DATA GATHERING USING M-COLLECTOR IN WSNs

Gnanasampath L[#1], Dhinesh J[#2], Deepthi Dhivya D[*3], Sivaranjani S[*4],A.R.Surendheran[*5]

[#]*Department of Computer Science and Engineering, Anna University*
K.S.R.College of Engineering, Tiruchengode-637 215,
Namakkal district, Tamil Nadu,India
[1]`splgnanasampath@gmail.com`
[3]`deepthi.dhivya@gmail.com`
[5]*surenderan26@gmail.com*
[*] *K.S.R.College of Engineering, Tiruchengode-637 215,*
Namakkal district, Tamil Nadu,India
[2]`j.dhinesh1414@gmail.com`
[4]`sivaranjanirms@gmail.com`

Abstract- In the existing system we have multiple M-collectors to collect data from sensor nodes. Each M-collector will collect data from sensors which are in its range. Data will be transmitted while an M-collector comes near to another M-collector. This takes much time for the transmission. There may be a chance of collision while transmitting data between the collectors. This is a multi hop transmission. To avoid this we are going to replace the multiple M-collectors with single M-collector and giving a rendezvous point for each group. Each rendezvous point will gather data from sensors which are in its range. The data which are in rendezvous point will be transmitted to the M-collector while the M-collector comes near to the range of rendezvous point. M-collector collect all the data from rendezvous point and transmit it to the base station. This system will increase the network life time by minimizing data gathering delay in wireless sensor network.

Key words: M-collector, rendezvous point, mobility, data gathering

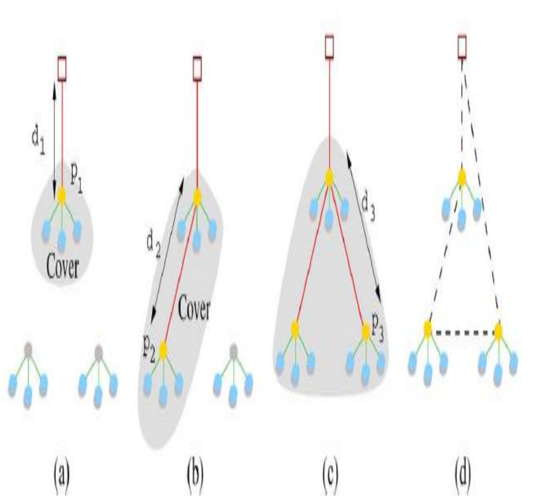
INTRODUCTION

Wireless sensor networks (WSNs) have tremendous range of applications, such as medical treatment, outer-space exploration, battlefield surveillance, emergency response, etc which is emerged as a new information gathering paradigm. Without a preconfigured structure sensors are deployed into a large scale sensing field. Nodes are discovered using the sensor nodes which are placed near to those nodes and it can be organized them into a network and used to monitor the environment. Sensing the field and uploading data to the data drop are two major challenges for consuming energy of the sensor. Mobile base stations are the destination of information. They collect the data sensed by sensor nodes either directly or through intermediate nodes

A mobile data collector could be a mobile robot or a vehicle with the transceiver to gather data from the sensors. It starts its travel from the mobile base station, navigates the network, collects the data from each node and uploads it to the mobile base station. Multiple M-collectors are used to navigate through shorter sub tours. Each M-collector travels through the group and gathers the data. So it causes time delay.

HEURISTIC ALGORITHMS FOR THE SINGLE-HOP DATA-GATHERING PROBLEM

The SHDGP is NP-hard, we will now develop a heuristic algorithm to solve the problem approximately. First, it is interesting to compare the SHDGP with a similar problem, i.e., the covering salesman problem (CSP).[3] Current and Schilling defined the CSP and proved its NP-hardness. The problem they considered is obtaining the shortest tour of a subset of all cities such that every city not on the tour is within some predetermined distance dist of a city that is on the tour. If the transmission range of each sensor could be modeled as a disk-shaped area, the SHDGP can be simplified to the CSP by setting dist in the CSP equal to the transmission range of sensors. A heuristic solution was introduced by dividing the problem into two NP-hard sub problems. First, find a minimum vertex cover and then determine the shortest tour of all vertices in the cover. The first step of their heuristic is to minimize the number of stops (same as polling points in the SHDGP), but it does not consider the lengths of the edges connecting these stops.



