Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 502381, 12 pages
http://dx.doi.org/10.1155/2013/502381

Research Article

Layer-Based Data Aggregation and Performance Analysis in Wireless Sensor Networks

Hongju Cheng,[1] Yongzhao Chen,[1] Naixue Xiong,[2] and Feifei Li[1]

[1] *College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China*
[2] *School of Computer Science, Colorado Technical University, Colorado, CO 80907, USA*

Correspondence should be addressed to Naixue Xiong; nxiong@coloradotech.edu

Received 3 June 2013; Accepted 24 July 2013

Academic Editor: Bin Wang

Due to the similarity and correlation among sensed data in wireless sensor network, it is an important way to reduce the number of packets transmitted with data aggregation technology so as to prolong the network lifetime. However, data aggregation is still a challenging issue since quality-of-service, such as end-to-end delay, is generally considered as a severe criterion required in many applications. We focus on the minimum-latency data aggregation problem and proposed a new efficient scheme for it. The basic idea is that we first build an aggregation tree by ordering nodes into layers, and then we proposed a scheduling algorithm on the basis of the aggregation tree to determine the transmission time slots for all nodes in the network with collision avoiding. We have proved that the upper bound for data aggregation with our proposed scheme is bounded by $(15R + \Delta - 15)$ for wireless sensor networks in two-dimensional space. Extensive simulation results have demonstrated that the proposed scheme has better practical performance compared with related works.

1. Introduction

Recent advances in microelectronics, low-power embedded modulators, and wireless networking have led to the emergence of the wireless sensor network [1]. All these sensor nodes are self-organized and cooperated in similar way as the ad hoc network. Such characteristics make it possible to deploy sensor network to obtain information about the covered area in an inaccessible location. It is expected that sensor networks open new vistas for many potential applications. Data aggregation [2] is generally considered as an important method in the sensor networks by aggregating and forwarding the raw data which is originated from multiple sources. In this way, the data collection from all nodes in the network is in fact done with the aggregation tree, in which the sink serves as the root and the nonleaf nodes will aggregate all raw data from their children and forward the result to their parents. This helps to minimize the traffic in the network, reduce the energy consumption, and increase the network lifetime accordingly.

Quality-of-Service (QoS), such as end-to-end delay, is generally considered as a severe criterion required in many applications. When data is collected by local nodes in the network, the sensed data is generally required to reach the root within a given time delay. To provide end-to-end delay guarantee is pioneering and challenging with the data aggregation problem in wireless sensor networks due to two separate observations. Firstly, the nonleaf nodes on the aggregation tree cannot forward to their parents until they have gathered all data from their descendants, which is possible to increase end-to-end delay especially in case that the aggregation tree is rather high. Secondly, due to the broadcasting characteristic of wireless communication, the exposed/hidden problem is the main factor which not only reduces capacity of the wireless network but also increases one-hop transmission delay. A careful scheduling algorithm concerned with the transmission time slots for all nodes in the network is rather necessary and important for data aggregation problem aiming at providing minimum end-to-end delay while utilizing the data aggregation scheme and eliminating the above collision problem.

The minimum-latency data aggregation problem in two-dimensional wireless sensor networks is well studied and proved to be NP hard [3]. Chen et al. had designed

a Δ—approximation algorithm for this problem, where Δ denotes the maximum node degree in the network [3]. Huang et al. introduced a constant approximation algorithm with the latency bounded by $(23R + \Delta - 18)$ [4] by building the aggregation tree with the help of maximum independent set (MIS), where R denotes the network radius. Wang et al. improved the idea of Huang and proposed a scheduling algorithm with the latency bound as $(15R + \Delta - 15)$ [5]. In this work, we have proved that the theoretic bound for data aggregation with our proposed scheme is also $(15R + \Delta - 15)$ for wireless sensor networks in two-dimensional space However, the proposed scheme has better performance compared with the previously mentioned related works [4, 5] especially in dense networks.

With the rapid development of applications in three-dimensional networks, such as underwater acoustic monitoring [6, 7], underground tunnels [8], space sensor network [9], and aerosphere pollution monitoring [10], it is interesting and challenging to study the minimum-latency data aggregation problem with three-dimensional wireless sensor networks too. Most of the current works in three-dimensional sensor networks aimed at providing connectivity, coverage, or location solutions [11]. In this work, we have also extended our efficient scheme to the case in three-dimensional network. We also demonstrated its practical performance compared with works originated from the two-dimensional networks.

The rest of this paper is organized as follows. we introduce the related works in Section 2 and present preliminary in Section 3. Section 4 introduces a new minimum-latency data aggregation (MDA) scheme. Section 5 has analyzed the correctness and performance of MDA scheme. In Sections 6 and 7, we present simulation results and conclusion.

2. Related Works

Data aggregation is considered as an important solution for the wireless sensor networks. The main goal of data-aggregation algorithms is to gather and aggregate data in an energy-efficient manner so that network lifetime is enhanced [2]. Krishnamachari et al. illustrated the impact of data aggregation by comparing its performance with traditional end-to-end routing schemes [12]. The optimal data aggregation problem was proven to be NP hard, and heuristic algorithms were proposed to gather data from multiple sources to the sink.

The process of aggregating data could reduce the transmission of data in the network, hence to reduce energy consumption. Most previous works have mainly focused on energy-saving issue and it has been investigated in [12, 13]. Wireless sensor networks often use tree topologies. This is not only because tree's structure is suitable for a network with one sink node, but also their simplicity is very attractive when network resources are limited. There are some papers that build a data aggregation tree to control the delay [3–5, 14–22].

Broadcast and data aggregation are the most fundamental and useful applications in wireless sensor networks. Data aggregation is sometimes referred to converge cast. Annamalai et al. designed a heuristic algorithm [14] which constructs a tree for both broadcast and converge cast. Simulation results show that latency of this algorithm is very high, and this algorithm has high requirements for hardware of wireless sensor. Upadhyayula et al. [15] designed a heuristic algorithm for converge cast alone and purposed to reduce the latency and energy consumption. This algorithm constructs a tree using a greedy approach where new nodes are added to the tree so that weight on the branch to which it is added is less.

Chen et al. first proved that the minimum data aggregation time problem is NP hard and designed a Δ—approximation algorithm for this problem [3], where Δ denotes the maximum node degree in the network. This algorithm is centralized and has a high latency, which makes this algorithm impractical. In paper [16], the authors focused on the latency performance of data aggregation and considered applications for which the total delay of the sensed data is important instead of the maximum delay. Wan et al. [17] designed a distributed algorithm to construct a dominating tree. The algorithm is to construct a maximal independent set and then select connected nodes to construct a dominating tree. Wan et al. [18] constructed three aggregation schedules of latency $(15R + \Delta - 4)$, $(2R + O(\log R) + \Delta)$, and $((1 + O(\log R/\sqrt[3]{R}))R + \Delta)$.

Huang et al. proposed a scheduling algorithm with the latency bound $(23R + \Delta - 18)$ [4], where R is the network radius and Δ is the maximum node degree. However, this algorithm has obvious errors in the first-fit algorithm so that the schedules are not collision-free in many cases. We will modify this algorithm and compare with it in simulation. Yu et al. proposed a distributed scheduling algorithm, named DAS, the latency bound $(24D + 6\Delta + 16)$ [19], where D is the network diameter. Xu et al. [20, 21] constructed a data aggregation tree using an existing approach with a small modification. Then, they presented an efficiently centralized algorithm and a distributed scheduling implementation. They proved that the latency bound is at most $(16R + \Delta - 14)$. They focused on data aggregation scheduling problem and proved that the lower latency bound under any interference model is $\max\{\log n, R\}$ in paper [20], where n is the network size. Wang et al. designed a scheduling algorithm, Peony-tree-based data aggregation (PDA), and proved the latency bound is $(15R + \Delta - 15)$ [5].

