Ad Hoc Networks 11 (2013) 1287–1296

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Near optimal scheduling of data aggregation in wireless sensor networks

Pei Wang [a,*], Yuan He [b], Liusheng Huang [a]

[a] Department of Computer Science, University of Science and Technology of China, China
[b] Department of Computer Science, Hong Kong University of Science and Technology, China

ARTICLE INFO

Article history:
Available online 13 January 2011

Keywords:
Wireless sensor networks
Data aggregation
Latency

ABSTRACT

Due to the large-scale ad hoc deployments and wireless interference, data aggregation is a fundamental but time consuming task in wireless sensor networks. This paper focuses on the latency of data aggregation. Previously, it has been proved that the problem of minimizing the latency of data aggregation is NP-hard [1]. Many approximate algorithms have been proposed to address this issue. Using maximum independent set and first-fit algorithms, in this study we design a scheduling algorithm, Peony-tree-based Data Aggregation (PDA), which has a latency bound of $15R + \Delta - 15$, where R is the network radius (measured in hops) and Δ is the maximum node degree. We theoretically analyze the performance of PDA based on different network models, and further evaluate it through extensive simulations. Both the analytical and simulation results demonstrate the advantages of PDA over the state-of-art algorithm in [2], which has a latency bound of $23R + \Delta - 18$.

1. Introduction

A wireless sensor network (WSN) consists of a large number of sensor nodes which communicate with each other via their RF transceivers. Owing to the ad hoc deployments and self-organizing characteristics of sensors, WSNs are suitable for a wide variety of applications such as scientific observation, environmental monitoring, health care, and military surveillance [3–10]. In those applications, the operation of data aggregation is often used for querying information like event numbers.

Shortening the latency of data aggregation is a fundamental requirement in WSNs, especially in real-time applications [11]. In this paper, we focus on reducing the latency of data aggregation by designing a good schedule. This is a challenging issue mainly because of the intrinsic interference in WSNs. When two interfering signals are sent simultaneously, neither of them can be received correctly. Certainly, we can schedule the transmissions of data aggregation to avoid interference. The latency of data aggregation, however, will thus be increased inevitably. Without a good scheduling scheme, data aggregation will result in an impractically high latency.

Previously, it has been proved that the problem of minimizing the latency of data aggregation is NP-hard [1]. Many scheduling algorithms have been proposed to reduce the latency of data aggregation [1,2,12–14]. Chen et al. in [1] propose a data aggregation algorithm which has an approximation factor of $\Delta - 1$, where Δ is the maximum node degree. The scheduling algorithm proposed by Wan et al. in [2] has a latency bound of $23R + \Delta - 18$, where R is the network radius (measured in hops). The objective of our study in this paper targets approaching the optimal schedule of data aggregation. We design a scheduling algorithm for data aggregation which has a latency bound of $15R + \Delta - 15$.

Our proposed algorithm, Peony-tree-based Data Aggregation, or PDA in short, works as follows. Initially, a data aggregation tree called peony tree is constructed by using maximum independent set and first-fit algorithms. The data aggregation process mainly consists of two phases: local aggregation and global aggregation. Step (1), in local aggregation, raw data at the leaf nodes are transmitted to

* Corresponding author. Tel.: +86 15105695634.
E-mail address: wangpei@mail.ustc.edu.cn (P. Wang).

1570-8705/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2011.01.003

their corresponding parent nodes for aggregation. Step (2), in global aggregation, data gathered in local aggregation are further aggregated layer by layer in a bottom-up manner. The latency bounds of the two phases are $\Delta - 1$ and $15R - 14$, respectively. We also theoretically analyze the performance of PDA and deduce the latency bound based on a more generic network model, which does not assume equal communication and interference ranges.

The rest of this paper is organized as follows. In Section 2, we present the related works and briefly compare them with our scheme. Section 3 introduces the terminology, models, and assumptions in this paper as well as the problem formulation. In Section 4 we elaborate the scheduling algorithm of data aggregation. Section 5 presents the proofs of correctness and theoretical performance analysis, followed by the simulations results in Section 6. We conclude the work in Section 7.

2. Related works

The problem of data aggregation is a well investigated problem in the field of wireless sensor networks [15–23]. For improving the real-time ability of wireless sensor networks, it is important to design a low latency data aggregation scheduling. Many scheduling algorithms have been proposed [1,2,12–14]. Theoretically, Chen et al. in [1] prove that the problem of minimizing the latency of data aggregation is NP-hard. They also design an algorithm for data aggregation which has an approximation factor of $\Delta - 1$, where Δ is the maximum node degree. As the latency of data aggregation is at least R (the network radius measured in hops), the algorithm proposed in [1] yields a multiplicative result of Δ and R. Zheng and Barton [12] prove that the optimal data aggregation rate is $\Theta(log(N)/N)$, where N is the number of nodes in the network, when operating in fading environments with power path-loss exponents that satisfy $2 < \alpha < 4$. Ye et al. [13] study the optimal forwarding instants and model the problem as a decision process that determines the optimal decision policies at the sensor nodes. Kesselman and Kowalski [14] design an algorithm which has a latency bound of $O(log(N))$. They assume that each node can learn the distance to the closest neighbor and has a special collision detection capability, although such conditions cannot always be guaranteed in WSNs.

Among all the existing works, the proposal of Huang et al. in [2] is the closest to our work. They propose a scheduling algorithm based on maximum independent set, which has a latency bound of $23R + \Delta - 18$. They mainly focus on a special scenario, however, in which the communication range equals the interference range. Huang's results are the state-of-arts on scheduling algorithms of data aggregation.

Compared with the previous studies, the contributions of this work are as follows:

- We design a new scheduling algorithm which minimizes the number of connective nodes (the details are presented in Section 4.1) and reduces the aggregation latency. It has a latency bound of $15R + \Delta - 15$, which is the lowest by far, to the best of our knowledge.
- We address the issue of scheduling data aggregation based on a more generic network model, in which the assumption of equal communication range and interference range is removed.

