Construction of Optimal Data Aggregation Trees for Wireless Sensor Networks

Deying Li[1,2], Jiannong Cao[1], Ming Liu[1], Yuan Zheng[1]
[1]Internet and Mobile Computing Lab, Department of Computing
Hong Kong Polytechnic University
Hung Hom, KLW, Hong Kong
{csdli | csjcao | csmliu| csyuazheng}@comp.polyu.edu.hk
[2]School of Information
Renmin University of China, Beijing, China

***Abstract*— This paper considers the problem of constructing data gathering trees in a wireless sensor network for a group of sensor nodes to send collected information to a single sink node. Sensors form application-directed groups and the sink node communicates with the group members, called source nodes, to gather the desired data using a multicast tree rooted at the sink node [7]. The data gathering tree contains the sink node, all the source nodes, and some other non-source nodes. Our goal of constructing such a data gathering tree is to minimize the number of non-source nodes to be included in the tree so as to save energies of as many non-source nodes as possible. It can be shown that the optimization problem is NP-hard. We first propose an approximation algorithm with a performance ratio of four, and then give a distributed algorithm corresponding to the approximation algorithm. Extensive simulations are performed to study the performance of the proposed algorithm. The results show that the proposed algorithm can find a tree of a good approximation to the optimal tree and has a high degree of scalability.**

Keywords-wireless sensor networks; data collection and aggregation;

I. INTRODUCTION

The advances in wireless networking and microelectronics leads to the emergence of wireless sensor networks (WSNs), which consists of a large number of low-cost sensor devices [1]. The sensor nodes are capable of sensing, data processing, and wireless communicating with each other. They coordinate to establish a multi-hop ad hoc network, monitor specified tasks, and cooperatively transmit sensory data to the sink node, which is also called the base station. WSNs can be easily deployed in physical environments to collect information from an area of interest in a robust and autonomous manner. A wide range of civil and military applications has been proposed for WSNs [2, 3].

A WSN have certain characteristics that distinguish it from other types of wireless networks. Among these properties, energy efficiency is an overriding factor in designing protocols for WSNS because, in many applications (e.g. environment monitoring and battlefield surveillance), after deployment, recharging or replacing a large number of sensor batteries would be unaffordable expensive and time consuming. Usually, the sensor node is capable of operating in an active mode or in a low-power stand-by mode. So, to increase the lifetime of a sensor network, we should put redundant sensors to sleep and awaken them when they are needed.

This paper considers the problem of constructing data aggregation trees in a wireless sensor network for a group of source nodes to send sensory data to a single sink node. It has been noticed that bout 75% of energy consumption of a sensor node is on communication [4]. A significant portion of the communication in WSNs is for query dissemination and sensory data transmission. Given an application, a set of sensor nodes, called *source nodes*, are defined to sense the environment. Systematic dissemination of queries to and collection of the sensory data from the source nodes are two important functions of the sink node. Generally speaking, the optimal sets of paths for sending information from the source nodes to the sink node do not necessarily form a tree. However, if we consider data aggregation, there exists an optimal set of paths that form a tree [5]. In data aggregation, instead of sending every message to the sink node, intermediate nodes combine their data with those received from the children nodes to compute an aggregated value, and then send only a single message with the aggregated value to the sink node. Therefore, data aggregation is usually performed along a tree, called *aggregation tree* or *convergecast tree* [6]. One of the main objectives in designing such a data gathering tree is to conserve the sensor energies so as to maximize their lifetime.

In this paper, sensors form data collection groups and the sink node communicates with the group members, which are the source nodes, to gather the desired data using a multicast tree rooted at the sink node [7]. The data-aggregation tree contains the sink node, all the source nodes, and some other non-source nodes. The non-source nodes in the tree simply forward the received data without the function of aggregation. Our goal is to minimize the number of non-source nodes in the tree to save energies of as many non-source nodes as possible. We formulate the optimization problem which can be shown to be NP-complete. Then, we first propose an approximation algorithm with 4 performance ratio, and then give a distributed algorithm corresponding to the approximation algorithm. Simulation results show that the proposed algorithm can find a tree of a good approximation to the optimal tree and has a high degree of scalability.

The rest of paper is organized as follows. Section II presents some related work. Section III specifies the system model and model the problem which can be shown to be NP-complete. Section V describes a greedy algorithm and Section VI gives a 4-approximation algorithm. Section VII proposes a distributed algorithm corresponding to the approximation algorithm. Section VIII presents the simulation results. Section IX concludes the paper.