DATA GATHERING WITH MULTIPLE M-COLLECTORS

For some large-scale applications, each data-gathering tour may take such a long time that a single M-collector may not be sufficient to visit the transmission ranges of all sensors before their buffers overflow. A possible solution[4] to this problem is to allow some sensors to relay packets from other nodes to the mobile data collector. Thus, the M-collector does not need to visit the transmission range of Every single sensor and the length of each tour can be reduced. However, the drawback of using relay is that some relaying nodes may fail faster than others. To avoid unbalanced network Lifetime, we will stay with the one-hop data-gathering scheme by utilizing multiple M-collectors. The data-gathering algorithm with multiple M-collectors can be described as follows. First, find the polling point set P by running the spanning tree covering algorithm. Then, find the minimum spanning tree T(V,E) on polling points. We refer to the minimum spanning tree on polling points as the spanning covering tree. Let Lmax be the upper bound on the length of any subtour, which guarantees the data to be collected before sensors run out of storage. Note that Lmax could depend on a lot of factors, such as the buffer size, the data acquisition rate of sensors, and the moving speed of M-collectors. Let t(v) denote the subtree of T, which is rooted at vertex v and consists of all child vertices of v and edges connecting them in T. Let Parent{v} be the parent vertex of v in T. Let Weight{v} represent the sum of all link costs in the subtree t(v) rooted at v. Then, calculate the weight values of all vertices in T. Repeatedly remove subtrees from T until no vertex is left in T. To build a subtree t in each loop, start from the deepest leaf vertex of the remaining T, and let it be the root Root(t) of the sub tree t. Check the weight of Parent(Root(t)), and let Root(t) = Parent(Root(t)) if Weight(Parent(Root(t))) ≤ Lmax/2. Otherwise, add all child vertices of Root(t) and edges connecting them in T into t and remove t from T. Here, Weight (Parent (Root (t))) also denotes the total edge length of sub tree t. After removing the sub tree, upgrade the weight value of each vertex in the remaining T. The algorithm terminates when T is empty. Then T is decomposed into a set of sub trees.

The total length of any sub tree t, which is denoted by Lt, is no more than Lmax/2. Finally, the subtour on polling points of each subtree can be determined by running the approximation algorithm for the TSP. Let Lt apx be the length of the approximated subtour on points in subtree t. In the 2-approximation algorithm for the TSP, the approximated tour is obtained by duplicating all edges of the minimum spanning tree and then finding an Eulerian circle in it. Hence, Lt apx is no more than two times the length of the minimum spanning subtree t Lt, that is,Lt apx ≤ 2 × Lt. As discussed earlier, Lt is bounded by Lmax/2.

Thus, we have Lt apx ≤ 2 × Lt ≤ Lmax, which means that the length of any subtour obtained by the data-gathering algorithm with multiple M-collectors is no more than the upper bound on the length of a subtour Lmax.

The data-gathering algorithm with multiple M-collectors:

1) Build the spanning covering tree;

2) decompose the spanning covering tree into a set of sub trees;

3) Find an approximate shortest subtour on the points of each subtree; and

4) sensing data collected from sensors are forwarded to the nearest Mcollector to the data sink. The complexity of the data-gathering algorithm with multiple M-collectors. Both outer and inner loops need to be executed at most O(M) rounds, where M is the number of candidate polling points. The loop takes O(M2) time, and the operations before the loops take another O(NM +M2) time. Thus, the total computational complexity is O(NM +M2).

```
Data Gathering Algorithm with Multiple M-collectors
Find the polling point set $P$
Find the spanning covering tree $T$ on all polling points in $P$
For each vertex $v$ in $T$, calculate the weight value $Weight(v)$
while $T \neq \Phi$
  Find the deepest leaf vertex $u$ in $T$
  Let the root of the subtree $t$, $Root(t) = u$
  while $Weight(Parent(Root(t))) \leq \frac{L_{max}}{2}$
    $Root(t) = Parent(Root(t))$
  end while
  Add all child vertices of $Root(t)$ and edges connecting
    them into $t$ and remove $t$ from $T$
  Update weight value of each remaining vertex in $T$
end while
```