3. Preliminaries

3.1. Network model

We consider a WSN consisting of one sink node s and N sensor nodes. Every sensor node is equipped with only one radio. All nodes share a common wireless channel to communicate. We also assume omni-directional antennas and unit-disk model. A node is only able to communicate with nodes within the communicate range r_c and be interfered by nodes within the interference range r_i. A node u can receive data correctly if and only if there is exactly one node sending data in the interference range r_i of u. Otherwise, collision happens and node u cannot receive data correctly. Our algorithm has no assumptions on r_c and r_i. For ease of presentation, we first assume $r_c = r_i$ when elaborating the algorithm in Section 4. We will remove this assumption in Section 5.

As for the process of data aggregation, we partition time into slots. A node can either send or receive a data packet in a time slot. The latency of data aggregation in this paper refers to the number of time slots passed during the entire course of data aggregation. Consider a sending node set A and its corresponding receiving node set B. We say data can be aggregated from A to B in a time slot if and only if all nodes in B can receive data correctly when all nodes in A transmit data simultaneously.

Now we formulate the problem of data aggregation. We use an undirected graph $G = (V,E)$ to represent a WSN of N nodes, where V denotes the set of nodes and E denotes the set of edges. There is an edge between node u and node v if and only if the Euclidean distance between them is not greater than r_c. A feasible data aggregation schedule is a sequence of node-disjoint sending sets $\{S_1, S_2, \ldots, S_K\}$, where $\bigcup_{i=1}^{i=K} S_i = V - \{s\}$. All nodes in S_i can transmit data simultaneously without interference in the ith time slot. After all nodes in S_K finish transmitting data in the Kth time slot, all data are aggregated to the sink node. That is, the latency of data aggregation with the above schedule is K. Denote the corresponding receiving set of S_i by R_i. It can be seen that $R_i \subseteq (\bigcup_{l=i+1}^{l=K} S_l) \bigcup \{s\}$.

3.2. Independent set and concurrent set

Node u and node v are independent if and only if they are not within the interference range of each other. If all nodes in X are independent from each other, we call node set X is an independent set. When node u and node v are not independent from each other, we call u and v cover each other. We will use independent set to select dominative nodes in the aggregation tree.

Consider two links $l_1{:}p \to q$ and $l_2{:}u \to v$ where p, q, u and v are four nodes. Link l_1 and link l_2 are independent if and only if p is independent from v and q is independent from u. When link l_1 and link l_2 are independent, node p

and node u can transmit simultaneously without interference.

Let $L = (V_S, V_R, f)$ be a link set, where V_S is the sending set, V_R is the receiving set, and f is a mapping function from V_S to V_R. If all links in L are independent from each other, link set L is an independent link set. Thus, transmissions through the links in an independent link set can be executed simultaneously without interference. Here we call the sending set V_S as a concurrent set. Later in this paper, we will use concurrent set to divide the nodes into multiple sets and correspondingly schedule data transmissions into different time slots.

For easy statement, we list important symbols used in Table 1.

4. Peony-tree-based Data Aggregation (PDA)

PDA mainly consists of three componentsPDA mainly consists of three steps, peony tree construction, local aggregation, and global aggregation.

4.1. Peony tree construction

We construct a tree called peony tree for data aggregation, which includes all the nodes in the network. First, an approximately maximum independent set is formed to cover all the nodes. We call nodes in the maximum independent set as dominative nodes. Some additional nodes are then selected to connect dominative nodes so that all dominative nodes are reachable from the sink node. The selected nodes are called connective nodes. Each of the other nodes in the network selects a neighboring dominative node as its parent, and the peony tree is constructed. A node which is neither a dominative node nor a connective node is called a white node. Therefore we have three types of nodes in total. Different from the algorithm proposed in [2] which selects connective nodes arbitrarily, we try to select the minimum number of connective nodes to avoid collision and reduce the aggregation latency.

4.1.1. Dominative nodes selection

Algorithm 1. Maximum Independent Set Construction

Input: A communication graph $G = (V,E)$.
Output: A maximum independent set D of G.

```
Construct a breadth first search tree T_BFS;
Divide all nodes on T_BFS into layers L_0,L_1,L_2,...,L_R;
D = φ;
for i = 0 to R do
    while L_i ≠ φ do
        Select a node u ∈ L_i;
        if u ∉ {p|p ∈ N(v), v ∈ D} then
            D = D ∪ {u};
            L_i = L_i − {u} − N(u);
            L_{i+1} = L_{i+1} − {u} − N(u);
        else
            L_i = L_i − {u};
```

Table 1
Terms and notations.

Notations	Descriptions
u,v,p,q	node
l_1,l_2	link
K	aggregation latency delay
$S_1,\ldots,S_i,\ldots,S_K$	concurrent set during slot i,$1 \leqslant i \leqslant K$
Δ	Maximum node degree
R	Depth of breadth first search tree
$N(u)$	u's neighbor nodes
$P(u)$	u's parent node peony tree search tree

Initially, we construct a common breadth first search tree T_{BFS} rooted at the sink node s. Denote the depth of T_{BFS} by R. All nodes in the network are thus divided into layers $L_0,L_1,\ldots,L_R$, according to their depths in T_{BFS}. $L_0 = \{s\}$. L_i exactly contains nodes in layer i of T_{BFS}.

We denote the maximum independent set by D. At the beginning, $D = \phi$. To minimize the aggregation latency, dominative nodes in D are selected layer by layer in a top-down manner. For a node u in L_i, if u is independent from all the other nodes in D, u will be added into D, u and its neighbors are removed from all layers as well. Otherwise, u will be removed from L_i. Repeat this procedure until L_i is empty. When there is no node in L_i, continue the procedure in L_{i+1}. After the procedure finishes at L_R, the resulting set D is a maximum independent set, *i.e.* the dominative nodes are selected. The corresponding pseudo code is given in Algorithm 1, where $N(u)$ is the neighbor nodes set of u. After every node broadcast its ID in "hello" messages, any node u can get its neighbor nodes $N(u)$.

4.1.2. Determination of connective nodes and white nodes

The connective nodes and dominative nodes together constitute the backbone of the network. According to the selection of dominative nodes, a connective node may cover one or more dominative nodes. When connective nodes are selected arbitrarily [2], the number of connective nodes are often more than necessary, reducing the rate of data aggregation. Based on such observation, we select minimum number of connective nodes by using first-fit algorithm.