II. RELATED WORK

One of the key challenges in data gathering is how to conserve the sensor energies so as to maximize their lifetime. To this end, directed diffusion [8] was proposed as one of the basic paradigms for sensing environmental phenomenon. It incorporates data-centric routing and application-specific processing such as data aggregation. The key idea is to combine data from different sensors to eliminate redundant transmissions, and provide a rich, multi-dimensional view of the environment being monitored.

Yu *et al* [9] studied the problem of scheduling packet transmissions for data gathering on a given data aggregation tree. The objective is to minimize the overall energy dissipation of the sensor nodes in the aggregation tree subject to the latency constraint. They use a non-monotonic transmission energy model with the example modulation scaling technique [10], which is based on the observation in [11] that in many channel coding schemes, the transmission energy can be significantly reduced by lowering transmission power and increasing the duration of transmission.

Intanagonwiwat *et al* [12] studied energy efficiency of greedy aggregation, which is different from the previous diffusion approach for the opportunistic aggregation on a lowest latency tree. Greedy aggregation constructs a Greedy Incremental Tree (GIT) as follows: a shortest path is established for only the first source to the sink whereas each of the other sources is incrementally connected at the closest node on the existing tree. Simulation showed that the greedy aggregation saved energy cost considerably over the opportunistic aggregation without an adverse impact on latency or robustness.

Kalpakis *et al* [13] studied the problem of maximum lifetime data aggregation. Given the locations of sensors and the sink node, and the available energy at each sensor, the problem is how to produce a set of trees for collecting data from all the sensors and sending them to the sink node such that the system lifetime is maximized, where the lifetime of the system is defined as the number of rounds until the first sensor is drained of its energy. The problem is modeled using integer programming with linear constraints.

H. Du et al [14] addressed a real-time scenario where the data aggregation must be performed within a specified latency constraint. In order to minimize the overall energy cost of the sensor nodes subject to the latency constraint, an algorithm was proposed to construct a data aggregation tree with theoretically upper bounded energy cost under the latency constraint.

This paper targets at minimizing the number of active nodes to be included in the aggregation tree, assuming that each sensor node in the network has the same transmission power. For a given multicast in a unit-disk graph, we find a Steiner tree spanning all the nodes in the multicast group such that the number of Steiner nodes is minimized. To our knowledge, there is no known result on this problem.

III. SYSTEM MODEL AND PROBLEM STATEMENT

A WSN is abstracted as a connected undirected graph with the sink nodes and all other sensor nodes as the set of vertices and the bi-directional wireless communication links as the set of edges. This graph can be arbitrary depending on the deployment of sensor nodes. We assume that there is only one sink node, and every node has the same transmission range r. Thus, a WSN can be modeled by a unit disk graph (UDG). UDG is a well-known model for representing sensor networks and their underlying Markov chain [15]. A random UDG $G = (V, E)$ can be constructed as follows. Place $N = |V|$ nodes uniformly at random in a square area and then connect every pair of nodes at Euclidean distance less than or equal to r. Here, each vertex in vertex-set V represents a sensor node or the sink node in the network. The existence of an edge $(u, v) \in E$ between two nodes u and v indicates that the nodes can reach each other, i.e. the distance between u and v is at most the transmission range r.

We denote a data collection request by (s, P), where s is the sink node and $P \subset V$ is a set of source nodes from which s wants to collect the data of its interest. We abstract the underlying data gathering process as a *data aggregation tree* T defined for (s, P). T is a tree of G rooted at s that includes every sensor node in P and the sensed data packets flow from sensor nodes towards s in T. Data aggregation is performed by any non-sink and non-leaf node of T (called an *internal node* hereafter). We assume that each internal node in T is able to do data aggregation which is performed only after all input information is available, either received from the children or generated by local sensing. We further assume that the size of aggregated data, which will be sent to the parent node, remains unchanged. For simplicity, we do not consider the extra costs in time and energy for data aggregation. We only consider the energy cost for sending the data since it is much larger than that for receiving data [16]. Given a data aggregation tree T, since every node, except the sink, will send the data to its parent node once (and exactly once without collision), the energy cost of T is then defined as the sum of energy costs of all sensor nodes in T, or formally as,

$$C(T) = \sum_{u \in T - \{s\}} p(u).$$

Now, we can formally define the problem studied in this paper as follows: given a UDG $G=(V, E)$ and a data gathering request (s, P), construct a data aggregation tree T for (s, P) such that the energy cost $C(T)$ is minimized. In fact, the problem can be formulated as finding a Steiner tree of a UDG with the minimum number of Steiner nodes. We define this problem, called ST-MSN, as follows:

Problem (ST-MSN): Given a unit-disk graph $G=(V, E)$, where V is a set of nodes in Euclidean plane, and a subset

$M = \{s\} \cup P$ of V, where s is a sink node and P is a set of source nodes, find a Steiner tree interconnecting nodes in M with the minimum number of Steiner nodes.

It can be shown that Problem (ST-MSN) is NP-hard. Due to limit in space, here we omit the proof.

IV. An Greedy Algorithm

Since the ST-MSN problem is NP-complete, we need to find a good polynomial time approximation algorithm. In this section, we propose a greedy algorithm for ST-MSN.

Suppose $\{s\} \cup P$ is a multicast group in $G=(V, E)$, where s is the sink node. The greedy algorithm constructs a multicast tree by iteratively adding source nodes to the existing tree till the tree includes all the source nodes and the sink node. Initially, the tree includes only the sink node. Each time the algorithm finds a source node among the remaining source nodes which is closest to the existing tree, and then adds the shortest path between that source node and the existing tree to the tree. This process continues until all the source nodes have been included in the tree. Before giving a formal algorithm, we first define some notations.

Definition: Suppose $G=(V, E)$ is graph, u is a node in V, T is a subgraph of G not including u, the distance between u and T, $d(u, T)$, is defined as

$$d(u,T) = \min_{v \in T}\{d(u,v)\}$$

The shortest path between u and T is the shortest path between u and v where $v \in T$ and $d(u, T)=d(u, v)$.

The greedy algorithm is formally presented as follows:

Algorithm: Greedy Algorithm

Input: $G=(V,E)$ and $\{s\} \cup P$ (s = sink node and $P \subset V$)

Output: Multicast tree spanning on $\{s\} \cup P$

Set $T = \{s\}$, $Q = P$

While $Q \neq \Phi$

Find $u \in Q$ such that $d(u,T) = \min_{u \in Q}\{d(u,T)\}$

$T = T \cup \{\text{the shortest path between } u \text{ and } T\}$

$Q = Q - \{u\}$

The above greedy algorithm takes $O(n^4)$ time. Because finding the shortest path takes $O(n^3)$ time, each loop takes time $O(n^3)$. There are $|P|$ loops, where $|P|$ is at most n, therefore the algorithm takes a total time of $O(n^4)$.

V. An Approximation Algorithm

In order to further reduce the computational complexity and improve the quality of output solution, we design another heuristic here. We propose an approximation algorithm based on minimum spanning tree to construct the data aggregation tree, called the *AT algorithm.*

The AT algorithm includes three steps: (1) Construct an auxiliary graph which is a weighted complete graph on P; (2) Compute a minimum spanning tree on the auxiliary graph; (3) Replace each edge in the minimum spanning tree by the corresponding shortest path in the original graph. The AT algorithm is formally described as follows.

Algorithm: AT Algorithm

Input: a UDG $G=(V, E)$ and a multicast group $\{s\} \cup P$

Output: A Steiner tree spanning on $M=\{s\} \cup P$

Step1: Construct an auxiliary graph which is a complete weighted graph $H = (P, E(P))$, where P is the set of nodes, and $\forall u \in P, v \in P$, there is an edge $(u, v) \in E(P)$ with a weight equal to the number of edges in the shortest path between u and v in the original unit-disk graph G.

Step2: Compute a minimum spanning tree T in H.

Step3: Replace each edge of T with a corresponding path in G. Then delete cycles to get an aggregation tree.

To illustrate the steps of the proposed approximation algorithm, let us consider the example in Figure 1. A unit disk graph G with 15 nodes is given, where the set of black nodes, $M = \{3,8,9,12,13\}$, is the multicast group of the source nodes. Figure 2(a) shows the weighted complete graph on the multicast group, where the weight of an edge (a, b), $\forall a,b \in V(G)$, is the number of edges in the shortest path between the nodes a and b on the original graph G. Figure 2(b) shows a minimum spanning tree corresponding to Figure 2(a). Figure 3(c) shows the corresponding Steiner tree spanning all the nodes in M, where the Steiner tree is obtained by changing every edge on the minimum spanning tree to corresponding shortest path on G.