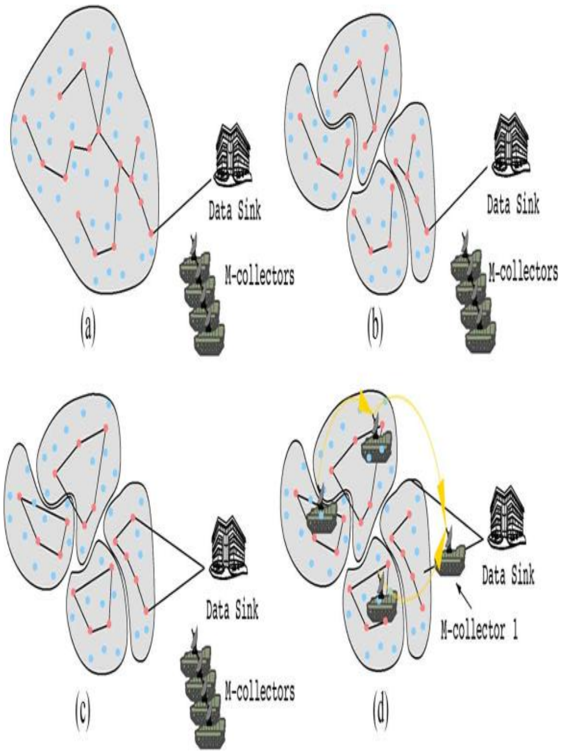


Fig.2 Data gathering using multiple M-Collectors a) Build the spanning tree b) Decompose into sub trees c) Find the shortest path in each group d) collects data from each group and forwards to the nearest M-collector to the base station

```
Spanning Tree Covering Algorithm
Create an empty set $P_{curr}$
Create a set $U_{curr}$ containing all sensors
Create a set $L$ containing all candidate polling points
while $U_{curr} \neq \Phi$
  Find a polling point $l \in L$, which minimizes $\alpha = \frac{cost\{nb(l)\}}{|nb(l) \bigcap U_{curr}|}$
  Cover sensors in $nb(l)$
  Add the corresponding polling point of $nb(l)$ into $P_{curr}$
  Remove the corresponding polling point of $nb(l)$ from $L$
  Remove sensors in $nb(l)$ from $U_{curr}$
end while
Find an approximate shortest tour on polling points in $P_{curr}$
```

SINGLE M-COLLECTOR SYSTEM

This paper aims at increasing the network life time and reducing the cost. It can be done so by introducing a back-up device in each group. In each sub tour place, the rendezvous nodes are placed. WSNs contain many partitions and the data generated in each sensor can be accumulated at designated sensors. These designated nodes buffer collected data until they are relayed to a mobile data collector. A similar method can also be used in connected networks to reduce the communication load (and energy consumption). Solutions that propose data gathering by a mobile device constitute the class of rendezvous-based solutions. Data is relayed over multiple hops before being delivered to the mobile device. It collects the data from all sensors. Nearby sensors directly transmit the data to base station. Nodes are created and grouped based on the transmission range specified. Back-up device is selected based on two constraints,

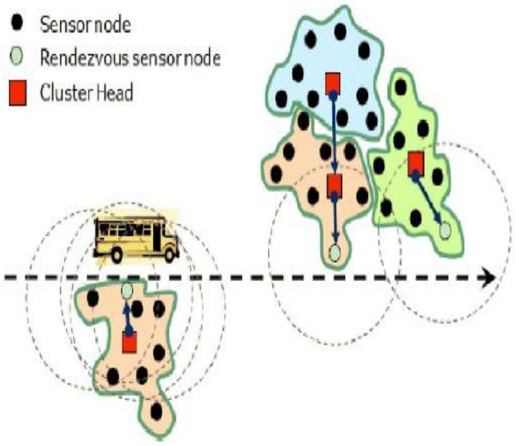
1) It should have the highest energy among all the sensors in the group.

2) It should be intermediate to both the mobile collectors and the group.

