

A Framework for Overall Storage Overflow Problem to Maximize the Lifetime in WSNs

Guoliang Song, Chen Zhang^(✉), Chuang Liu, and Yuna Chai

Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
s_guoliang@foxmail.com, chenzhanghit@outlook.com, chuangliuhit@gmail.com,
chaiyuna@outlook.com

Abstract. Storage overflow problem in wireless sensor networks is a new and challenging issue, wherein data-collecting base station is not available while more data items are generated than available storage space in the entire network. In this paper, we consider overall storage overflow problem in WSNs, the goal of which is to maximize the minimum remaining energy of data node (the node with overflow data) in order to prolong the lifetime of the sensor network. For overall storage overflow problem, we propose a two-step solution. A degree-constrained data aggregation algorithm is presented, and then we further propose a data replication algorithm which is a unified method, integrating data aggregation and data redistribution. Extensive simulations show that our proposed algorithms significantly outperform than existing algorithms especially in extending the lifetime of the sensor network.

Keywords: Wireless sensor networks · Overall storage overflow · Data aggregation · Data redistribution

1 Introduction

In recent years, wireless sensor networks (WSNs) have been widely used in various fields. Many of them are deployed in remote area or challenging environments to collect large volumes of data for a long period of time, such as ocean monitoring, volcano eruption monitoring and climate change. Due to the inaccessible and hostile environments, it is not feasible to deploy long-term base station with power outlets. Therefore, the generated data is first stored inside the sensor network for a period of time, and then collected by periodic visit of the robots or data mules. In the challenging environment, however, uploading opportunities would be unpredictable and rare, a major problem is how to store the massive amount of data inside the network comprising of nodes with limited storage space and limited energy. In the sensor network, sensor nodes are randomly deployed in the area and then each node collects data independently. When events of the interest take place, sensor nodes close to them may collect data more frequently than nodes far away, therefore these nodes may run out of their storage space quickly than others. After a period of time, some nodes may deplete their storage space and generate overflow data, while other nodes may have available storage space. There are two level of data overflow in the sensor network.

1. **Partial Storage Overflow:** In this level of data overflow, some nodes (denoted as data nodes) in the sensor network deplete their own storage space while other nodes (denoted as storage nodes) still have available storage space. And the total size of available storage space is greater than or equal to the total size of overflow data. If uploading opportunities are not available, the newly generated data at data node can not be stored, causing data loss. In order to avoid data loss, data redistribution are proposed. The idea of data redistribution is that redistributing overflow data from data node to storage node, such that any data node does not have any overflow data.
2. **Overall Storage Overflow:** This is a more serious situation, where the total size of overflow data exceeds the total size of available storage space in the network. To overcome overall storage overflow problem, it needs two step: data aggregation [1] and data redistribution. Data aggregation for overall storage overflow problem is to reduce the size of overflow data, such that the overflow data can fit into the available storage space. After data aggregation, overall storage overflow problem becomes partial storage overflow problem and it can be solved by data redistribution.

Therefore overall storage overflow problem is more serious and complicated compared to partial storage overflow problem. In this paper, we focus on overall storage overflow problem. For this problem, we consider different sensor nodes may have different remaining energy, especially data node which usually has low remaining energy as collecting massive data consumes a lot of energy. The contributions of this paper are as follows:

1. We first study overall storage overflow problem in WSNs for maximizing the minimum remaining energy of data node. To our best knowledge, the problem has not been addressed by any of existing research.
2. We propose a data aggregation algorithm and a data replication algorithm. Data replication algorithm is a unified method which integrate data aggregation and data redistribution.
3. Extensive experiments have been conducted to verify that our algorithm achieves higher lifetime than existing approaches.

The rest of this paper is organized as follows. In Sect. 2, we present related work. In Sect. 3, we introduce overall storage overflow problem. Sections 4 and 5, we introduce data aggregation and unified method for overall storage overflow problem respectively. And we also give its corresponding algorithms. In Sect. 6, we compare the proposed algorithms with existing algorithms and discuss the performance. Section 7 concludes the paper with future work.

2 Related Work

Storage overflow problem in wireless sensor network is relatively new research topic. Tang et al. [1] study overall storage overflow problem in base station-less sensor networks. They solve the problem by data aggregation and data

redistribution. And they address data aggregation for overall storage overflow is equivalent to multiple traveling salesman walks problem (MTSW). Alhakami et al. [2] proposed a unified method that is based upon data replication techniques for overall storage overflow problem. Both above work assume that the energy of each node is infinity, ignoring different nodes may have different remaining energy.