Some works [22–27] concerned with the delay control. Yuan et al. designed a Multi-level Fusion Synchronization Protocol to achieve the desired trade-offs between the credibility and the aggregation latency [22]. Some works have investigated the energy latency tradeoff [23–25]. Given a deadline, they minimized the amount of missed data [23], minimized overall energy dissipation of sensor nodes [24], or minimized the maximum energy consumption [25]. Many applications of sensor networks require real-time data aggregation. Chipara et al. proposed dynamic conflict-free query scheduling [26], a novel scheduling technology based on TDMA, which is a natural choice for real-time sensor network applications.

Event-triggered data aggregation refers to no node need to send data until a relevant, unpredicted event occurs in the network. A distributed TDMA scheduling protocol for data aggregation is proposed in paper [27], which is called DATP.

The sensor nodes send dummy packets in order to determine whether they are interfered with each other, if not, they are assigned the same time slot.

In this paper, we aimed at the minimum latency data aggregation problem with the data aggregation tree. We designed a novel approach to build the aggregation tree, which selects the maximum independent set in even layers. We have proved that the upper bound for data aggregation with our proposed scheme is bounded by $(15R + \Delta - 15)$ for wireless sensor networks in two-dimensional space. We also simulated the case in three-dimensional wireless sensor networks. The results have demonstrated that the proposed scheme has better performance compared with related works.

3. Preliminary

In this section, we will describe the system model, related terms and give a detailed problem definition.

3.1. Network Model. We consider a wireless sensor network with a sink node s and N sensor nodes. Each node is equipped with a wireless radio by which a node can receive/transmit data packet. Assume that all nodes have the same transmission range r_c; the topology of wireless sensor network can be represented as an undirected graph $G = (V, E)$, where V is the set of all nodes and $E \subseteq V^2$ is the set of undirected links. E is defined as $E = \{(u, v) \in V^2 \mid d_{uv} \leq r_c\}$, in which d_{uv} is the Euclidean distance between node u and v, r_c is the transmission radius.

3.2. Interference Model. Here, we use the symbol time slot to denote the period that one node is used to send or receive data packet. The radio generally works in half-duplex mode and thus one node cannot transmit and receive packet simultaneously due to broadcast characteristic of wireless communication, which is generally named as the hidden/exposed terminal problems. Here, we assume that the interference range is identical to the transmission range [28].

3.3. Related Terms

3.3.1. Data Aggregation. We assume that the network is designed with simple data aggregation function, such as max, min, and average. In this way, the final result after aggregation generally has the same property as the incoming originally data, such as packet length and prior. And all these results can be sent or received during one single time slot operation.

3.3.2. Independent Set (IS). Given an undirected graph $G = (V, E)$ and $I \subseteq V$, I is called an independent set if any two nodes in I are not adjacent. It is obvious that two nodes $u, v \in$ IS denotes that both u and v are not within the transmission range of each other. A maximum independent set is used to represent the independent set with maximum number of nodes for a given graph G.

3.3.3. Connected Dominating Set (CDS). Given an undirected graph $G = (V, E)$ and $D \subseteq V$, D is called a dominating set if D induces a connected subgraph of G and every node in G either belongs to D or is adjacent to a node in D.

3.3.4. Concurrent Set (CS). The concurrent set denotes a set of nodes, in which each node can transmit without conflict with the transmission of other nodes.

3.4. Data Aggregation Scheduling. With the terms mentioned previously, the data aggregation scheduling problem in the wireless sensor networks is defined to find a sequence of concurrent set $S_1, S_2, \ldots, S_T$ so that the latency is minimized, where $\bigcup_{i=1}^{T} S_i = V - \{s\}$. For any $i \neq j$, $S_i \cap S_j = \emptyset$. This problem has been proved to be NP hard [14].

4. Proposed Scheme

In this section, we give the detailed description of our proposed scheme for the minimum-latency data aggregation (MDA) problem. The basic idea is that we firstly construct a data aggregation tree by dividing the nodes into layers and then design a scheduling scheme in which each node is assigned with one time slot for transmission while collision is avoided. The notations used in the algorithm are summarized in Notations.

4.1. Data Aggregation Tree Construction

4.1.1. Initialization. For a given network graph, initially we can divide all nodes into R layers with the breadth-first searching algorithm which starts from the sink. The sink node s is in the layer 0. L_i denotes the set of the nodes that are i-hop away from the sink s. Here we use L_i to denote the set of nodes which are i-hop away from the sink s. The sink node s is in the layer 0.

4.1.2. Independent Set Construction for Each Layer. Here, we use the independent set to construct the data aggregation tree. The idea is described as follows. Initially, we add the sink node s to the independent set in layer 0. Obviously, there is only node in layer 0, and no conflict occurs. Let $D_0 = \{s\}$, in which D_0 denotes the independent set of layer 0. Generally, we select independent set D_i in layer i if i is an even, and the process starts from layer 0. In layer i, we check each node in sequence to find whether it conflicts with nodes in set D_j $(j = 0, 2, \ldots, i - 2)$. If not, the node is added to D_i; otherwise, we will move it to layer $(i + 1)$. The corresponding pseudocode is given in Algorithm 1. After the independent sets D_i $(i = 0, 2, 4, \ldots)$ have been selected, we can obtain the total independent set as $D = \cup D_i$ $(i = 0, 2, 4, \ldots)$.

4.1.3. Construct the Aggregation Tree. The basic idea is that we construct a connected dominating set as the aggregation tree based on the previous independent set. Obviously, sink node s in layer 0 is the root of data aggregation tree. We add s to D_0. We find that D_2 with Algorithm 1 and the nonindependent nodes are moved to layer 3. Each node u in layer i, $i > 2$ can check its neighbors in upper layer and adds them to

Input: Graph G, $L_1, L_2, \ldots, L_R$, layer number i;
Output: Maximum Independent set D_i, D, and L_i.
(1) $D_i = \emptyset$;
(2) for each node $u \in L_i$;
(3) if $u \notin \{x \mid (x, v) \in E, v \in D\}$, then
(4) $D_i = D_i \cup \{u\}$;
(5) else $L_i = L_i - \{u\}$; $L_{i+1} = L_{i+1} \cup \{u\}$;
(6) end if
(7) end for

Algorithm 1: Layer independent set construction.

the set $upper(u)$. Node u is moved to layer $(i + 1)$ in case that $|upper(u)| = 0$. In each even layer, we build the layer maximum independent set and move other nodes down to the next layer. The next step is to build the children list for each node. Firstly, node u in layer $(i-1)$ checks the neighbors in layer i and adds the neighbor to the set $lower(u)$. Repeat the previous process and the data aggregation tree can be built finally.

From the process of aggregation tree construction, we can clearly see that the children of nodes in IS are either leaf nodes or connecting nodes, while the children of the connecting node are nodes in IS. The pseudocode of the data aggregation tree construction process is presented in Algorithm 2.

4.2. Data Aggregation Scheduling. Nodes are scheduled according to their roles on the tree. We firstly schedule leaf nodes and then the nodes in connected dominating set.