The rendezvous based solutions and the proposed protocol selects Polling Points that are in close proximity with the Mobile collector trajectory. Mobile collector is used to collect data from groups of SNs. During a training period, all the WSN edge nodes located within the range of Mobile collector routes are appointed as Polling Points and build paths connecting them with the remainder of sensor nodes. Those paths are used by remote nodes to forward their sensory data to Polling Points. The movement of mobile robots is controllable which is impractical in realistic urban traffic conditions. Most importantly, no strategy is used to appoint suitable nodes as Polling Points while selected Polling points are typically associated with uneven numbers of SNs. The secret meeting based solutions are presented for variable as well as fixed Mobile collector trajectories. The solution presented for fixed Mobile collector track seeks to determine a segment of the Mobile collector track shorter than a certain bound such that the total cost of the trees connecting source nodes with PP is minimized. The whole algorithm is performed centrally at the BS. Apart from, a number of other Polling Point-based solutions that assume variable Mobile collector trajectory have been proposed. These works determine the Mobile collector trajectory in such way that certain optimization criteria (e.g., minimum energy consumption for transferring the data to PPs) are met while obeying certain constraints (e.g., the MC trajectory length should be lower than a certain threshold). A common characteristic of all techniques described above is that the routing structures that carry data from SNs to PPs are built once and are used without any modification for the whole lifetime of the WSN. Most of these works are centralized approaches that try to minimize an energy related cost function without paying proper attention to the selection of nodes that will serve as PPs. Specifically, they do not take into account the contact time of a PP with the MC during which it can send the buffered data. Also, there is no special focus on the amount of data the PPs receive from the other nodes of the network. So, a heavily loaded PP that is in contact with the MC for only a short time may not manage to transfer all buffered data and this gradually may lead to buffer overflow or very long delivery delays. Also, they do not examine the proximity of the selected PPs and as a result, frequent collisions could arise due to concurrent transmissions from nearby PPs when the MC is approaching these PPs. Apparently, this considerably reduces the actual data delivery rate to the MC. Note also that many of the previous works provide an on time delivery guarantee by bounding the length of MC trajectory. The main trade-off that should be considered is between the delivery delay tolerated and the energy consumption due to multi hop routing to the PPs. Another issue in all previous schemes is that there is no provision in case that PPs run out of energy. In that case, all SNs that send their data to these PPs cannot send their data to MC any longer. A local or even a global rebuilding of the routing structures may be required in order to bypass dead PPs

GROUPING

The large-scale deployment of WSNs and the need for data aggregation necessitate efficient organization of the network topology for the purpose of balancing the load and prolonging the network lifetime. Grouping has proven to be an effective approach for organizing the network in the above context. Besides achieving energy efficiency, clustering also reduces channel contention and packet collisions, resulting in improved network throughput under high load. Our clustering algorithm borrows ideas from the algorithm of Che ne tal. To build a cluster structure of unequal clusters. The clustering algorithm in constructs a multisized cluster structure, where the size of each cluster decreases as the distance of its cluster head from the base station increases. We slightly modify the approach of to build clusters of two different sizes depending on the distance of the cluster heads(CH) from the MC's trajectory. Specifically, SNs located near the MC trajectory are grouped in small sized clusters while SNs located farther away are grouped in clusters of larger size. The CHs near the MC trajectory are usually burdened with heavy relay traffic coming from other parts of the network. By maintaining the clusters of these CHs small, CHs near the MC trajectory are relatively relieved from intra cluster processing and communication tasks and thus they can afford to spend more energy for relaying inter cluster traffic to PPs.



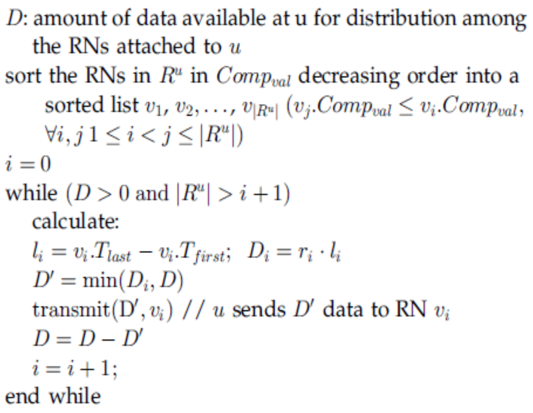
Data forwarding paths in MobiCluster

DATA AGGREGATION AND FORWARDING TO THE RNS

The steady phase of MobiCluster protocol starts with the periodic recording of environmental data from sensor nodes with a Tr period. The data accumulated at individual source nodes are sent to local CHs (intra cluster communication) with a Tc period (typically, Tc is a multiple of Tr). CHs perform data processing to remove spatial-temporal data redundancy, which is likely to exist since cluster members are located maximum two hops away. CHs then forward filtered data toward remote CH they are attached to. Alongside the inter cluster path, a second-level of data filtering may apply.