Tang et al. [3] also study how to minimize the total energy consumption in the process of data redistribution, and address it as a minimum cost flow problem. Hou et al. [4] study how to maximize the minimum remaining energy of the nodes after data preservation, such that the data can be preserved for maximum amount of time. And Takahashi et al. [5] try to preserve the data inside the network for maximum possible time, by distributing the data items from low energy nodes to high energy nodes. Xue et al. [6] consider different data may have different importance and priority, and study how to preserve data with maximum priority. They address the core of the problem is a maximum weighted flow problem and propose a time efficient heuristic algorithm. A network flow perspective of data preservation problem in sensor networks is given in [7]. All above work, however, do not address overall storage overflow problem and they just try to redistribute overflow data as much as possible. In this paper, we consider the different remaining energy of nodes, and try to maximize the minimum remaining energy of data node in the process of data aggregation in order to prolong the lifetime of the network.

There are active research that focused on data aggregation. Kuo et al. [8] studies how to construct a data aggregation tree that minimizes the total energy cost of data transmission, while Chen et al. [9] study the construction of a data gathering tree to maximize the network lifetime. Yan et al. [10] and Lee et al. [11] consider the aggregation delay in the process of data aggregation and propose data aggregation scheduling scheme to minimize latency in duty-cycled WSNs. Some other work use mobile base stations collect aggregated data [12, 13]. However data aggregation for overall storage overflow problem significantly differs from above data aggregation. The above data aggregation in wireless sensor network is used to collect data items from different sensor nodes, in order to reduce number of transmissions and energy consumption. Data aggregation for overall storage overflow problem is to aggregate the overflow data, so that the overflow data can be stored in the available storage space. In Sect. 3, we introduce the process of data aggregation for overall storage overflow problem in detail.

3 Overall Storage Overflow Problem

The wireless sensor network consists of many nodes, we denote the node with overflow data as data node, and the node with available storage space as storage node. To aggregate data, one or more data nodes (called initiators) send their overflow data to other data nodes. When a data node (called an aggregator) receives the data, it aggregates its own overflow data, then forwards the initiators entire overflow data to another data node, which becomes an aggregators

and aggregates its own overflow data, and so on so forth. This continues until enough aggregators are visited such that the total size of overflow data is equals to or is slightly less than total available storage in the network. Each aggregator can aggregate its own overflow data only once. If an aggregator receives another initiator's overflow data, it just transfers it to other data node. And if a storage node receives the initiator's overflow data, it simply relays it. After the aggregation, the initiators' overflow data become zero, and the last aggregator has both its own aggregated data and the entire overflow data from initiator. Some data nodes which neither an initiator nor an aggregator are not involved in data aggregation, they have original overflow data which is not aggregated.

Network Model. The sensor network can be modeled as an undirected graph $G = (V, E)$, where $V = \{1, 2, \dots, |V|\}$ is set of $|V|$ sensor nodes, and E is set of $|E|$ edges. Every sensor node can transmit and receive data, but its transmission range is limited. $\forall v_i, v_j \in V$, there exists an edge $(v_i, v_j) \in E$ in graph G if and only if node v_i and v_j are in each other transmission range. Assume that each node has same transmission range and there are p data nodes, denoted as v_d . Thus the number of storage nodes (denoted as v_s) is $|V| - p$. We consider that each data node has same size of overflow data and each storage node has same available storage space. Let R denote the size of overflow data in bits at each data node, and let m denote the available storage space in bits at each storage node. For overall storage overflow problem, it satisfies the following equation.

$$p \times R > (|V| - p) \times m \quad (1)$$

Feasible Overall Storage Overflow. In order to reduce the overflow data to the size which can be stored by the available storage capacity, enough number of aggregators should be visited. Let q denotes the number of aggregators, and r represents the size of overflow data after data aggregation, which based on a spatial correlation model [14], indicating that the size of redundant overflow data between any two data nodes is $R - r$. The feasibility of data aggregation can be derived in [1].

$$q = \lceil \frac{p \times R - (|V| - p) \times m}{R - r} \rceil = \lceil \frac{p \times (R + m) - |V| \times m}{R - r} \rceil \quad (2)$$

There is at least one initiator, and the maximum number of aggregators is $p - 1$. Therefore, the valid range of p is

$$\frac{|V|m}{m + R} < p \leq \lfloor \frac{|V|m - R + r}{m + r} \rfloor \quad (3)$$

Example 1. Figure 1 is an example of overall storage overflow problem in a linear sensor network with five nodes. Figure 1(a) is data aggregation step for overall storage overflow problem. Node A , C and D are data nodes, while B and E are storage nodes. Each data node has 2 units overflow data, and each storage node has 2 units available storage space. There are total 6 units of overflow data while

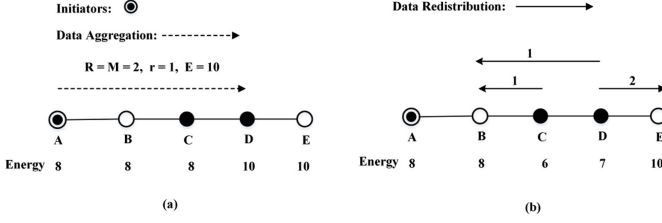


Fig. 1. A sensor network with overall storage overflow problem. (a) Data aggregation step. (b) Data redistribution step.

there are only 4 units of available storage space, causing overall storage overflow problem. We assume that $r = 1$. The number of aggregators q is calculated as 2 by using Eq. 2. Therefore the number of initiator is one. One possible aggregation walk is A, B, C, D and node A is the initiator. After data aggregation, the size of overflow data at A, C and D are 0, 1 and 3 respectively.

