Approaching the Optimal Schedule for Data Aggregation in Wireless Sensor Networks

Pei Wang[1], Yuan He[2], and Liusheng Huang[1]

[1] Department of Computer Science
University of Science and Technology of China, China
[2] Department of Computer Science
Hong Kong University of Science and Technology, China

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]. 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 $23R + \Delta - 18$.

Keywords: wireless sensor networks, data aggregation, latency.

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 [3,4,5,6,7,8,9]. 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. 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.

G. Pandurangan et al. (Eds.): WASA 2010, LNCS 6221, pp. 26–35, 2010.

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 [1,2,10].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 a latency bound of $(\Delta - 1)R$, where Δ is the maximum node degree and R is the network radius measured in hops. Kesselman and Kowalski [10] design an algorithm which has a latency bound of $O(log(N))$. However, 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 Wan *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. Wan's results are the state-of-arts on scheduling algorithms of data aggregation.

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. In local aggregation, raw data at the leaf nodes are transmitted to their corresponding parent nodes for aggregation. 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. Section 2 introduces the terminology, models, and assumptions in this paper as well as the problem formulation. In Section 3 we elaborate the scheduling algorithm of data aggregation. Section 4 presents the proofs of correctness and theoretical performance analysis, followed by the simulations results in Section 5. We conclude the work in Section 6.

2 Preliminaries

2.1 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 3. We will remove this assumption in Section 4.

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 $\{S_1, S_2, ..., S_K\}$, 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 $R_i \subseteq (\bigcup_{l=i+1}^{l=K} S_l) \bigcup \{s\}$.

2.2 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 $l_1 : p \rightarrow q$ and $l_2 : u \rightarrow v$ 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.

3 Peony-Tree-Based Data Aggregation (PDA)

PDA mainly consists of three components, the construction of peony tree, local aggregation, and global aggregation.

3.1 Construction of Peony Tree

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.

1. Dominative nodes selection. 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 $L_0, L_1, ..., L_R$, according to their depths in T_{BFS}. Specially, $L_0 = \{s\}$.

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, add u into D and remove u and its neighbors from all layers. Otherwise, remove u 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.

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 $D_0, D_1, ..., D_R$ 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.

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 $2(R-2)+1+1+1=2R-1$.

3.2 Local Aggregation

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 child 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 $(i = 1, 2, ..., k)$, where k is the number of the node-disjoint maximum concurrent subsets of W. Therefore, 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 $(i = 1, 2, ..., k)$ are iteratively determined by using first fit algorithm.

The first fit algorithm for constructing node-disjoint maximum concurrent sets works as follows. 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 $U_1, U_2, ..., U_h$, 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 $\{l : v \to f(v) \mid v \in U_i\}$, 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.

3.3 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 them by $UD_{i,1}, UD_{i,2}, ..., UD_{i,m_i}$ and

$UC_{i,1}, UC_{i,2}, ..., UC_{i,n_i}$ 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 $n_R = 0$.

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 $(\sum_{l=R}^{l=i+1}(m_l + n_l) + j + k)$th time slot. Dominative nodes in $UD_{i,j}$ transmit data in the $(\sum_{l=R}^{l=i+1}(m_l + n_l) + n_i + j + k)$th time slot. We can imply that the latency of global aggregation is $\sum_{l=R}^{l=1}(m_l + n_l)$ slots.

4 Discussion

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

4.1 Correctness

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

Proof. Due to the page limit, we skip the proof here.

Theorem 2. *The maximum number of dominative neighbors for a connective node is 5.*

Proof. Due to the page limit, we skip the proof here.

Theorem 3. *The maximum number of connective neighbors of a dominative node is 12.*

Proof. Due to the page limit, we skip the proof here.

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

Proof. Recall that our schedule consists of local aggregation phase and global aggregation phase. As the latency bound of local aggregation phase is $\Delta - 1$ according to Theorem 1, we analyze the latency bound of global aggregation phase before we get the latency bound of data aggregation on a peony tree.

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, according to Theorem 2, a connective node has at most 5 dominative neighbors and one of them is its parent. So the number of dominative children of a connective node is at most 4. Consequently, the latency bound of data aggregation from dominative nodes in an even layer to connective nodes in the upper layer is 4.

Second, according to Theorem 3, a dominative node has at most 12 connective neighbors and one of them is its parent node. So the number of connective children of a dominative node is at most 11. Consequently, 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 $11(R-2)+12+4(R-1)=15R-14$.

Therefore, the latency bound of data aggregation on a peony tree is $(\Delta-1)+(15R-14)=15R+\Delta-15$.

4.2 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 5. *(Wegner Theorem [11]) The area of the convex hull of n ($n \geq 2$) non-overlapping unit-radius circular disks is at least*

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

Theorem 6. *$\Gamma(\rho)$, the upper bound of the number of independent nodes in a radius of ρ, satisfies the condition that*

$$2\sqrt{3}(\Gamma(\rho)-1)+(2-\sqrt{3})\lceil\sqrt{12\Gamma(\rho)-3}-3\rceil+\pi \leq (2\rho+1)^2\pi \tag{2}$$

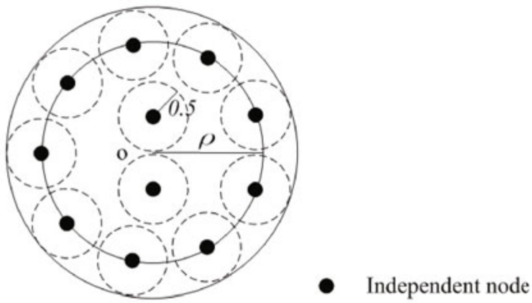


Fig. 1. The maximum number of independent node in a radius of ρ

Proof. Consider independent nodes in a disk of radius ρ centered at point o as shown in Fig. 1. 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 5, the area of the convex hull which covers all the small disks of radius 0.5 is at least $(2\sqrt{3}(\Gamma(\rho)-1)+(2-\sqrt{3})\lceil\sqrt{12\Gamma(h)-3}-3\rceil+\pi)/4$. 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 inequation 2.