Denote the peony tree by $T = (V,E'\}$, where V is the node set and E' is the edge set. We may divide the dominate nodes into layers D_0, $D_1,\ldots$, D_R according to their hops to the sink node. For a dominative node u in D_i, select a node v in L_{i-1}, which is a neighbor node of u. Node v is set as the parent of all the dominative nodes in D_i which cover v. Remove all the children of node v from D_i. Then a dominative node p in D_{i-1} which covers v is set as the parent of v. Repeat this procedure until D_i is empty. When there is no node in D_i, repeat the procedure in D_{i-1}. The procedure continues until no dominative node remains in D_2. At last, set s as the parent of all connective nodes in L_1. Thus all the connective nodes are selected.

Those nodes which are neither dominative nodes nor connective nodes are set as white nodes. Denote the set of white nodes by W. For each white node w in W, a dominative node is selected as its parent, which has the lowest depth among all the dominative nodes covering w.

Algorithm 2. Peony tree construction

Input: A communication graph $G = (V,E)$, the set of dominative nodes D.
Output: Peony tree $T = (V,E^{'})$ on G, the set of connective nodes C, the set of white nodes W.
Divide all nodes in D into layers $D_0, D_1, \ldots, D_R$ on T_{BFS};
$E^{'} = \phi, C = \phi, W = V - D$;
for $i = R$ to 2 **do**
  while $D_i \neq \phi$ **do**
    Select a dominative node u in D_i;
    Select a node $v \in \{v|v \in N(u), v \in L_{i-1}\}$;
    $C = C \bigcup \{v\}$;
    $E^{'} = E^{'} \bigcup \{l{:}q \leftrightarrow v|q \in D_i, q \in N(v)\}$;
    $D_i = D_i - \{q|q \in D_i, q \in N(v)\}$;
    $W = W - \{v\}$;
    Select a node $p \in \{q|q \in D_{i-1}, q \in N(v)\}$;
    $E^{'} = E^{'} \bigcup \{l{:}v \leftrightarrow p\}$;
$E^{'} = E^{'} \bigcup \{l{:}s \leftrightarrow q|q \in C, q \in L_1\}$;
for Each node in $w \in W$ **do**
  Select a node $u \in \{u|u \in D, u \in N(w)$, $Depth(u) = min\{Depth(v)|v \in D, v \in N(w)\}\}$;
  $E^{'} = E^{'} \bigcup \{l{:}u \leftrightarrow w\}$;

The pseudo code of peony tree construction is presented in Algorithm 2.

Given the depth of T_{BFS} as R, according to the construction of peony tree, there are both connective nodes and dominative nodes in each layer of T_{BFS}, except in layer 0, layer 1, and Layer R. Specifically, sink node s is the unique dominative node in layer 0. There are no dominative nodes in layer 1. There are no connective nodes in layer R of T_{BFS}, either. Consequently, the depth of the peony tree is $2(R - 2) + 1 + 1 + 1 = 2R - 1$.

4.2. Local aggregation

Algorithm 3. Maximum concurrent sets construction

Input: Link set $L = (V_S, V_R, f)$.
Output: Node-disjoint maximum concurrent sets $U_1, U_2, \ldots, U_h$.
$i = 1$;
while $V_S \neq \phi$ **do**
  $U_i = \phi$;
  for each node $u \in V_S$ **do**
    if $f(u) \notin \{w|w \in N(v), v \in U_i\}$ and $u \notin \{w|w \in N(f(v)), v \in U_i\}$ **then**
      $U_i = U_i \bigcup \{u\}$;
      $V_S = V_S - \{u\}$;
  $i = i + 1$;

During the local aggregation phase, white nodes transmit the sensory data to their parents (the corresponding dominative nodes) for aggregation. Meanwhile, a dominative node takes aggregation operations when all its children nodes have transmitted data.

Because transmissions on dependent links should transmit data in different time slots, to avoid interference and minimize the latency in local aggregation, the set of white nodes W is divided into k node-disjoint maximum concurrent sets. Denote these node-disjoint maximum concurrent sets by $UW_i (i = 1, 2, \ldots, k)$, where k is the number of the node-disjoint maximum concurrent subsets of W. Nodes in UW_i can transmit data without interference in the ith time slot. Then the latency of data aggregation in the phase of local aggregation is k. UW_i $(i = 1, 2, \ldots, k)$ are iteratively determined by using first-fit algorithm.

Algorithm 3 is the pseudo code for constructing node-disjoint maximum concurrent sets. Consider a link set $L = (V_S, V_R, f)$, where V_S is sending node set, V_R is its corresponding receiving node set, and f is a mapping function from V_S to V_R. Denote all maximum concurrent sets by $U_1, U_2, \ldots, U_h$, where h is the number of maximum concurrent sets. For each node u in V_S, if link $l{:}u \to f(u)$ is independent from all links in $\{l{:}v \to f(v)|v \in U_i\}$, add u into U_i and remove u from V_S. When there is no more node to add into U_i, repeat the above procedure to form the next maximum concurrent set U_{i+1} until no node remains in V_S.

4.3. Global aggregation

Global aggregation involves two types of transmissions. One is from a dominative node to its parent connective node, and the other is from a connective node to its parent dominative node. Global aggregation starts with the transmission from the dominative nodes in L_R to their parent connective nodes in L_{R-1}. The above two types of transmissions run alternately until the data are aggregated from the connective nodes in layer 1 to the sink node.

To avoid interference and minimize the latency of global aggregation, we divide dominative nodes and connective nodes in each layer of T_{BFS} into node-disjoint maximum concurrent sets. Denote these node-disjoint maximum concurrent sets by $UD_{i,1}, UD_{i,2}, \ldots, UD_{i,m_i}$ and $UC_{i,1}, UC_{i,2}, \ldots, UC_{i,n_i}$ respectively, where m_i is the number of maximum concurrent subsets of D_i and n_i is the number of maximum concurrent subsets of C_i. Specially, s is sink node and the unique one in layer 0, $m_0 = 0$. Because there is no dominative nodes in layer 1, $m_1 = 0$. Similarly, we have $n_R = 0$. All node-disjoint maximum concurrent sets are selected layer by layer in a bottom-up manner by first-fit algorithm as shown in Algorithm 3.