After data aggregation, the next step is data redistribution. Data redistribution is to decide how to redistribute overflow data from data node to storage node. This has been shown to be a minimum cost flow problem [3], which can be solved efficiently. Figure 1(b) shows data redistribution step post data aggregation. After data aggregation, the total size of overflow data is equal to available storage space. One possible data redistribution solution is redistributing C's 1 unit of data to B, D's 1 unit of data to B via C and D's 2 units of data to E. Finally, any data node does not have overflow data.

4 Data Aggregation for Overall Storage Overflow Problem

4.1 Data Aggregation Formulation

In the sensor network, all sensor nodes have a limited energy source, typically in the form of a battery. It is awkward and unreasonable to replace the energy source of node. For data node, it costs more energy than storage node as it collects and saves massive data items. The remaining energy of the data node is generally lower than the residual energy of the storage nodes. It is therefore significant to save the data node's energy for prolonging the lifetime of network (the time until the first node depletes its energy in the network).

Let $V_{DN} = \{DN_1, DN_2, \dots, DN_p\}$ denotes the set of data nodes and there is a set of aggregation walks: $W = \{W_1, W_2, \dots, W_a\}$, where each walk W_i ($1 \leq i \leq a$) start from a distinct initiator. Each node have its own energy E_i . Let E'_i denote node's residual energy after data aggregation. Then,

$$E'_i = E_i - \sum_{j=1}^a C_{i, W_j} \quad (4)$$

where C_{i,W_j} is the energy cost of node i in the aggregation walk W_j by transmitting or receiving data items. If node i is not in the aggregation walk W_j , $C_{i,W_j} = 0$. The objective of data aggregation is to find a set of aggregation walk $W = \{W_1, W_2, \dots, W_a\}$, such that the minimum energy among all data node V_{DN} is maximized post aggregation, while saving as much energy as possible.

$$\max_W \min_{1 \leq i \leq p} E'_{DN_i} \quad (5)$$

under the energy constraint that each node can not spend more energy than its own energy, $E'_i \geq 0, \forall i \in V$.

4.2 Data Aggregation Algorithm

Since the main participant in the process of data aggregation is the data node, we firstly transform the original sensor network $G(V, E)$ into an aggregation network $G'(V', E')$. In the aggregation network $G'(V', E')$, V' is set of p data node in V . For any two data node $v_i, v_j \in V'$, if there exists an edge $(v_i, v_j) \in E$, we add the same edge in $G'(V', E')$, thus $(v_i, v_j) \in E'$. Otherwise we find the shortest paths between node v_i and v_j , and add a new edge in the aggregation network. Its weight of the new added edge is the cost of the shortest path between those two nodes. Therefore the aggregation network is a complete graph. In this paper, we introduce a new variable to represent the weight of edge between data nodes in the network.

Definition 1 (Quality of Edge). For any two data nodes, $v_i, v_j \in V$ and $e_{ij} = (v_i, v_j) \in E$, $Q(e_{ij}) = \frac{d_{ij}^2}{E_{ij}}$, where d_{ij} is the distance between node v_i and v_j , and $E_{ij} = \min(E_{v_i}, E_{v_j})$ is the minimum energy between node v_i and v_j .

The quality of edge is proportional to the square of the distance between two data nodes in the edge. The idea behind the quality of the edge is that the larger quality of the edge, the two node in this edge will have less energy or longer distance, then less likely the edge will be selected as data aggregation path. We use the quality represent the weight of edge to help us to select the aggregation path. In the aggregation network, if two data nodes are not directly connected in the original sensor network, the distance d_{ij} is the cost of the shortest path between those two nodes, and E_{ij} is the minimum energy of node in the found

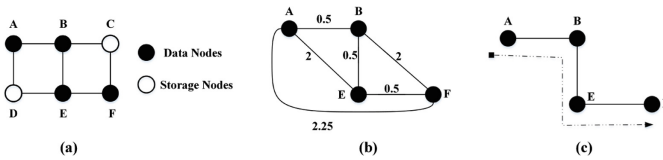


Fig. 2. (a) Original sensor network G . (b) Aggregation network G' . (c) Aggregation walk.

shortest path. Figure 2(a) shows the original wireless sensor network. Figure 2(b) is the corresponding aggregation network G' . In the original grid sensor network Fig. 2(a), we assume that the distance between any pair of connected nodes is 1 and the energy of every node is 2. The quality of edge is marked on every edge in Fig. 2(b) according to Definition 1.