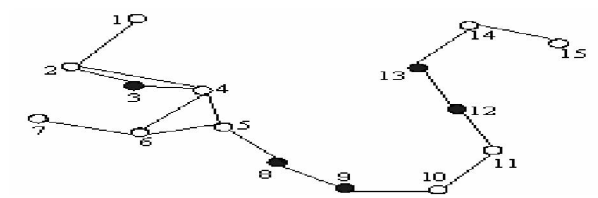


Figure 1. An Example Unit Disk Graph G

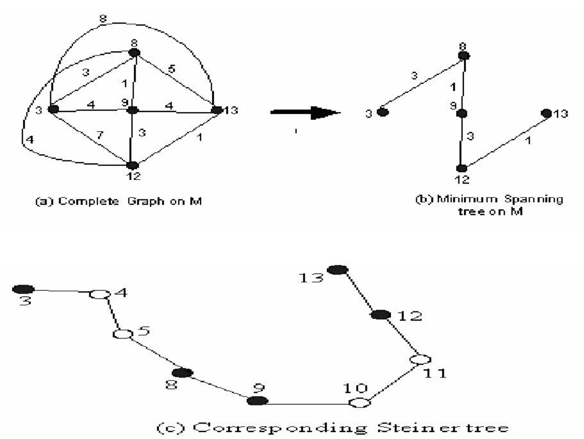


Figure 2. Steps in the AT algorithm

The proposed AT algorithm can achieve a very good performance ratio of four and the time complexity of AT algorithm is $O(n^3)$. Again, due to limit in space, the proofs are omitted here.

VI. DISTRIBUTED VISION FOR AT ALGORITHM

In the AT algorithm, the global information is assumed to be known before computing the data-gathering tree. In this section, we give a distributed vision of the AT algorithm, which does not require the sensor nodes to know the topology of the network.

All the nodes have a unique identity, but they do not know the identity of any other node. Each node knows the links incident to it. All the nodes in the multicast group $\{s\} \cup P$ are aware of their group membership and the sink node knows itself as the sink node. For the duration of the algorithm, the network is assumed to be reliable, i.e., there is no link or node failure in the network.

The main idea of the distributed AT algorithm can be summarized in two steps, shown below.

Algorithm: Distributed AT Algorithm:

Step 1: Using the distributed algorithm in [17] to find all-pairs shortest path in the network.

Step 2: After finishing step 1, the sink node knows all-pairs shortest paths in the multicast group, which contains all source nodes. Using the auxiliary graph method mentioned in section VI, the sink node can find minimum Steiner spanning tree on the multicast group.

In more details, the distributed AT algorithm consists of five parts, where parts 1-4 are similar to the algorithm in [17].

Part 1. Find a spanning tree, T, of the un-weighted graph G. Initially, all the nodes are inactive. Here, we use the algorithm given by Awerbuch [18] to find a spanning tree, T, of the underlying un-weighted graph. This part needs $O(|E|+V\log_2|V|)$ time. After the spanning tree is found, the leaf nodes are marked Active and non-leaf nodes are marked Available. In addition, the spanning tree algorithm ensures that every node can identify the links incident on it.

Part 2. Each node determines the identities of its neighbors in the graph G. This can be accomplished by letting all active and available nodes to broadcast their own identities along each link incident on them. The message complexity of this step is $2|E|$.

Part 3. Determine the All-Pairs Shortest-Distance matrix. Each node constructs a distance matrix which has row and column labels corresponding to the node and its neighbors. Distance information is transmitted along the tree edges starting at each leaf node. As the algorithm proceeds, new columns and rows are added to the distance matrix and existing distance information is updated. At the end of this part, exactly one or two nodes will contain the Shortest-Distance matrix, D, of the entire graph G, and each (i,j) entry of D will hold the shortest distance between node i and node j. The message complexity and the time complexity of this part are both $O(n)$.

Part 4. Broadcast the All-Pairs Shortest-Distance matrix to the sink node. When a non-sink node receives a message containing the matrix, it simply forwards this message by broadcast. Only the sink node maintains the all-pairs Shortest-Distance matrix. This part has a time complexity of $O(|V|)$ and message complexity of $O(|V|)$.