4.2.1. Aggregation of Leaf Nodes. In this process, we are to assign time slots for each leaf nodes on the data aggregation tree, and the leaf nodes can send data packet to their parent during the assigned time slots. In order to avoid interference and minimize the latency at this phase, leaf nodes are divided into k concurrent sets and nodes in each set can transmit data simultaneously with interference avoided, which is denoted as $S_1, S_2, \ldots, S_k$, where k is the number of the concurrent sets. Nodes in S_i are scheduled to transmit data in the ith time slot. In the following, we will introduce that the process leaf nodes are separated into different concurrent sets. Firstly, there is no concurrent sets and $k = 0$. Then, we choose a node x from L_R and try to insert x inserted into one concurrent set. Obviously, we need to create a new set and let $k = 1$, and we can insert x into S_1. Then, we pick another node y in layer R and determine whether it conflicts with any node in current concurrent set S_i $(i = 1, 2, \ldots, k)$ or not. If S_i is found, y is inserted into it; otherwise, a new concurrent set is necessary, and we have k increased by one and insert y into the new set. This process continues until all leaf nodes are assigned to a set.

The pseudocode of concurrent set construction is presented in Algorithm 3.

4.2.2. Aggregation of Connected Dominating Nodes. When all leaf nodes are assigned time slots to transmit data packet to their parents, we are to schedule the time slots for nonleaf nodes on the aggregation tree. Starting from the bottom of the tree, the algorithm assigns the sending time to all connected dominating nodes. According to the tree construction process, the dominating nodes are located in even layers and connecting nodes in odd layer. Hence, the algorithm will construct concurrent set in every layer. That is, the dominating nodes in layer d are divided into m_d sets if d is even. Similarly, the connected nodes in odd layer c are also divided into n_c sets. Due to the fact that the latency of the aggregation of leaf nodes is k, we schedule all connected dominating nodes from the tree bottom to the root. Generally, for each node $u \in L_i$, if u is assigned to the jth set in layer i, the scheduled time slot for u is $(k + j + \sum_{d=H}^{i-2} m_d + \sum_{c=H}^{i-1} n_c)$ in case that i is even, where H denotes the depth of the data aggregation tree. And the result is $(k + j + \sum_{c=H}^{i-2} n_c + \sum_{d=H}^{i-1} m_d)$ in case that i is odd.

4.2.3. Data Aggregation Scheduling Algorithm. After the previous two steps, the sending time of all nodes in the network are assigned. The pseudocode of the data aggregation scheduling is presented in Algorithm 4. As we can observe from the algorithm details, the time of the sink s received is latency $= k + \sum_{i=H}^{i=1}(m_i + n_i) + 1$. Particularly, we have $n_i = 0$ in case that $\mathrm{mod}(i, 2) = 1$, and otherwise $m_i = 0$.

4.3. Example Demonstration. In this section, we use an example to demonstrate the process of data aggregation tree construction in detail. Figure 1(a) shows the topology of a random network with 20 nodes, in which each node is represented by a cycle, and the node identification is marked below the cycle. There is a link/edge between two nodes if they are within the transmission range of each other. Here, we assume that the sink node; that is, s is located at the center of the network.

Initially, we can organize the network topology into layers, Algorithm 1, and nodes in each layer can be observed in Figure 1(b). For example, there are only one node, that is, s, in layer 0, and five nodes, that is, 1, 2, 3, 4, and 5 in layer 2. There are totally 5 layers in this example with the network radius as 4. It is obvious that the layer number for each node denotes the distance from the node to the sink.

The second process is to build the MIS for each even layer. The process is carried out in increasing order of the layers. Firstly, there is only one node in layer 0, and the node s is added to the independent set for layer 0, that is, $D_0 = \{s\}$, which is marked as black in Figure 1(b). Note that nodes in layer 1 are one hop away from node s, and thus they are not possible to be included into the IS, and they are added into the aggregation tree with their parent as s. The next step is to select IS for layer 2. With Algorithm 2, we can obtain $D_2 = \{6, 8, 9, 11, 12, 13\}$ which is also marked black in the figure; 7, 10, and 14 are dependent nodes and they are moved down to layer 3 (the process is simulated via dash line in Figure 1; then we can select parent nodes form nodes in D_2, which are marked as gray in Figure 1(b). In this way, the tree construction process for previous three layers is finished.

Now, consider nodes in layer 3. Note that 7, 10, and 14 are moved from previous layer to layer 3, and it is possible

Input: Graph $G = (V, E)$;
Output: Data aggregation tree $T = (V', E')$, the dominated set $D_i, i = 2, 4, \ldots$, the set of connective nodes $C_i, i = 1, 3, \ldots$
(1) $V' = V$, $E' = \emptyset$;
(2) Breadth first search graph G with the root as s;
(3) Divide all node of V' into layers $L_0, L_1, \ldots, L_R$;
(4) $D_1 = \{s\}$;
(5) for each node $x \in L_1$, $par(x) = s$; $E' = E' \cup \{(x, s)\}$;
(6) for $i = 2$ to R do
(7) calculate $upper(u)$, $u \in L_i$
(8) if $|upper(u)| = 0$ and $u \in L_i$, then $L_i = L_i/u$, $L_{i+1} = L_{i+1} \cup \{u\}$;
(9) if i is even, construct the layer independent set D_i, and sequence the $lower(y)$ with decreasing order of the set size as $y_1, y_2, \ldots, y \in L_{i-1}$, and the set size as $y_1, y_2, \ldots, y \in L_{i-1}$;
(10) $j = 1$;
(11) while L_i != empty do
(12) for each node $u \in lower(y_j)$
(13) $par(u) = y_j$, $E' = E' \cup \{(u, y_j)\}$, $L_i = L_i/u$;
(14) if i is even then $C_i = C_i \cup y_j$;
(15) end for
(16) $j = j + 1$;
(17) end while
(18) end for

Algorithm 2: Data aggregation tree construction.

Input: Data aggregation tree T, G;
Output: Concurrent Set $S_1, S_2, \ldots, S_k$.
(1) initialize $S_i = \emptyset$, $i = 0, 1, \ldots$;
(2) for each layer i from R to 1;
(3) select one leaf node u from layer i;
(4) $j = 1$;
(5) if $S_j = \emptyset$, then $S_j = S_k \cup \{u\}$, go to step 3;
(6) for each $v \in S_j$,
(7) if $(u, par(v)) \notin E$ and $par(u) \neq par(v)$, then $S_j = S_j \cup \{u\}$, break;
(8) end for
(9) if $u \notin S_j$, then $j = j + 1$, go to step 5;
(10) end for

Algorithm 3: Concurrent set construction.

Input: Data aggregation tree $T = (V', E')$, depth of the data aggregation tree H;
Output: Sending time $t(u)$ for each node u, and *latency*.
(1) for each node $u \notin CDS$, run Algorithm 3 to construct the concurrent set $S_1, S_2, \ldots, S_k$;
(2) if $u \in S_j$ then $t(u) = j$;
(3) $latency = k$;
(4) for $i = H$ to 1
(5) if i is even, then
(6) all nodes $u \in L_i$, construct the concurrent set $SD_1, SD_2, \ldots, SD_m$;
(7) if $u \in SD_j$, then $t(u) = latency + j$, $latency = latency + m$;
(8) else
(9) all nodes $u \in L_i$, construct the concurrent set $SC_1, SC_2, \ldots, SC_n$;
(10) if $v \in SC_j$, then $t(v) = latency + j$, $latency = latency + n$;
(11) end if
(12) end for

Algorithm 4: Minimum-latency data aggregation.