Similarly with local aggregation, we allocate different time slots for different maximum concurrent sets. Data are aggregated layer by layer in a bottom-up manner. A node transmits data after all its children transmit data. Due to the fact that the latency at local aggregation phase is k, we schedule all dominative nodes and connective nodes as follows. Connective nodes in $UC_{i,j}$ transmit data in the $\left(\sum_{l=R}^{l=i+1}(m_l + n_l) + j + k\right)$ th time slot. Dominative nodes in $UD_{i,j}$ transmit data in the $\left(\sum_{l=R}^{l=i+1}(m_l + n_l) + n_i + j + k\right)$th time slot. We can imply that the latency of global aggregation is $\sum_{l=R}^{l=1}(m_l + n_l)$ slots.

4.4. An example

Fig. 1 presents an example of aggregation tree construction. Fig. 1a shows the original topology of a WSN. Fig. 1b shows the selected dominative nodes, as well as the covering relation between the dominative nodes and non-dominative nodes. Fig. 1c shows an aggregation tree constructed by the algorithm proposed in [2], which selects connective node arbitrarily. There are five connective nodes in Fig. 1c. On the same network topology, Fig. 1d shows a peony tree constructed by the proposed PDA, which has only three connective nodes. Further, using the same aggregation rule mentioned above, the data aggregation latency of Fig. 1d is 11, while the latency of Fig. 1d is 9. The number attached to each node in Fig. 1c and d indicates the sequence number of the time slot allocated for the node to transmit data. Clearly, our algorithm effectively reduces the number of connective nodes and the latency of data aggregation.

5. Discussion

In this section, we first prove the latency bound of our algorithm based on the network model described in Section 3.1 with the assumption $r_c = r_i$, and then we remove the constraint and deduce the latency bound based on a generic network model.

5.1. Correctness

Theorem 1. *Consider data transmissions through a link set $L = (V_S, V_R, f)$, where V_S is set of node sending data, V_R is set of node receiving data and f is the mapping function from V_S to V_R. G is the undirected graph which consists of V_S and V_R. The latency bound of transmitting from V_S to V_R equals to the maximum node degree of nodes in V_R on G.*

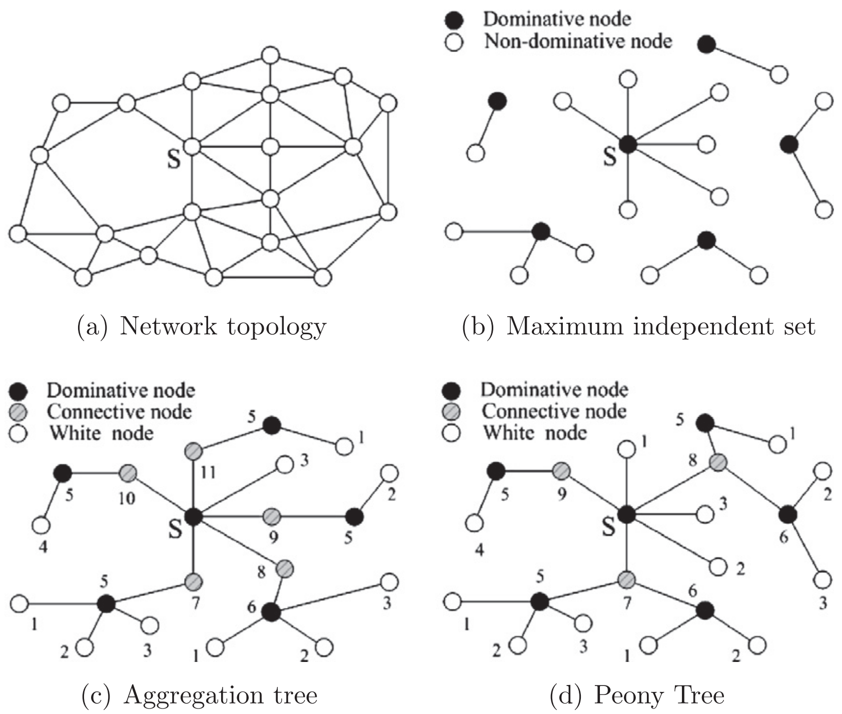
Proof. To minimize the latency and avoid the interference, divide all nodes in V_S into several node-disjoint maximum concurrent sets. Denote the maximum node degree of nodes in V_R on G by Δ. We use V_u to denote the set of sending nodes which are neighbors of node u and M_u to denote the number of sending nodes in V_u that have not transmitted data.

First we prove that M_u decreases by at least 1 after each allocated time slot. Suppose w is a sending node in V_u. Let U_c be the maximum concurrent set for the current time slot.

1. If link $w \to f(w)$ is independent from all links in $\{v \to f(v) | v \in U_c\}$, w can be added in the current maximum concurrent set U_c and can transmit data in the time slot allocated for U_c. Then M_u decreases by 1 after the slot.
2. Otherwise, according to the definition of concurrent set, there must be another sending neighbors of u which are transmitting data. M_u also decreases by at least 1 after the current time slot. □

To summarize (1) and (2), it requires at most Δ time slots for the nodes in V_u to transmit data to V_R. The latency bound of transmitting from V_S to V_R equals to Δ.

Theorem 2. *The latency bound of local aggregation is $\Delta - 1$.*



(a) Network topology (b) Maximum independent set (c) Aggregation tree (d) Peony Tree

Fig. 1. Peony tree construction.

Proof. Given the maximum node degree Δ, a dominative node has at most Δ neighbors and one of them must be a connective node. Thus at most $\Delta - 1$ neighbors of a dominative node u are white nodes. According to Theorem 1, the latency bound of local aggregation is $\Delta - 1$. □

Theorem 3. *The latency bound of global aggregation is $15R - 14$.*

Proof. Recall that in a peony tree, dominative nodes are in the even layers while connective nodes are in the odd layers, so there are two transmission patterns in global aggregation. One is data transmission from a dominative node to a connective node, and the other is data transmission from a connective node to a dominative node.