Algorithm 1. Degree-Constrained Data Aggregation Algorithm

Input: $G(V, E)$ and the number of aggregators q

Output: The set of aggregation walks W and $E'_{min_{DN}}$

Notations:

e_i : the edge in the aggregation network graph;

$v_{e'_i}$: one node of the edge e_i ;

$v_{e''_i}$: the other node of the edge e_i ;

$rf(v_i)$: the number of reference of the node v_i ;

$Q(e_i)$: the quality of the edge e_i ;

$E'_{min_{DN}}$: the minimum remaining energy of data node;

- 1: Transform $G(V, E)$ into $G'(V', E')$;
 - 2: Calculate the quality of each edge in $G'(V', E')$;
 - 3: Sort edges' quality in $G'(V', E')$, $Q(e_1) \leq Q(e_2) \leq \dots \leq Q(e_N)$;
 - 4: **for** $1 \leq j \leq |V'|$ **do**
 - 5: Initialize node $rf(v_j) = 2$;
 - 6: **end for**
 - 7: $W = \phi$, $count = i = 1$;
 - 8: **while** $count \leq q$ **do**
 - 9: **if** $rf(v_{e'_i}) > 0$ and $rf(v_{e''_i}) > 0$ and (e_i in W will not induce cycle) **then**
 - 10: $W = W \cup \{e_i\}$;
 - 11: $rf(v_{e'_i}) --$;
 - 12: $rf(v_{e''_i}) --$;
 - 13: $count ++$;
 - 14: **end if**
 - 15: $i ++$;
 - 16: **end while**
 - 17: **for** $1 \leq j \leq |W|$ **do**
 - 18: Aggregate data along W_j from one end which has the smaller quality of edge;
 - 19: **end for**
 - 20: find the minimum remaining energy of data node $E'_{min_{DN}}$;
 - 21: **return** W and $E'_{min_{DN}}$;
-

Degree-Constrained Data Aggregation Algorithm. Now we present an approximation algorithm for data aggregation. It works as follows. Line 1 transforms the original wireless sensor network graph into the aggregation network graph. Line 2 calculates the quality of each edge in the aggregation network according to definition 1. Line 3 sorts all the edges' quality into nondecreasing order. Lines 4–6 initialize the number of reference of each node to be 2. That is, the degree of each node in the set of aggregation walks will not exceed 2. The while loop in lines 8–16 check if each edge in W is cycleless and the number

of reference of each node is greater than 0. If yes, add it into W . This continues until q edges are added into W . After that, it starts from one end which has the smaller quality of edge and aggregate overflow data via visiting the rest nodes. Figure 2(c) shows the aggregation walk which generated by the algorithm corresponding to aggregation network graph Fig. 2(b).

Time Complexity. Due to space constraints, the analysis is omitted. The time complexity of this algorithm is $O(|V|^3)$.

5 Integrating Data Aggregation and Data Redistribution

To overcome overall storage overflow problem in the sensor network, we have a two-step solution. But this solution does not necessarily achieve good performance. A unified method is proposed [2] which is based on data replication techniques. Data replication technology for overall storage overflow problem is that using storage node which is on the aggregation walk to replicate part or all overflow data of initiator in the process of data aggregation. However the total size of replicated data on any storage node along any aggregation walk cannot exceed this node's available storage space. As it does not consume extra energy, it saves a lot of energy in the step of data redistribution.

Example 2. Figure 1 shows the example of two-step solution for overall storage overflow problem. Considering that initial energy E of each node is 10, and the energy cost is 1 by transmitting 1 data item. Thus, data aggregation and redistribution cost is 6 and 5 respectively, and the residual energy of each node is marked under the node in Fig. 1. The total energy cost of two-step solution is 11 and the minimum residual energy of data node is 6. Figure 3 illustrate the data replication technology with the same sensor network which described in Fig. 1. It shows that when initiator node A sends its 2 units of data passing storage node B , it replicates 1 unit of the data (marked in parentheses) and stores at B . Therefore, next in data redistribution step, node D only needs redistribute 2 units of data to node E . Finally the total energy cost is 9 which has an 18% improvement compared to two-step solution, while the minimum residual energy of data node is 7, having a 17% improvement.

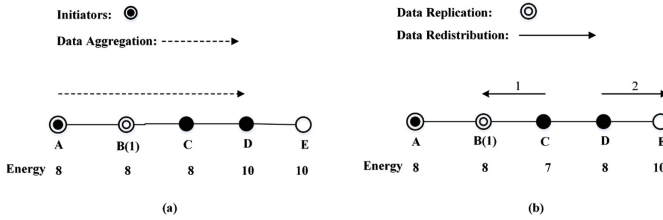


Fig. 3. Data replication technology for same sensor network in Fig. 1.

Since our Degree-Constrained Data Aggregation Algorithm can find the set of aggregation walks and data redistribution can be solved by minimum cost flow algorithm, the challenges of data replication technology are how to select initiator for each aggregation walk and how many units of data to replicate at storage nodes which in the aggregation walk. In each aggregation walk, the initiator has two choices. For example, in Fig. 3(a) the initiator can be node A or node D which can lead to different energy consumption. Observing that the last aggregator has more overflow data than other data nodes after data aggregation and having more available storage nodes around the last aggregator would make the data redistribution more energy-efficient. Therefore we select the node which surrounded by less storage nodes as the initiator in each aggregation walk. For the second challenge, we give below definition.