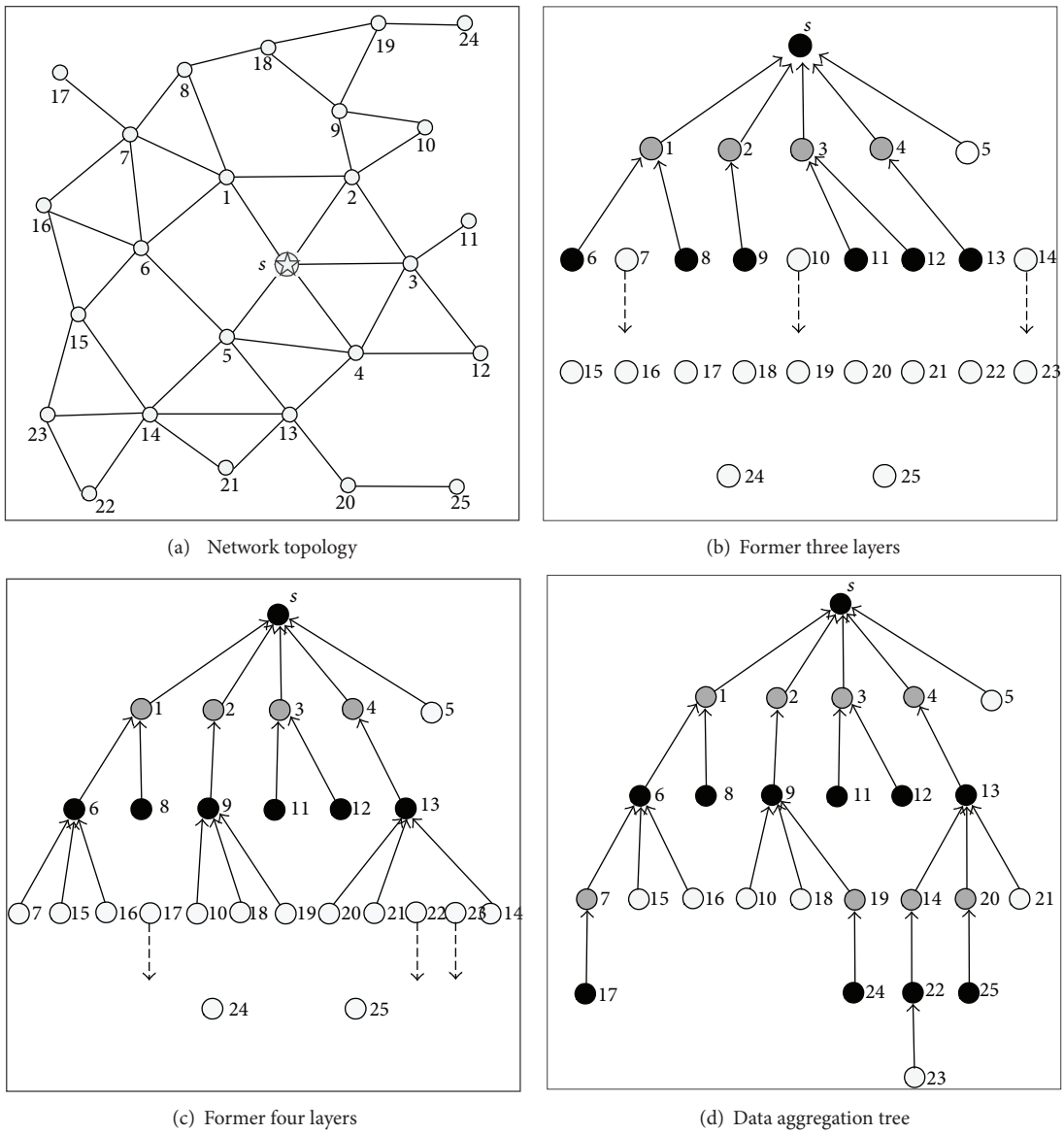


Figure 1: An example to demonstrate the process of MDA algorithm.

that some nodes in layer 3 cannot find their parents in upper layer, that is, layer 2. With Algorithm 2, we calculate the set upper for each node in layer 3, and we have $|upper(17)| = 0$, $|upper(22)| = 0$, $|upper(23)| = 0$. As we can observe from Figure 1(b), 17 can connect to sink s via 7; however, 17 cannot find proper parent in layer 2 since 7 is moved to layer 3. The case is similar to 22 and 23, and all these three nodes are moved down to layer 4, which is marked with dash line in Figure 1(b). After these processes are done, we can select parent for nodes in layer 3 and add them into the tree.

There are five nodes, that is, 17, 22, 23, 24, and 25 in layer 4. We firstly check whether they can find their parents in upper layers before building the independent set. It can be seen that set upper is not empty for all nodes in layer 4; secondarily, we build the MIS for layer 4, and $D_4 = \{17, 22, 24, 25\}$ which is marked black; 23 is moved down to layer 5; finally, we choose parent for nodes in layer 4, and 7, 14, 19, and 20 are selected accordingly and included in set C_3 which is marked gray. In this way, the tree construction for former four layers is finished which is illustrated in Figure 1(c).

Note that there is only 23 in layer 5 with parent 22 in layer 4, and it can be inserted into the tree directly. So far, all nodes in the network are included in the aggregation tree with the height as 5. The final result is demonstrated in Figure 1(d).

The last process is to schedule the aggregation process on the tree with Algorithms 3 and 4. Following the idea of our scheduling scheme, the leaf nodes on the tree are scheduled firstly. It starts from the final layer, and all leaf nodes are scheduled into collision-free time slots. As we can see from Figures 1(a) and 1(d), these leaf nodes can be divided into two separate sets, {23, 15, 10, 21, 5} and {16, 18}. Nodes in the first set are assigned to the first time slot, and nodes in the second

set are assigned to second time slot. In this way, we use only 2 time slots to ensure that all leaf nodes can send to their parents while collision is avoided. Then, we are to schedule the nonleaf nodes on the tree. It can be seen that nodes in $D_4 = \{17, 24, 22, 25\}$ can be scheduled with one time slot, that is, slot 3. The process is carried out in the same way until all nodes are scheduled. And we can see that in time slot 11 the sink node can collect all data from all nodes in the network.

5. Performance Analysis

Lemma 1. *Given the original network radius R, the depth of the data aggregation tree by our MDA algorithm does not exceed* $(2R - 1)$.

Proof. As shown in the process of the tree construction, nodes in layer 0 and layer 1 are not moved down.

In layer 2, we select the layer maximum independent set D_2 and move the nonindependent nodes to layer 3. L_2' denotes the set of the nodes which moved from layer 2 to layer 3. Since each node in L_2' is adjacent to at least one node in D_2, it finds its parent in D_2. Layer 3 is the base for nodes in original layer 2.

In layer 3, some nodes may not find their neighbors in D_2. Their neighbors have been moved to the set L_2' in the same layer with them. Nodes in layer 3 cannot find their neighbors as their parent in upper layer, which should be moved to layer 4. L_3' denotes the set of nodes which move from layer 3 to layer 4. Hence, each node in L_3' can find its neighbor as its parent in L_2'. We should select the layer maximum independent set D_4 in layer 4, so some nodes in L_3' may be as the nonindependent nodes moved to layer 5. L_3'' denotes the set of nodes which move from layer 3 to layer 5. Since each node in L_3'' is adjacent to one node in D_4 at least, it finds its parent in D_4. Layer 5 is the lowest layer the nodes in original layer 3 can be moved.