Part 5. Sink node multicasts a message to every node of the multicast tree along the tree edges. The sink node can find the final Steiner tree based on the information acquired from Part 4. Finally, the sink node sends a multicast message to each node of the multicast tree along the tree edges. This part needs $O(|V|)$ messages and $O(|V|)$ time.

The distributed AT algorithm terminates in $O(|V|)$ time and has a message complexity of $O(|E|+|V|\log|V|)$, where $G=(V, E)$. The proof is omitted here.

VII. PERFORMANCE EVALUATION

We now describe the the performance evaluation of the proposed approximation algorithm. There are three objectives: (a) to study the deviation of the results of the AT algorithm (AT) from the optimal algorithm using exhaustive search (OA); (b) to study the scalability of the AT algorithm when the number of sensor nodes increases; (3) to compare the performance of the AT algorithm with the Greedy algorithm(GA).

A. Simulation Setup

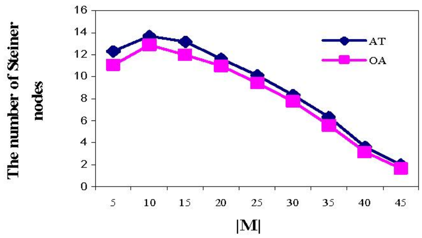
The simulation is conducted in a 1000×1000 2-D free-space by independently and uniformly allocating N nodes into the area. All nodes have the same transmission range R. For any pair of nodes, if their distance is no more than R, there exists an edge between the two nodes. The results presented here are the averages of 100 separate simulation runs. In each run, we place N nodes in the square and select M nodes to form a multicast group, both done randomly. Any topology that is not connected is discarded. For the first goal, we first run the AT algorithm and determine all Steiner nodes, and then get the optimal solution by exhaustive search. For the second goal, in order to study the scalability of our approximation algorithm(AT), we run the approximation algorithm(AT) and determine all Steiner nodes for large values.

B. Evaluation Resutls

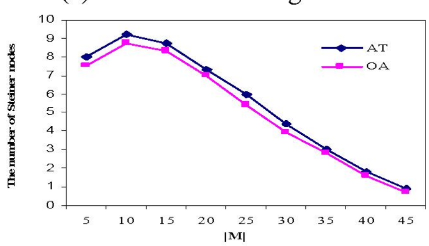
The approximation AT algorithm and the optimal OA algorithm are simulated and their performances are compared. Figures 3(a)-(d) show the number of Steiner nodes in the Steiner tree constructed on the multicast group as a function of M. The values of N and R are fixed while the value of M varies. In the four figures, N=50, and R is 80, 100, 125, 150, respectively. As we can see, the number of Steiner nodes in Steiner trees increases as M increases, but when M increases to some value, the number of Steiner nodes decreases with the growth of M. This is because when M reaches a certain value, with the growth of M, the number of the transmitting nodes in a multicast group that can connect each other increases, therefore, the number of Steiner nodes decreases.

We can observe from Figure 3(a)-(d) that the result of the AT algorithm is close to the optimal solution. This shows that our approximation algorithm has a very good performance.

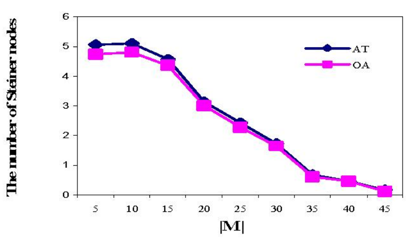
The second set of experiments is carried out for measuring the scalability of the proposed AT algorithm. For given transmission range values R = 80, 100, 120, we show the simulation results with the number of nodes N = 500, 1000, respectively. Figure 4(a)-(c) show the number of Steiner nodes as the function of M. We see that the number of Steiner nodes when N=500 is very close to the number of Steiner nodes when N=1000, so the scalability of our AT algorithm is very good.