Definition 2 (Demand Number of Storage Node). For any storage node u on any aggregation walk, let $N(u)$ be all its one-hop neighbor nodes. For each data node $v \in N(u) \cap V_{DN}$, let $D_{u,v}$ represent the distance between node u and node v , and $V_{SN} = V - V_{DN}$ denotes the set of storage nodes. The demand number $d(u)$ of storage node u , $d(u) = \sum_{v \in N(u) \cap V_{DN}} \frac{1}{\sum_{q \in N(v) \cap V_{SN}} \frac{D_{u,v}^2}{D_{q,v}^2}}$.

Noted that $\sum_{q \in N(v) \cap V_{SN}} \frac{D_{u,v}^2}{D_{q,v}^2} > 0$ since node v has at least one neighboring storage node u . And if each node v just has one neighboring storage node u , the value of $d(u)$ is equal to the number of data nodes which surround node u . The idea behind $d(u)$ is that the less number of data nodes surrounding u and the more number of storage nodes surrounding such data nodes with shorter distance, the more unites of data items should be replicated at storage node u . Next we give a data replication algorithm, it works as follows. In each aggregation walk, the initiator sends all its overflow data to the next node along the walk. If a data node receives the overflow data, it just aggregates its own data and then sends the received data to the next node. For a storage node u receiving the data, firstly it calculates its own demand number $d(u)$. And then calculate the amount of data to be replicated as $\min(\frac{z}{d(u)}, z, s)$, where z is the rest of overflow data which has not been replicated and s represents the available storage space of this node. Finally node u replicates the calculated units of data in its storage space and relays the entire overflow data to the next node along the aggregation walk. This continues until the last aggregator receives the overflow data. The last aggregator aggregates its data and keeps the rest units (may be zero) of overflow data which have not been replicated. Finally, the aggregated overflow data is redistributed to storage node which has available storage space by minimum cost flow algorithm [3].

Time Complexity. Due to space constraints, the analysis is omitted. The time complexity of this algorithm is $O(|V|^2|E|\log(|V|C))$, where $C = \max\{R+r, m\}$.

Algorithm 2. Data Replication Algorithm**Input:** The sensor network G , and the set of aggregation walk W **Output:** Minimum remaining energy of data node $E''_{min_{DN}}$ **Notations:** $|W_i|$: the number of nodes on the aggregation walk W_i ; $mcfa$: minimum cost flow algorithm; $V_j.space$: available storage space of storage node V_j ;1: $a = |W|$;2: **for** $1 \leq i \leq a$ **do**3: Let V_{ini} and V_{agg} be the initiator and the last aggregator on the aggregation walk W_i respectively;4: V_{ini} sends all its overflow data to the next node along W_i ;5: $z = R, b = |W_i|$;6: **for** $2 \leq j \leq (b - 1)$ **do**7: **if** ($V_j \in W_i$ is data node) **then**8: V_j aggregates its own data with overflow data of V_{ini} ;9: **else**10: $s = V_j.space$;11: **if** ($s > 0$ and $z > 0$) **then**12: Calculate $d(V_j)$;13: $t = \min(\frac{z}{d(V_j)}, z, s)$;14: Replicate t units of overflow data on V_j ;15: $z = z - t$;16: **end if**17: **end if**18: V_j sends the entire overflow data of V_{ini} to the next node along W_i ;19: **end for**20: V_{agg} aggregates its own data and keeps the rest z units of data of V_{ini} ;21: **end for**22: $E''_{min_{DN}} = mcfa(G)$;23: **return** $E''_{min_{DN}}$;

6 Performance Evaluation

This section presents the effectiveness of our proposed algorithms for overall storage overflow problem. Extensive experiments were performed in Java. In our experiment, we adopt first order radio model [15]. For node u send R -bit data to its neighbor v over their distance d , the transmission energy cost at u is $E_t(R, d) = E_{elec} \times R + \epsilon_{amp} \times R \times d^2$, and the receiving energy cost at v is $E_r(R) = E_{elec} \times R$, where $E_{elec} = 100$ nJ/bit and $\epsilon_{amp} = 100$ pJ/bit/m². 50 and 100 sensor nodes are scattered randomly across a 1000×1000 m² network, in which no two nodes can be in the same location. The transmission range of each node is 250 m. For data node, the initial energy is randomly around 600 J–700 J, while the initial energy of each storage node is randomly around 900 J–1000 J. Unless otherwise mentioned, the sensor network consists of 50 nodes, and $R = m = 1$ MB. To eliminate the impact of randomness, each experiment scenario is repeated 100 times.

6.1 Performance of Data Aggregation Algorithm

For data aggregation algorithm, we compare the performance of our Degree-Constrained Data Aggregation algorithm (denoted as DCDA) with STF-Walk [1] and LP-Walk [1] algorithm. STF-Walk algorithm is a $(2 - \frac{1}{q})$ -approximation data aggregation algorithm, while LP-Walk is a novel heuristic algorithm.