In layer 4, because there may be some nodes whose neighbors are all in the set L_3'', they should be moved to layer 6 with the nodes in L_3'' moving to layer 5. L_4' denotes the set of nodes which moves from layer 4 to layer 6. We should select the layer maximum independent set D_6 in layer 6, so some nodes in L_4' may be as the nonindependent nodes moved to layer 7. L_4'' denotes the set of nodes which moves from layer 4 to layer 7. Because each node in L_4'' is adjacent to at least one node in D_6, it finds its parent in D_6. Layer 7 is the base layer that nodes in original layer 4 can be moved.

After analyzing the nodes moving cases in the four layers, we can get the number of layers each node is displaced from its original position depending on its neighbor in upper layer. When a node was moved to an even layer, the layer independent set should be selected in this even layer, so the node may be as a nonindependent node moved to the next layer. Then, the node in the odd layer can find its neighbor in the maximum independent set; the node will no longer be moved. In this way, it would move into its final position in an odd layer. Therefore, in the worst case, a node's moving layer is the moving layer of its neighbor in upper layer plus 2. For a node u in layer k, if all its neighbors in layer $(k - 1)$ have been moved to layer $(2(k-1)-1)$, at worst, they have to be moved to layer $(2k - 3)$. We can infer that the final layer which u moved to is equal to the sum of the layer $(2k - 3)$ which neighbors of u moved to plus 2, that is, layer $2(k - 1)$. Given the original network radius R, in the worst case, the nodes in layer R may be moved to layer $(2R - 1)$. □

Lemma 2. *The latency of aggregation from leaf nodes is* $\Delta - 1$.

Proof. Given the maximum node degree Δ, the latency of aggregation from leaf nodes is $(\Delta - 1)$ [4, 5]. □

Theorem 3. *The latency bound of the connected dominating nodes aggregation is* $15R - 14$.

Proof. Now, we will estimate the data aggregation of connected dominating nodes. According to Lemma 1, the depth of data aggregation tree is at most $(2R - 1)$. In the tree, the number of layers of the dominative nodes except the sink s is $(R-1)$. The number of layers of the connective nodes is $(R-1)$, because connective nodes cannot be in layer R. According to [4, 5], the latency bound of data aggregation from dominative nodes to connective nodes is 4, and the latency bound of data aggregation from connective nodes to dominative nodes is 11. Particulary, the sink node s is the root of the tree. Thus, connective nodes in layer 1 are its children and it takes at most 12 time slots to finish the transmission.

Based on the previous analysis, the latency bound of the connected dominating nodes aggregation is $4(R-1) + 11(R-2) + 12 = 15R - 14$. □

Theorem 4. *The total latency bound of data aggregation is* $15R + \Delta - 15$.

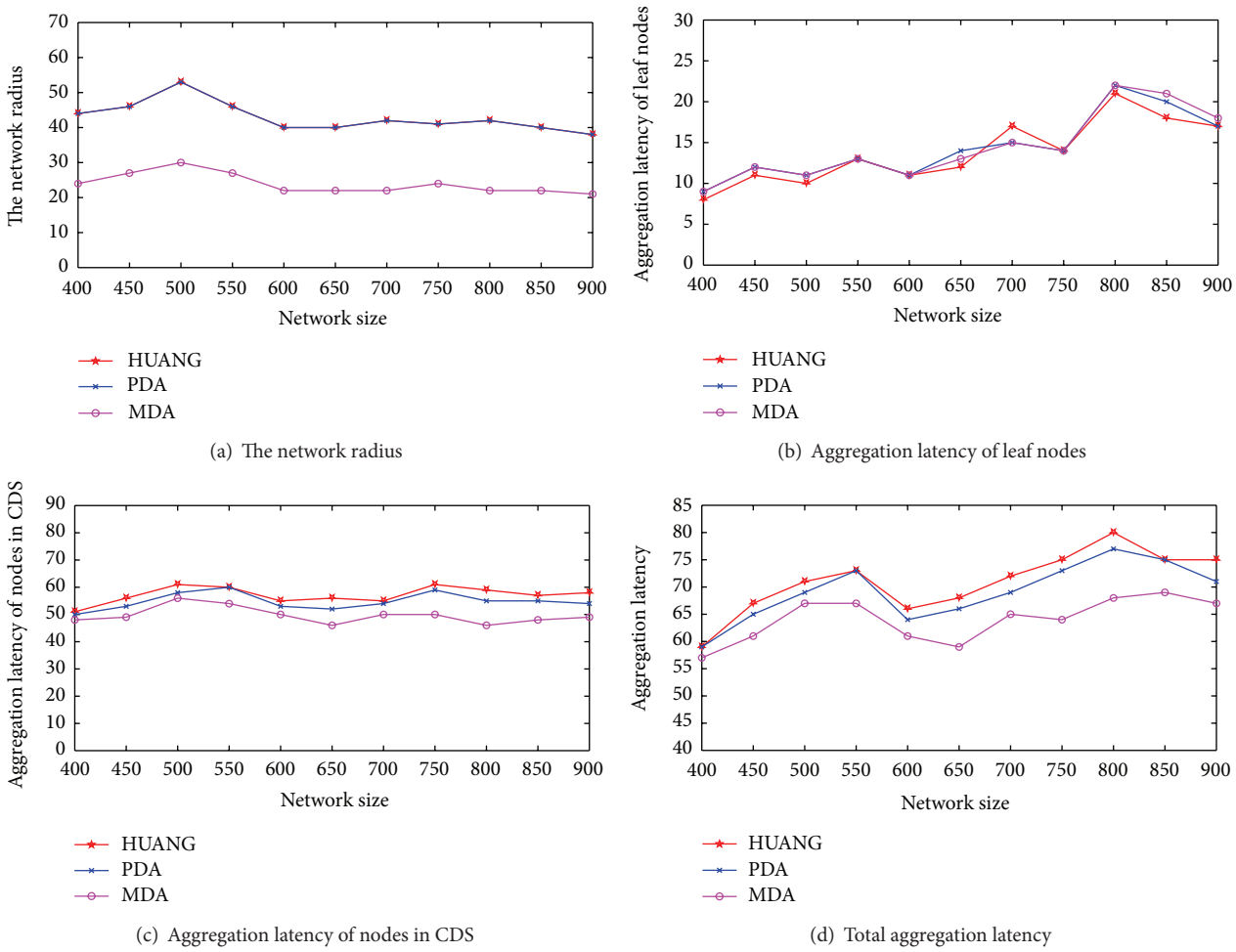
Proof. According to the MDA, the total latency is the sum of latencies of leaf nodes aggregation and connected dominating nodes aggregation. The total latency bound of data aggregation is $(\Delta - 1) + (15R - 14) = 15R + \Delta - 15$. □

Theorem 5. *The time complexity with the MDA algorithm is* $O(Rn^2 + n\Delta)$, *in which n is the node number, R is the network radius, and* Δ *is the maximum node degree.*

Proof. Initially, we use the breadth search algorithm to construct the layer structure for a given network, and the complexity with Algorithm 1 is $O(n^2)$.

During the tree construction process of Algorithm 2, we check each node whether they can find their parent, and the time complexity is $O(n^2)$. Secondarily, we select independent set for even layers with the rest nodes excluded in the IS moved down, and the time complexity for this operation is $O(n)$. Finally, each node will select its parent and be added into the aggregation tree; the time complexity is $O(n^2)$. Assume the network radius as R, the time complexity for the tree construction is $O(Rn^2)$.

During the scheduling process, we first schedule the leaf nodes and then nonleaf nodes. Note that we schedule these nodes from layers far away from the sink, and the collision conflict occurs only in case that the transmission is carried



(a) The network radius

(b) Aggregation latency of leaf nodes

(c) Aggregation latency of nodes in CDS

(d) Total aggregation latency

Figure 2: Simulation results in two-dimensional networks with different network sizes.

out simultaneously by neighbors of its parent, and thus the scheduling time complexity is $O(n\Delta)$.