Upon reaching the end CH u, filtered data are forwarded to u's local PPs in a pipeline fashion. In the case that multiple PPs exist in that cluster, data are not equally distributed among them. Instead, the CH favors the data delivery by the most suitable PPs, those with highest competence value (Compval). Data distribution among PPs should ensure that each PP will be able to accommodate its assigned data, i.e., to deliver all its buffered data and not experience an outage. Hence, CH u sorts the PPs in its Ru set in Compval decreasing order and delivers to each PP node vi 2 Ru the maximum amount of data Di it can accommodate, minus an “outage prevention allowance” amount O. The Di value is calculated taking into account the PP’s data rate ri and the length li of the time interval [vi.Tfirst, vi.Tlast] that vi remains within the MC’s range. The process is repeated for each vi 2 Ru until all data available at u are distributed among its PPs. The algorithm executed by each CH u for distributing data to the PPs attached to it.

ALGORITHM:
DATA_ DISTRIBUTION



```
$D$: amount of data available at u for distribution among
    the RNs attached to $u$
sort the RNs in $R^u$ in $Comp_{val}$ decreasing order into a
    sorted list $v_1, v_2, \ldots, v_{|R^u|}$ $(v_j.Comp_{val} \leq v_i.Comp_{val}$,
    $\forall i, j\ 1 \leq i < j \leq |R^u|)$
$i = 0$
while $(D > 0$ and $|R^u| > i + 1)$
    calculate:
    $l_i = v_i.T_{last} - v_i.T_{first};\quad D_i = r_i \cdot l_i$
    $D' = \min(D_i, D)$
    transmit$(D', v_i)$ // $u$ sends $D'$ data to RN $v_i$
    $D = D - D'$
    $i = i + 1;$
end while
```

COMMUNICATION BETWEEN PPs AND MOBILE COLLECTOR

MobiCluster protocol involves the delivery of data buffered to PPs to MC. Data delivery occurs along an intermittently available link. Hence, a key requirement is to determine when the connectivity between an PP and the MC is available. Communication should start when the connection is available and stop when the connection no longer exists, so that the PP does not continue to transmit data when the PP is no longer receiving it. To address this issue, an acknowledgment-based protocol between PPs and MC are to be produced. The MC, in all subsequent path traversals after the setup phase, periodically broadcasts a POLL packet, announcing its presence and soliciting data as it proceeds along the path. The POLL is transmitted at fixed intervals T poll (typically equal to T beacon). This POLL packet is used by PPs to detect when the MC is within connectivity range. The PP receiving the POLL will start transmitting data packets to the MC. The MC acknowledges each received data packet to the PP so that the PP realizes that the connection is active and the data were reliably delivered. The acknowledged data packet can then be cleared from the PP’s cache. More details about the communication protocol between PPs and MC can be found in Appendix C, available in the online supplemental material. The last phase of MobiCluster protocol involves the delivery of data buffered to PPs to MC. Data delivery occurs along an intermittently available link; hence, a key requirement is to determine when the connectivity between PP and the MC is available. Communication should start when the connection is available and stop when the connection no longer exists, so that the PP does not continue to transmit data when the MC is no longer receiving it. The MC, in all subsequent path traversals after the setup phase, periodically broadcasts a POLL packet, announcing its presence and soliciting data as it proceeds along the path. The POLL is transmitted at fixed intervals Tpoll (typically equal to Tbeacon). This POLL packet is used by PPs to detect when the MC is within connectivity range. The PP receiving the POLL will start transmitting data packets to the MC. The MC acknowledges each received data packet to the PP so that the PP realizes that the connection is active and the data were reliably delivered. The acknowledged data packet can then be cleared from the PP’s cache.

CONCLUSIONS

Thus our paper reduces the cost for M-collector. We are using single M-collector instead of multiple M-collectors. This will reduce the much amount of cost. Then the time delay. The travel distance will be much reduced when compared to multi M-collectors deployed field. The future extension will consist of activating sleep mode in sensors while the sensor is idle.