We compare DCDA algorithm with STF-Walk algorithm where considering $r/R = 0.5$ and the whole valid range of $p \in [26, 33]$. Figure 4(a) shows the total aggregation cost of STF-Walk and DCDA algorithms. With the increase of the number of data nodes p , total aggregation costs of both STF-Walk and DCDA increase. It's obviously that the DCDA algorithm yields less cost than STF-Walk. This is because STF-Walk algorithm visits some edges twice in the process of data aggregation, while DCDA algorithm tries to visit some edges which has smaller weight instead of edges' second visiting. Figure 4(b) shows the minimum remaining energy of data node after data aggregation corresponding to Fig. 4(a). The minimum remaining energy of data node decrease with increase p in both DCDA and STF-Walk algorithms. As DCDA algorithm considers different node with different remaining energy and selects the data node with higher priority which has higher remaining energy to participate in the process of data aggregation, the minimum remaining energy of data node in DCDA algorithm is always higher than the remaining energy in STF-Walk algorithm. And the performance difference between DCDA and STF-Walk algorithm gets bigger with the increase of number of data nodes.

LP-Walk algorithm which is a novel heuristic algorithm outperforms STF-Walk algorithm in total energy consumption. We adopt $r/R = 0.3$ and 0.7 , for $r/R = 0.3$, the valid range of p is from 26 to 37, while $p \in [26, 29]$ for $r/R = 0.7$. Figure 5(a) is the aggregation energy cost by varying r/R and p . And Fig. 5(b) is corresponding the minimum remaining of data node after data aggregation. It shows that for the same p , with the increase of r/R , the total aggregation cost for both DCDA and LP-Walk algorithm increase and the minimum remaining energy of data node post aggregation decrease. The reason is that less redundant

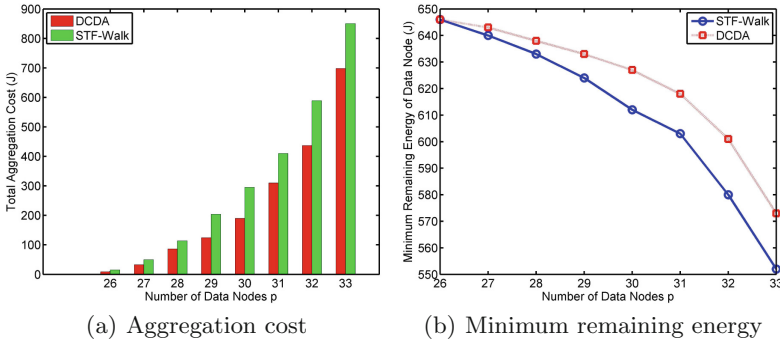


Fig. 4. Comparing DCDA with STF-Walk by varying p where $r/R = 0.5$.

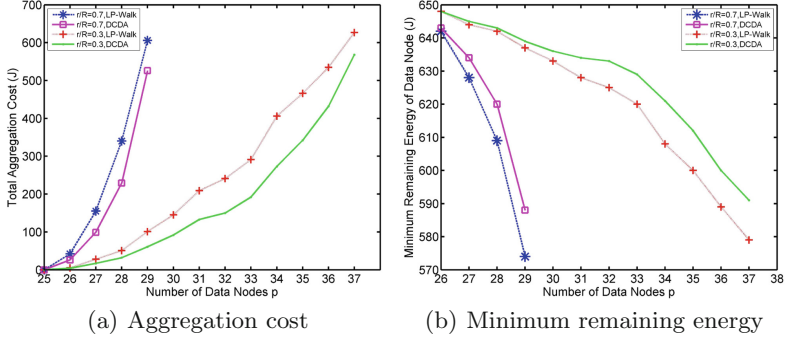


Fig. 5. Comparing DCDA with LP-Walk by varying p and r/R .

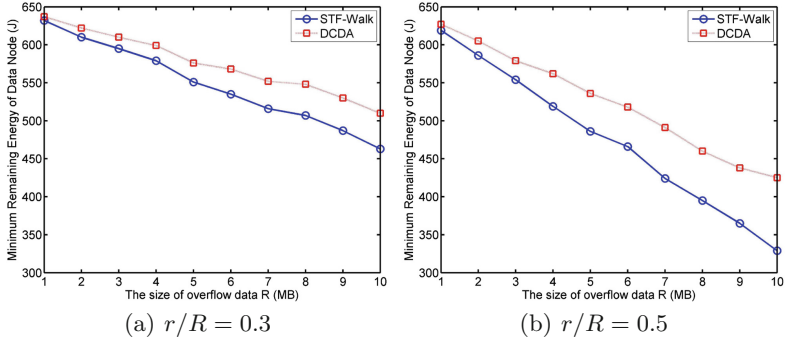


Fig. 6. Comparing DCDA with LP-Walk in minimum remaining energy of data node by varying R where $p = 30$.