In this way, the time complexity with our MDA algorithm is $O(n^2) + O(n^2) + O(Rn^2) + O(n\Delta) = O(Rn^2)$.

Based on the previous analysis, we could know that the total latency bound of data aggregation obtained by MDA is the same as [5] in two-dimensional space, which is the best result we have ever known. Besides, our proposed algorithm could achieve better performance by reducing the height of the aggregation tree without increasing the time complexity. The later experimental results also validate the efficiency of the method. □

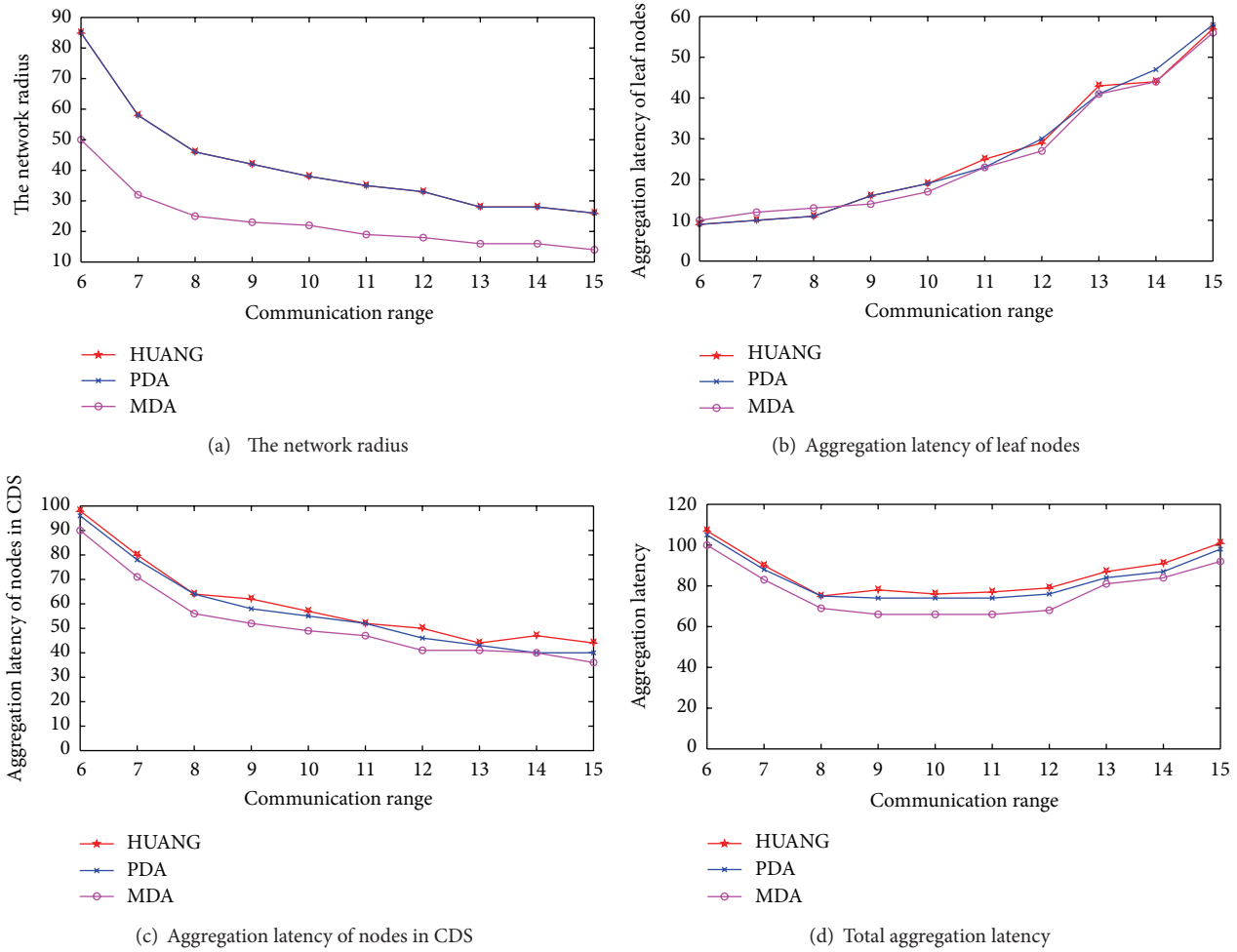
6. Simulation Results

Our simulation is accomplished by generating a random wireless sensor network in MATLAB software. We evaluate the performance of the proposed MDA and related in two- and three-dimensional networks.

6.1. Simulation Results in Two-Dimensional Networks. In this simulation part, the network topology is randomly generated by placing nodes in a fixed region of size 100 m × 100 m. We compare our MDA with the algorithm proposed by Huang et al. in [15] (denoted as HUANG in short in the figures) and PDA proposed by Wang in [19].

6.1.1. Impact of Network Size. The first group of simulations estimates the impact of network size. The transmission range of each sensor is fixed to 10 m. The aggregation latency is measured when the network size varies from 300 to 800. We compare its average performance by building 11 different network topologies.

Figure 2(a) compares the network radius after constructing the data aggregation tree by using these algorithms. As mentioned in the previous section, the worst case for the tree height with the MDA algorithm is $2R - 1$. However, it can be seen that the upper bound is seldom met in the simulations. This is because nodes are not always moved during the tree construction process with our MDA algorithm, which leads to the reduction of tree height. However, the results with HUANG and PDA are almost always $2R - 1$. It is seen that the network radius of HUANG and PDA is approximately twice as much as MDA. Our algorithm shows great improvement on the tree height as we can observe from Figure 2(a).