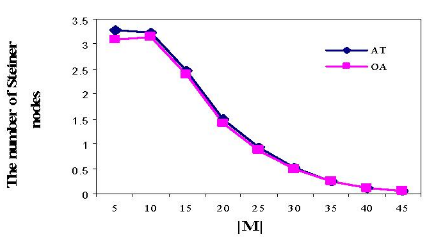
(a) Transmission Range R=80



(b) Transmission Range R=100



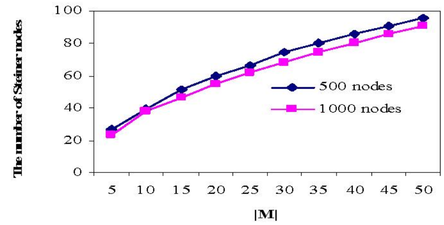
(c) Transmission Range R=125



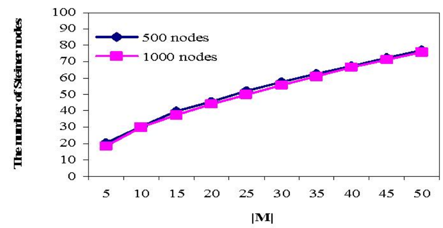
(d) Transmission Range R=150

Figure 3 Comparing AT with OA

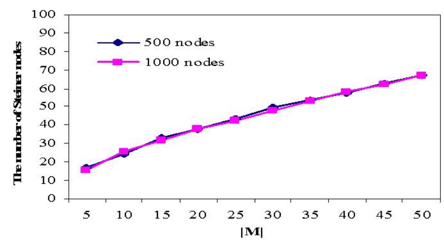
The third set of experiments is carried out for comparing the AT algorithm with the GA algorithm. For given transmission range values R = 80, 100, 120, we show the results with the number of nodes N = 500, 1000, respectively. Figure 5(a)-(c) show the number of Steiner nodes as a function of M when N=500, and Fig. 6(a)-(c) show the results when N=1000. We can see that the number of Steiner nodes in AT is very close to the Steiner nodes in GA algorithm. Although the Greedy algorithm has no bound ratio, its performance is also very good.



(a) Transmission Range R=80

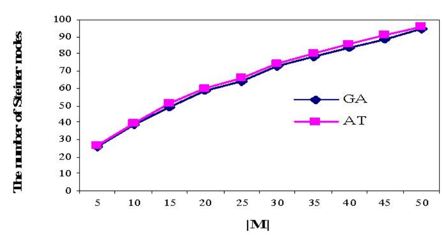


(b) Transmission Range R=100

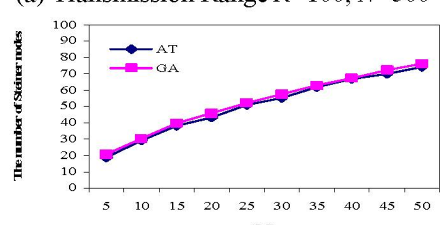


(c) Transmission Range R=120

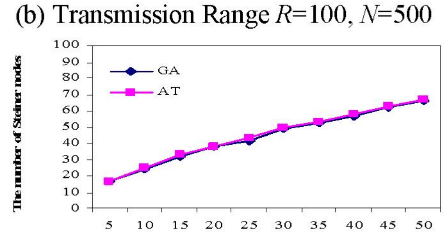
Figure 4 the Scalability of AT algorithm



(a) Transmission Range R=100, N=500

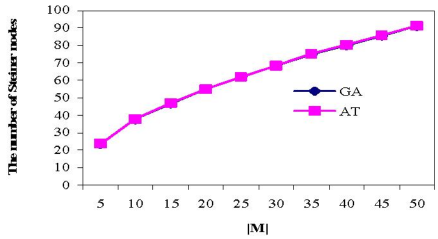


(b) Transmission Range R=100, N=500

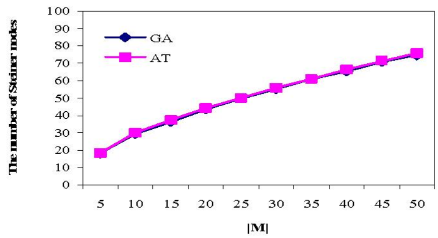


(c) Transmission Range R=120, N=500

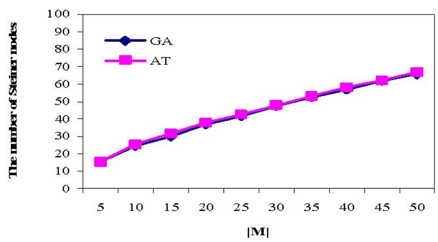
Figure 5 Comparing AT with GA when N=500