REFERENCE

1. B. Sauser. (2007). Networking the Hudson River. [Online]http://www.technologyreview.com/Infotech/19309/page1/?a=f
2. I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, “Data collection, storage and retrieval with an underwater sensor network,” in *Proc.ACM SenSys*, 2005, pp. 154–165.
3. J. T. Current and D. A. Schilling, “The covering salesman problem,” *Transp. Sci.*, vol. 23, no. 3, pp. 208–213, Aug. 1989.
4. M. Ma and Y. Yang, “SenCar: An energy efficient data gathering mechanism for large scale multihop sensor networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 10, pp. 1476–1488, Oct. 2007.
5. E. Ekici, Y. Gu, and D. Bozdag, “Mobility-based communication in wireless sensor networks,” *IEEE Commun. Mag.*, vol. 44, no. 7, pp. 56–62,Jul. 2006.
6. G. Xing, T. Wang, W. Jia, and M. Li, “Rendezvous design algorithm for wireless sensor networks with a mobile base station,” in *Proc. ACM Mobihoc*, May 2008, pp. 231–240.

7. M. Zhao and Y. Yang, "Bounded relay hop mobile data gathering in wireless sensor networks," *IEEE Trans. Comput.*, vol. 61, no. 2, pp. 265– 277, Feb. 2012.

8. P. Skraba, H. Aghajan, and A. Bahai, "RFID wakeup in event driven sensor networks," in *SigComm Poster Session*, Portland, OR, Aug. 2004. [Online]. Available: http://conferences.sigcomm.org/sigcomm/2004/ posters/skraba.pdf

[9]. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energyefficient communication protocols for wireless microsensor networks," in *Proc. HICSS*, Maui, HI, Jan. 2000, pp. 1–10.

[10]. O. Younis and S. Fahmy, "Distributed clustering in adhoc sensor networks:A hybrid, energy-efficient approach," in *Proc. IEEE INFOCOM*, Hong Kong, China, 2004, pp. 629–640.

[11] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, "Max–min Dcluster formation in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, 2000, pp. 32–41.

[12] X. Liu, J. Cao, S. Lai, C. Yang, H. Wu, and Y. Xu, "Energy efficient clustering for WSN-based structural health monitoring," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 2768–2776.

[13] S. Tang, J. Yuan, X. Li, Y. Liu, G. Chen, M. Gu, J. Zhao, and G. Dai, "DAWN: Energy efficient data aggregation in WSN with mobile sinks," in *Proc. IWQoS*, Jun. 2010, pp. 1–9.

[14] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," in *Proc. 22nd IEEE Intern. Parallel Distrib. Symp.*, Miami, FL, Apr. 2008, pp. 1–9.

[15] Z. Zhang, M. Ma, and Y. Yang, "Energy efficient multihop polling in clusters of two-layered heterogeneous sensor networks," *IEEE Trans, Comput.*, vol. 57, no. 2, pp. 231–245, Feb. 2008.

[16] G. Zhou, T. He, J. Stankovic, and T. Abdelzaher, "RID: Radio interference detection in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2005, pp. 891–901.

[17] G. Zhou, T. He, and J. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *Proc. 2nd Int. Conf. MobiSys*, Jun. 2004, pp. 125–138.

[18] B. Gavish, "Formulations and algorithms for the capacitated minimal directed tree problem," *J. ACM*, vol. 30, no. 1, pp. 118–132, Jan. 1983.

[19] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[20] D. S. Johnson, "Approximation algorithms for combinatorial problems,"*J. Comput. Syst. Sci.*, vol. 9, no. 3, pp. 256–278, Dec. 1974.

[21] J. T. Current and D. A. Schilling, "The covering salesman problem,"*Transp. Sci.*, vol. 23, no. 3, pp. 208–213, Aug. 1989.

[22] E. Arkin and R. Hassin, "Approximation algorithms for the geometric covering salesman problem," *DAMATH: Discrete Applied Math. Combinatorial Operations Res. Comput. Sci.*, vol. 55, 1994.

[23] S. S. Skiena, "Traveling salesman problem," in *The Algorithm Design Manual*. New York: Springer-Verlag, 1997, pp. 319–322.

[24] M. Ma and Y. Yang, "SenCar: An energy efficient data gathering mechanism for large scale multihop sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 10, pp. 1476–1488, Oct. 2007.