data between data nodes leads to more number of aggregator are visited, thus increasing aggregation energy cost and reducing the minimum remaining energy of data node. However DCDA algorithm outperforms LP-Walk algorithm in both aggregation cost and minimum remaining energy of data node for the same reason. In Fig. 6(a), $p = 30$ and $r/R = 0.3$, we vary R from 1 MB to 10 MB while in Fig. 6(b) $p = 30$ and $r/m = 0.5$. It's obviously that as the increase of the size of overflow data, the minimum remaining energy of data node post aggregation in both two algorithm decrease, and the minimum remaining energy in Fig. 6(b) decrease faster than that in Fig. 6(a). This is because with the increase of r/R , the more number of aggregators should be visited, it costs more energy for data nodes. However the minimum remaining energy of data node in DCDA algorithm is still higher than that in LP-Walk algorithm and the performance difference get bigger with the increase of R . Therefore DCDA algorithm is more energy efficient and can prolong the lifetime of the sensor network compared to LP-Walk algorithm.

6.2 Performance of Data Replication Algorithm

In this experiment, two-step solution adopts our data aggregation algorithm (DCDA) and minimum cost flow algorithm. We compare the performance of data replication algorithm and two-step solution. In Fig. 7, $r/R = 0.5$, we vary p from 26 to 33. Figure 7(a) shows data redistribution energy cost and Fig. 7(b) presents minimum remaining energy of data node. It is obviously that replication algorithm performs better than two-step solution in both data redistribution energy cost and minimum remaining energy of data node. In Fig. 7(a), data redistribution energy cost decrease with increase of data node p . This is because with the increase of p , the more overflow data is aggregated and less overflow data is redistributed. In general, the minimum remaining energy of data node decrease with increase of data node. However, in some case, the minimum remaining energy of data node increase with the increase of data node. It maybe that algorithms find relatively short paths in the process of data aggregation and redistribution, leading to less energy cost. Figure 8 investigates the effect

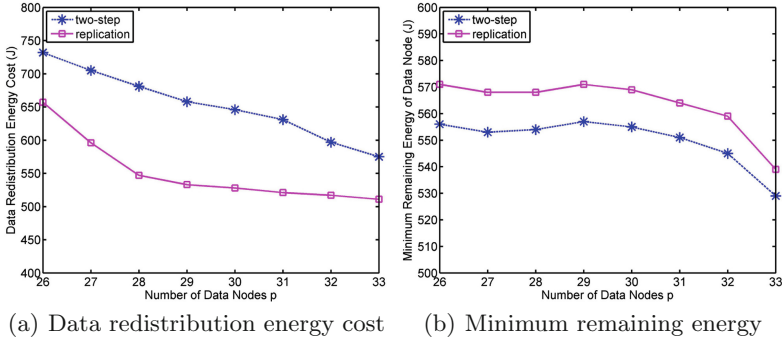


Fig. 7. Comparing data replication algorithm with two-step solution by varying p where $r/R = 0.5$.

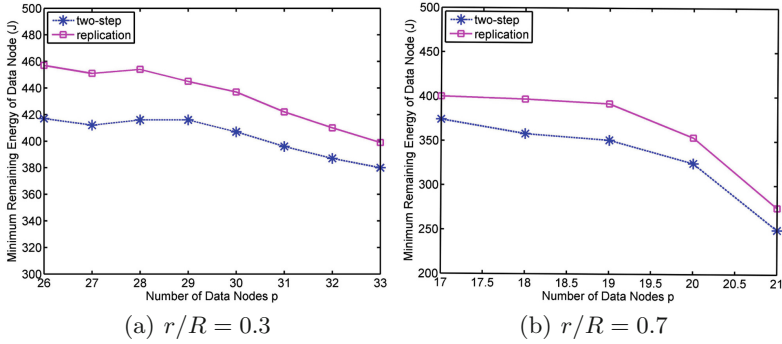


Fig. 8. Comparing replication algorithm with two-step solution in minimum remaining energy of data node by varying p where $R/m = 5$.

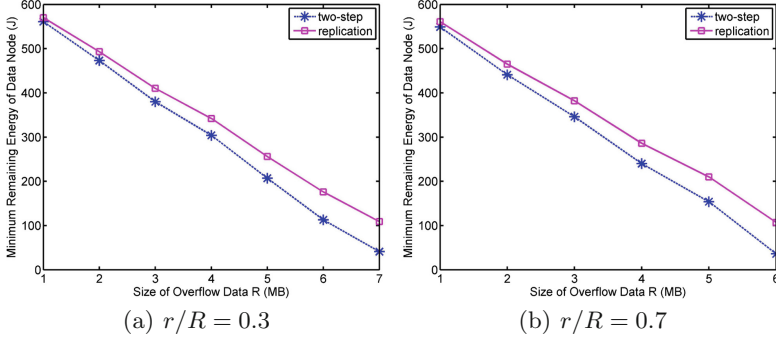


Fig. 9. Comparing replication algorithm with two-step solution in minimum remaining energy of data node by varying R where $p = 28$.

of $R/m = 5$ and $m = 1$ MB. Figure 8(a) shows the performance of data replication algorithm and two-step solution in minimum remaining energy of data node where $r/R = 0.3$ while in Fig. 8(b) $r/R = 0.7$. It shows the same trend as Fig. 7(b). For data replication algorithm, it replicates data items at storage nodes, saving a lot of energy of sensor nodes during data redistribution. And it still has advantage over two-step solution.