First, we prove that the latency bound of data aggregation from dominative nodes in an even layer to connective nodes in the upper layer in peony tree is 4. Note that a connective node has at most five dominative neighbors as illustrated in Fig. 2. One of the dominative neighbors is the parent of the connective node, so the number of dominative children of a connective node is at most 4. According to Theorem 1, the latency bound of data aggregation from dominative nodes in an even layer to connective nodes in the upper layer is 4.

Second, we show that the latency bound of data aggregation from connective nodes in an odd layer to dominative nodes in the upper layer in peony tree is 11. Because a dominative node has at most 12 connective neighbors as illustrated in Fig. 3. One of them is the parent node of the dominative node, so the number of connective children of a dominative node is at most 11. According to Theorem 1, the latency bound of data aggregation from connective nodes in an odd layer to dominative nodes in the upper layer is 11. Specially, the sink node is a dominative node without parent. The latency bound of aggregation from connective nodes in layer 1 to the sink node is 12.

In the peony tree, the number of layers which contain connective nodes is $R - 1$. The number of layers which contain dominative nodes except sink node s is also $R - 1$. Basing on the above analysis, the latency bound of global aggregation is $11(R - 2) + 12 + 4(R - 1) = 15R - 14$. □

Theorem 4. *The latency bound of data aggregation on a peony tree is $15R + \Delta - 15$.*

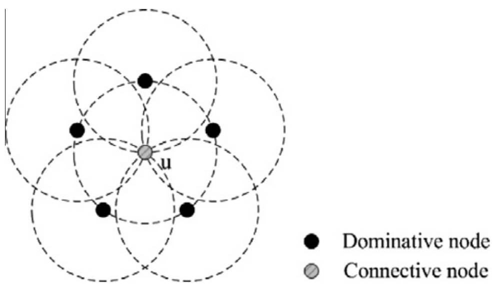


Fig. 2. The maximum number of dominative neighbor is 5.

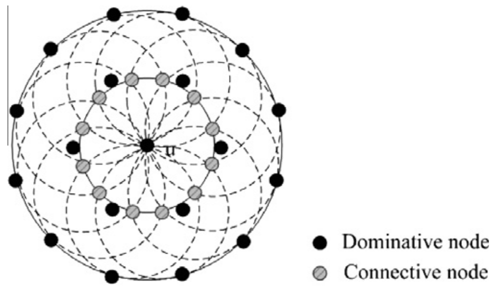


Fig. 3. The maximum number of connective neighbor is 12.

Proof. According to our scheduling algorithm, the total latency of data aggregation is the sum of latencies of local aggregation and global aggregation. According to Theorems 2 and 3, the latency bound of data aggregation using our algorithm is $(\Delta - 1) + (15R - 14) = 15R + \Delta - 15$. □

Theorem 5. *The computational complexity of PDA is $O(N^2 + \Delta N)$, where N is the number of nodes and Δ is the maximum node degree.*

Proof. Let us review the construction of PDA. Initially, PDA construct a breadth first search tree for construction. A dominative node set is formed by maximum independent node set and first-fit algorithm. Thereby, connective node set is selected carefully to construct peony tree. Lastly, maximum concurrent sets are calculated for data aggregation scheduling.

As the neighbor information $N(u)$ is used for node u, we first analyze the computational complexity of forming $N(u)$. $N(u)$ can be gotten by scanning every node v in the network and check whether v is u's neighbor. In this way, the neighbor information of all nodes can be calculated in $O(N^2)$. □

Nodes in PDA is initialized in the form of a breadth first search tree, which can be consturcted in $O(N^2)$.

According to Algorithm 1, dominative nodes are constructed by maximum independent set and first-fit algorithm. Because every node is exactly accessed once, its computational complexity is $O(N)$.

Peony tree is constructed by Algorithm 2. Every dominative node selects a connective node. Because dominative nodes, connective nodes and white nodes are exactly accessed once, its computational complexity is also $O(N)$.

The data aggregation scheduling is calculated by Algorithm 3. Because there is at least one node sending packet during a slot and a sending node can only be interfered by at most Δ neighbors of receiving node. The computational complexity in this step is $O(N \times \Delta)$.

Consequently, the computational complexity of PDA is $O(N^2) + O(N^2) + O(N) + O(N) + O(N \times \Delta) = O(N^2 + \Delta N)$

5.2. On generic model

Most existing works base their schemes on the assumption $r_c = r_i$, so far we also discuss based on this assumption. Indeed, our scheme is not constrained by the assumption, and in this section we remove it.

As data aggregation is conducted layer by layer, nodes suffer interference only from nodes in the same layer when they transmit data simultaneously. As mentioned above, there are three types of transmissions during the course of data aggregation, below we will discuss separately. For simplicity, we normalize the communication range $r_c = 1$ and the interference range $r_i = \rho$.

Denote the upper bound of the number of independent nodes in a radius of ρ by $\Gamma(\rho)$. Before deducing the latency of data aggregation, we first present two theorems related to $\Gamma(\rho)$.

Theorem 6. *(Wegner Theorem [24]) The area of the convex hull of $n(n \geqslant 2)$ non-overlapping unit-radius circular disks is at least*

$$2\sqrt{3}(n-1) + (2-\sqrt{3})\left\lceil\sqrt{12n-3}-3\right\rceil + \pi \qquad (1)$$

Theorem 7. *$\Gamma(\rho)$, the upper bound of the number of independent nodes in a radius of ρ, satisfies the condition that*

$$2\sqrt{3}(\Gamma(\rho)-1) + \left(2-\sqrt{3}\right)\lceil\sqrt{12\Gamma(\rho)-3}-3\rceil + \pi \leqslant (2\rho+1)^2\pi \qquad (2)$$

Proof. Consider independent nodes in a disk of radius ρ centered at point o as shown in Fig. 4. Place a small disk with radius 0.5 centered at each independent node. Because the distance between any two independent nodes is greater than 1 ($r_c = 1$), all the small disks are non-overlapping. According to Theorem 6, the area of the convex hull which covers all the small disks of radius 0.5 is at least $\left(2\sqrt{3}(\Gamma(\rho)-1) + \left(2-\sqrt{3}\right)\lceil\sqrt{12\Gamma(h)-3}-3\rceil + \pi\right)/4$. Because all the small disks are in a big disk of radius $h + 0.5$ centered at point o, the convex hull which covers all the small disks is also in the big disk. The area of the convex hull is not greater than that of the big disk, so $\Gamma(\rho)$ satisfies the in Eq. (2). □

Now we deduce the latency of data aggregation on generic model.