(a) The network radius

(b) Aggregation latency of leaf nodes

(c) Aggregation latency of nodes in CDS

(d) Total aggregation latency

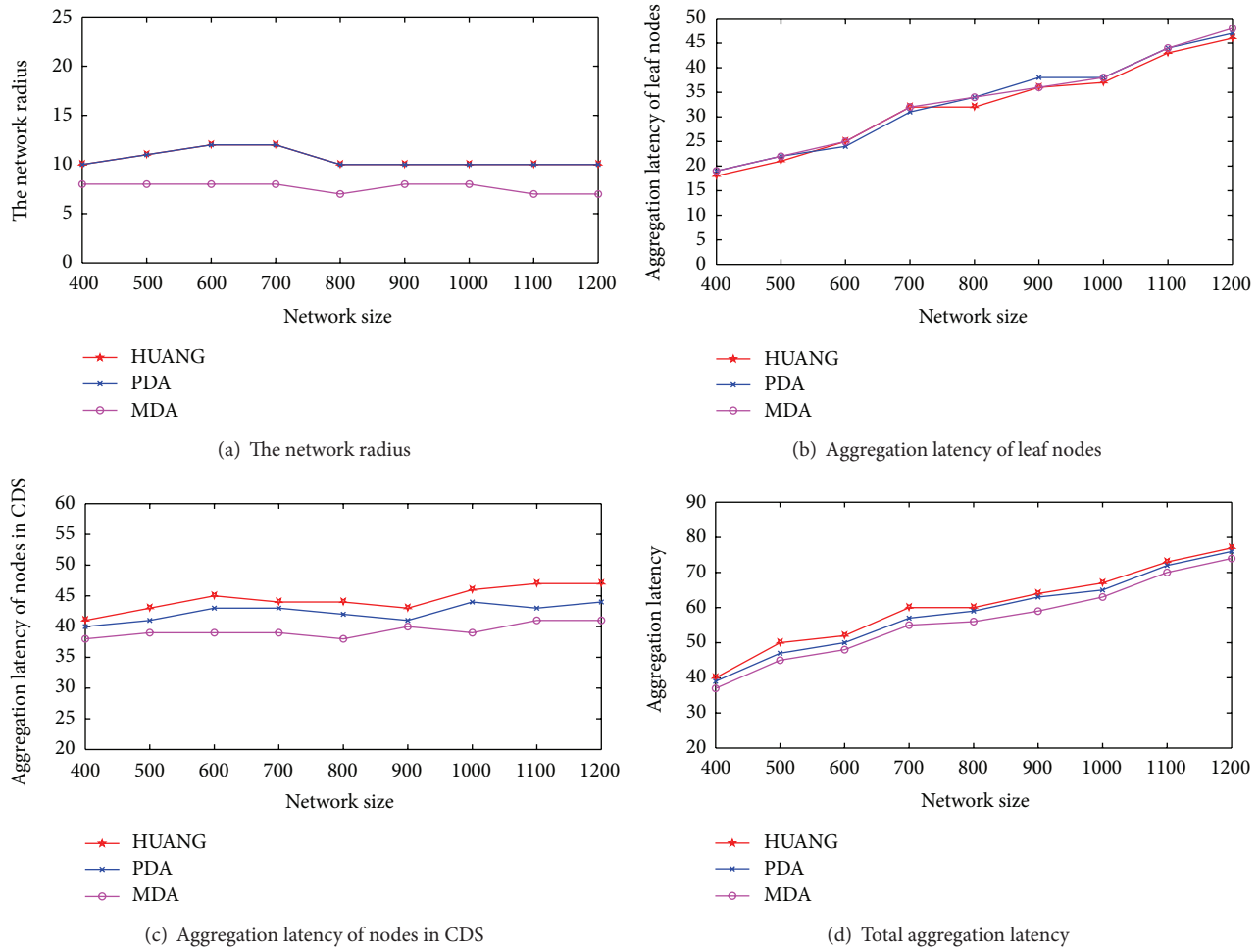
Figure 3: Simulation results in two-dimensional networks with different communication ranges.

Figure 2(b) shows the simulation result of aggregation latency of leaf nodes. Due to the fact that independent set is selected only in even layers with our MDA, the number of connected dominating nodes is less than the other two algorithms, and thus the number of leaf nodes is larger than the other two algorithms. So, it is reasonable that, in some case, the delay for leaf nodes scheduling costs more time slots with our MDA algorithm, which can be seen in case that network size is 850.

Figure 2(c) compares the aggregation latency of nodes in the connected dominating sets. As mentioned previously, the MDA algorithm only selects dominating nodes in even layers, and thus it leads to less number of dominating nodes compared with the other two algorithms. And accordingly, the required number of time slots for aggregation is generally smaller.

The total latency for the data aggregation process is calculated with the previous two parts. As we can see from Figure 2(d), our MDA algorithm runs better than HUANG and PDA, which is more significant in case that the network size is very large.

6.1.2. Impact of Communication Range. The second group of simulations estimates the impact of communication range. The network size is fixed to 1000 while the communication range varies from 6 m to 15 m. Figure 3(a) shows the network radius after constructing the data aggregation tree by using HUANG, PDA, and MDA. With the communication range becoming larger, the node degree increases, and accordingly the radius decreases. Figure 3(c) compares the latency of aggregation from leaf nodes. The latency of aggregation from leaf nodes decreases when the communication range becomes larger. The total latency is the sum of latencies of leaf nodes aggregation and connected dominating nodes aggregation. The smaller the transmission range, the more independent nodes, the more connected dominating nodes accordingly. Hence, the total latency is mainly determined by the latency of aggregation from connected dominating nodes. When the transmission range becomes larger, the node degree increases; thus, the number of leaf nodes increases. Now, the total latency depends on latency of aggregation from leaf nodes. The latency is large at both ends of curves and relative small in middle, as shown in Figure 3(d).



(a) The network radius

(b) Aggregation latency of leaf nodes

(c) Aggregation latency of nodes in CDS

(d) Total aggregation latency

Figure 4: Simulation results in three-dimensional networks with different network sizes.

6.2. Simulation Results in Three-Dimensional Networks. In the part, the network topology is randomly generated by placing nodes in a $100 \times 100 \times 100\,m^3$ cube. We compare our MDA with HUANG and PDA when applied to the three-dimensional networks.

6.2.1. Impact of Network Size. The third group of simulations estimates the impact of network size in three-dimensional networks. Similar to the first group of simulations, the results of this group are shown in Figure 4.

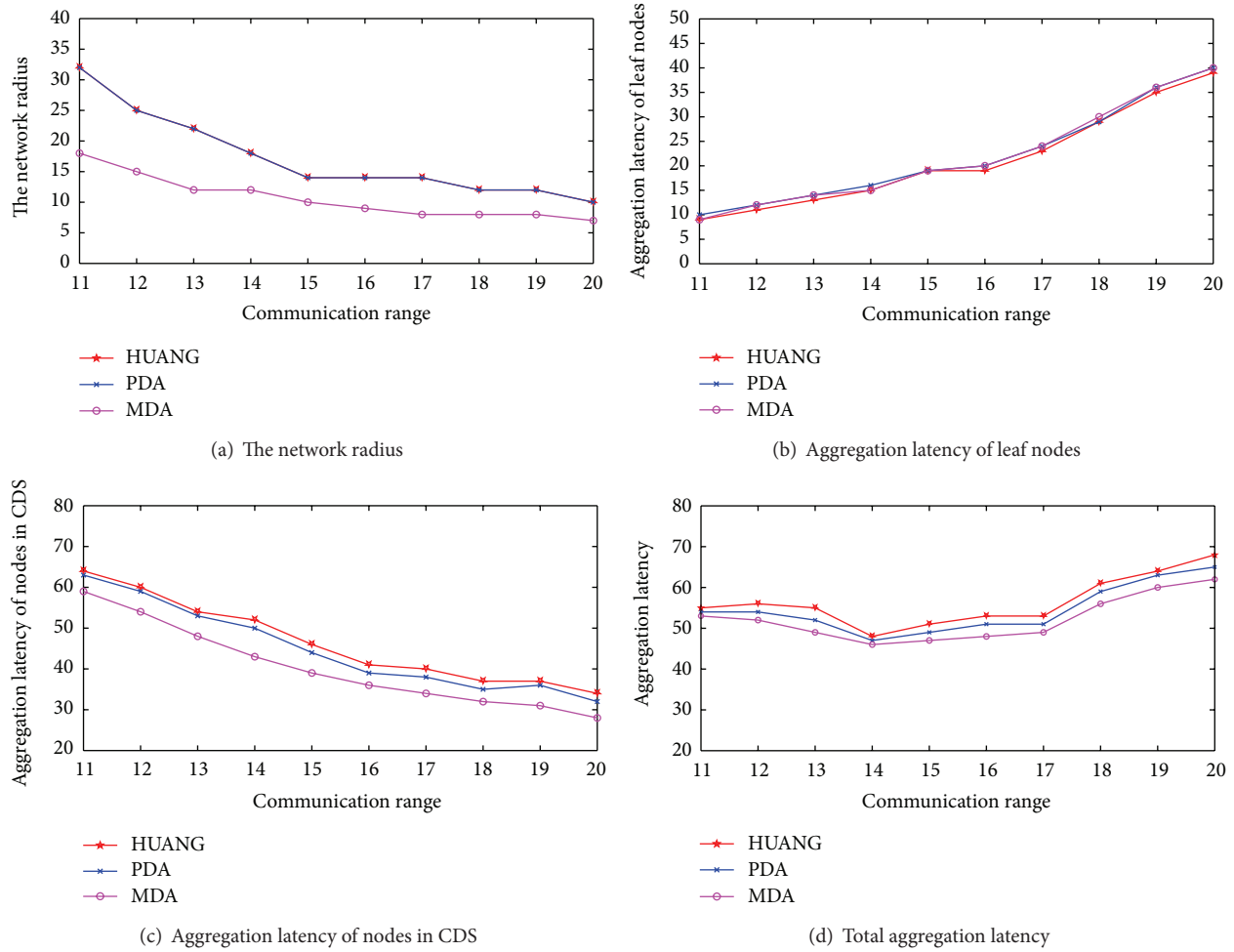
6.2.2. Impact of Communication Range. The last group of simulations estimates the impact of communication range in three-dimensional networks. Similar to the first group of simulations, the results of this group are shown in Figure 5.