(a) Transmission Range R=80, N=1000



(b) Transmission Range R=100, N=1000



(c) Transmission Range R=120, N=1000

Figure 6 Comparing AT with GA when N=1000

VIII. CONCLUSIONS

This paper studies how to construct an optimal multicast tree for data aggregation in WSNs involving minimum number of non-source nodes. The objective is to save energies of as many non-source nodes as possible. We first proposed a greedy algorithm, and then proposed approximation algorithm based on minimum spanning tree that can construct a data aggregation tree with the performance ratio of four. We also give a distributed version of the approximation algorithm. Extensive simulations have been performed to evaluate the performance of the proposed algorithms, and the obtained results validated that the approximation algorithm can construct a tree of a good approximation to the optimal tree and has a high degree of scalability.

ACKNOWLEDGMENT

This research is partially supported by the Hong Kong Polytechnic University under the research grants G-YX28 and G-YE57 and the National Natural Science Foundation of China.

REFERENCES

[1] D. Culler, D. Estrin, and M. Srivastava, “Overview of sensor networks”, Computer, Vol. 37, No. 8, pp. 41-49, 2004.

[2] C.-Y. Chong and S. P. Kumar, “ Sensor Networks:Evolution, Opportunities, and Challenges”, Proceedings of the IEEE, Vol. 91, No. 8, pp.1247–1256, 2003.

[3] D. Estrin et al. Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers The National Academies Press, Washington DC, 2001.

[4] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, “Wireless sensor networks for habitat monitoring", *The 1st ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, USA, pp. 88-97, Sept 2002.

[5] A. Goel and D. Estrin, “Simultaneous Optimiation for Concave Costs: Single Sink Aggregation or Single Source Buy-at-Bulk”, Proc. 14th ACM-SIAM Symposium on Discrete Algorithms, 2003.

[6] H. Karl, M. Lobbers, and T. Neberg, “A data aggregation framework for wireless sensor networks”, Technical report TKN-03-016, Telecommunication Network Group, Technical University Berlin, Sept. 2003.

[7] S. Tilak, N.B. Abu-Ghazaleh, and W. Heinzelman, “A Taxonomy of Wireless Micro-Sensor Nework Models”, Mobile Computing and Communications Review, Vo. 1, No. 2.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks”, The 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Boston, MA, Aug 2000.

[9] Yang Yu, B. Krishnamachari, and V. K. Prasanna, “Energy-latency tradeoffs for data gathering in wireless sensor networks”, The 23rd Conference of IEEE Communication Society (INFOCOM), Hong Kong, SAR China, Mar 2004.

[10] C. Schurgers, O. Aberhorne, and M. B. Srivastava, “Modulation scaling for energy-aware communication systems”, International Symposium on Low Power Electronics and Design (ISLPED), Huntington Beach, CA, Aug 2001.

[11] B. Prabhakar, E. Uysal-Biyikoglu, an dA. E. Gamal, “Energy-efficient transmission over a wireless link via lazy packet scheduling”, The 20th Conference of IEEE Communication Society (INFOCOM), Anchorage, Alaska, April 2001.

[12] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, “Impact of network density on data aggregation in wireless sensor networks”, The 22nd International Conference on Distributed Computing Systems (ICDCS), Austria, July 2002.

[13] K. Kalpakis, K. Dasgupta, and P. Namjoshi, “Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks”, The International Journal of Computer and Telecommunications Networking, Vol. 42, No. 6, pp. 697-716, 2003.

[14] H. Du, H. Hu, and X. Jia, “Energy Efficient Routing and Scheduling for Real Time Data Aggregation in Wireless sensor networks”, accepted by Computer Communications, 2005.

[15] B.A. Clark and C.J. Colbourn, “Unit Disk Graph”, Discrete Mathematics, Vol. 86, pp. 165-177, 1991.

[16] W. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocols for wireless microsensor networks”, *The 34th Hawaiian International Conference on Systems Science* (*HICSS*), Island of Maui, Hawaiian, Jan 2000.

[17] S. Kanchi and D. Vineyard “An Optimal Distributed All pairs shortest path problem” http://www.kettering.edu/acad/scimath/compsci/techrept/2000/2000-5.pdf

[18] B. Awerbuch, “Optimal distributed algorithms for minimum-weight spanning tree, counting, leader election and related problems”, in *Proc. 19th ACM Symp. On Theory of Computing, ACM, New York*, 1987, pp.230-240