1. *Transmissions from white nodes to dominative nodes*
   When $\rho \leqslant 1$, a dominative node receives data from at most $\Delta - 1$ white neighbors. Hence, the latency bound of transmissions from white nodes to dominative nodes is still $\Delta - 1$.
   When $\rho > 1$, a dominative node can be interfered by all the white nodes in the interference range, no matter whether or not they are neighbors. Thus the latency bound is affected by the maximum number of white nodes in a radius of ρ. As there are at most $\Gamma(\rho)$ dominative nodes in a radius of ρ and a dominative node has at most $\Delta - 1$ white neighbors, the latency bound of transmissions from white nodes to dominative nodes is $\Gamma(\rho)(\Delta - 1)$.
2. *Transmissions from dominative nodes to connective nodes*
   For a connective node u, there are at most $\Gamma(\rho)$ dominative nodes in a radius of ρ and one of them is its parent. Consequently, the latency bound of transmissions from dominative nodes to connective nodes is $\Gamma(\rho) - 1$.
3. *Transmissions from connective nodes to dominative nodes*
   Because a connective node connects dominative nodes in the adjacent layer, the number of connective nodes in a radius of ρ is not more than the number of dominative nodes in a radius of $1 + \rho$. Hence, a dominative node has at most $\Gamma(1 + \rho)$ connective neighbors. Meanwhile, one of those connective nodes is the parent of the dominative node. The latency bound of transmissions from connective nodes to dominative nodes is $\Gamma(1 + \rho) - 1$. Specially, the latency from connective nodes in layer 1 to the sink node is at most $\Gamma(1 + \rho)$.

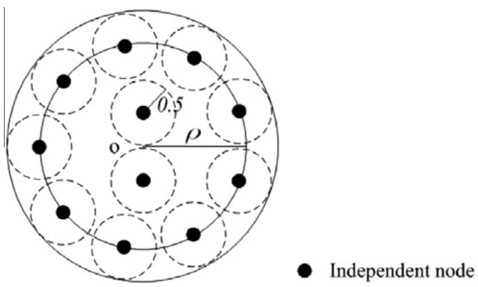


Fig. 4. The maximum number of independent node in a radius of ρ.

Theorem 8. *The latency bound of data aggregation in a peony tree is*

$$(\Gamma(\rho) + \Gamma(1+\rho) - 2)(R-2) + \Gamma(\rho)\Delta + \Gamma(1+\rho) - 1 \qquad (3)$$

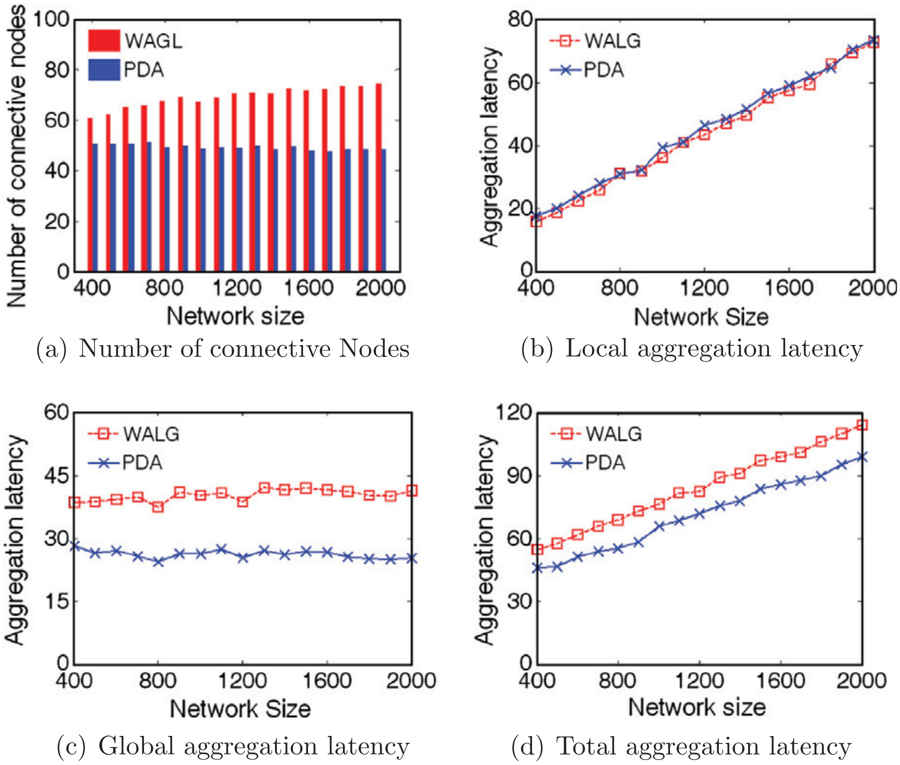
Proof. Recall that there are no connective node in layer R and no dominative node in layer 1 of the tree. The latency from connective nodes in layer 1 to sink node is $\Gamma(1 + \rho)$. So the latency bound of data aggregation on peony tree is $\Gamma(\rho)(\Delta - 1) + (\Gamma(\rho) - 1) + ((\Gamma(\rho) - 1) + (\Gamma(1 + \rho) - 1))(R - 2) + \Gamma(1 + \rho) = (\Gamma(\rho) + \Gamma(1 + \rho) - 2)(R - 2) + \Gamma(\rho)\Delta + \Gamma(1 + \rho) - 1$. □

6. Performance evaluation

In the simulations, we let all sensor nodes have the identical communication range r_c and identical interference range r_i. For comprehensive evaluation, r_c and r_i are tuned in different groups of simulations. Sensor nodes are randomly deployed into a 100 m × 100 m area for 20 times to generate different network topologies. We compare PDA with the algorithm proposed by Huang et al. in [2] (denoted as WALG in short in the figures) by measuring the number of time slots to aggregate all the sensory data in a WSN.