Figure 9 presents the effect of the size of overflow data where $p = 28$. We vary R from 1 MB to 7 MB. With the increase of R , it becomes more challenging since there are more overflow data. In Fig. 9(a), $r/R = 0.3$ while $r/R = 0.7$ in Fig. 8(b). We observe that with the increase of R , the minimum remaining energy of data node decrease linearly. And with the same R , the minimum remaining energy of data node in $r/R = 0.7$ is lower than that in $r/R = 0.3$. This is because with the increase of r/R , more aggregators are visited, it costs more energy. In Fig. 8(b), the sensor network can not finish data aggregation and data redistribution work as some data node deplett their energy when $R = 7$ MB. However, data replication algorithm outperforms than two-step solution. And the performance difference gets larger with the increase of R . This again demonstrates the effectiveness of replication algorithm.

7 Conclusions and Future Work

In this paper, we study overall storage overflow problem in wireless sensor network, the goal of which is to maximize the minimum energy of data node. To our best knowledge, the problem has not been addressed by any of existing research. And we propose energy-efficient data aggregation and data replication algorithms. Via extensive simulations, it shows that our algorithms can effectively prolong the lifetime of sensor network compared with existing algorithms. As for future work, we will consider that different data nodes may have different size of overflow data. And in order to adapt to large scale sensor networks, we will design distributed algorithms.

References

1. Tang, B., Ma, Y., Alhakami, B.: Dao: overcoming overall storage overflow in base station-less sensor networks via energy efficient data aggregation. Technical report, California State University Dominguez Hills
2. Alhakami, B., Tang, B., Han, J., Beheshti, M.: DAO-R: integrating data aggregation and offloading in sensor networks via data replication. In: 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1–7 (2015)
3. Tang, B., Jaggi, N., Wu, H., Kurkal, R.: Energy-efficient data redistribution in sensor networks. *ACM Trans. Sensor Netw.* **9**(2), 1–28 (2013)
4. Hou, X., Sumpter, Z., Burson, L., Xue, X.: Maximizing data preservation in intermittently connected sensor networks. In: 2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS), pp. 448–452 (2012)
5. Takahashi, M., Tang, B., Jaggi, N.: Energy-efficient data preservation in intermittently connected sensor networks. In: 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 590–595 (2011)
6. Xue, X., Hou, X., Tang, B., Bagai, R.: Data preservation in intermittently connected sensor networks with data priority. In: 2013 10th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), pp. 122–130 (2013)
7. Tang, B., Bagai, R., Nilofar, F., Yildirim, M.B.: A generalized data preservation problem in sensor networks—a network flow perspective. In: International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW), pp. 275–289 (2014)
8. Kuo, T.W., Lin, K.C.J., Tsai, M.J.: On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: NP-completeness and approximation algorithms. *IEEE Trans. Comput.* **65**(10), 3109–3121 (2016)
9. Chen, Z., Yang, G., Chen, L., Xu, J.: Constructing maximum-lifetime data-gathering tree in wsns based on compressed sensing. *Int. J. Distrib. Sensor Netw.* **12**(5), 1–11 (2016)
10. Yan, X., Du, H., Ye, Q., Song, G.: Minimum-delay data aggregation schedule in duty-cycled sensor networks. In: Yang, Q., Yu, W., Challal, Y. (eds.) WASA 2016. LNCS, vol. 9798, pp. 305–317. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42836-9_28
11. Lee, T., Kim, D.S., Choo, H., Kim, M.: A delay-aware scheduling for data aggregation in duty-cycled wireless sensor networks. In: 2013 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), pp. 254–261 (2013)
12. Liu, C., Du, H., Ye, Q.: Sweep coverage with return time constraint. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2016)
13. Alia, O.M.: Dynamic relocation of mobile base station in wireless sensor networks using a cluster-based harmony search algorithm. *Inf. Sci.* **385–386**, 76–95 (2017)
14. Cristescu, R., Beferull-Lozano, B., Vetterli, M., Wattenhofer, R.: Network correlated data gathering with explicit communication: NP-completeness and algorithms. *IEEE/ACM Trans. Netw.* **14**(1), 41–54 (2006)
15. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: 33rd Annual Hawaii International Conference on System Sciences (2000)

Combinatorial Optimization and Applications
11th International Conference, COCOA 2017, Shanghai,
China, December 16-18, 2017, Proceedings, Part I
Gao, X.; Du, H.; Han, M. (Eds.)
2017, XVIII, 483 p. 125 illus., Softcover
ISBN: 978-3-319-71149-2