7. Conclusions

Data aggregation is an import technology used to reduce the energy consumption in the wireless sensor networks. In this paper, we focused on the minimum latency data aggregation problem in wireless sensor networks and proposed a novel minimum-latency data aggregation (MDA) algorithm to build the aggregation tree as well as scheduling scheme for the node transmission in the network. We proved that the theoretical latency bound for MDA in the plane is $(15R + \Delta - 15)$. We have also simulated the case in three-dimensional wireless sensor networks. Extensive simulation results have demonstrated that our algorithm has good performance compared with the related algorithms. In the future work, we are to develop a distributed version of the proposed MDA algorithm with energy considered by constructing a load balance aggregation tree. Furthermore, we will extend our work to the multisink wireless sensor networks.

Notations

$dist(u, v)$: Euclidean distance between u and v
$N(u)$: The set of neighbors for node u
R: The network radius
H: Depth of the data aggregation tree
Δ: Maximum node degree
L_i: The set of nodes that are i-hop away from the sink s
$upper(u)$: The set of neighbors in upper layer of node u



(a) The network radius

(b) Aggregation latency of leaf nodes

(c) Aggregation latency of nodes in CDS

(d) Total aggregation latency

Figure 5: Simulation results in two-dimensional networks with different communication ranges.

$lower(u)$: The set of neighbors in lower layer of node u
$tag(u)$: The tag whether u's sending time is determined
$par(u)$: The parent node of node u on the data aggregation tree.

Acknowledgments

This work is supported by the National Science Foundation of China under Grants nos. 61370210, 61103175, the Fujian Provincial Natural Science Foundation of China under Grants nos. 2011J01345, 2013J01232, and the Development Foundation of Educational Committee of Fujian Province under Grant no. 2012JA12027.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] R. Rajagopalan and P. K. Varshney, "Data-aggregation techniques in sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.

[3] X. Chen, X. Hu, J. Zhu, X. Jia, J. Wu, and Y. He, "Minimum data aggregation time problem in wireless sensor networks," in *Mobile Ad-Hoc and Sensor Networks*, vol. 3794, pp. 133–142, Springer, Berlin, Germany, 2005.

[4] S. C.-H. Huang, P.-J. Wan, C. T. Vu, Y. Li, and F. Yao, "Nearly constant approximation for data aggregation scheduling in wireless sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 366–372, May 2007.

[5] P. Wang, Y. He, and L. Huang, "Near optimal scheduling of data aggregation in wireless sensor networks," *Ad Hoc Networks*, vol. 11, pp. 1287–1296, 2013.

[6] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[7] D. Pompili and T. Melodia, "Three-dimensional routing in underwater acoustic sensor networks," in *Proceedings of the 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '05)*, pp. 214–221, ACM, Montreal, Canada, October 2005.

[8] D. Wu, L. Bao, and R. Li, "A holistic approach to wireless sensor network routing in underground tunnel environments," *Computer Communications*, vol. 33, no. 13, pp. 1566–1573, 2010.

[9] X. Hong, M. Gerla, R. Bagrodia, T. J. Kwon, P. Estabrook, and G. Pei, "The mars sensor network: efficient, energy aware communications," in *Proceedings of the IEEE Military Communications Conference (MILCOM '01)*, vol. 1, pp. 418–422, October 2001.

[10] W. Tsujita, A. Yoshino, H. Ishida, and T. Moriizumi, "Gas sensor network for air-pollution monitoring," *Sensors and Actuators B*, vol. 110, no. 2, pp. 304–311, 2005.

[11] V. Ravelomanana, "Extremal properties of three-dimensional sensor networks with applications," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 246–257, 2004.

[12] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, pp. 575–578, 2002.

[13] Y. Wu, S. Fahmy, and N. B. Shroff, "On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 1013–1021, April 2008.

[14] V. Annamalai, S. K. S. Gupta, and L. Schwiebert, "On tree-based convergecasting in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '03)*, vol. 3, pp. 1942–1947, 2003.

[15] S. Upadhyayula, V. Annamalai, and S. K. S. Gupta, "A low-latency and energy-efficient algorithm for convergecast in wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 6, pp. 3525–3530, December 2003.

[16] C. Joo, J.-G. Choi, and N. B. Shroff, "Delay performance of scheduling with data aggregation in wireless sensor networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, March 2010.

[17] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 3, pp. 1597–1604, June 2002.

[18] P.-J. Wan, S. C.-H. Huang, L. Wang, Z. Wan, and X. Jia, "Minimum-latency aggregation scheduling in multihop wireless networks," in *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, pp. 185–193, ACM, New Orleans, La, USA, May 2009.

[19] B. Yu, J. Li, and Y. Li, "Distributed data aggregation scheduling in wireless sensor networks," in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM '09)*, pp. 2159–2167, April 2009.

[20] X. Xu, X. Y. Li, X. Mao, S. Tang, and S. Wang, "A delay-efficient algorithm for data aggregation in multihop wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 163–175, 2011.

[21] X. Xu, S. Wang, X. Mao, S. Tang, and X. Li, "An improved approximation algorithm for data aggregation in multi-hop wireless sensor networks," in *Proceedings of the 2nd ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC '09)*, pp. 47–56, New Orleans, La, USA, May 2009.

[22] W. Yuan, S. V. Krishnamurthy, and S. K. Tripathi, "Synchronization of multiple levels of data fusion in wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 1, pp. 221–225, December 2003.

[23] S. Hariharan and N. B. Shroff, "Maximizing aggregated revenue in sensor networks under deadline constraints," in *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (CDC/CCC '09)*, pp. 4846–4851, December 2009.

[24] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. 244–255, March 2004.

[25] L. Becchetti, P. Korteweg, A. Marchetti-Spaccamela, M. Skutella, L. Stougie, and A. Vitaletti, "Latency constrained aggregation in sensor networks," in *Algorithms—ESA 2006*, Y. Azar and T. Erlebach, Eds., vol. 4168, pp. 88–99, Springer, Berlin, Germany, 2006.

[26] O. Chipara, C. Lu, and J. Stankovic, "Dynamic conflict-free query scheduling for wireless sensor networks," in *Proceedings of the 14th IEEE International Conference on Network Protocols (ICNP '06)*, pp. 321–331, November 2006.

[27] M. O. Díaz-Anadón and K. K. Leung, "TDMA scheduling for event-triggered data aggregation in irregular wireless sensor networks," *Computer Communications*, vol. 34, no. 17, pp. 2072–2081, 2011.

[28] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.