6.1. Impact of network size

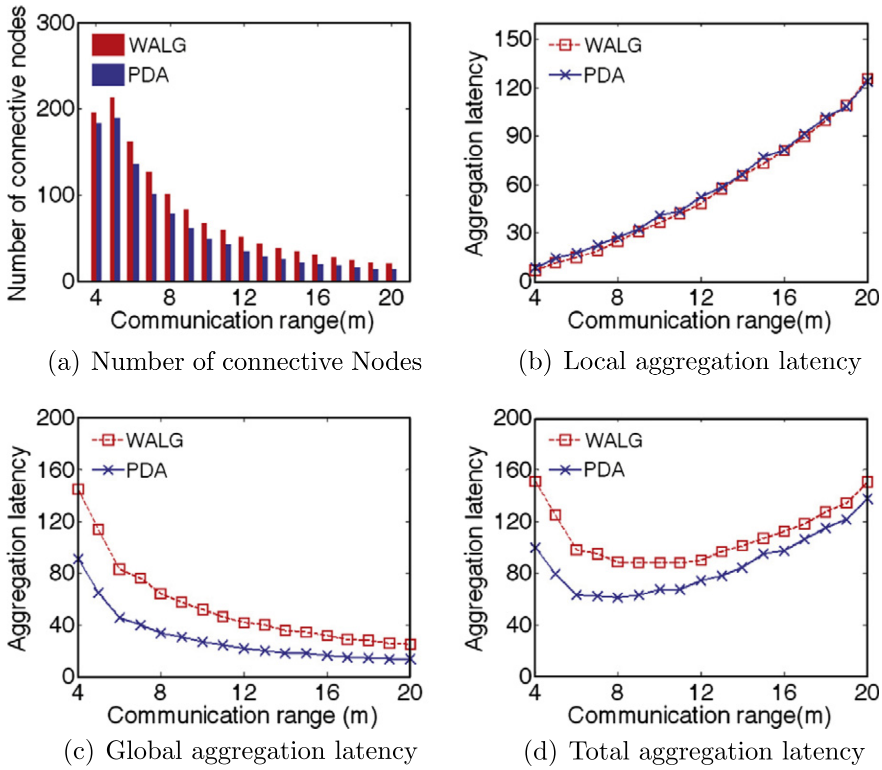
The first group of simulations examines the impact of network size. Let $r_c = r_i = 10$ m, and 20 different topologies are generated, with network sizes varying from 400 to 2000.



(a) Number of connective Nodes (b) Local aggregation latency

(c) Global aggregation latency (d) Total aggregation latency

Fig. 5. Comparison with different network sizes.



(a) Number of connective Nodes (b) Local aggregation latency

(c) Global aggregation latency (d) Total aggregation latency

Fig. 6. Comparison with different communication ranges.

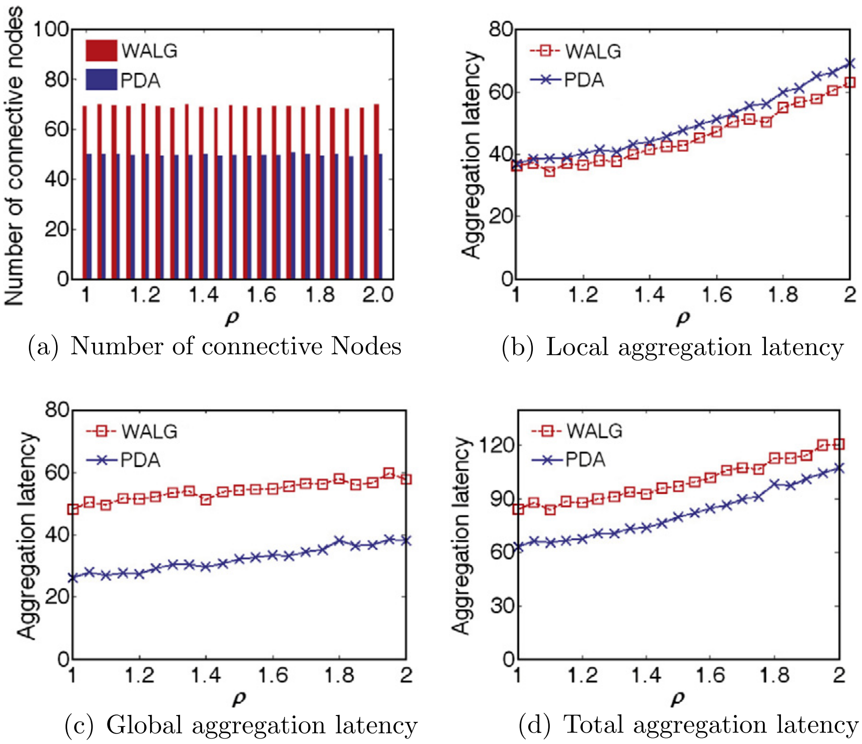


Fig. 7. Comparison with different ratios of interference range to communication range.

Fig. 5a plots the number of connective nodes selected by PDA and WALG (note that the numbers of dominative nodes selected by the two algorithms are same, so we do not show them in the figures). We can see the number of connective nodes selected by PDA is much less than that of WALG. Also, WALG is slightly affected by the network size, while PDA performs consistently with different network sizes for the reason that PDA choose connective nodes carefully.

Fig. 5b and c compare the latencies of local aggregation and global aggregation by using PDA and WALG. For both algorithms, the local aggregation latency linearly increases with the network size, while the global aggregation latency nearly remains constant. Moreover, the global aggregation latency by using PDA is about 25% shorter than that of using WALG, because PDA selects less connective nodes.

Overall, the total aggregation latency by using PDA is about 20% shorter than that of using WALG, as shown in Fig. 5d.

6.2. Impact of communication range

We now fix the network size to 1000 and $r_c = r_i$. The aggregation latency is measured when r_c varies from 4 m to 20 m. Because less dominative nodes are selected with a larger communication range, the number of connective nodes apparently decreases when r_c increases, as shown in Fig. 6a. As a result, the global aggregation latency also decreases, as shown in Fig. 6c. Similar with the first group of simulations, we can see performance gaps between PDA and WALG from the results shown in Fig. 6b–d, respectively.