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

1. Transmissions from white nodes to dominative nodes. When $\rho \leq 1$, 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 $\Gamma(1 + \rho)$.

Theorem 7. *The latency bound of data aggregation in a peony tree is*

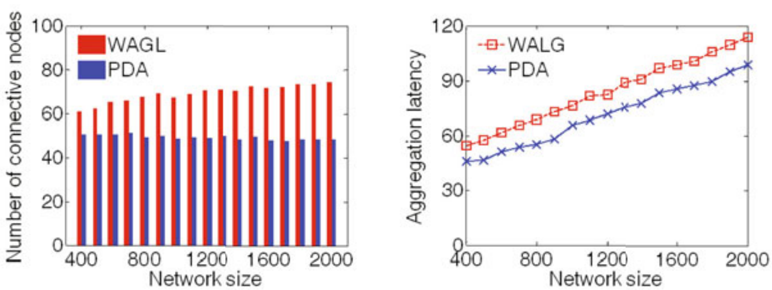
$$(\Gamma(\rho) + \Gamma(1 + \rho) - 2)(R - 2) + \Gamma(\rho)\Delta + \Gamma(1 + \rho) - 1 \tag{3}$$

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 $\Gamma(\rho)(\Delta - 1) + (\Gamma(\rho) - 1) + ((\Gamma(\rho) - 1) + (\Gamma(1 + \rho) - 1))(R - 2) + \Gamma(1 + \rho) = (\Gamma(\rho) + \Gamma(1 + \rho) - 2)(R - 2) + \Gamma(\rho)\Delta + \Gamma(1 + \rho) - 1$.

5 Performance Evaluation

In the simulations, we let all sensor nodes have the identical communication range r_c and identical interference range r_i, where $r_c = r_i = 10m$. Sensor nodes are randomly deployed into a $100m \times 100m$ area for multiple times to generate different network topologies. 20 different topologies are generated with network sizes (measured by the number of nodes) varying from 400 to 2000. We compare PDA with the algorithm proposed by Wan *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.

Figure 2(a) 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. The total aggregation latency by using PDA is about 20% shorter than that of using WALG, as shown in Fig. 2(b).



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

Fig. 2. Comparison with different network sizes

6 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.

References

1. Xujin, C., Xiaodong, H., Jianming, Z.: Minimum data aggregation time problem in wireless sensor networks. In: 1st International Conference on Mobile Ad-hoc and Sensor Networks, pp. 133–142. IEEE Press, Wuhan (December 2005)
2. Scott, C.-H.H., Peng-Jun, W., Chinh, T.V., et al.: Nearly constant approximation for data aggregation scheduling in wireless sensor networks. In: Proceedings of INFOCOM, pp. 366–372. IEEE Press, Anchorage (May 2007)
3. Ian, F.A., Weilian, S., Yogesh, S., Erdal, C.: Wireless sensor networks: a survey. Computer Networks 38(4), 393–422 (2002)
4. Mo, L., Yunhao, L.: Underground structure monitoring with wireless sensor networks. In: Proceedings of ACM/IEEE IPSN, pp. 69–78. ACM Press, Cambridge (April 2007)
5. Robert, S., Alan, M.M., Joseph, P., et al.: An analysis of a large scale habitat monitoring application. In: Proceedings of ACM SenSys, pp. 214–226. ACM Press, Baltimore (November 2004)
6. Mo, L., Yunhao, L.: Iso-map: Energy-efficient contour mapping in wireless sensor networks. IEEE Transactions on Knowledge and Data Engineering (TKDE) 22(5), 699–710 (2010)
7. Mo, L., Yunhao, L., Lei, C.: Non-threshold based event detection for 3d environment monitoring in sensor networks. IEEE Transactions on Knowledge and Data Engineering (TKDE) 20(12), 1699–1711 (2008)
8. Kebin, L., Mo, L., Yunhao, L., et al.: Passive diagnosis for wireless sensor networks. In: Proceedings of ACM SenSys, pp. 113–126. ACM Press, Raleigh (November 2008)
9. Lufeng, M., Yuan, H., Yunhao, L., et al.: Canopy closure estimates with greenorbs: Sustainable sensing in the forest. In: Proceedings of ACM Sensys, pp. 99–112. ACM Press, Berkeley (November 2008)
10. Alexander, K., Dariusz, R.K.: Fast distributed algorithm for convergecast in ad hoc geometric radio networks. J. Parallel Distrib. Comput. 66(4), 578–585 (2006)
11. Wegner, G.: Über endliche kreispackungen in der ebene. Studia Scientiarium Mathematicarium Hungarica 21, 1–28 (1986)