transmission slot
14: \( Z = Z \setminus \{t_2(v)\} \)
15: \( T_r(v) \leftarrow \cup_{x \in Ch(v)} \{t_2(x), T_3(x)\} \)
16: while \( T_r(v) \neq \emptyset \) and \((|TS(v)| < ReqTS(v))\) do
17: \( t_3^1(v) \leftarrow \min\{t \in T_r(v)\} \)
18: if \( \exists z, z = \min\{r \in Z, r > t_3(v), t_3(v) \in T_r(v), r \notin TS(CS(v))\}\) then
19: \( T_3(v) \leftarrow T_3(v) \cup \{z\} \)
20: \( TS(v) \leftarrow TS(v) \cup \{z\} \)
21: end if
22: \( T_r(v) \leftarrow T_r(v) \setminus \{t_3^1(v)\} \)
23: end while

from node V to node H. In general, the transmission slot assigned to a parent node is always greater than all of its children. Therefore, the type one transmission slot assigned to a node \( v, T_1(v) \), is presented in lines 2 of Algorithm 3.

Rule #2: “Each internal node in a current data collection tree can report its own data early such that no collision happens.” After assigning the type one transmission slot to forward data from the leaf node, the node continues to be assigned one time slot to send its own data. In Fig. 1(a), after being assigned time slot 4 as the first type, node R is assigned time slot 1 to send its own data. Similarly, after being assigned time slots 7, 4, 4, nodes A, D, G then are assigned time slots 1, 1, 3, respectively, to send their own data. In general, type two transmission slot of node \( v, t_2(v) \), can be obtained as in lines 6–11 of Algorithm 3.

Rule #3: “Upon receiving the children’ own data early, a parent node is responsible to forward those data as early as possible.” If we consider each child node \( x \) of \( v \), node \( x \) could send its own data early at \( t_2(x) \), node \( v \) therefore has to be assigned a transmission slot to forward that data immediately. Secondly, if node \( x \) also has to forward data to some others nodes which can send their own data early, the data should be continuously forwarded by \( v \) as soon as possible. Therefore, we put all the early receiving transmission slots of \( v \) into list \( T_r(v) \), thus \( T_r(v) = \cup_{x \in Ch(v)} \{t_2(x), T_3(x)\} \), and sort the values in this set in ascending order. We then assign the transmission slots to forward all early data packets received by those transmission slots. As in lines 14–23, type three transmission slots of a node \( v \) are obtained. For example, node G in Fig. 1(a) can send its own data at \( t_2(G) = 3 \). Therefore, node A is a parent of G responsible to forward that data early, and time slot 4 is assigned to forward that data, \( T_3(A) = \{4\} \).

4.3 The Adaptive Network Traffic Mechanism

In this section, we describe an adaptive mechanism that enables sensor nodes to detect the completion of data transmission in their sub-tree. Note that in the same round of the DETA schedule, the first type time slot which is assigned to the parent is always greater than of all its children, which is represented in (1). Moreover, from (2)–(3), the first type of transmission slot that is assigned to a node is greater than the two remaining types in each schedule round. Therefore, if the parent node \( v \) does not receive any data from node \( x \) at any \( t_1(x) \in T_1(x) \), and \( t_1(x) \) is greater than the maximum value in \( T_3(x) \), node \( v \) can go into sleep mode in all subsequent receiving transmission slots in \( T_1(x) \). The base station, instead of listening until the end of schedule, concludes data collection once it infers that
performance evaluation. However, we do not lose the energy in all the earlier transmission slots because the earlier transmission slot of the parent node also can be used to forward data from its children, e.g., if a child node is assigned time slot 1 to send its own data, then a parent node will be assigned time slot 2 to send its own data; thus, if a parent node does not have data to send, the time slot of the parent node will be used to forward data from its children. In Fig. 5, we can see that the effect of that disadvantage is not too big, we only spend at most 2.8% more energy.

5.2.2 The Impact of Network Density

To investigate the impact of network size, we increase the number of nodes to vary from 100 to 600 in the same 100m x 100m region, each sensor also has a 15m transmission range. We evaluate the proposed algorithm with both: full and dynamic traffic patterns. In full traffic networks, each sensor node has one data to send to the base station in a sampling interval. Fig.3 illustrates the improvement of our scheme compared with the TPO scheme in terms of data collection delay. Our proposed algorithm can reduce data collection delay significantly. The improvement varies from 6.4% to 17.2%. In particular, a salient feature of the DETA is that it consumes the same energy consumption with the TPO in the case of full traffic. This is because they use the same amount of total transmission slots to collect all data, as in Fig. 6. Secondly, we assume that only 50% nodes in the network have data to report to the base station. The impact of increasing the number of nodes in the fixed sensing area are presented in Fig. 4 and Fig. 7. Note that our scheme can achieve 6.7% to 18.5% improvement compared with the TPO scheme in terms of data collection delay as in Fig. 4. As seen in Fig. 7, the effect of the disadvantage of the DETA algorithm regarding energy consumption is very small; the DETA scheme only spends at most 2.7% more energy.

6. Conclusion

In this paper, we have presented a TDMA scheduling algorithm to schedule all nodes in the network to send their data to the base station with minimum delay. In addition, we designed an adaptive mechanism to enable sensor node to go to sleep early according to the current data distribution in a sampling interval, thereby reducing data collection delay and energy consumption. The simulation results show that our proposed scheme achieves better performance than the existing schemes in terms of data collection delay.

Acknowledgment

This research was supported in part by MSIP (NIPA,KEIT) and MEST(NRF), Korean government, under ITRC (NIPA-2013-(H0301-13-3001)), IT R&D Program [10041244, SmartTV 2.0 Software Platform], and PRCP (2012-0005861), respectively.
Fig. 2: Data collection delay: varying traffic patterns

Fig. 3: Data collection delay: varying number of nodes when all nodes have data to send

Fig. 4: Data collection delay: varying number of nodes when 50% of nodes have data to send

Fig. 5: Energy consumption: varying traffic patterns

Fig. 6: Energy consumption: varying number of nodes when all nodes have data to send

Fig. 7: Energy consumption: varying number of nodes when 50% of nodes have data to send

References