6.3. Impact of the ratio of interference range to communication range

The ratio of the interference range to the communication range is denoted by ρ. In the last group of simulations, the network size is fixed to 1000, and $r_c = 10$ m. $r_i = r_c \times \rho$. The aggregation latency is measured when ρ varies from 1 to 2. Since r_c keeps constant, the number of connective nodes in the network keeps nearly unchanged. Meanwhile, PDA selects much less connective nodes than WALG, as shown in Fig. 7a.

According to Theorem 8, when the interference range increases, both the local aggregation latency and the global aggregation latency increase, as shown in Fig. 7b–d.

7. Conclusion

We study the issue of minimizing the latency of data aggregation in WSNs and propose a scheduling algorithm named PDA. PDA has a latency bound of $15R + \Delta - 15$, where R is the network radius (measured in hops) and Δ is the maximum node degree. To the best our knowledge, this latency bound is by far the lowest one. Moreover, we remove the constraint of equal communication and interference ranges used in most existing works and discuss the latency bound of PDA based on a generic network model. We compare PDA with the state-of-arts algorithm proposed in [2] through simulations and the results show that PDA significantly outperforms existing designs.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.adhoc.2011.01.003.

References

[1] X. Chen, X. Hu, J. Zhu, Minimum data aggregation time problem in wireless sensor networks, in: Proceedings of MSN.
[2] S.C.-H. Huang, P.-J. Wan, C.T. Vu, Y. Li, F.F. Yao, Nearly constant approximation for data aggregation scheduling in wireless sensor networks, in: Proceedings of IEEE INFOCOM.
[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (2002).
[4] M. Li, Y. Liu, Underground structure monitoring with wireless sensor networks, in: Proceedings of ACM/IEEE IPSN.
[5] R. Szewczyk, A.M. Mainwaring, J. Polastre, J. Anderson, D.E. Culler, An analysis of a large scale habitat monitoring application, in: Proceedings of ACM SenSys.
[6] T. He, S. Krishnamurthy, J.A. Stankovic, T.F. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, B.H. Krogh, Energy-efficient surveillance system using wireless sensor networks, in: Proceedings of ACM MobiSys.
[7] M. Li, Y. Liu, Iso-map: Energy-efficient contour mapping in wireless sensor networks, IEEE Transactions on Knowledge and Data Engineering (TKDE) 22, May 2010.
[8] Y.L. Mo Li, L. Chen, Non-threshold based event detection for 3d environment monitoring in sensor networks, IEEE Transactions on Knowledge and Data Engineering (TKDE) 20, December 2008.
[9] Y.L.M.L.Z.G.F.H. Kebin Liu, Mo Li, Passive diagnosis for wireless sensor networks, in: Proceedings of ACM SenSys.
[10] Y.L.J.Z.S.T.X.-Y.L.G.D. Lufeng Mo, Yuan He, Canopy closure estimates with greenorbs: Sustainable sensing in the forest, in: Proceedings of ACM Sensys.
[11] L. Klingbeil, T. Wark, A wireless sensor network for real-time indoor localisation and motion monitoring, in: Proceedings of ACM/IEEE IPSN.
[12] R. Zheng, R.J. Barton, Toward optimal data aggregation in random wireless sensor networks, in: Proceedings of IEEE INFOCOM.
[13] Z. Ye, A. Abouzeid, J. Ai, Optimal policies for distributed data aggregation in wireless sensor networks, in: Proceedings of IEEE INFOCOM.
[14] A. Kesselman, D.R. Kowalski, Fast distributed algorithm for convergecast in ad hoc geometric radio networks, Journal of Parallel and Distributed Computing 66 (2006).
[15] S. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong, Tag: a tiny aggregation service for ad-hoc sensor networks, in: Proceedings of USENIX OSDI.
[16] J. Considine, F. Li, G. Kollios, J.W. Byers, Approximate aggregation techniques for sensor databases, in: Proceedings of ICDE.
[17] S. Nath, P.B. Gibbons, S. Seshan, Z.R. Anderson, Synopsis diffusion for robust aggregation in sensor networks, ACM Transactions on Sensor Networks 4 (2008).
[18] A. Manjhi, S. Nath, P.B. Gibbons, Tributaries and deltas: efficient and robust aggregation in sensor network streams, in: Proceedings of SIGMOD.
[19] J. Benson, T. O'Donovan, U. Roedig, C. Sreenan, Opportunistic aggregation over duty cycled communications in wireless sensor networks, in: Proceedings of IPSN.
[20] A. Anandkumar, L. Tong, A. Swami, A. Ephremides, Minimum cost data aggregation with localized processing for statistical inference, in: Proceedings of IEEE INFOCOM.
[21] B. Yu, J. Li, Y. Li, Distributed data aggregation scheduling in wireless sensor networks, in: INFOCOM 2009, IEEE.
[22] J. Shi, R. Zhang, Y. Liu, Y. Zhang, Prisense: privacy-preserving data aggregation in people-centric urban sensing systems, in: Proceedings of IEEE INFOCOM.
[23] C. Joo, J.-G. Choi, N. Shroff, Delay performance of scheduling with data aggregation in wireless sensor networks, in: Proceedings of IEEE INFOCOM.
[24] G. Wegner, Über endliche kreispackungen in der ebene, Studia Scientiarium Mathematicarium Hungarica 21 (1986).

Pei Wang received his B.S. degree in computer science from University of Science and Technology of China in 2005. His current research interests include data aggregation and opportunistic routing in wireless sensor networks.

Yuan He received his BE degree in University of Science and Technology of China in 2003, his ME degree in Institute of Software, Chinese Academy of Sciences in 2006, and his PhD degree in Hong Kong University of Science and Technology. He now works as a PostDoc Fellow with Prof. Yunhao Liu in the Department of Computer Science and Engineering at Hong Kong University of Science and Technology. His research interests include sensor networks, peer-to-peer computing, and pervasive computing. He is a student member of the IEEE, the IEEE Computer Society, and ACM.

Liusheng Huang received the M.S. degree in computer science from University of Science and Technology of China in 1988. He is currently a professor and Ph.D. supervisor of the Department of Computer Science and Technology at University of Science and Technology of China. His research interests are in the areas of wireless sensor network, distributed computing, and information security. He is the senior membership of China Computer Federation.